



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA


Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

DISEÑO Y DESARROLLO DE UN SELECCIONADOR
ENERGÉTICO DE PROTONES

Trabajo Fin de Grado

Grado en Ingeniería Mecánica

AUTOR/A: Esteban Blasco, David

Tutor/a: Santiago Praderas, Víctor Manuel

Cotutor/a externo: SEIMETZ, MICHAEL

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

ANÁLISIS, MEJORA Y ADECUACIÓN DE UN SELECCIONADOR ENERGÉTICO DE PROTONES

Trabajo Fin de Grado
Grado en Ingeniería en Ingeniería Mecánica

AUTOR/A: Esteban Blasco, David
Tutor/a: Santiago Praderas, Víctor
Cotutor/a externo: Seimetz, Michael
CURSO ACADÉMICO: 2022/2023

RESUMEN

Debido a los altos costes y el gran tamaño asociado a los actuales aceleradores de protones, es necesario estudiar nuevas tecnologías que permitan dotar de una mayor accesibilidad a la aceleración de protones para aplicaciones científicas y oncológicas. Así pues, el empleo de láseres pulsados para la aceleración de protones reduce considerablemente el tamaño de la instalación de un acelerador, pero produce un haz de protones con un espectro energético muy ancho. Por ello, es necesario el desarrollo de un seleccionador energético de protones para seleccionar los protones más interesantes para una aplicación determinada. Las aplicaciones a corto plazo se basan en estudios sobre la fluencia de protones lograda mediante láseres pulsados, pero para poder utilizar la tecnología en estudios oncológicos es necesario obtener un haz monoenergético capaz de penetrar una profundidad dada en cada caso.

Palabras clave: seleccionador energético de partículas, acelerador de protones, imanes permanentes, protonterapia, campos magnéticos.

RESUM

A causa dels alts costos i la gran grandària associada als actuals acceleradors de protons, és necessari estudiar noves tecnologies que permeten dotar d'una major accessibilitat a l'acceleració de protons per a aplicacions científiques i oncològiques. Així doncs, l'ús de làsers premuts per a l'acceleració de protons redueix considerablement la grandària de la instal·lació d'un accelerador, però produeix un feix de protons amb un espectre energètic molt ample. Per això, és necessari el desenvolupament d'un seleccionador energètic de protons per a seleccionar els protons més interessants per a una aplicació determinada. Les aplicacions a curt termini es basen en estudis sobre la *fluència de protons aconseguida mitjançant làsers premuts, però per a poder utilitzar la tecnologia en estudis oncològics és necessari obtindre un feix monoenergètic capaç de penetrar una profunditat donada en cada cas.

Paraules clau: seleccionador de enèrgetic de partícules, accelerador de protons, imants permanents, protonterapia, camps magnètics

ABSTRACT

Due to the high costs and the large size associated with current proton accelerators, it is necessary to study new technologies that allow proton acceleration to be more accessible for scientific and oncological applications. Thus, the use of pulsed lasers for proton acceleration considerably reduces the size of an accelerator setup but produces a proton beam with a very broad energy spectrum. Therefore, it is necessary to develop a proton energetic selector to select the most interesting protons for a given application. Short-term applications are based on studies on the fluence of protons achieved by pulsed lasers, but in order to use the technology in oncological studies it is necessary to obtain a monoenergetic beam capable of penetrating a given depth in each case.

Key words: energetic particle sorter, proton accelerator, permanent magnets, protontherapy, magnetic fields.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

MEMORIA

**Análisis mejora y adecuación de un seleccionador
energético de protones**

ÍNDICE DE LA MEMORIA

1. Objeto del proyecto.....	6
2. Consideraciones previas.....	7
2.1. Revisión del estado del arte.....	7
2.2. Tipos de aceleradores de partículas	8
2.2.1. Aceleradores electrostáticos.....	8
2.2.2. Aceleradores por radio frecuencia (RF)	9
2.2.3. Acelerador betatrón	12
2.3. Aceleración de protones mediante láseres pulsados	13
2.3.1. Introducción.....	13
2.3.2. Aplicaciones y ventajas de la tecnología.....	16
2.4. Antecedentes.....	16
3. Factores a considerar	19
3.1. Condiciones del encargo	19
3.1.1. Requisitos generales	19
3.1.2. Requisitos de adecuación.....	20
3.2. Normativa	21
4. Estudio de las alternativas	22
4.1. Alternativas.....	22
4.1.1. Alternativa 0: no actuación.....	22
4.1.2. Alternativa 1: empleo de varillas roscadas.....	23
4.1.3. Alternativa 2: empleo de perfiles en U sin radio de curvatura	24
4.1.4. Alternativa 3: empleo de perfiles de tipo C.....	24
4.1.5. Alternativa 4: empleo de placas metálicas	25
4.2. Descarte de alternativas	25
4.3. Presupuesto de las alternativas a escoger y elección de la solución final.....	26
4.3.1. Aspectos a tener en cuenta en la elaboración del presupuesto	26
4.3.2. Presupuesto de las alternativas a escoger	27
4.3.3. Elección de la solución final.....	27
5. Descripción de la solución adoptada y justificación detallada de las características y de los componentes de la solución adoptada.....	27
6. Proceso constructivo	28
7. Valoración de la solución adoptada y aspectos a solucionar.....	30
7.1. Primea hipótesis	31
7.2. Segunda hipótesis	33
8. Adecuación.....	34

ÍNDICE DE LA MEMORIA

8.1. Estudio sobre la influencia de la posición de los imanes en el haz. Configuración óptima para los imanes considerando campo magnético nulo en los huecos.	35
8.2. Estudio del campo magnético en los huecos. Configuración óptima para los imanes considerando el campo magnético simulado en los huecos.	41
8.3. Medidas del campo magnético	44
8.4. Simulaciones efectuadas con el campo magnético medido	56
8.4.1. Construcción del mapa magnético del campo BZ	56
8.4.2. Construcción de los mapas magnéticos de los campos B_x Y B_y . Efectos de B_y en el comportamiento del haz.	59
8.5. Variación del campo magnético con la temperatura.....	63
8.6. Efectos del aire en la propagación de los protones.....	65
9. Modificaciones.....	66
10. Experimentación y toma de resultados.....	69
11. Conclusiones.....	73
12. Bibliografía	74

ÍNDICE DE ILUSTRACIONES

Imagen 1. I3m en el Campus de Vera de la UPV (News, 2022).	6
Imagen 2. Esquema de funcionamiento de un tubo de rayos catódicos.	7
Imagen 3. Generador de alto potencial eléctrico en cascada de Cockroft-Walton (Jimeno, Presente, pasado y futuro de los aceleradores de partículas (Parte 1), 2020).	8
Imagen 4. Esquema de un Acelerador Lineal (Infante, 2012).	10
Imagen 5. Primer Ciclotrón.....	11
Imagen 6. Ciclotrón (Infante, 2012).	11
Imagen 7. Primer Betatrón de Kerst (History, 1940).	12
Imagen 8. Esquema de funcionamiento de la aceleración de protones mediante láseres pulsados (Millán, 2017).	14
Imagen 9. <i>Regla de la mano derecha</i> (Fernandez J. L., s.f.).	15
Imagen 10. Seleccionador energético de partida.....	16
Imagen 11. Campo magnético de un imán de herradura (Vectores de Campo Magnético, s.f.).	17
Imagen 12. Foto de uno de los 10 imanes C que componen el seleccionador.	17
Imagen 13. Mínima distancia entre imanes.	18
Imagen 14. Contacto entre imanes y guías.	18
Imagen 15. Esquema de la configuración de partida.	19
Imagen 16. Imanes a diferentes alturas.....	23
Imagen 17. <i>Esquema de uso de perfiles en forma de C.</i>	25
Imagen 18. <i>Perfil en U sin radio de curvatura.</i>	27

ÍNDICE DE LA MEMORIA

Imagen 19. <i>Lámina de aluminio.</i>	27
Imagen 20. Empleo de una brida como ayuda para desatornillar el tornillo más próximo a los bloques de neodimio que fija los imanes en el eje X.	28
Imagen 21. Explicación esquemática complementaria a la Imagen 16.	29
Imagen 22. Placa de metal descentrada.	29
Imagen 23. Placa de metal centrada. (Fuente: Elaboración propia)	29
Imagen 24. Representación de la repulsión que ejercen los imanes entre sí.	30
Imagen 25. Representación de esfuerzos de las vigas longitudinales.	33
Imagen 26. Seleccionador energético de protones con los pilares introducidos.	33
Imagen 27. Mapa del campo magnético de un Imán en la dirección del eje Z.	34
Imagen 28. <i>Trazado de partículas sin campo magnético entre imanes. Con espacios de 0,8 cm entre imanes y 1,4 cm entre bloques, siendo x constante.</i>	36
Imagen 29. <i>Trazado de partículas sin campo magnético entre imanes. Con espacios de 0,8 cm entre imanes y 1,4 cm entre bloques.</i>	37
Imagen 30. <i>Trazado de partículas sin campo magnético entre imanes. Con espacios de 0,5 cm entre imanes y 1,4 cm entre bloques.</i>	38
Imagen 31. <i>Trazado de partículas sin campo magnético entre imanes. Con espacios de 0,8 cm entre imanes y 3,8 cm entre bloques.</i>	39
Imagen 32. <i>Mapa magnético de la simulación sin campo magnético entre imanes. Con espacios de 0,8 cm entre imanes y 3,8 cm entre bloques.</i>	40
Imagen 33. <i>Mapa magnético de la simulación con suma de campo magnético entre imanes, y espacios de 0,8 cm entre imanes y 3,8 cm entre bloques.</i>	42
Imagen 34. <i>Mapa magnético de la simulación con suma del campo magnético entre imanes, y espacios de 0,8 cm entre imanes y 3,8 cm entre bloques.</i>	42
Imagen 35. Montaje del seleccionador para la toma de medidas.	44
Imagen 36. Compartimento cero de Gauss o “cero Gauss Chamber”.	45
Imagen 37. Introducción de la sonda en el área a medir.	46
Imagen 38. Áreas a medir.	47
Imagen 39. Inclinación o pérdida de la horizontalidad del carrito debido al momento que genera el desequilibrio de las fuerzas peso.	47
Imagen 40. Flexión de la viga debido al aumento del voladizo (y del momento flector).	48
Imagen 41. Reducción de la flexión del brazo que lleva la dirección del eje Y a partir de una tira de cinta adhesiva.	49
Imagen 42. Representación del recorrido de la sonda a través de la dirección del eje Y de la superficie a medir.....	49
Imagen 43. <i>Campo Bx medido manualmente.</i>	50
Imagen 44. <i>Campo By medido manualmente.</i>	50
Imagen 45. <i>Campo Bz medido manualmente.</i>	51
Imagen 46. <i>Campo Bx medido sobre el segundo bloque según la ruta de la “Tabla 4”.</i>	52
Imagen 47. <i>Campo By medido sobre el segundo bloque según la ruta de la “Tabla 4”.</i> (Fuente: Elaboración propia a partir del código recogido en “2. Mapas magnéticos” del “ANEXO IV”)	52
Imagen 48. <i>Campo Bz medido sobre el segundo bloque según la ruta de la “Tabla 4”.</i>	53
Imagen 49. <i>Campo Bx medido sobre el segundo bloque según la ruta de la “Tabla 5”.</i>	53
Imagen 50 <i>Campo By medido sobre el segundo bloque según la ruta de la “Tabla 5”.</i>	54

ÍNDICE DE LA MEMORIA

Imagen 51 <i>Campo Bz medido sobre el segundo bloque según la ruta de la “Tabla 5”.</i>	54
Imagen 52. <i>Medidas del campo magnético sobre el primer bloque según la ruta de la “Tabla 5”.</i>	55
Imagen 53 <i>Medidas del campo magnético sobre el primer bloque según la ruta de la “Tabla 6”.</i>	55
Imagen 54. <i>Mapa magnético construido a partir de las medidas del campo magnético con paso estrecho realizadas para los 2 bloques del seleccionador.</i>	56
Imagen 55. Campo Bz total del seleccionador obtenido copiando medidas de un único imán en cada bloque.	57
Imagen 56. Campo Bz total del seleccionador obtenido copiando medidas de cada imán a completar.	58
Imagen 57. Trazado de partículas a partir del campo magnético medido.	58
Imagen 58. Descomposición vectorial del campo magnético en los bordes del imán.	59
Imagen 59. Superposición (trazado de partículas) – (campo By medido).	60
Imagen 60. Campo By medido sobre el primer bloque.	61
Imagen 61. Trazado de partículas en el plano YZ considerando el campo magnético que aparece en la “Imagen 60”.	61
Imagen 62. Campo By y trazado de partículas en el plano YZ a partir de este (Resultados 1).	62
Imagen 63. Campo By y trazado de partículas en el plano YZ a partir de este (Resultados 2).	62
Imagen 64. Superposición del trazado de partículas en el plano YZ con diferentes sentidos de By.	63
Imagen 65. Superposición del trazado de partículas en el plano XY un campo magnético de intensidad $B_z \times 0,99$.	64
Imagen 66. Trazado de partículas en el Plano XY con pérdida de energía en aire a partir del campo magnético de la “Imagen 56”.	65
Imagen 67. <i>Mapa magnético más aproximado al medido realmente.</i>	67
Imagen 68. <i>Trazado de partículas a partir del campo recogido en la “Imagen 67”.</i>	67
Imagen 69. Montaje del seleccionador en la cámara de vacío.	68
Imagen 70. Montaje del colimador de grafito de 1 mm.	69
Imagen 71. Seleccionador nivelado y orientado correctamente delante de la fuente de protones.	69
Imagen 72. Campo magnético de los imanes C.	70
Imagen 73. Modificación del montaje del pinhole circular.	70
Imagen 74. Resultado de las películas radiocrómicas.	71

ÍNDICE DE TABLAS

Tabla 1. Energías del haz según el espesor de la hoja de tungsteno.	20
Tabla 2. Coste de las alternativas 2 y 4.	27
Tabla 3. Posición de los imanes en el eje X.	37
Tabla 4. <i>Posición final de los protones en el eje X sin campo magnético entre imanes. Con espacios de 0,8 cm entre imanes y 3,8 cm entre bloques.</i>	39
Tabla 5. <i>Posición final de los protones en el eje X con suma de campo magnético entre imanes, y espacios de 0,8 cm entre imanes y 3,8 cm entre bloques.</i>	43
Tabla 6. <i>Ruta de medida del segundo bloque con paso estrecho.</i>	51
Tabla 7. <i>Ruta de medida con paso ancho.</i>	53
Tabla 8. <i>Ruta de medida del segundo bloque con paso ancho.</i>	55
Tabla 9. <i>Ruta de medida para la parte de los imanes extremos que queda por medir.</i>	56
Tabla 10. <i>Posición final de los protones en el eje X según simulación de "Imagen 57".</i>	59
Tabla 11. <i>Datos técnicos sobre los bloques de neodimio (LDA., s.f.).</i>	63
Tabla 12. <i>Posición final de los protones de la "Imagen 65"</i>	64
Tabla 13. <i>Posición final de los protones de la "Imagen 68"</i>	67

ÍNDICE DE GRÁFICAS

Gráfica 1. <i>Campo Bz que afecta a un protón de 8 MeV al atravesar el mapa magnético de la "Imagen 32"</i>	41
Gráfica 2. <i>Campo Bz que afecta a un protón de 8 MeV al atravesar el mapa magnético de la "Imagen 33"</i>	43
Gráfica 3. <i>Comparación de resultados.</i>	72

1. Objeto del proyecto

El objetivo de este TFG es el de la revisión, mejora y adecuación, de un seleccionador energético de protones para un Acelerador Lineal (LINAC por sus siglas en inglés) de protones, con el fin de ensayarlo en el CNA (Centro Nacional de Aceleradores) ubicado en Sevilla y poder montarlo en futuros experimentos.

La actividad se va a desarrollar para un equipo de investigación del i3m (Instituto de Instrumentación para Imagen Molecular), cuyo ámbito de trabajo es la aceleración de protones mediante láseres pulsados. El nombre del equipo es LAIA (Láser Acceleration of Ions and Applications) y está estudiando los efectos radiobiológicos de estos protones con cultivos celulares, pero se enfrenta a diversos retos, los cuales tiene por objeto este TFG.

El i3m es un centro de investigación mixto creado en 2010 por la Universidad Politécnica de Valencia (UPV) y el Consejo Superior de Investigaciones Científicas (CSIC) con sede en el Campus de Vera de la UPV.



Imagen 1. *i3m en el Campus de Vera de la UPV (News, 2022).*

2. Consideraciones previas

2.1. Revisión del estado del arte

La aceleración de partículas es un campo de la física que ha experimentado un gran desarrollo a lo largo de los últimos siglos, y que ha permitido importantes avances en diversas áreas de la investigación, desde la física fundamental hasta la medicina.

La historia de la aceleración de partículas se remonta al siglo XIX, cuando se descubrió el electrón y se empezó a estudiar su comportamiento en campos eléctricos y magnéticos. Así pues, puede considerarse que el tubo de rayos catódicos, desarrollado por Crookes en 1875 como dispositivo de visualización, es el primer acelerador de partículas (de baja energía) conocido. Los rayos catódicos fueron descubiertos por primera vez por Plücker en 1858, al aplicar una descarga eléctrica sobre gases enrarecidos presentes en un tubo de vacío y observar el brillo que se formaba en las paredes.

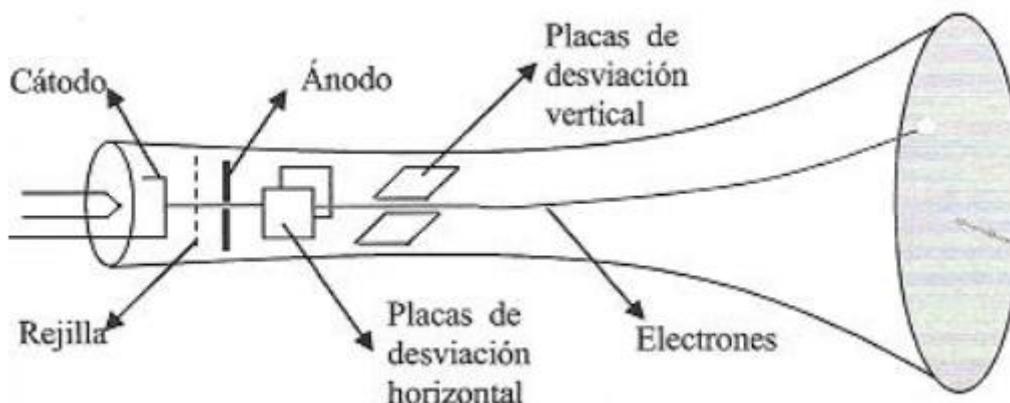


Imagen 2. Esquema de funcionamiento de un tubo de rayos catódicos.
(Fuente: <http://rsefalicante.umh.es/TemasCampoElectrico/Electrico12.htm>)

Plücker se percató de la modificación de la posición del brillo que producía la aplicación de campos magnéticos, creados con un electroimán, a los rayos, lo que le permitió avanzar con descubrimientos importantes en el campo de la espectroscopia de gases. No obstante, fue Thomson en 1897 quien determinó la relación carga/masa de los rayos catódicos, midiendo la desviación producida por un campo eléctrico y la cantidad de energía, tras la exposición de la Ley de Lorentz por su autor, Hendrick Lorentz, en 1895 (Martínez D. P., 2021), que estudiaremos más adelante. Además, Thomson dedujo el carácter corpuscular¹ de los rayos y su naturaleza eléctrica negativa, sin pasar por alto la baja masa que debía poseer el electrón. Es así como descubrió el electrón y desarrolló un nuevo modelo atómico. Fue finalmente Millikan, en 1913, quien determinó por separado el valor de la carga y la masa del electrón mediante el experimento de la gota de aceite (Callejo J. C., Rodríguez Perez, & Santa Marta Pastrana, 2013).

Por otro lado, Rutherford demostró la existencia de un núcleo cargado positivamente en el interior de los átomos, publicando así su modelo atómico en 1911, gracias a un experimento que pretendía estudiar la dispersión de las partículas α , emitidas por un

¹ Compuestos de partículas cargadas

elemento radiactivo (polonio), cuando atravesaban finas láminas de diversos elementos como el aluminio o el oro. En el experimento se observó que, contrariamente a lo que cabría esperar con el modelo atómico de Thomson, muchas de las partículas presentaban una modificación importante de su trayectoria, o incluso, llegaban a rebotar.

Más tarde, en 1919, Rutherford logró inducir una reacción nuclear a partir de partículas α .

Estos avances acrecentaron el interés por la aceleración de partículas para poder estudiar el núcleo atómico y superar la barrera coulombiana del núcleo² con el objetivo de seguir produciendo reacciones nucleares artificiales. Arrancaba así el comienzo de la carrera por construir aceleradores de partículas que fuesen capaces de producir mayores energías en las partículas. Es por ello que, a partir de la década de 1920, se empezaron a desarrollar los primeros aceleradores de partículas de alta energía. Para entender mejor el interés por aumentar la energía de las partículas, en el anexo I se recogen algunos conceptos importantes.

2.2. Tipos de aceleradores de partículas

De la infinidad de aceleradores de partículas de alta energía que se han desarrollado a lo largo de la historia, es posible hacer la siguiente distinción atendiendo al mecanismo de aceleración:

2.2.1. Aceleradores electrostáticos

En 1928, gracias al invento del primer generador en cascada en 1920 por Heinrich Greinacher, los físicos británicos John Douglas Cockcroft y Ernest Walton comenzaron a desarrollar un generador electrostático en cascada que llegó a alcanzar 700kV en 1932. Gracias a él, consiguieron acelerar protones a 400 keV ese mismo año logrando la primera reacción nuclear ($\text{Li} + \text{p} \rightarrow 2\text{He}$). Sentaban así el comienzo de la era de los aceleradores de partículas de alta energía.



Imagen 3. *Generador de alto potencial eléctrico en cascada de Cockcroft-Walton (Jimeno, Presente, pasado y futuro de los aceleradores de partículas (Parte 1), 2020).*

² Fuerza de repulsión electrostática entre los protones en el núcleo atómico.

Este tipo de acelerador lograba grandes voltajes y se podían introducir mejoras, pero pronto alcanzaba el nivel de saturación en el que un aumento de la energía de las partículas implicaba un aumento mucho mayor del tamaño del acelerador.

Es por ello que se empezó a desarrollar otro tipo de aceleradores electrostáticos basados en corriente continua (DC: Direct Current). En 1931, Van de Graaff construyó el primer generador electrostático de alta energía que permitió la construcción de su acelerador y derivó en el acelerador Tandem Van de Graf, que lograba energías más altas a un costo de una menor intensidad de corriente. No obstante, estos aceleradores seguían alcanzando el nivel de saturación por debajo de una energía de 1 GeV, por lo que, aunque son útiles para estudios en física de partículas, seguían teniendo limitaciones en términos de la energía máxima que podían alcanzar.

El principal campo de aplicación de los aceleradores electrostáticos es el de la Espectrometría de Masas con Aceleradores (AMS, por sus siglas en inglés), pero también se utilizan en medicina, debido a que la alta sensibilidad alcanzada permite administrar radiación a un paciente con una precisión y control muy elevados, lo que ayuda a reducir los efectos secundarios y minimizar el daño a los tejidos sanos. Además, otras ramas científicas como la física nuclear o la astrofísica también se ven beneficiadas.

2.2.2. Aceleradores por radio frecuencia (RF)

Estos aceleradores se comenzaron a desarrollar prácticamente de forma simultánea a los aceleradores electrostáticos, pero consiguen mayores energías que los anteriores ya que en lugar de utilizar una diferencia de potencial constante, aceleran las partículas mediante la sucesión de campos eléctricos separados entre sí que invierten su polaridad con una frecuencia dada (radiofrecuencia). El principio de funcionamiento es el mismo que el de los aceleradores electrostáticos ya que es una diferencia de potencial la que acelera las partículas, pero a diferencia de los anteriores, que utilizan un alto potencial eléctrico en una sola etapa, los aceleradores por radiofrecuencia utilizan una sucesión de diferencias de potencial sucesivas entre cavidades. Estas diferencias de potencial es lo que se conoce como Potencial Eléctrico de Radiofrecuencia (REP, por sus siglas en inglés).

Dentro de estos aceleradores se puede distinguir a su vez entre:

2.2.2.1. Aceleradores lineales (LINACs: Linear Accelerators)

En 1924, el físico Gustaf Ising³ propuso por primera vez el empleo de campos eléctricos variables con respecto al tiempo sobre tubos de deriva para acelerar partículas cargadas, pero fue el ingeniero Rolf Wideröe en 1928 quien logró acelerar iones

³No confundir con Ernst Ising, hijo de Gustaf Ising (mercader), quien, también en 1924, estudió el carácter binario de una cadena lineal de momentos magnéticos que interactúan entre sí (Caro, y otros, 1999).

Consideraciones previas

negativos de potasio a 50 keV con un oscilador de 1 MHz y 25 kV, dando lugar a la creación del primer acelerador lineal de electrones (Rohlf, s.f.).

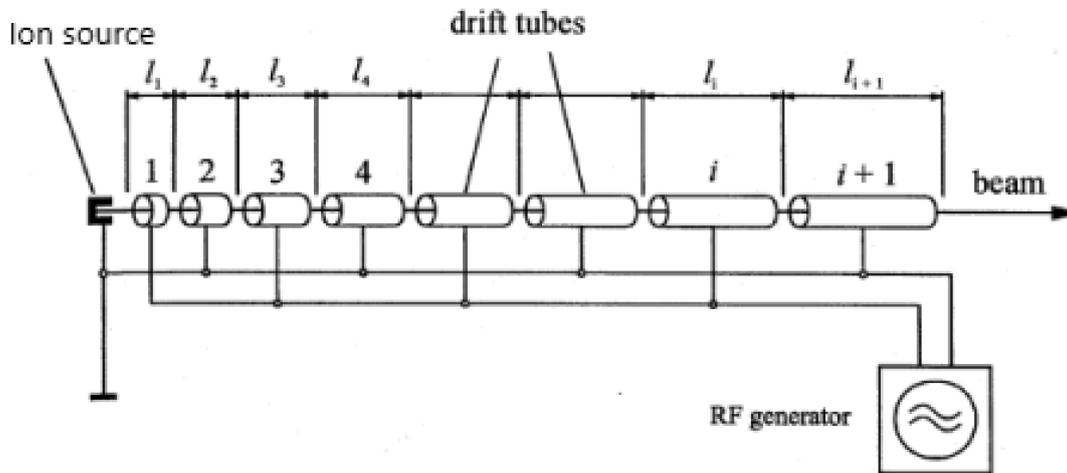


Imagen 4. Esquema de un Acelerador Lineal (Infante, 2012).

Posteriormente, Luis Walter Alvarez construyó el primer acelerador lineal de protones, también conocido como de Iones pesados, con el que alcanzó una energía de 32 MeV en 1946 (Jimeno, Pasado, presente y futuro de los aceleradores de partículas (Parte 2), 2020).

Las aplicaciones de estos aceleradores van desde la medicina hasta la investigación en física de partículas, pasando por la seguridad nacional al ser útiles para detectar sustancias como explosivos o estupefacientes.

2.2.2.2. Aceleradores circulares

En 1929 Ernest O. Lawrence, inspirado por Ising y Wideröe, concibió el ciclotrón y logró acelerar protones hasta 80 keV en 1931, junto a su estudiante Stan Livingston, con un ciclotrón de apenas 10 cm, hecho de alambre y cera, al aplicarle un potencial de unos 2000 V. Más tarde, en 1932 y con un ciclotrón un poco más elaborado, Lawrence alcanzó los 1.25 MeV, energía suficiente para provocar la desintegración del núcleo atómico (Varela, 2015).

Consideraciones previas

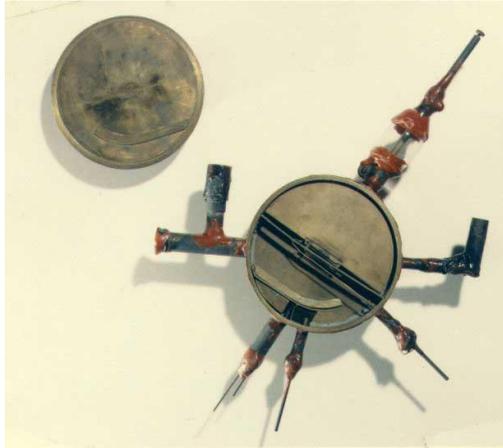


Imagen 5. *Primer Ciclotrón.*

(Ruiz L. M., Ernest Lawrence y los inicios de la Gran Ciencia (I): Nace el ciclotrón, 2013)

Este tipo de aceleradores utilizan cavidades de radiofrecuencia, al igual que los anteriores LINAC, pero se le suma la aplicación de un campo magnético fijo para curvar la trayectoria de los haces y lograr una trayectoria circular que permite que, con una única cavidad, los haces se aceleren constantemente, con una frecuencia de giro fija, tal y como queda explicado en la imagen 6.

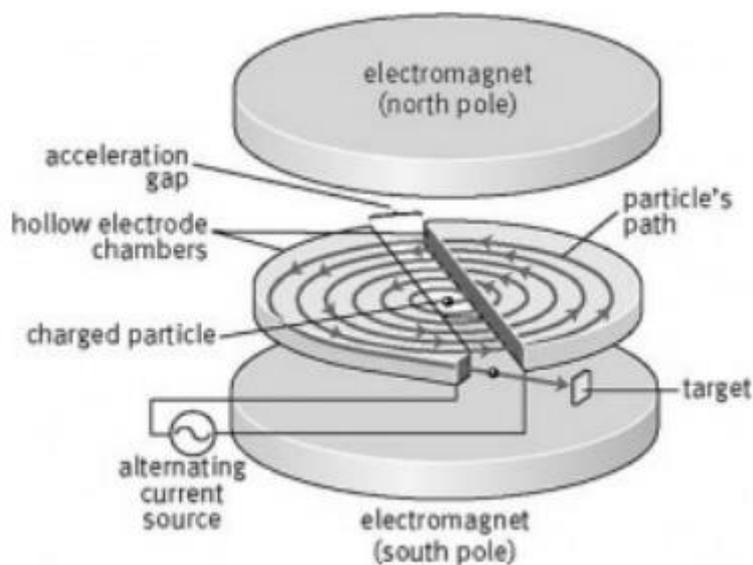


Imagen 6. *Ciclotrón (Infante, 2012).*

2.2.3. Acelerador betatrón

Este tipo de aceleradores se basan en el principio de la ley de Faraday, que enuncia que un campo magnético variable puede generar un campo eléctrico perpendicular a la dirección de cambio del campo eléctrico. El betatrón fue el primer acelerador de partículas capaz de acelerar electrones a altas energías utilizando campos magnéticos. Esta rama de la aceleración de partículas tiene sus comienzos sobre la década de 1920 al igual que las otras, pero dio los primeros frutos un poco más tarde.

Aunque Donald W. Kerst es generalmente reconocido como el inventor del betatrón, Wideröe desempeñó un papel crucial al sentar las bases teóricas y conceptuales para su diseño. En 1922 Wideröe esbozó el concepto del ray-transformer, un acelerador de partículas basado en la interacción de partículas cargadas con campos eléctricos oscilantes y derivó en la regla 2-1, que es una regla empírica que establece una relación entre la frecuencia de oscilación del campo eléctrico aplicado y la velocidad de las partículas cargadas en el ray-transformer.

Dos años después, Wideröe añadió la condición para la estabilidad radial y en 1927 fabricó un modelo de betatrón, pero no funcionó y es por ello que abandonó su camino y construyó un acelerador lineal, del que hemos hablado en el apartado "2.2.2.1. Aceleradores lineales por radiofrecuencia (LINACs: Linear Accelerators)".

La historia siguió su curso sin ningún avance importante en este tipo de aceleradores hasta 1940 cuando Kerst construyó un betatrón capaz de acelerar electrones hasta 2.2 MeV. Más tarde, en 1950, este físico estadounidense desarrolló el mayor betatrón del mundo capaz de alcanzar los 300 MeV.



Imagen 7. *Primer Betatrón de Kerst (History, 1940).*

Las aplicaciones de este tipo de acelerador son principalmente médicas ya sea para obtener rayos X y rayos gama, o para tratamientos oncológicos.

Es necesario destacar que algunos de estos aceleradores poseen imanes antes de entrar al acelerador para seleccionar las partículas dependiendo del cociente e/m según la ley de Lorentz. Es por esto por lo que, su principal mérito, es acelerar cualquier tipo de partícula cargada con una pequeña dispersión energética. Esto sienta un precedente al objeto de este trabajo.

2.3. Aceleración de protones mediante láseres pulsados

2.3.1. Introducción

Estas vías de aceleración han seguido estudiándose y han resultado en otros muchos aceleradores con aplicaciones muy diversas, pero nos centraremos ahora en la tecnología que nos ocupa.

El mecanismo de aceleración que se está estudiando, descubierto alrededor del año 2000, utiliza láseres pulsados ultracortos y ultraintensos como medio para la aceleración de partículas (protones). La interacción de estos pulsos láser con la materia, en concreto con una lámina sólida delgada. Cuando el pulso láser (campo electromagnético de alta intensidad) alcanza la lámina, la materia se transforma en plasma altamente ionizado, y se genera una corriente de electrones que se mueven a velocidades relativistas (cerca de la velocidad de la luz) en un solo periodo de oscilación del campo electromagnético. Esta transferencia de energía del pulso a los electrones los expulsa de la superficie del blanco debido a la fuerza ponderomotriz⁴. Esta aceleración de electrones dentro del blanco genera radiación de rayos X y un campo eléctrico muy fuerte en la superficie trasera del blanco. Este campo eléctrico acelera los iones generados, incluidos los protones, mediante el mecanismo conocido como Target Normal Sheath Acceleration (TNSA) (Poole, y otros, 2018).

La presencia de hidrógeno en la superficie del blanco, proveniente de contaminantes como moléculas de agua o hidrocarburos, es responsable de la aceleración de protones (Millán, 2017).

⁴ Fuerza que experimenta una partícula cargada cuando se encuentra expuesta a un campo electromagnético.

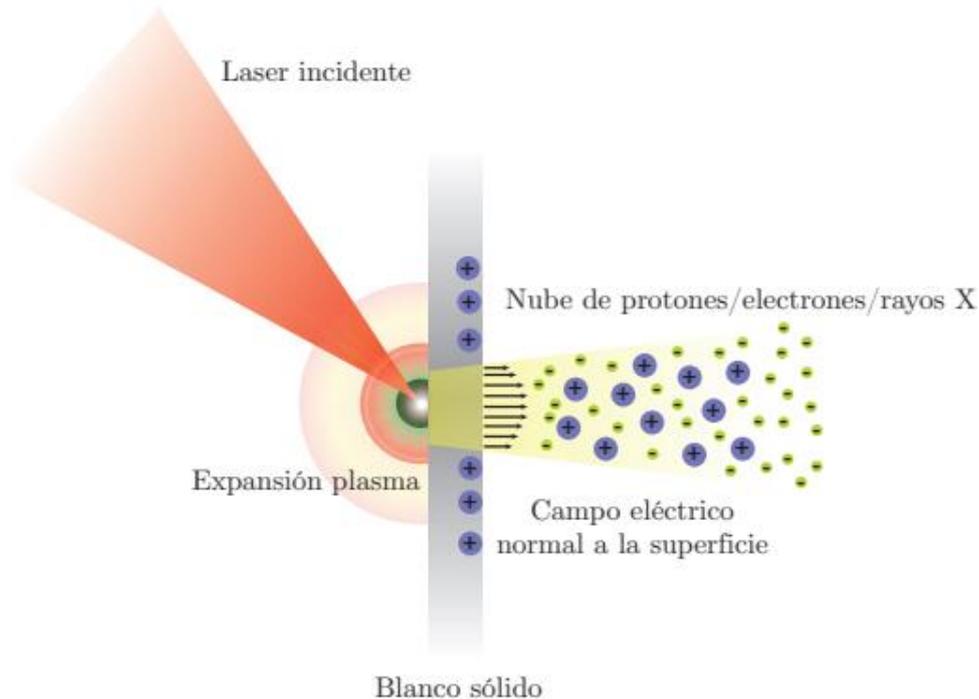


Imagen 8. Esquema de funcionamiento de la aceleración de protones mediante láseres pulsados (Millán, 2017).

Esta tecnología puede presentar ciertas ventajas frente a los aceleradores actuales de las mismas características, pero produce un haz con una distribución energética y angular más amplia. Por ello, es necesario focalizar los protones y separarlos por energías, si se desea conseguir un haz que presente el máximo número de protones con una energía concreta, por ejemplo, por sus posibles aplicaciones médicas (Torralba, y otros, 2022).

El estudio del focalizador de protones, que tiene por objeto el enfoque del haz de protones para mejorar la densidad de partículas, requerirá del empleo de imanes cuadrupolares, pero no es objeto de este TFG. No obstante, el funcionamiento del seleccionador o separador se introduce a continuación.

El principio de funcionamiento que utilizará el seleccionador energético desarrollado es el mismo que utiliza un espectrómetro de masas, y se basa en la relación carga/masa de las partículas a estudiar y la Ley de Lorentz.

La ley de Lorentz establece que: “una partícula cargada q que circula a una velocidad \vec{v} por un punto en el que existe una intensidad de campo magnético \vec{B} , sufrirá la acción de una fuerza denominada fuerza de Lorentz cuyo valor es proporcional al valor de q , \vec{v} y \vec{B} ” (Fernandez J. L., s.f.).

El valor de dicha fuerza se obtiene por medio de la siguiente expresión (1):

$$\vec{F} = q \cdot \vec{v} \times \vec{B} \quad (1)$$

Por lo tanto, para una partícula determinada acelerada a una velocidad \vec{v} que se hace pasar a través de un campo magnético \vec{B} , la fuerza de Lorentz tendrá un valor concreto en función de la carga de la partícula. Esta fuerza adquirirá la dirección perpendicular al

Consideraciones previas

movimiento de la partícula y la dirección del campo magnético, según la regla de la mano derecha:

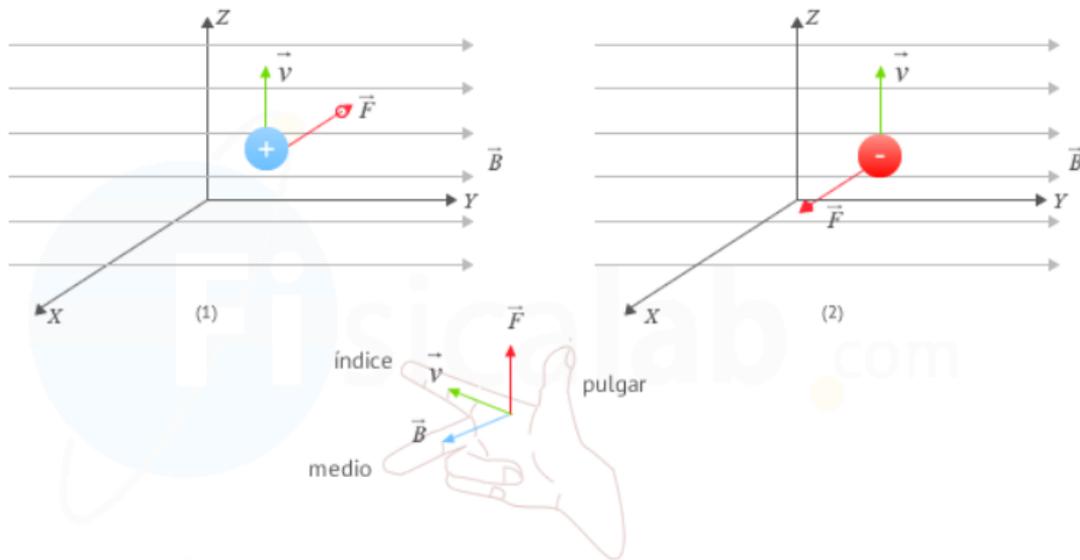


Imagen 9. Regla de la mano derecha (Fernandez J. L., s.f.).

Por otro lado, la manera en que la partícula responde a la fuerza de Lorentz viene determinada por su masa, pues si tenemos en cuenta la 2ª ley de Newton, que establece que (2):

$$F = m \cdot a \quad (2)$$

Donde:

- F es la fuerza neta aplicada a la partícula (Fuerza de Lorentz)
- m es la masa de la partícula
- a es su aceleración

Es posible determinar la aceleración que experimenta una partícula y modifica su trayectoria como (3):

$$a = \frac{F}{m} = \frac{q}{m} \cdot \vec{v} \times \vec{B} \quad (3)$$

En el caso de la espectrometría de masas se aceleran partículas hasta una velocidad dada para hacerlas pasar a través de un campo magnético. Más adelante se registra el lugar de impacto de las partículas, por lo que conocidos los valores de \vec{B} y \vec{v} es posible determinar la relación carga/masa de la partícula, por lo que es posible conocer con exactitud el tipo y número de partículas de la materia a estudiar.

No obstante, en la situación que nos concierne, la relación carga/masa es constante, pero varía la velocidad de un conjunto de protones, por lo que la aceleración que afectará a los protones será distinta pudiéndose separar así los protones por energías. En nuestro caso esas partículas serán protones.

2.3.2. Aplicaciones y ventajas de la tecnología

Las aplicaciones de esta tecnología son principalmente científicas y oncológicas. Aunque se está todavía lejos de conseguir utilizarla en protonterapia, es interesante puesto que actualmente los aceleradores de protones son muy costosos y grandes, dificultando así el acceso de los pacientes que lo requieren al tratamiento de protonterapia, que presenta ciertas ventajas terapéuticas (dosimétricas) frente a la radioterapia convencional.

Estas ventajas se deben a que los protones producen una mayor radiación a medida que pierden su energía por lo que calculando la energía mínima necesaria para que los protones lleguen al tejido tumoral es posible suministrar una mayor radiación sin afectar al tejido sano. Además, al ser los protones más pesados que los electrones su dispersión al entrar al cuerpo humano es menor, lográndose una liberación más precisa y localizada de la radiación.

2.4. Antecedentes

Se parte de un seleccionador energético de protones compuesto por 2 bloques de campos magnéticos opuestos formados por 5 imanes C cada uno. Estos imanes están constituidos, a su vez, por un total de 20 bloques de neodimio con forma de prisma rectangular enfrentados 10 a 10.

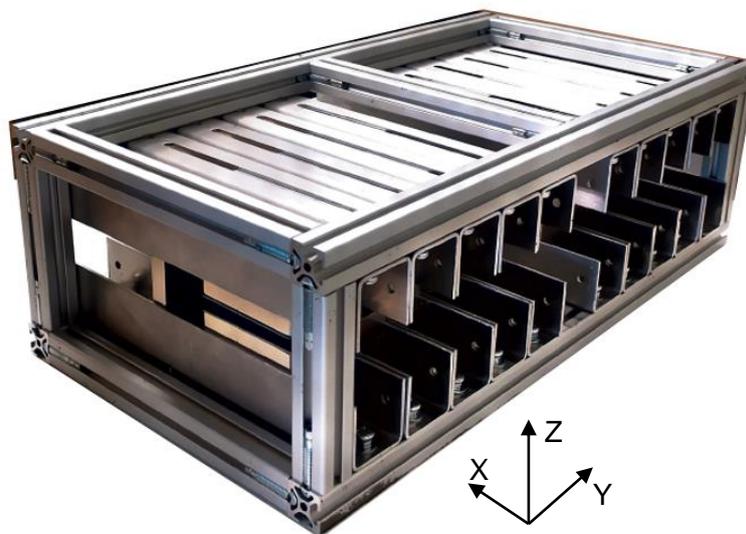


Imagen 10. *Seleccionador energético de partida.*
(Fuente: *Elaboración propia*)

Con el sistema de referencia que se ha escogido el sentido de la fuerza en la dirección X que se obtiene mediante la ley de Lorentz es opuesto a al sentido de la dirección positiva del eje X de nuestro sistema de referencia. No obstante, este aspecto ya se encuentra corregido en las imágenes del campo magnético en la dirección del eje Z de este TFG para que al aplicar la regla de la mano derecha se obtengan los resultados que se esperan.

De este modo, los bloques enfrentados producen un campo magnético similar al que produce un imán de herradura:

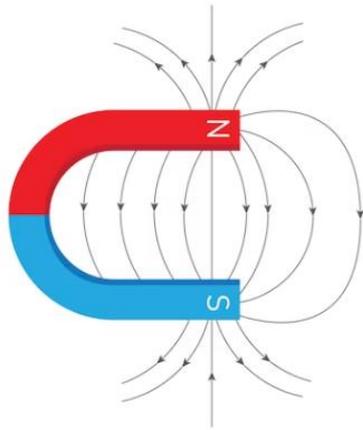


Imagen 11. Campo magnético de un imán de herradura (Vectores de Campo Magnético, s.f.).

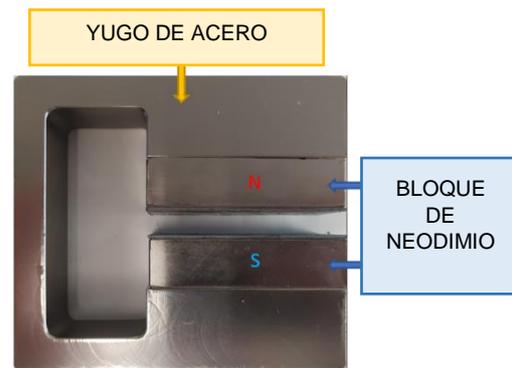


Imagen 12. Foto de uno de los 10 imanes C que componen el seleccionador. (Fuente: Elaboración propia)

Aunque como veremos en el apartado “9. Experimentación y toma de resultados” esto no es del todo cierto.

Para lograr que los imanes enfrentados generen un campo magnético con una componente principal (B_z) sin que se suelden magnéticamente entre sí, se han utilizado piezas de acero (material ferromagnético) como la que se muestra en la “Imagen 12”. Cabe destacar que los prismas rectangulares no se encuentran adheridos a los yugos de acero de manera estructural, siendo el campo magnético el único principio físico que los mantiene unidos. Es por ello que al manipular el seleccionador es necesario ser especialmente cuidadosos con los imanes, puesto que, ya sea por el campo magnético de los imanes adyacentes o por fuerzas aplicadas por el operario, es posible desacoplar alguno de los bloques de neodimio de su yugo de acero provocando la unión entre imanes enfrentados.

Para conseguir un campo más uniforme y facilitar la separación de los imanes enfrentados en caso de unión, cada imán de neodimio lleva adherida magnéticamente una placa de hierro en su cara vista. Además, estas placas protegen los imanes ya que son bastante frágiles.

Es necesario destacar también que la mínima distancia alcanzable entre imanes de distintos bloques es de 1 cm, pues la estructura del seleccionador posee una viga transversal intermedia que sirve de barrera para evitar que la atracción de los imanes centrales produzca su unión.

Además, la mínima distancia a la que se pueden acercar imanes de un mismo bloque es de $0,5 \pm 0,1$ cm (medido con una regla con una escala de medición precisa de hasta los milímetros) puesto que las guías de los imanes tienen un espesor de 0,2 cm y no se produce un contacto plano entre guías, ni guías e bloques de neodimio (“Imagen 13”).

Consideraciones previas

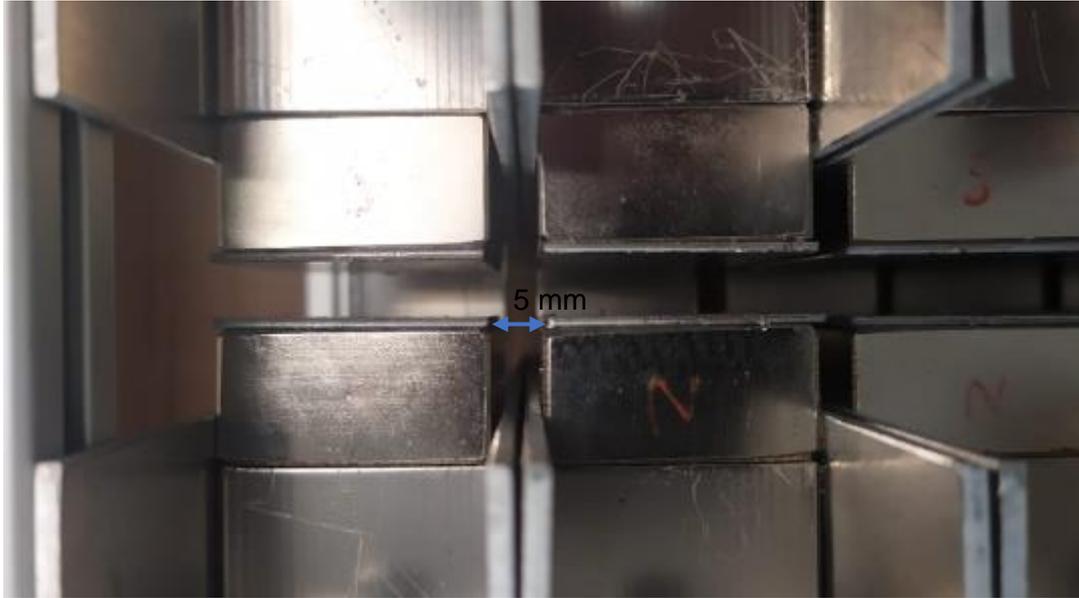


Imagen 13. *Mínima distancia entre imanes.*
(Fuente: *Elaboración propia*)

Estos imanes presentaban ciertos problemas de precisión en su integración con el soporte, pero tras estudiarlo se ha llegado a la conclusión de que el problema viene causado porque la curvatura de doblado de los perfiles en U que mantienen a los imanes en su posición impide un contacto firme entre las superficies planas de imán y soporte, pues el contacto se produce entre las esquinas del imán y la curvatura del doblado del soporte.



Imagen 14. *Contacto entre imanes y guías.*

Además, el ajuste de la posición de los imanes se realiza de forma manual mediante tornillos, siendo este un sistema poco eficiente y preciso, por lo que sería interesante mejorarlo de cara a realizar ensayos en aceleradores para adaptar la configuración del seleccionador a las energías del haz de protones de manera más rápida y sencilla.

Por otro lado, inicialmente el seleccionador está montado con una configuración óptima para separar haces de protones acelerados mediante láseres pulsados, cuyo intervalo energético es de entre 2 MeV y 8 MeV, según se aprecia en la "Imagen 15".

Factores a considerar

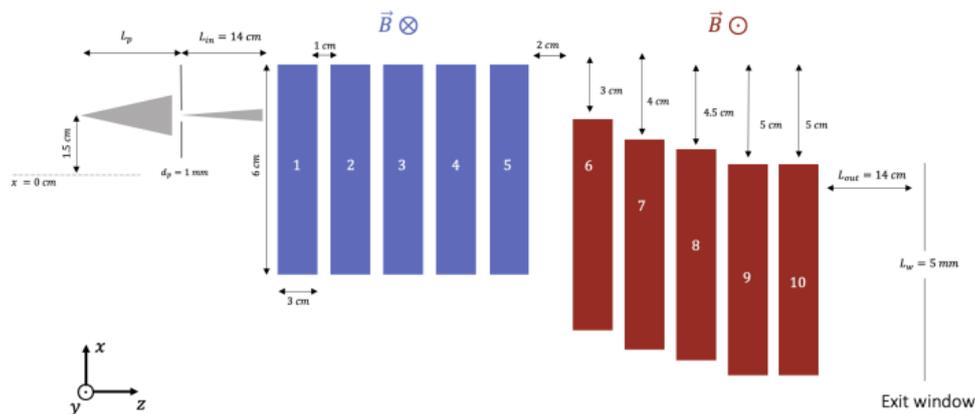


Imagen 15. Esquema de la configuración de partida.
(Fuente: i3m)

Los estudios y simulaciones que permitieron escoger esta configuración fueron realizados por Alicia Reija Vecino en 2021 mediante el empleo de un código de Python (“ANEXO II”) que le permitió modelar el comportamiento del haz, a partir de sus características y el campo magnético generado por los imanes del seleccionador, teniendo en cuenta su disposición.

3. Factores a considerar

3.1. Condiciones del encargo

3.1.1. Requisitos generales

- Los materiales de las partes estructurales deben ser no magnéticos o paramagnéticos⁵ para evitar interferencias con el campo magnético generado por los imanes.
- Se requiere un elemento estructural que sostenga lateralmente los bloques de neodimio en su lado largo para prevenir que los campos magnéticos adyacentes los desplacen del yugo de acero.
- Se debe utilizar un elemento de separación entre el primer y segundo bloque de imanes para evitar la unión de los imanes situados en la parte central del seleccionador, ya que sus campos magnéticos opuestos generan una fuerte atracción que se suma a la repulsión del resto de imanes de su bloque.
- El seleccionador debe ser de configuración variable para adaptarse a las condiciones requeridas en cada ensayo, lo que implica que todos los imanes del seleccionador deben tener libertad de movimiento sobre el plano horizontal (Plano XY).
- El dispositivo debe ser capaz de funcionar en condiciones de vacío y ser fácilmente transportable, ya que en algunos experimentos se ubicará dentro de la cámara de vacío del acelerador.

⁵ Materiales que tienen una respuesta magnética débil cuando se les aplica un campo magnético.

3.1.2. Requisitos de adecuación

En segundo lugar, para poder adecuar el seleccionador energético de protones es necesario conocer el espectro energético de los protones a la entrada del mismo, así como las características del haz. En este sentido, aunque el fin último es separar los protones acelerados mediante láseres pulsados, el objetivo más próximo es estudiar el funcionamiento del dispositivo en el CNA (Acelerador Ciclotrón 18/9 MeV, 2023) en cuyo ciclotrón se alcanza una energía fija de 18 MeV. No obstante, es posible añadir hojas de tungsteno en la salida del acelerador que absorben parte de la energía para rebajarla y lograr unas condiciones más interesantes de cara a la aplicación real del seleccionador. Así pues, cuanto mayor sea el espesor de la hoja, más energía se pierde y más se abre la distribución energética, obteniendo las energías recogidas en la siguiente tabla:

Tabla 1. Energías del haz según el espesor de la hoja de tungsteno.

ESPESOR W (μm)	ENERGÍA (MeV)
0	13.5 ± 0.2 *
50 ± 8	12.0 ± 0.6
100 ± 16	10.3 ± 0.7
150 ± 20	8.4 ± 0.8
200 ± 24	6.1 ± 1.0
250 ± 28	3.1 ± 1.5
300 ± 30	1.9 ± 1.6

La distancia en aire entre la brida de vacío y nuestro colimador puede variar para el experimento por lo que habría que volver a calcular los valores exactos, pero estas energías pueden usarse de manera orientativa. Por lo tanto, se ha decidido que se realizarán medidas a una energía de 8,4 MeV o 6,1 MeV para estudiar la separación obtenida entre los haces de diferentes energías, teniéndose los siguientes requisitos:

- Espectro Energético de entre 5,1 MeV y 9,2 MeV.
- Se debe estudiar la pérdida de energía de los protones en el aire y cómo afecta esta al funcionamiento del seleccionador puesto que en este acelerador el seleccionador no irá ubicado en la cámara de vacío.
- Atendiendo a la fluencia de protones y su dispersión angular se expone que en el ciclotrón se obtendrán haces paralelos cuyas características vienen dadas por la anchura del colimador montado y no son factores a tener en cuenta en las simulaciones ni objeto de estudio en los ensayos.
- Necesidad de conocer el tipo de imanes a emplear y sus características para poder realizar las simulaciones necesarias. En este sentido, se parte de una medida del campo magnético en la dirección vertical (Z) generado por uno de los imanes permanentes que compondrán el seleccionador energético de partículas final.

Finalmente, el presupuesto de la investigación es limitado, por lo que resulta necesario pensar en abaratar lo máximo posible las soluciones adoptadas.

* Sin W (0 μm) no son 18 MeV porque hay una hoja de Al además y un poco de pérdida en aire.

3.2. Normativa

Cabe mencionar en este apartado la Ley 31/1995, de 8 de noviembre, de Prevención de Riesgos Laborales, que constituye las garantías y responsabilidades para proporcionar cierto nivel de protección de la salud de los trabajadores en relación con los riesgos derivados de las condiciones de trabajo (BOE, 1997).

Compuesto por 12 artículos, los dos primeros establecen las disposiciones generales de esta Ley 31/1995, de 8 de noviembre, mientras que los siguientes artículos, hacen referencia a las obligaciones del empresario. Especial mención al artículo 6, que alude a aspectos más técnicos a cerca de las medidas preventivas, con el objetivo de garantizar la seguridad y salud en los lugares de trabajo, con la pretensión de evitar riesgos para los trabajadores (BOE, 1997).

Además, España ha ratificado Convenios de la Organización Internacional del Trabajo, que mantienen relaciones con la seguridad y salud en los lugares de trabajo, y que constituye nuestro ordenamiento jurídico interno. Hacemos así alusión al Convenio número 155 de la OIT, sobre la seguridad y salud de los trabajadores, de 22 de junio de 1981, ratificado por España el 26 de julio de 1985, así como el Convenio número 148 de la OIT, sobre el medio ambiente de trabajo, de 20 de junio de 1977, ratificado por nuestro país el 24 de noviembre de 1980 (BOE, 1997).

También se han establecido Directivas, concretamente mencionaremos la Directiva 89/654/CEE, de 30 de noviembre, en el ámbito de la Unión Europea diversos criterios en materia de seguridad, salud en los lugares de trabajo, además de medidas de protección contra accidentes y situaciones de riesgo (BOE, 1997).

De la misma manera, cabe mencionar el Área de Prevención de FREMAP, donde se presenta un Manual que pretenda dar a conocer los riesgos relativos a una oficina y las medidas preventivas disponibles para los trabajadores con el objetivo de esquivarlos. Esta información se establece con relación al entorno del puesto de trabajo, a la manipulación manual de cargas, posturas de trabajo, y también en función del tipo de puesto de trabajo (Social, 2014).

Por otro lado, atendiendo a la normativa aplicable en lo referente a la exposición a radiación y campos magnéticos se tiene:

- Directiva 97/43/Euratom del Consejo, de 30 de junio de 1997, en relación con la protección de la salud ante los riesgos procedentes de las radiaciones ionizantes en exposiciones médicas.
- Directiva 90/641/Euratom del Consejo, de 4 de diciembre de 1990, a propósito de la protección operacional de los trabajadores exteriores con amenaza de exposición a radiaciones ionizantes por actuación en zona controlada.
- Directiva 89/618/Euratom del Consejo, de 27 de noviembre de 1988, referente a la información de la comunidad sobre las medidas de protección sanitaria aplicables y acerca de conducta a seguir en caso de emergencia radiológica.
- Directiva 2003/122/Euratom del Consejo, de 22 de diciembre de 2003, en cuanto al control de fuentes radioactivas selladas de actividad elevada y de las fuentes huérfanas.
- La Directiva 97/43/Euratom del Consejo, de 30 de junio de 1997, en el Real Decreto 815/2001, de 13 de julio, acerca del fundamento del uso de las radiaciones ionizantes para la protección radiológica de las personas con ocasión de exposiciones médicas.

- ISO 150 20553:2006, “Protección de la Radiación – Monitoreo de trabajadores ocupacionalmente expuestos a riesgos de contaminación interna con material radiactivo”, en relación a aquellos trabajadores bajo amenaza de exposición radioactiva a partir del monitoreo de la absorción potencial y real, además del control de la exposición.
- UNE-EN ISO 14146:2021, sobre la protección radiológica, concretamente haciendo alusión a los criterios y límites para la evaluación periódica de los servicios de dosimetría.
- IEEE Std 1308-1994TM. “IEEE Recommended Practice for Instrumentation: Specifications for Magnetic Flux Density and Electric Field Strength Meters- 10 Hz to 3 kHz”.

4. Estudio de las alternativas

Dados los problemas existentes en el actual seleccionador energético de protones, que pueden resumirse en:

- Dificultad de modificación de la configuración
- Escasa uniformidad en la altura de los imanes

Se proponen las siguientes alternativas para dar con la solución de diseño adecuada. La configuración óptima será estudiada posteriormente.

4.1. Alternativas

Las alternativas propuestas se detallan a continuación de manera enunciativa a la espera de ser estudiadas en el siguiente apartado (“4.2. Descarte de alternativas”).

4.1.1. Alternativa 0: no actuación

Esta alternativa consiste en la no actuación sobre el problema y resulta económica y sencilla de implementar.

En este sentido, si bien modificar la configuración del seleccionador actual resulta en un proceso largo y tedioso, no es especialmente problemático pues una vez montada la configuración deseada, no es necesario modificarla hasta que se necesite otro tipo de configuración para unas condiciones energéticas distintas. Además, no es requerida una alta precisión en el posicionamiento de los imanes a lo largo de sus ejes de libertad por lo que montarlos de manera manual no representa un problema importante.

No obstante, la posición vertical de los imanes sí debe ser corregida para lograr una mayor fluencia de protones, pues el espacio por el que ha de pasar el haz se reduce si hay imanes a diferentes alturas tal y como se muestra en la “Imagen 16”:



Imagen 16. *Imanes a diferentes alturas.*
(Fuente. Elaboración propia)

Es por esto que esta alternativa no es aceptable como solución final y se enuncian las alternativas recogidas a continuación.

4.1.2. Alternativa 1: empleo de varillas roscadas

Esta solución consiste en diseñar un seleccionador totalmente nuevo que utilice una serie de varillas roscadas unidas a los imanes mediante carros con pinzas para modificar la su posición a partir del giro de las varillas. Esta solución es especialmente interesante si se acompaña de motores para controlar el giro de las varillas roscadas pues sería posible entonces determinar la posición exacta de los imanes en cada momento. Además, facilitaría y agilizaría el proceso de montaje.

Para llevarla a cabo, sería necesario utilizar, idealmente, un microcontrolador y 20 motores si se desea dotar de libertad de movimiento a cada imán en las dos direcciones del plano XY. No obstante, se podrían emplear únicamente 2 motores para viabilizar su construcción adaptándola a las necesidades del proyecto.

De esta manera se separarían los 2 bloques actuales en bloques independientes, quedando el primer bloque fijo y el segundo anclado al carrito que modificaría su posición a lo largo del plano XY. En relación con esto, es importante destacar que sería interesante situar el comienzo del intervalo de movimiento del segundo bloque en el eje Y al menos a 1 cm del primer bloque para evitar que en el caso más desfavorable el último imán del primer bloque y el primero del segundo llegaran a unirse.

Por otro lado, el montaje de ambos bloques debe permitir el movimiento de los imanes a lo largo del eje X del seleccionador, por lo que podría utilizarse un montaje similar al actual, pero seguiría siendo necesario mejorar el posicionamiento vertical, para lo que pueden usarse alguna de las alternativas propuestas en los siguientes apartados.

Aunque no se utilizasen motores para localizar los imanes en la posición deseada, el empleo de varillas roscadas en lugar de tornillos sobre guías para fijar los imanes facilita el posicionamiento puesto que mover los imanes para fijarlos en otra posición es una tarea difícil debido a las fuerzas que estos generan. Además, esta alternativa facilitaría la adaptación del seleccionador energético a las diferentes energías requeridas debido a que, el factor determinante en la separación energética lograda es la distancia existente entre bloques de imanes de campos magnéticos opuestos, que a su vez determina el posicionamiento sobre el eje X de los imanes, tal y como se puede apreciar más abajo en el apartado “8. Adecuación”. Sin embargo, con el montaje actual es necesario desanclar todas las guías de uno o ambos bloques para separar los bloques entre sí, y repetir el proceso sobre los imanes si se desea modificar su posición sobre el eje X.

4.1.3. Alternativa 2: empleo de perfiles en U sin radio de curvatura

Esta opción consiste en utilizar otra clase de perfiles en U con la anchura de los imanes (3cm) pero sin radio de curvatura. Estos perfiles pueden solucionar el problema de la regulación de la altura de los imanes, pero se seguiría empleando el posicionamiento manual de los mismos utilizando tornillos sobre guías para fijarlos.

Es una solución sencilla y limpia que sólo requiere de la sustitución de los perfiles actuales por otros y no añade elementos al seleccionador. No obstante, este tipo de perfiles son más difíciles de encontrar, y sería necesario subcontratar procesos de mecanizado para cortarlos y crear los orificios y carriles sobre los que se colocarán los tornillos que mantienen los imanes en la posición deseada.

4.1.4. Alternativa 3: empleo de perfiles de tipo C

Una alternativa similar a la anterior, pero, al utilizar como soporte un perfil de tipo C, el radio de curvatura del perfil no interaccionaría con los bordes del imán. No obstante, estos perfiles dificultarían la restricción lateral de los imanes puesto que sólo se produciría un contacto lineal entre imanes y soportes en sus caras laterales. De esta forma, es posible que fuese necesario que la línea de presión, o borde del perfil, se encontrase sobre el centro de las caras laterales o por encima de estas para que el momento generado por las fuerzas magnéticas y las fuerzas de reacción no separe los bloques de neodimio de su yugo de acero, tal y como se aprecia en la “Imagen 17”:



Imagen 17. Esquema de uso de perfiles en forma de C.
(Fuente: Elaboración propia)

Además, los yugos de acero quedan también restringidos lateralmente puesto que la parte inferior y superior del bloque están unidas mediante un elemento vertical en uno de sus extremos que contactaría con los bordes del perfil, pero es posible que se produzca un momento sobre el eje Y del imán debido a las fuerzas magnéticas.

Por tanto, sería necesario estudiar la magnitud del momento generado y si es necesario entonces introducir además algún otro elemento restrictivo, o si por contraposición, al restringir la lateralidad de los bloques de neodimio también quedan fijos los yugos de acero.

Al igual que en la solución anterior, esta alternativa requiera de la subcontratación de procesos de mecanizado.

4.1.5. Alternativa 4: empleo de placas metálicas

Esta solución trata de incorporar placas metálicas de una anchura inferior a la del hueco de los perfiles menos dos veces su radio de curvatura y un espesor superior ese radio de curvatura del perfil para asegurar el contacto entre caras planas.

De esta manera, sería posible garantizar un contacto superficial plano entre imanes y soportes restringiendo la oscilación vertical (estática). Esta es una opción similar a las anteriores pues sólo solucionaría la alineación vertical de la altura, sin embargo, dificultaría un poco los procesos de modificación del seleccionador al añadir piezas móviles al conjunto. Por otro lado, es la opción más económica y sencilla de implementar, pues el único proceso de mecanizado a llevar a cabo es el corte de las placas metálicas.

4.2. Descarte de alternativas

Después de considerar las anteriores opciones, es posible extraer las conclusiones que se detallan a continuación.

Atendiendo a la solución número 0, es posible descartarla puesto que es necesario igualar las alturas de todos los imanes y no sería posible sin ejecutar ninguna acción.

Por otro lado, la solución número 1 no cumple con las necesidades del proyecto puesto que es necesario que el seleccionador energético sea lo más sencillo y portable posible para realizar ensayos en los distintos aceleradores a los que se desplazará el grupo de investigación. Además, es necesario montarlo en la cámara de vacío de los

aceleradores, por lo que se debe buscar en todo momento la máxima reducción de tamaño posible para garantizar la versatilidad, pues en determinados experimentos es posible que sea necesaria la introducción de otros elementos dentro de la cámara. Teniendo esto en cuenta, no es necesario pasar a estudiar los costes de su implementación, pero se puede apreciar que esta alternativa necesita ser combinada con una de las otras, por lo que su coste será mayor que el de cualquiera de ellas.

Así pues, tras el estudio de las 2 primeras alternativas se ha decidido que no se dará solución al problema de la dificultad de modificación de la posición de los imanes del seleccionador, quedando este tema pendiente para futuras revisiones en un estado más avanzado de la investigación.

De entre las siguientes tres alternativas, es posible descartar, sin entrar en mucho detalle, la alternativa número 4, porque es más compleja y requiere ser estudiada, habiendo muchas variables que pueden invalidarla al generar nuevos problemas sobre el montaje del seleccionador. Además, no aporta ninguna ventaja competitiva frente a las otras dos restantes.

Por lo tanto, para dar con la solución que más se ajusta a las necesidades del seleccionador, es necesaria una comparación de las alternativas 3 y 5. A priori, podría parecer que la solución 3 es más conveniente que la 5 porque mantendría la simplicidad del montaje actual al sustituir los elementos actuales y no introducir elementos adicionales. Sin embargo, la solución número 5 es igualmente sencilla de implementar y no existe ninguna restricción en el número de piezas del seleccionador. Por consiguiente, y como tampoco existen restricciones temporales, para escoger la solución final se atenderá estrictamente a aspectos presupuestarios. Para poder llegar a una conclusión se ha generado un presupuesto, que recoge el coste estimado de cada solución de manera simplificada.

4.3. Presupuesto de las alternativas a escoger y elección de la solución final

4.3.1. Aspectos a tener en cuenta en la elaboración del presupuesto

Es necesario destacar que en la búsqueda de los componentes que integrarían ambas soluciones se ha determinado las siguientes restricciones para dar con una conclusión objetiva:

- El material de conformado de los componentes a emplear será el aluminio, por no presentar propiedades magnéticas.
- Las dimensiones de los componentes son estimadas, pero deberán adaptarse al fin último de la solución.
- Se realizará una comparativa de precios en la que al menos se recogerán 2 presupuestos distintos para cada componente en caso de que no compartan proveedor para asegurar una buena elección.

Además, este presupuesto es orientativo y nada tiene que ver el presupuesto final, pero sirve a modo de guía para decantarse por una solución.

Descripción de la solución adoptada y justificación detallada de las características y de los componentes de la solución adoptada

4.3.2. Presupuesto de las alternativas a escoger

Tabla 2. Coste de las alternativas 2 y 4.

<p>3. PERFILES EN U SIN RADIO DE CURVATURA</p>  <p>Imagen 18. Perfil en U sin radio de curvatura.</p>	<p>5. EMPLEO DE PLACAS METÁTLICAS</p>  <p>Imagen 19. Lámina de aluminio.</p>
<ul style="list-style-type: none"> - El precio ronda los 10€ por metro - Se necesitarían $2 \times 10 \times 0,225 = 4,5\text{m}$ por lo que sería necesario comprar 5 perfiles de 1m. 	<ul style="list-style-type: none"> - Aproximadamente 4,20 € por metro - Se necesitan 1,90m por lo que sería necesario comprar 2 planos de 1m
<p>COSTE TOTAL APROXIMADO: 50€</p>	<p>COSTE TOTAL APROXIMADO: 8,40€</p>

4.3.3. Elección de la solución final

Teniendo entonces en cuenta aspectos presupuestarios, se ha optado por la solución número 3.

5. Descripción de la solución adoptada y justificación detallada de las características y de los componentes de la solución adoptada

Rememorando y ampliando los aspectos tratados anteriormente, la solución consiste en utilizar unas placas metálicas para asegurar un contacto plano entre las esquinas de los imanes y el fondo de los perfiles en U para lograr una altura homogénea para todos los imanes.

Como los perfiles tienen un radio de curvatura aproximado de 1 mm, las placas deben de ser de una anchura inferior a 28 cm y un espesor mayor a 1mm para asegurar un contacto plano.

Además, la longitud de las placas tiene que ser inferior a 10 cm, que es la longitud de los imanes en el eje X, para que los tornillos que no interfieran con los tornillos que sostienen a los imanes en dicho eje.

Teniendo esto en cuenta se han escogido como materia prima 2 láminas de aluminio de 1000 x 25 x 2 mm, las cuales han sido cortadas con una longitud de 9 mm utilizando una sierra para metales manual y un sargento para sujetar la lámina a la mesa y facilitar el corte. Posteriormente se ha utilizado una navaja sin filo para eliminar las rebabas y se han colocado en su posición.

6. Proceso constructivo

Para montar las placas en su posición, es necesario desmontar las guías y los imanes del seleccionador. Además, se deben aflojar los tornillos que sostienen los pilares para aumentar la distancia entre las guías superiores e inferiores y dotarlas así del espacio necesario para introducir las placas de aluminio. No obstante, se ha detectado un problema durante el montaje, pues al permitir a los imanes cierto juego vertical, los campos magnéticos han producido alteraciones en su configuración inicial, ya que algunos se han unido entre ellos o se han girado. Para solucionarlo, se ha decidido que, para manipular el seleccionador energético, y siempre que se necesario aumentar el espacio vertical, se disminuirá la tensión de los tornillos de los pilares de un lado o del otro según se requiera, pero nunca simultáneamente. Además, los tornillos podrán ser aflojados, pero no se deberán liberar nunca de sus respectivas tuercas.

Por otro lado, como los tornillos que sostienen los imanes en la posición deseada a lo largo del eje X se basan en el contacto con los imanes para restringir el movimiento imponiendo una limitación morfológica, al llevar la llave hexagonal necesaria para atornillar o desatornillar el tornillo a su posición, esta es fuertemente atraída por el imán que sujeta dicho tornillo complicando bastante la introducción de la llave en la cabeza del tornillo. Dado este problema, se recomienda utilizar una brida como elemento seleccionador, tal y como se muestra en la “Imagen 20”:

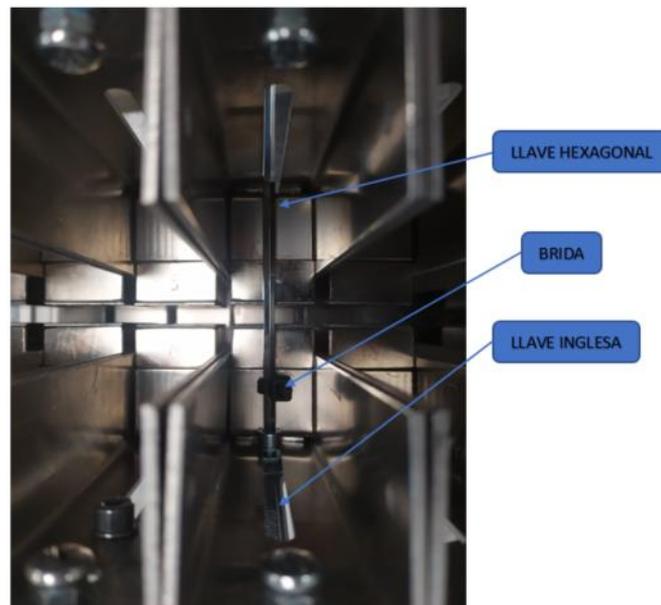


Imagen 20. Empleo de una brida como ayuda para desatornillar el tornillo más próximo a los bloques de neodimio que fija los imanes en el eje X.
(Fuente: Elaboración propia)

Y visto de perfil (“Imagen 21”):

Proceso constructivo

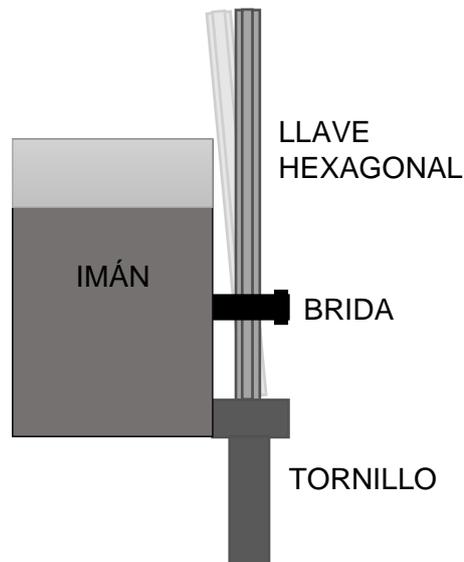


Imagen 21. Explicación esquemática complementaria a la Imagen 16.
(Fuente: Elaboración propia)

De esta manera, se simplifica mucho el proceso.

Además, cabe destacar que es necesario tener en cuenta que las fuerzas de atracción y repulsión de los imanes son de una magnitud considerable siendo interesante el empleo de herramientas, como destornilladores de punta plana, a modo de palanca para moverlos a la posición deseada.

Finalmente, a la hora de atornillar las vigas longitudinales a los pilares, es necesario prestar atención a la posición de las placas, pues deben de encontrarse aproximadamente en el centro de las guías para que la curvatura de los perfiles no les afecte:

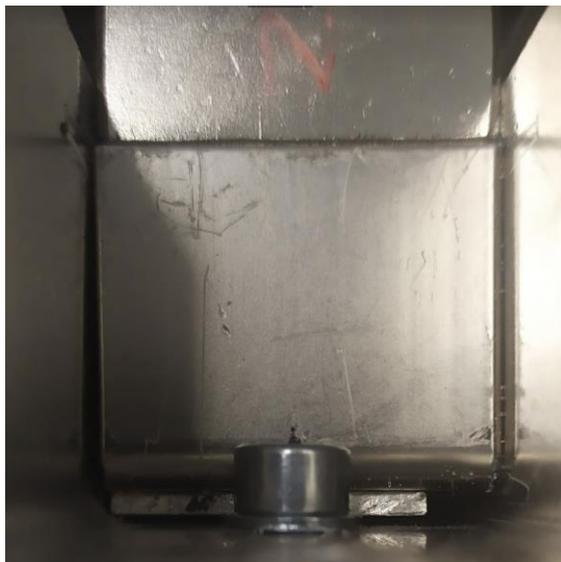


Imagen 22. Placa de metal descentrada.
(Fuente: Elaboración propia)

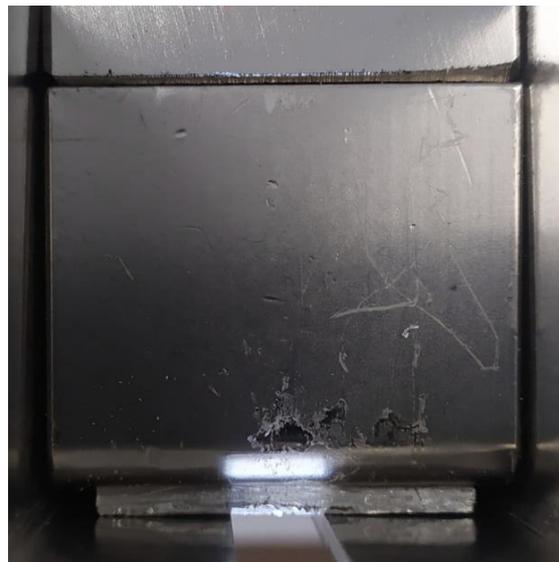


Imagen 23. Placa de metal centrada.
(Fuente: Elaboración propia)

7. Valoración de la solución adoptada y aspectos a solucionar

Una vez montado, se ha comprobado que el problema inicial se ha resuelto con éxito prácticamente en todos los imanes del seleccionador, pero se ha detectado una pequeña holgura vertical todavía en la parte central del seleccionador, aunque se ha reducido las diferencias entre alturas en general.

Para determinar el origen de esta holgura es necesario prestar atención a otro detalle, y es que, a medida que nos desplazamos hacia el centro, las placas de aluminio van ganando capacidad de deslizamiento sobre las guías transversales de los imanes (Eje X), hasta ganar una total libertad en la parte central. No obstante, cabe destacar que dicha libertad no está presente simultáneamente en las dos caras del imán, si no que aparece únicamente en una de las dos.

Teniendo en cuenta los aspectos anteriores, es posible deducir que las vigas longitudinales reducen los esfuerzos de compresión sobre los imanes a medida que nos alejamos de sus puntos de anclaje, llegando a producir una compresión nula en su parte central, al menos sobre una de las caras de los imanes. Para dar explicación a este comportamiento, es posible dar, principalmente, con dos posibles explicaciones, que pueden influir a la vez.

1. Los yugos de acero que componen los imanes tienen una altura con elevada tolerancia que produce una ausencia de contacto entre caras planas de imán y guía cuando un imán es demasiado bajo como para que la diferencia de alturas sea compensada por la compresión sobre las placas y guías de aluminio.
2. Las vigas longitudinales están sometidas a esfuerzos de flexión que producen una flecha máxima sobre su centro, lo que se traduce en un mayor espacio entre vigas superior e inferior dando lugar a una compresión nula.

No obstante, ninguna de los dos razonamientos explicaría por qué las placas de aluminio únicamente tienen juego en uno de los extremos del imán. Para dar explicación a esto es necesario tener en cuenta el campo magnético. Se observa que, si presionamos sobre el extremo fijo de un imán con juego, la placa de aluminio adquiere entonces la capacidad de moverse lateralmente, lo que indica que hay alguna fuerza que la presiona aumentando su fuerza de rozamiento e impidiendo así su movimiento. Dado que no hay ningún componente estructural presente que pueda generar este tipo de fuerzas, es posible deducir que es el campo magnético el que las produce tal y como se aprecia en la siguiente imagen:

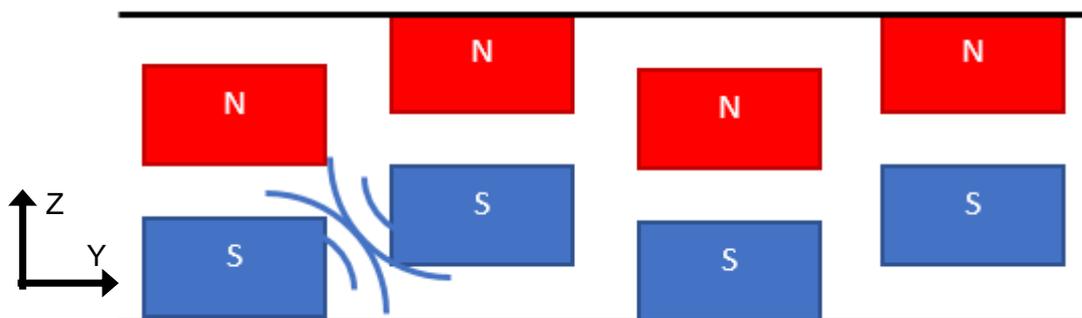


Imagen 24. Representación de la repulsión que ejercen los imanes entre sí.
(Fuente: Elaboración propia)

Como los imanes tienen libertad de movimiento en el eje Z, no se encuentran exactamente a la misma altura, por lo que se producen unas fuerzas de repulsión con una componente horizontal, que es anulada por las guías, y otra vertical, que es la que comprime una de las placas de aluminio, pero deja libre la placa opuesta.

Explicado este comportamiento es necesario estudiar las anteriores justificaciones para determinar qué fenómeno físico produce la holgura vertical en el centro del seleccionador. Ambas podrían dar explicación al problema observado, no obstante, sería difícil explicar mediante la primera hipótesis por qué las únicas placas con juego son las centrales, pues es muy improbable que, por azar, los imanes más altos hayan quedado en los extremos y los más bajos en el centro. Aun así, se ha comenzado estudiando esta primera explicación.

7.1. Primea hipótesis

Para comenzar, se ha medido la altura de todos los imanes. Para la realización de las medidas se ha utilizado un pie de rey digital que permite leer hasta las centésimas de milímetro con una precisión de 0,03 mm. Es necesario destacar que, para evitar posibles errores de calibración, este calibre permite referenciarlo a cero siempre que se desee, por lo que antes de comenzar las medidas se ha inicializado el cero con el pie de rey totalmente cerrado. De esta manera es posible minimizar, por ejemplo, los efectos de la temperatura sobre las medidas.

Las medidas han sido difíciles de realizar puesto que el material que conforma el calibre es ferromagnético, posiblemente acero, pudiendo afectar esto a la precisión, pues la atracción que sufren las mandíbulas del calibre se traduce en un aumento de la presión, que a su vez puede producir pequeñas deformaciones o la penetración del recubrimiento de los yugos de acero. Es por ello que únicamente se han realizado 3 medidas, pero es suficiente para el estudio que se pretende llevar a cabo.

Las medidas quedan:

IMANES	1ª MEDIDA (mm)	2ª MEDIA (mm)	3ª MEDIDA (mm)
1	90,17	90	90,03
2	90,02	89,91	89,74
3	90,4	90,17	90,12
4	90,56	90,1	90,26
5	90,23	89,93	89,96
6	90,11	89,8	89,87
7	90,4	90,21	90,12
8	89,86	89,74	89,86
9	90,3	90,05	89,92

Teniendo entonces en cuenta los valores máximos y mínimos para la altura medida:

- $h_{max} = 90,56$ mm
- $h_{min} = 89,78$ mm

Es posible establecer la tolerancia de la altura de los imanes como (4) (Metrología dimensional, 2017):

$$T = (h_{max} + 0,03) - (h_{min} - 0,03) = 0,88 \text{ mm} \quad (4)$$

Valoración de la solución adoptada y aspectos a solucionar

Se observa que existe una diferencia importante de alturas entre los diferentes imanes, e incluso entre partes distintas de un mismo imán, y dados los factores que se comentaron con anterioridad, es posible deducir que las medidas no son del todo fiables. No obstante, aunque sabemos que las piezas se han mecanizado en CNC (control numérico por computadora), no disponemos de la tolerancia real. Por lo tanto, para determinar si esas las diferencias dimensionales en la altura de un imán son influyentes o despreciables, debido a la compresión de las placas de aluminio de manera aproximada, es posible realizar los cálculos que se muestran a continuación.

Según la Ley de Hooke (Muñoz, y otros, 2020) se tiene que la tensión unitaria en la dirección del eje Z viene dada por la siguiente ecuación (5):

$$\sigma_z = E \cdot \varepsilon_z \quad (5)$$

Donde:

- Módulo elástico del aluminio: $E = 70 \text{ GPa}$
- Deformación: $\varepsilon_z = \frac{\Delta z}{z_0}$ (6)

Suponiendo entonces que:

- Se tuviesen únicamente dos imanes, uno con una altura constante sobre el límite inferior de la tolerancia y el otro con la máxima altura posible.
- La presión se ejerciese de manera uniforme en dirección del eje Z a partir de la compresión de dos planchas de rigidez infinita paralelas situadas a ambos lados de los imanes en la parte externa de las placas de aluminio.

Para que los imanes alineasen sus centros en la dirección del eje Z, sería necesario que las 2 placas de aluminio que se encuentran a ambos lados del imán más alto sufriesen una compresión igual a $T/2$ cada una.

Por lo tanto, suponiendo que ambas placas son idénticas, y teniendo en cuenta las ecuaciones (5) y (6), se tiene:

$$\varepsilon_z = \frac{0,88}{4} = 0,22 \quad (6)$$

$$\sigma_z = 70 \cdot 0,22 = 15,4 \text{ GPa} \quad (5)$$

Entonces para una sola placa se obtiene una fuerza (7):

$$F = \sigma_z \cdot A = (15,4 \cdot 1000) \cdot (90 \cdot 25) = 34650000 \text{ N} \quad (7)$$

Esta fuerza es imposible de suministrar mediante el apriete de los cuatro tornillos que comprimen las vigas, por lo tanto, las placas de aluminio no son capaces de mitigar los efectos de la alta tolerancia que se tiene para la altura de los imanes. No obstante, estos cálculos son muy rudimentarios y se utilizan suposiciones y datos que pueden ser bastante distintos de la realidad. Por lo tanto, dado que sí que existe flexión en las vigas (también en las guías, pero su momento de inercia es mucho mayor y la longitud menor, siendo la flecha prácticamente despreciable en comparación) y es muy probable que la compresión requerida sobre las placas de aluminio sea mucho menor, se estudiará la segunda hipótesis para darle solución y comprobar entonces si se soluciona el problema o se requiere de un estudio más detallado de la primera.

7.2. Segunda hipótesis

Esta segunda consideración establece que holgura se produce debido a los esfuerzos de flexión que afectan a la viga y producen una flecha máxima sobre la parte central tal y como se demuestra en la siguiente imagen:

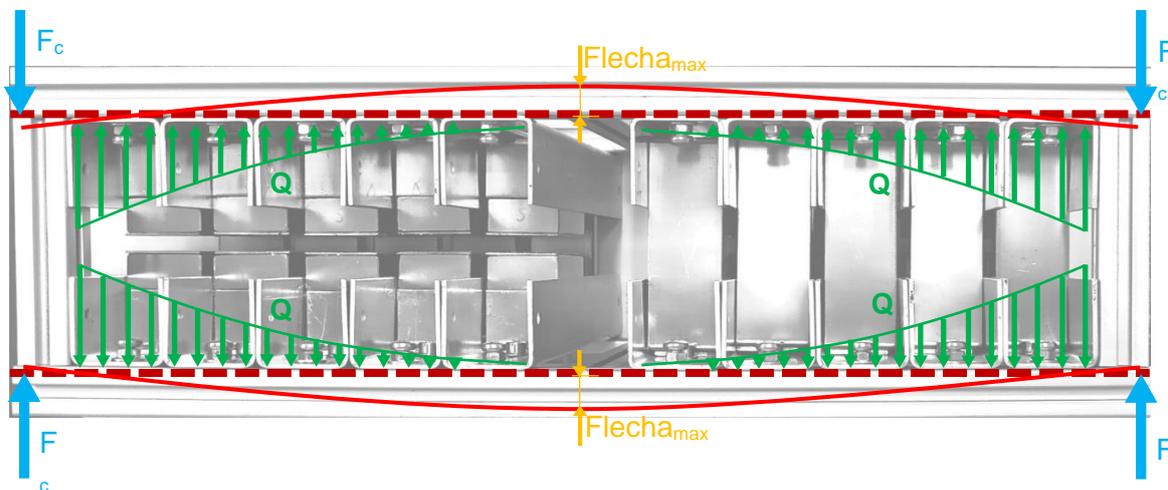


Imagen 25. Representación de esfuerzos de las vigas longitudinales.
(Fuente: Elaboración propia)

Caso ideal en el que todos los imanes tienen la misma altura y la compresión es ejercida por los cuatro tornillos de manera uniforme. Además, las placas de aluminio son idénticas y homogéneas, y las guías no tienen flexión ni adquieren momentos en la dirección del eje Y.

Las presiones se ejercen en los extremos de las vigas, que podría decirse que están en voladizo, generándose un momento que produce una flecha máxima en el centro de la viga. Para solucionar el problema se han introducido unos pilares a compresión en el centro que reducen el voladizo a la mitad, solucionando el problema, tal y como se aprecia a continuación:



Imagen 26. Selector energético de protones con los pilares introducidos.
(Fuente: Elaboración propia)

Nota. En esta fotografía falta el primer imán, pero se ha comprobado que la solución es válida incluso cuando se montan todos los imanes del selector.

Esta solución impone una nueva restricción al seleccionador, pues la mínima distancia entre bloques posible pasa de 1 cm a 2,5 cm aproximadamente, ya que, antes de introducir los pilares, eran las vigas transversales centrales las que hacían de tope, debido a su interacción con los tornillos que sujetan a los imanes en la dirección del eje X, pero ahora son directamente los extremos laterales de las guías de los imanes los que interactúan con los pilares centrales. No obstante, esto no supone ningún problema, tal y como veremos a continuación.

8. Adecuación

Para adaptar la posición de los imanes a las características del haz a estudiar, es necesario realizar algunos cálculos iterativos. Por ello, se han desarrollado varios códigos de Python que permiten simular el comportamiento de los protones en su paso por el campo magnético presente en el seleccionador. Cabe destacar que los datos del campo magnético de los que se parte para la realización de las simulaciones son las medidas efectuadas del magnetismo en la dirección Z sobre el plano de simetría de un imán en esta misma dirección, abarcando la superficie proyectada de este y el área formada por su lado largo y los 5 mm perpendiculares adyacentes a cada lado, tal y como se puede apreciar en la siguiente imagen:

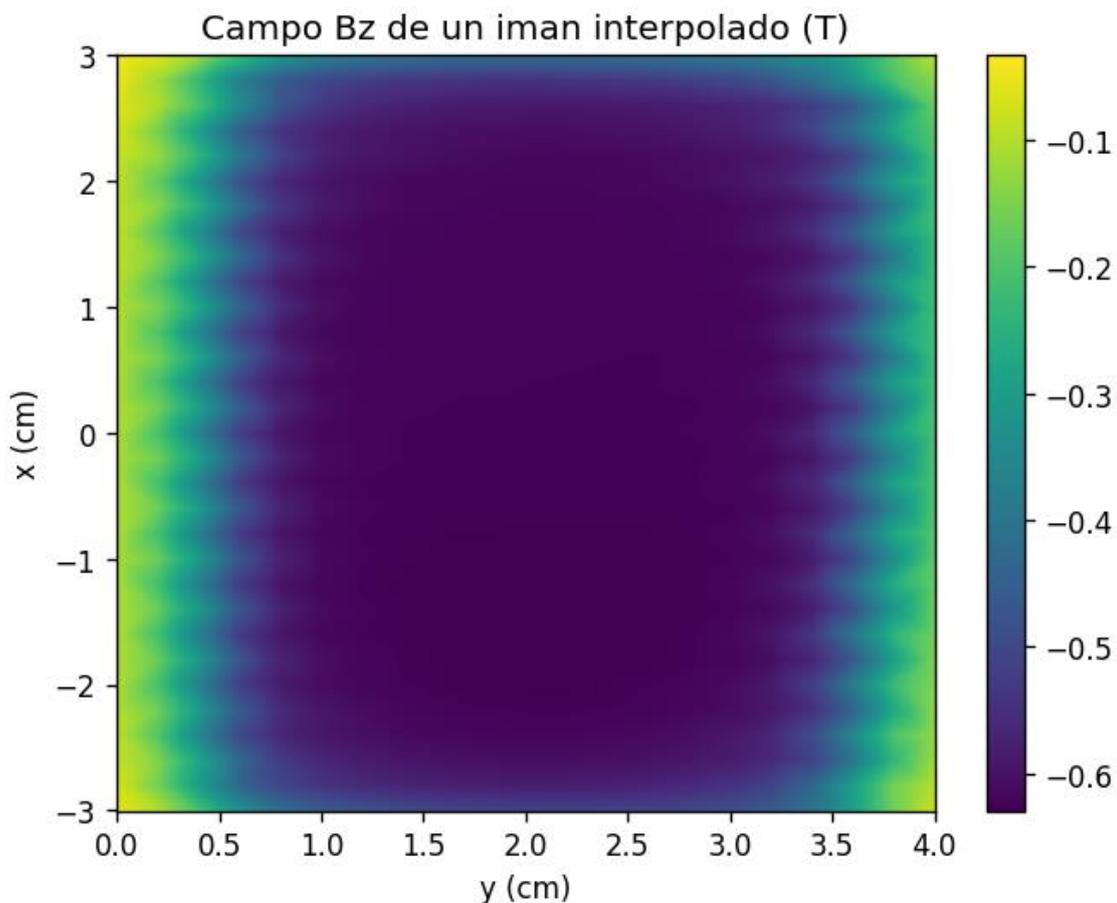


Imagen 27. Mapa del campo magnético de un Imán en la dirección del eje Z.
(Fuente. Elaboración propia a partir del código del "ANEXO II")

Nota. Si observamos con detenimiento la imagen superior es posible observar que el campo magnético no se distribuye de manera simétrica respecto a $x = 2$ cm, pero las diferencias son apreciables en los extremos del intervalo. Por tanto, aunque podría parecer que la medida se encuentra desplazada unos milímetros a la derecha, no se aplicará ninguna corrección puesto que el campo magnético que se tiene sobre la superficie del imán sí que parece estar centrado.

Se desconoce la temperatura a la que se han realizado estas medidas del campo magnético, pero no es necesario tenerla en cuenta para este estudio.

El código desarrollado por Alicia ("ANEXO II") considera un espacio de 1 cm entre imanes de un mismo bloque, por lo que no quedan espacios sin medir (0,5 mm posteriores un imán + 0,5 mm anteriores al siguiente imán = 1 cm), pero en el caso de aumentar o disminuir esta distancia es necesario determinar un método para el cálculo del campo magnético existente en esos huecos. Además, este código no permite variar fácilmente la posición de los imanes en el eje Y al introducirse ésta de manera manual para cada bucle.

Por otro lado, es posible que el campo magnético considerado en esta simulación no sea muy cercano a la realidad, al cumplirse para el campo magnético el principio de superposición. Además, no se está teniendo en cuenta el campo magnético existente en las direcciones X e Y, por lo que sería interesante medirlo para estudiar su influencia, aunque sabemos de antemano que es de una magnitud muy inferior.

En consecuencia, se van a llevar a cabo las siguientes acciones principales:

1. Se creará un código de Python para simular los efectos que tiene la posición de los imanes sobre el comportamiento del haz de protones, considerando un campo magnético nulo entre imanes.
2. Dada la configuración escogida anteriormente, se estudiará la influencia de los huecos considerando también el máximo campo magnético posible entre imanes.
3. Se escogerá una configuración final que será montada en el seleccionador.
4. Con esta configuración, se medirá el campo magnético real existente en el seleccionador en 3D (direcciones X, Y, Z del campo magnético) para volver a simular el comportamiento del haz con los nuevos datos.

8.1. Estudio sobre la influencia de la posición de los imanes en el haz. Configuración óptima para los imanes considerando campo magnético nulo en los huecos.

A continuación, se detallan las siguientes consideraciones a tener en cuenta previamente:

- En estas simulaciones no se tiene en cuenta el campo magnético existente entre imanes.
- El campo magnético que se ha utilizado para efectuar las simulaciones sólo tiene en cuenta la dirección Z.
- La longitud global sobre la que es posible distribuir los imanes del seleccionador en el eje Y es de 37,8 cm.

Adecuación

- La longitud en dirección X del seleccionador cumple de sobra con los requisitos de la separación por lo que no es necesario tenerla en cuenta.
- La separación entre imanes será la misma en ambos bloques y distinta entre bloques.
- Además, para estas simulaciones se tendrá en cuenta un intervalo energético de los protones de entre 4 MeV y 10 MeV con un paso de 1 MeV, que es algo más amplio que el que afectará finalmente al seleccionador, para operar con cierto margen.
- El código que ha permitido desarrollar las simulaciones se recoge en apartado “2. Simulaciones con campo magnético nulo entre imanes” del “ANEXO III”
- Finalmente, el posicionamiento de los imanes sobre el eje X asegura que todos los protones se ven afectados por la influencia de todos los imanes al pasar sobre ellos. En este sentido es interesante que el centro del haz quede centrado en cada imán para asegurar un campo magnético más uniforme, pero, tal y como se aprecia en la imagen superior, el campo magnético es bastante uniforme en toda la superficie del imán. La configuración óptima de los imanes en el eje X se estudiará más adelante.

En primer lugar, se distribuirán los imanes de manera la forma más uniforme posible y sin complicar mucho los cálculos a lo largo del seleccionador. Por ejemplo, con una distancia de 0.8 cm entre imanes de un mismo bloque y 1,4 cm entre bloques (sabiendo que la longitud de los imanes es de 3 cm), se tiene (“Imagen 28”):

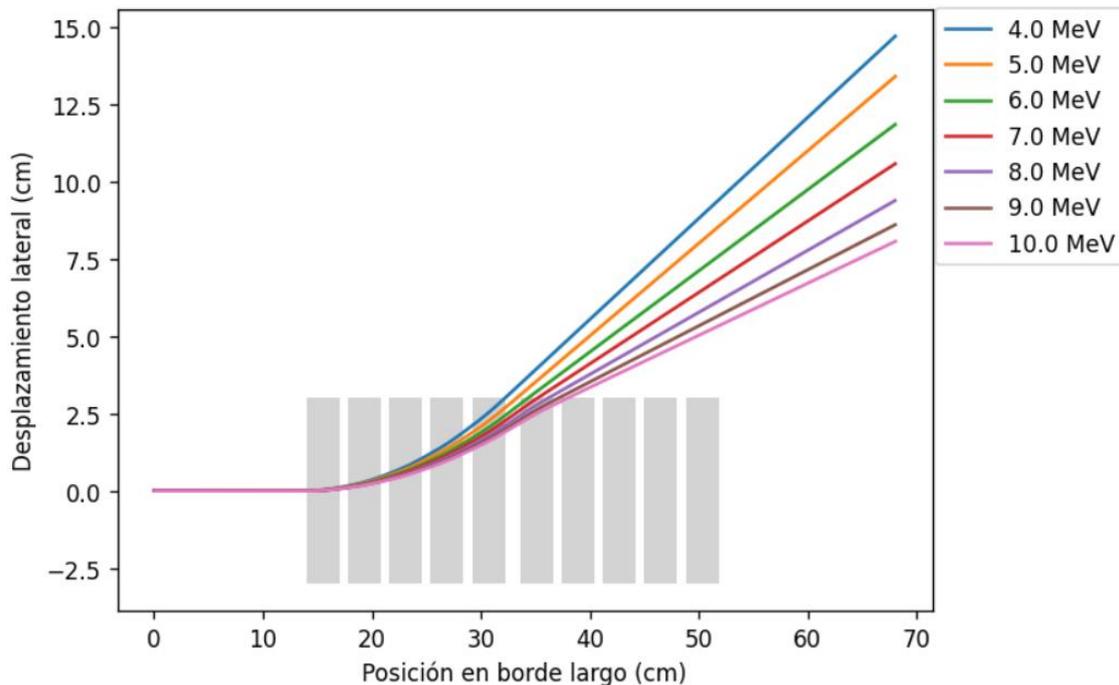


Imagen 28. Trazado de partículas sin campo magnético entre imanes. Con espacios de 0,8 cm entre imanes y 1,4 cm entre bloques, siendo x constante.

(Fuente: Elaboración propia desde “2.1. Trazado de partículas simple” del “ANEXO III”)

Es posible apreciar que los protones no llegan a pasar sobre el segundo bloque de imanes, siendo necesario modificar la posición de los mismos sobre el eje X. Tras varias simulaciones se obtienen los siguientes resultados (“imagen 29”):

Adecuación

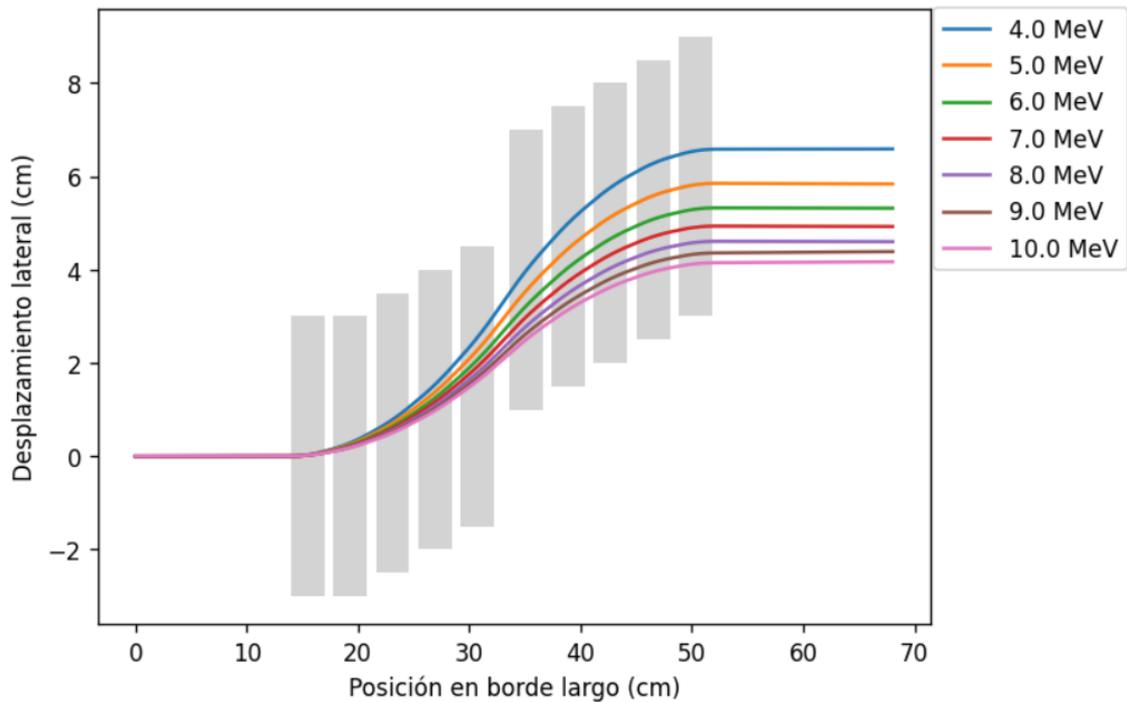


Imagen 29. Trazado de partículas sin campo magnético entre imanes. Con espacios de 0,8 cm entre imanes y 1,4 cm entre bloques.

(Fuente: Elaboración propia desde “2.1. Trazado de partículas simple” del “ANEXO III”)

Tabla 3. Posición de los imanes en el eje X.

POSICIÓN DE CADA IMÁN EN EL EJE X									
IMANES 1 ^{er} BLOQUE					IMANES 2 ^o BLOQUE				
1	2	3	4	5	6	7	8	9	10
0 cm	0 cm	0,5 cm	1 cm	1,5 cm	4 cm	4,5 cm	5 cm	5,5 cm	6 cm

Esta simulación se obtiene un haz más o menos centrado en todos los imanes sobre el eje X, por lo que se pasará ahora a estudiar los efectos de modificar la posición de estos en el eje Y.

Al aplicar el mismo campo magnético, pero de sentido opuesto en el primer y segundo bloque, el haz se separa y se desplaza, pero tiene un ángulo de salida paralelo al de la entrada, por lo que para optimizar la separación es necesario fijarse en la curvatura descrita por los protones desde su entrada a la salida del seleccionador. En este sentido, se observa que, en su paso a través de cada imán, la trayectoria de los protones se desvía debido a la ley de Lorentz, pero lo hacen de manera distinta debido a las diferentes relaciones E/m (siendo m una constante). Por tanto, al final del primer bloque, todos los protones salen con un ángulo, distinto entre ellos, pero igual, sea cual sea la longitud de los huecos, que se compensará en el segundo bloque para salir con el mismo ángulo con el que entraron a seleccionador.

Teniendo esto en cuenta, es posible deducir que:

- A igual espacio entre bloques, el haz se separa más cuanto mayor sea la longitud de los bloques, siendo la separación obtenida siempre mayor cuanto mayor sea la longitud del seleccionador.

Adecuación

- La separación obtenida se maximiza para el máximo espacio entre bloques sea cual sea la distancia del seleccionador.

Para comprobar estas enunciaciones se realizan las siguientes simulaciones:

A continuación, si mantenemos la separación entre bloques (1,4 cm), pero reducimos al máximo la distancia entre imanes (0,5 cm)⁶, la simulación queda ("Imagen 30"):

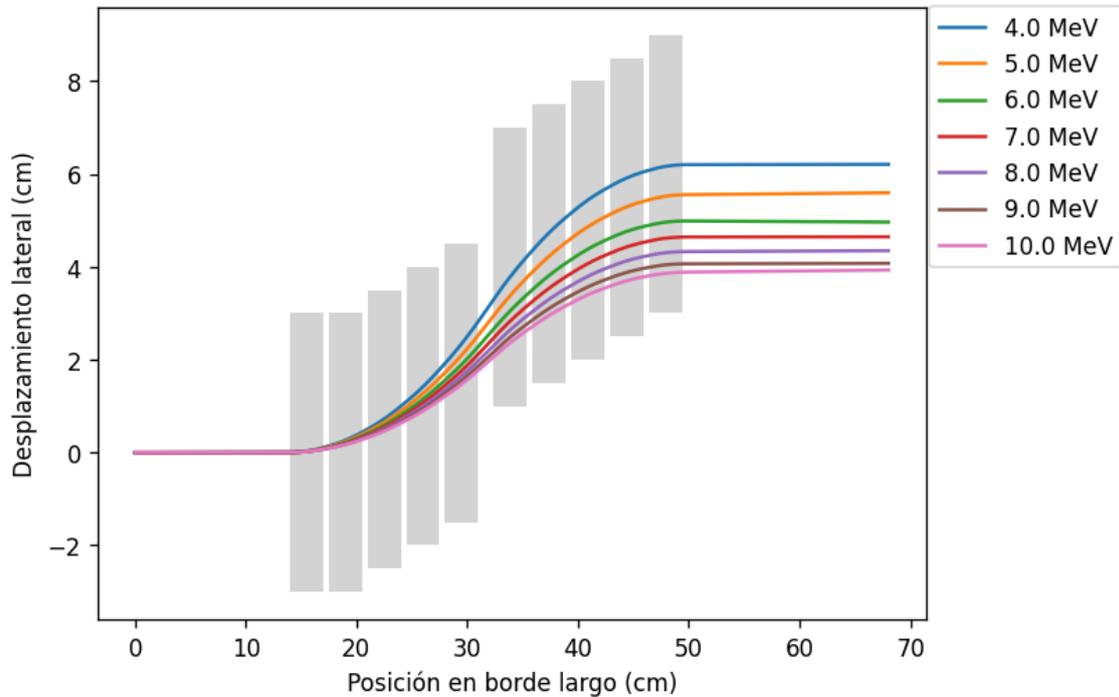


Imagen 30. Trazado de partículas sin campo magnético entre imanes. Con espacios de 0,5 cm entre imanes y 1,4 cm entre bloques.

(Fuente: Elaboración propia desde "2.1. Trazado de partículas simple" del "ANEXO III")

Nota. En esta simulación no se está aprovechando todo el espacio existente en la estructura, quedando la distancia entre el primer y último imán acortada de 37,8 cm a 35,4 cm.

Finalmente, si dejamos estable la distancia entre imanes al mínimo, aumentando la distancia entre bloques al máximo (3,8 cm), teniendo en cuenta la longitud máxima del seleccionador, se tiene ("Imagen 31"):

⁶ Visto en el apartado "2.4. Antecedentes"

Adecuación

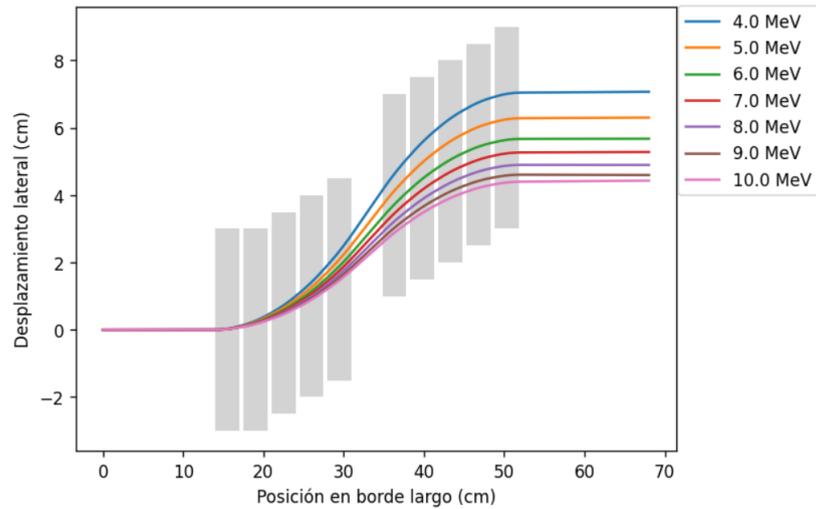


Imagen 31. Trazado de partículas sin campo magnético entre imanes. Con espacios de 0,8 cm entre imanes y 3,8 cm entre bloques.
(Fuente: Elaboración propia desde “2.1. Trazado de partículas simple” del “ANEXO III”)

La posición final de los imanes sobre el eje X en Y=68 cm (longitud escogida al azar para dar con una buena representación de las trayectorias) queda recogida en la siguiente tabla:

Tabla 4. Posición final de los protones en el eje X sin campo magnético entre imanes. Con espacios de 0,8 cm entre imanes y 3,8 cm entre bloques.

ENERGÍAS [MeV]	X_{FINAL} [cm]		
	1 ^a CONFIGURACIÓN	2 ^a CONFIGURACIÓN	3 ^a CONFIGURACIÓN
4	6,59	6,21	7,07
5	5,84	5,59	6,30
6	5,32	4,96	5,67
7	4,92	4,65	5,28
8	4,60	4,35	4,89
9	4,39	4,08	4,60
10	4,17	3,93	4,43
ΔX [cm]	2,42	2,28	2,64

Nota. Separación del haz $\rightarrow \Delta X = X_{FINAL}(4 \text{ MeV}) - X_{FINAL}(10 \text{ MeV})$

Si comparamos los valores de la segunda configuración con los de la primera y la tercera, se observa que la separación aumenta al aumentar la longitud del seleccionador, incluso aunque se mantenga constante la distancia entre bloques. Por otro lado, centrándonos en la primera y tercera configuración, es posible apreciar que, a igual longitud del seleccionador, la separación es mayor cuanto mayor sea la distancia entre bloques. De esta manera es posible verificar las anteriores deducciones.

Adecuación

Como se busca maximizar la separación del haz, nos quedaremos con la tercera configuración para seguir adelante. Además, teniendo en cuenta que estas simulaciones están sobre el límite inferior de desplazamiento en la dirección X del haz, al considerar un campo magnético nulo entre imanes, y el centro del haz se encuentra bastante centrado en los imanes o incluso en la primera mitad de estos, es posible partir de la posición de los imanes en el eje X utilizada en las simulaciones anteriores para el resto de simulaciones.

El campo magnético del selector queda representado en la siguiente imagen:

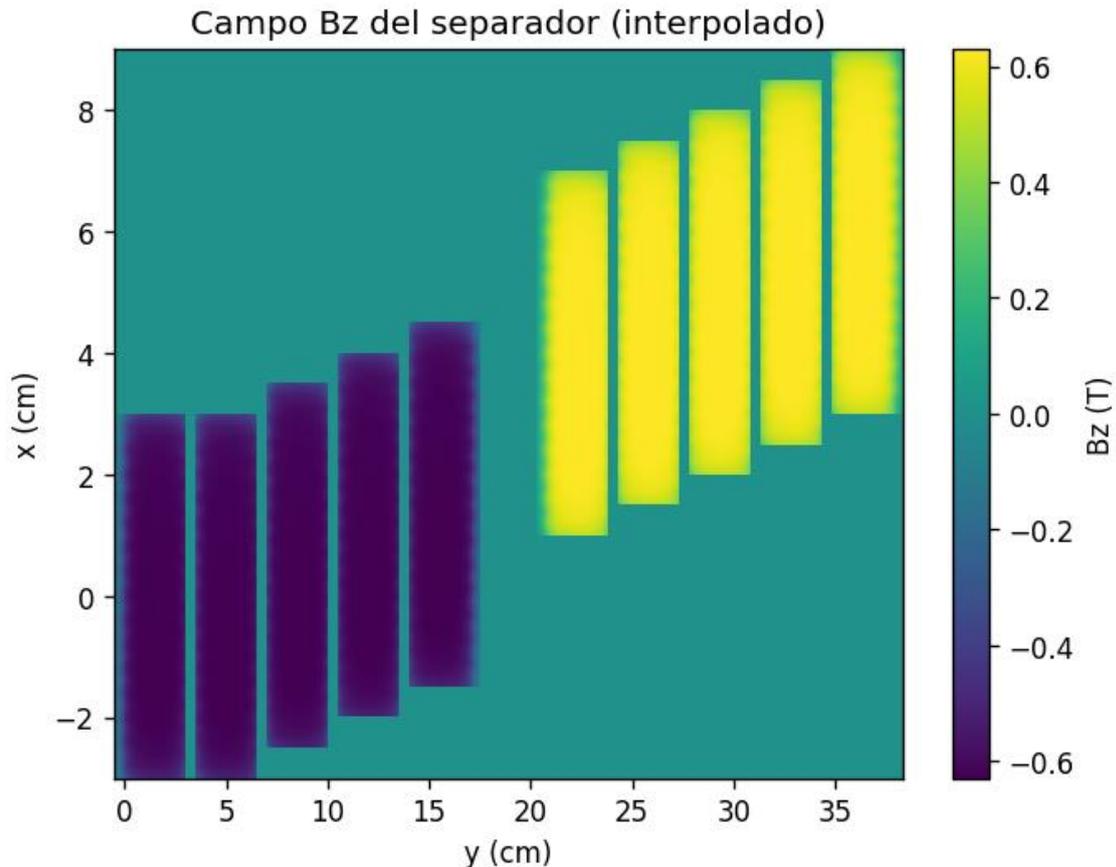
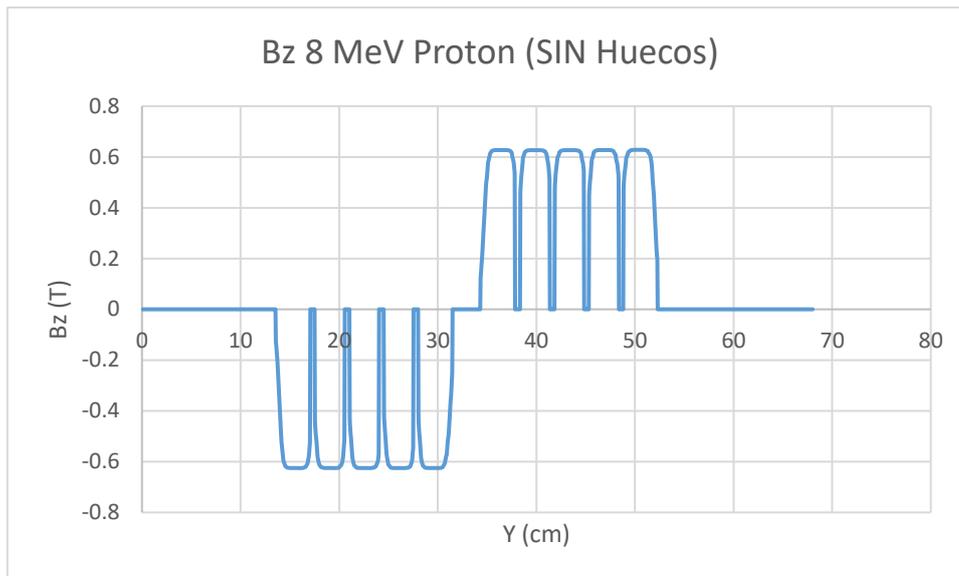


Imagen 32. Mapa magnético de la simulación sin campo magnético entre imanes. Con espacios de 0,8 cm entre imanes y 3,8 cm entre bloques.

(Fuente: Elaboración propia desde "2.2. Mapa magnético de la simulación" del "ANEXO III")

Para poder estudiar mejor el campo magnético que se está aplicando en la simulación, es posible representar el campo magnético que afecta a un protón de, por ejemplo, 8 MeV (Gráfica 1):



Gráfica 1. Campo B_z que afecta a un protón de 8 MeV al atravesar el mapa magnético de la "Imagen 32".

(Fuente: Elaboración propia a partir de los datos que proporciona "2.1. Trazado de partículas simple" del "ANEXO III")

Dado el anterior estudio, que es bastante ilustrativo sobre la influencia de la posición de los imanes en el comportamiento del haz, se estudiará a continuación la influencia de considerar campo magnético en los huecos.

8.2. Estudio del campo magnético en los huecos. Configuración óptima para los imanes considerando el campo magnético simulado en los huecos.

Para asegurar que la configuración anterior es válida, es necesario ahora efectuar algunas simulaciones considerando el campo magnético dado entre imanes de un mismo bloque, o en su defecto, un campo magnético de una magnitud superior a la real para asegurar el funcionamiento del dispositivo en el otro extremo más desfavorable.

Como no se tienen los datos reales, para asignar un valor de campo magnético a los huecos es necesario atender a dos fenómenos:

- Campo magnético de imanes en serie. Debido al principio de superposición del campo magnético, al unir dos imanes en serie el campo magnético resultante es la suma del campo que generan ambos imanes por separado. Realmente es un poco menor debido a que el contacto no es ideal.
- Campo magnético de imanes en paralelo. Los imanes en paralelo generan el mismo campo magnético que por separado, pero en un área mayor.

Dado que en nuestro caso no se tienen imanes ni en serie ni en paralelo al no encontrarse pegados, se ha desarrollado una simulación que utiliza la suma de campos magnéticos en los huecos para calcular las fuerzas que afectarán a las partículas, es decir, se suma el campo magnético que se tiene en el área comprendida por los 5 mm posteriores a cada imán con la que comprende a los 5 mm anteriores al siguiente imán. Se asegura, así, que el campo magnético real será menor que el que se está simulando.

Adecuación

De esta manera se obtiene un campo magnético como el que se muestra a continuación:

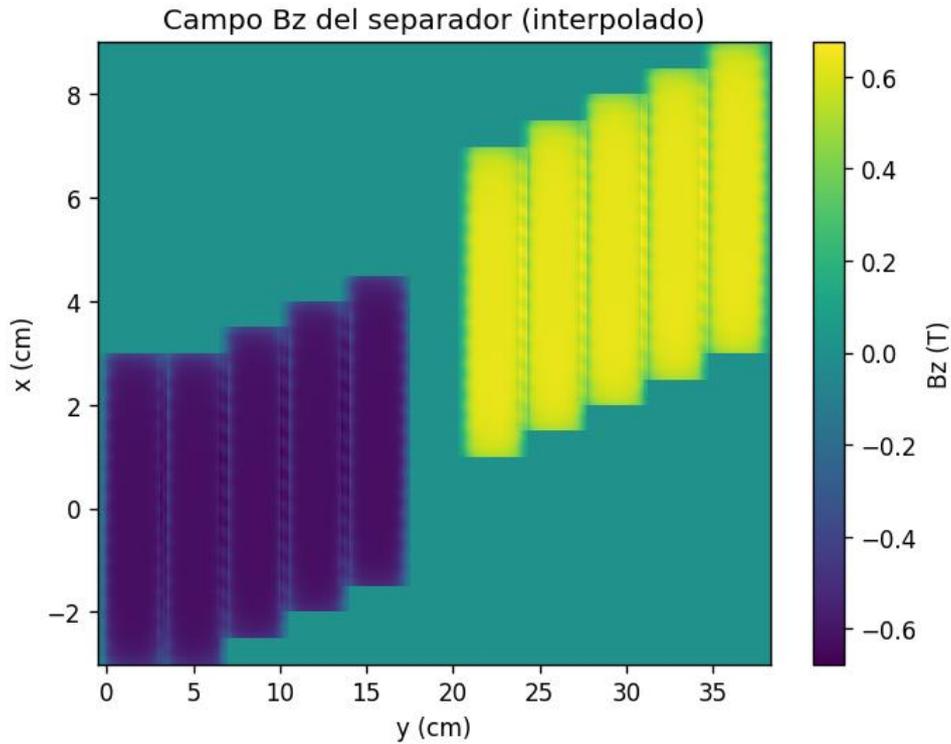


Imagen 33. Mapa magnético de la simulación con suma de campo magnético entre imanes, y espacios de 0,8 cm entre imanes y 3,8 cm entre bloques.
(Fuente: Elaboración propia desde “3.2. Mapa magnético de la simulación” del “ANEXO III”)

Esta consideración del campo magnético permite obtener los siguientes resultados:

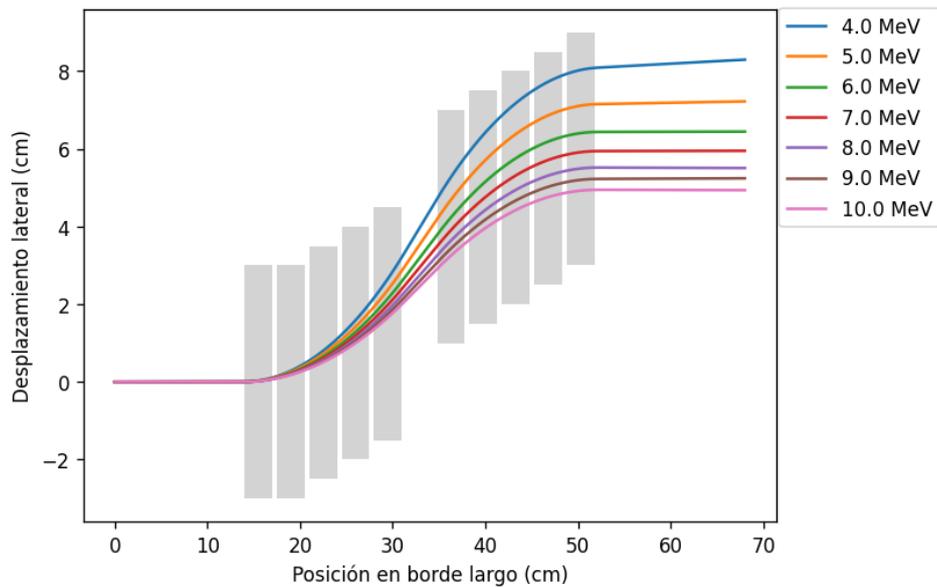


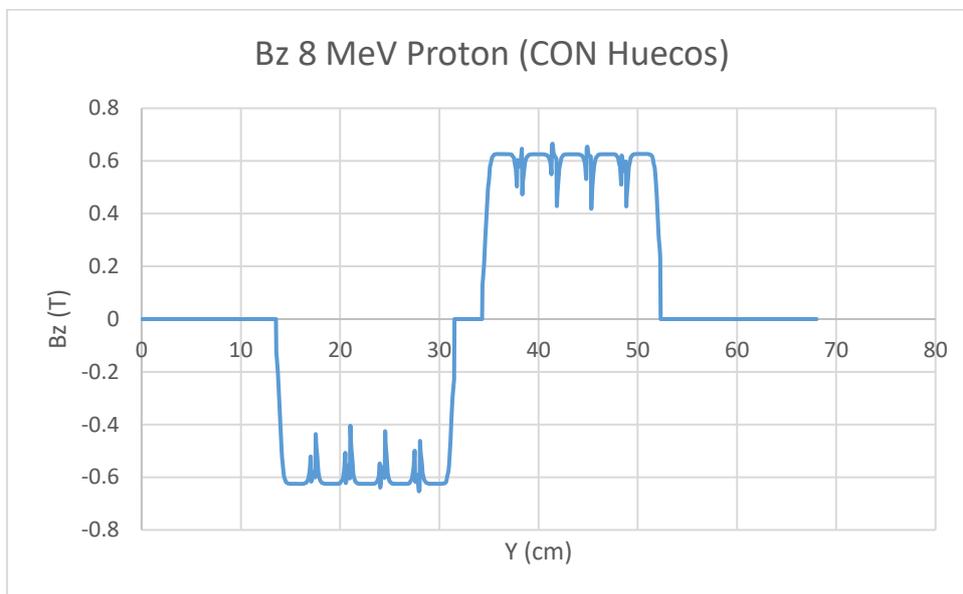
Imagen 34. Mapa magnético de la simulación con suma del campo magnético entre imanes, y espacios de 0,8 cm entre imanes y 3,8 cm entre bloques.
(Fuente: Elaboración propia desde “3.1. Trazado de partículas simple” del “ANEXO III”)

Tabla 5. Posición final de los protones en el eje X con suma de campo magnético entre imanes, y espacios de 0,8 cm entre imanes y 3,8 cm entre bloques.

ENERGÍAS [MeV]	4	5	6	7	8	9	10
X_{FINAL} [cm]	8,30	7,22	6,44	5,95	5,50	5,24	4,94
ΔX [cm]	3,36						

Se observa que, tal y como se esperaba, la trayectoria de los protones se ha desplazado a la parte superior de los imanes, pero no es necesario modificar la posición de estos en el eje X pues se encuentran todavía dentro del límite. Además, la separación conseguida en el haz ha aumentado bastante respecto a la que se tenía considerando campo magnético nulo en los huecos.

Si representamos el campo magnético que afecta a un protón de 8 MeV en esta simulación queda ("Gráfica 2"):



Gráfica 2. Campo B_z que afecta a un protón de 8 MeV al atravesar el mapa magnético de la "Imagen 33".

(Fuente: Elaboración propia a partir de los datos que proporciona "3.1. Trazado de partículas simple" del "ANEXO III")

De esta manera es posible confirmar que el campo magnético real será menor que el que se está aplicando en esta simulación, pues no es posible que en los huecos haya una magnitud del campo magnético mayor que en el centro de los imanes.

Tenidos en cuenta los dos límites del campo magnético y habiendo dado con una configuración válida para ambos, se procederá a montar el seleccionador con la configuración escogida para, posteriormente, medir el campo magnético real.

8.3. Medidas del campo magnético

Una vez montado el seleccionador. Se ha procedido a medir el campo magnético real producido por los imanes con esa configuración. Para ello, otro grupo de investigación del i3m, el MRI (Magnetic Resonance Imaging) se ha dispuesto a prestarnos su sonda y sus instalaciones, por lo que nos hemos trasladado con el seleccionador a su laboratorio.

En este laboratorio, se tiene una cámara acondicionada a una temperatura estable de 18 °C para que el campo magnético se mantenga constante, pues los bloques de neodimio son sensibles a la temperatura. Además, la temperatura es baja ya que a menor temperatura mayor campo magnético, lo que facilita la medida. La variación del campo magnético generado en función de la temperatura se estudiará más adelante.

Para la toma de medidas se utiliza una sonda de efecto Hall que permite tomar lecturas del campo magnético en las tres direcciones. Esta sonda está acoplada al extremo de un robot cartesiano de tres ejes que permite, a partir del giro de unas varillas roscadas acopladas a tres motores, mover la posición de la sonda de manera relativa. El primer motor mueve un carrito que porta al segundo motor y segundo carrito que porta al tercer conjunto. Los motores están conectados a una placa de desarrollo de hardware que los controla a través de un programa de Matlab al que es posible introducirle el desplazamiento en mm que deseamos en cada eje del robot.

En la siguiente imagen es posible observar el laboratorio con los elementos mencionados anteriormente.

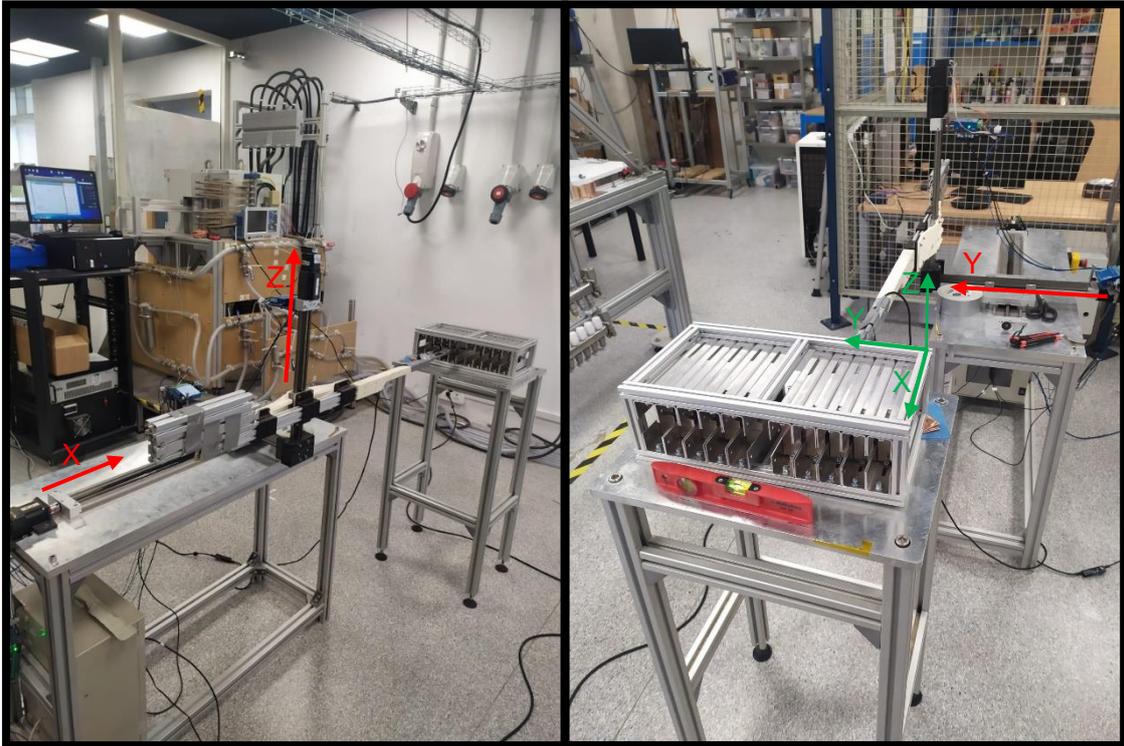


Imagen 35. Montaje del seleccionador para la toma de medidas.
(Fuente: Elaboración propia)

Nota. El sistema de Referencia se ha escogido igual para robot cartesiano y seleccionador.

Adecuación

Antes de comenzar las medidas, se ha calibrado la sonda utilizando un compartimento cero de Gauss o “cero Gauss chamber” (Rodríguez, 2022) como el que se muestra en la imagen inferior, que es un compartimento en el que es posible introducir la sonda para suprimir los campos magnéticos y referenciar la lectura de campo magnético a cero.

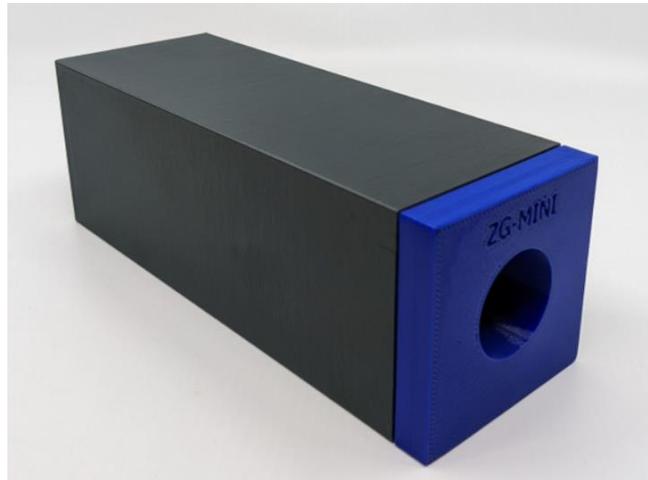


Imagen 36. Compartimento cero de Gauss o “cero Gauss Chamber”.
(Fuente: <https://www.magnetic-shield.com/mini-zero-gauss-chamber/>)

Sería necesario entonces restar el campo magnético terrestre, pero podría decirse que los imanes que se están utilizando son de medio campo, pues tienen una potencia mayor de 0,5 T (Almandoz, 2010). Por tanto, sería posible efectuar las medidas oportunas sin tener en cuenta el campo magnético terrestre, que es de aproximadamente 30000 nT ($\pm 40 \mu\text{T}$ a lo largo del día) teniendo en cuenta las referencias (Ministerio de transportes, 2019) y (Martínez S. C., 2019).

Las medidas se van a efectuar en cuatro lecturas distintas ya que no es posible recorrer toda la superficie a medir en una sola vez debido a las limitaciones morfológicas, pues se desea introducir la sonda en el espacio que separa los polos norte y sur de los imanes y para ello, sólo es posible acceder por un lado en el primer bloque y por el contrario en el segundo. Esto se debe a que los imanes cambian de orientación entre primer y segundo bloque, encontrándose la unión de los polos del yugo de acero en lados opuestos en cada bloque. Además, los pilares verticales de la estructura del seleccionador también imponen restricciones y, aunque se han desmontado los pilares centrales tal y como se aprecia en la siguiente imagen, no es posible desmontar los pilares de las esquinas.

Adecuación

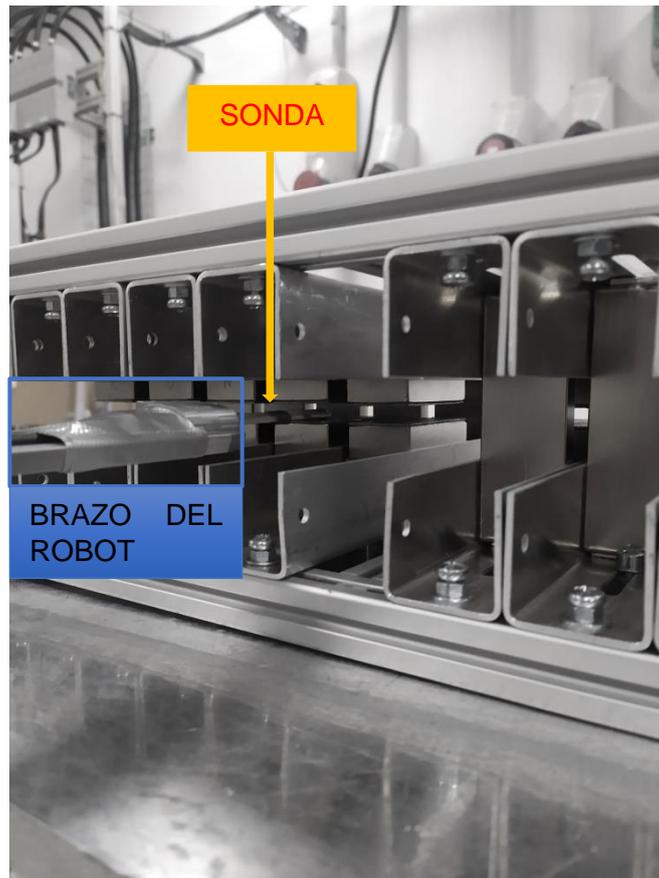


Imagen 37. *Introducción de la sonda en el área a medir.
(Fuente: Elaboración propia)*

Por ello, uno de los pilares de las esquinas de la estructura impide el paso del brazo robot al que va enganchada la sonda. Además, brazo del robot no cabe dentro del espacio a medir por lo que, teniendo en cuenta que el sensor de la sonda se encuentra a 6,5 cm del brazo, sólo podrán medirse por completo las superficies del primer y segundo imán. Las áreas a medir para construir un mapa total del campo magnético quedan representadas en la siguiente imagen:



Imagen 38. Áreas a medir.
(Fuente: Elaboración propia)

Nota. Las flechas indican el acceso de la sonda.

Por otro lado, es necesario nivelar el seleccionador y el robot que porta la sonda para que esta sea capaz de recorrer la superficie a medir sin chocar con nada en las cuatro lecturas a efectuar. Para ello, aunque al principio se utilizó un nivel para nivelar ambas partes en el plano horizontal, resulta imposible hacer que la sonda cubra el área de medida sin chocarse a partir de la nivelación de la mesa que porta el robot y la estructura del seleccionador. Este fenómeno se debe a que se produce una pérdida de la horizontalidad en el carrito que mueve la sonda en el eje X a medida que aumentan o disminuyen los momentos además de una flexión en la viga que desplaza la sonda en el eje Y a medida que aumenta el voladizo. En las siguientes imágenes se explica de manera más detallada lo expuesto anteriormente:

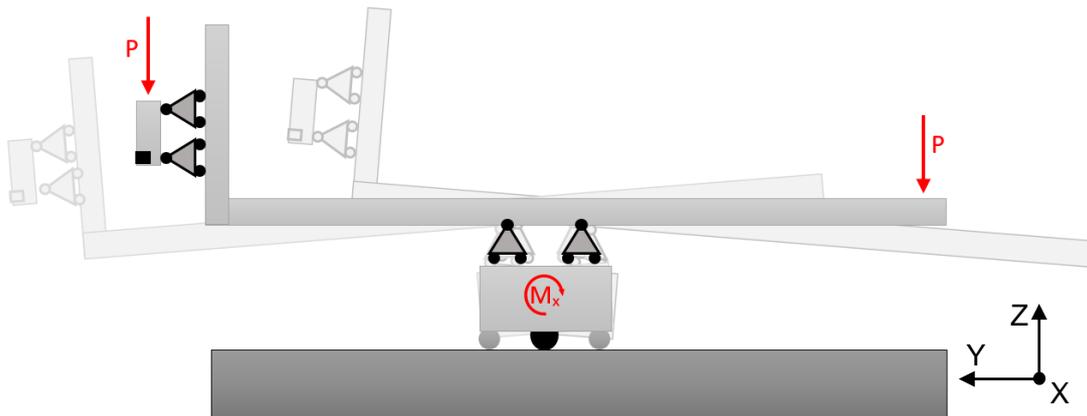


Imagen 39. Inclinación o pérdida de la horizontalidad del carrito debido al momento que genera el desequilibrio de las fuerzas peso.
(Fuente: Elaboración propia)

Adecuación

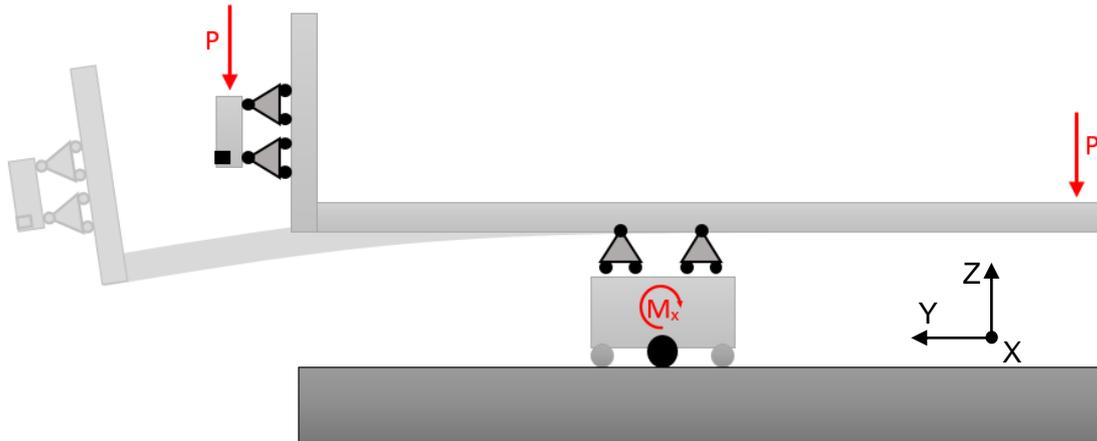


Imagen 40. Flexión de la viga debido al aumento del voladizo (y del momento flector).
(Fuente: Elaboración propia)

Por ello, se llevaron a cabo las siguientes acciones:

- En lugar de situar el centro del carrito en el centro del intervalo de movimiento, se ha decidido situarlo de manera que ya al inicio del movimiento se tenga un desequilibrio del peso en favor del lado en el que aumentará ese desequilibrio a medida que el robot complete su ruta. De esta forma se disminuye la holgura del carrito producida por los momentos que tienen lugar en la dirección del eje X puesto que, en lugar de pasar de momento positivo a negativo, se empieza con un momento negativo que aumenta a medida que el brazo se desplaza, lo que hace que el carrito se incline únicamente hacia un lado y hasta alcanzar el límite de inclinación.

Para que el procedimiento sea reproducible, se ha decidido situar la sonda en su límite de movimiento en el eje Y para luego desplazarla en sentido inverso el total de la distancia a medir, 18cm, más 0,2 cm de margen. De esta manera, aunque es difícil de medir la variación del ángulo de inclinación, se asegura que siempre será positivo.

- Se ha añadido una tira de cinta adhesiva que una los extremos libres de los brazos que llevan la dirección de los ejes Y y Z para formar un triángulo que permita reducir la flexión del brazo que porta la dirección del eje Y. No obstante, el triángulo que se forma no funciona de la misma manera en que lo hacen las estructuras articuladas. En este caso, la tira de cinta únicamente transmite esfuerzos de tracción, pero se reduce la flexión al contar con un ángulo rígido entre los brazos, tal y como se puede apreciar en la siguiente imagen:

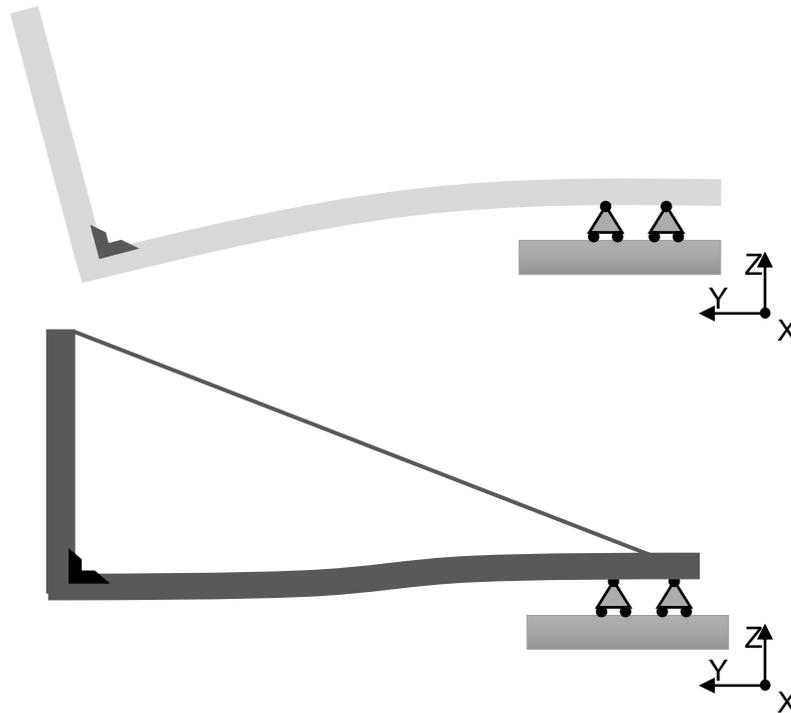


Imagen 41. Reducción de la flexión del brazo que lleva la dirección del eje Y a partir de una tira de cinta adhesiva.
(Fuente: Elaboración propia)

Se obtiene así una viga empotrada con giro nulo en el extremo del voladizo, reduciéndose así la flecha máxima.

- Además, el selector y el banco que porta al robot se han nivelado con ángulos contrarios para impedir que se bloquee la sonda, siendo el del banco de 2° y el de la sonda de -3° :



Imagen 42. Representación del recorrido de la sonda a través de la dirección del eje Y de la superficie a medir.
(Fuente: Elaboración propia)

A partir de las soluciones llevadas a cabo, es posible comenzar las medidas. Las primeras medidas fueron realizadas sobre el segundo bloque de manera manual, introduciendo directamente los steps de posición al motor y apuntando los resultados en un Excel. De esta manera se observó que el campo magnético no es nada estable en los bordes y espacios entre imanes, donde tiene menor magnitud y varía rápidamente entre un rango más o menos estable, pero es más fácil de leer en el centro de los imanes. Por su parte, el sentido del campo magnético en las direcciones de los ejes X e Y se estudiará más adelante en el siguiente punto.

Adecuación

Los intervalos para la toma de datos no han sido constantes si no que se ha ido moviendo la sonda con intervalos más grandes o más pequeños dependiendo del lugar de la medida y la precisión requerida formando algo así como una malla adaptativa. De esta forma, tras más de 160 medidas manuales, no se ha conseguido una gran resolución pues el tamaño de malla es demasiado grande, tal y como es posible apreciar en las siguientes imágenes:

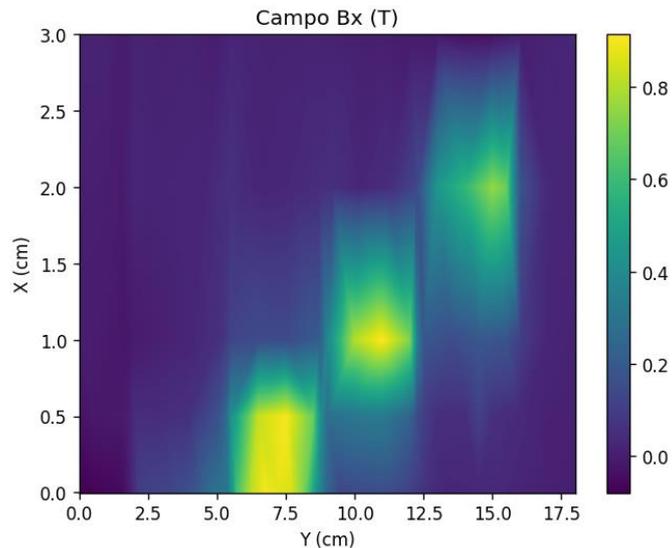


Imagen 43. Campo B_x medido manualmente.

(Fuente: Elaboración propia a partir del código recogido en "2. Mapas magnéticos" del "ANEXO IV")

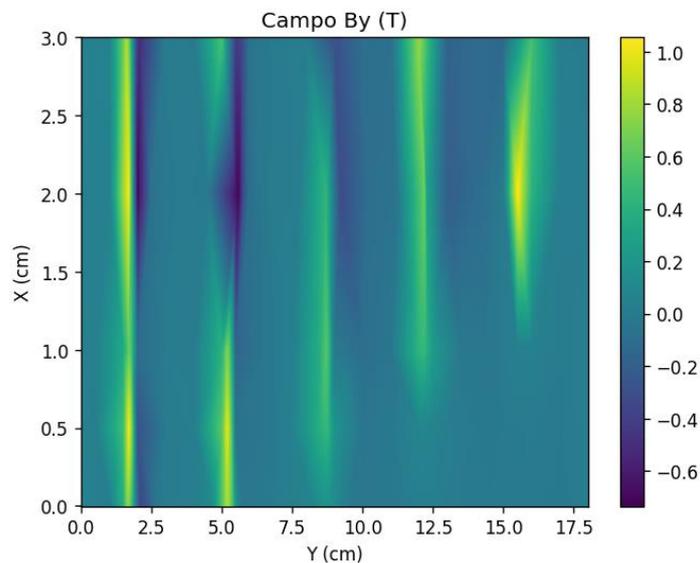


Imagen 44. Campo B_y medido manualmente.

(Fuente: Elaboración propia a partir del código recogido en "2. Mapas magnéticos" del "ANEXO IV")

Adecuación

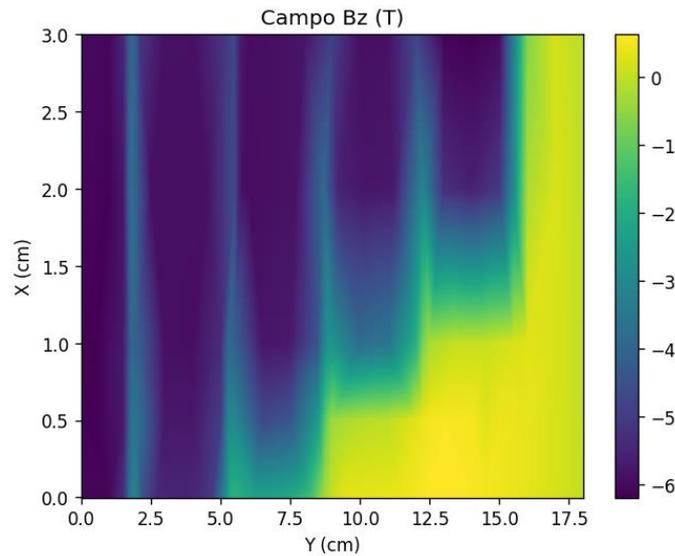


Imagen 45. Campo Bz medido manualmente.

(Fuente: Elaboración propia a partir del código recogido en “2. Mapas magnéticos” del “ANEXO IV”)

No obstante, y gracias a la ayuda de Pepe Borreguero Morata (doctorando), se consiguió adaptar un programa de Matlab que utilizan los integrantes del equipo de investigación que nos prestó sus instalaciones a las características de nuestra ruta, ya que ellos mapean volúmenes cilíndricos, al ser un grupo que se dedica a la resonancia magnética, y se necesitaba medir un plano por nuestra parte.

Por tanto, se procedió a mapear el campo magnético de forma automática introduciéndole al robot que desplaza la sonda los puntos inicial y final, y los intervalos de parada para la toma de medidas en cada dirección. Para empezar, se utilizó la siguiente ruta:

Tabla 6. Ruta de medida del segundo bloque con paso estrecho.

	MÍNIMO	MÁXIMO	INTERVALO
Y (cm)	0	18	0.1
X (cm)	0	7	0.25

*Tiempo de Parada = 3 segundos

Con esta ruta se pudo mapear con bastante precisión los imanes que se encuentran a la salida del seleccionador tal y como se puede ver en las siguientes imágenes:

Adecuación

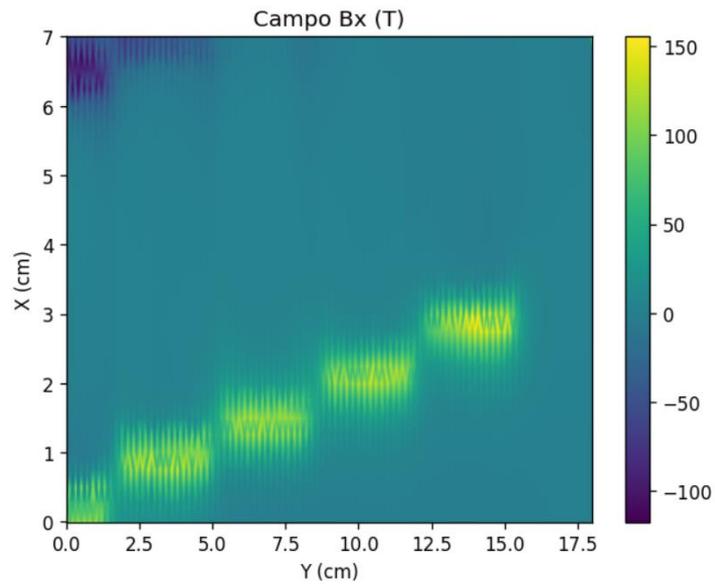


Imagen 46. Campo Bx medido sobre el segundo bloque según la ruta de la "Tabla 4".
(Fuente: Elaboración propia a partir del código recogido en "2. Mapas magnéticos" del "ANEXO IV")

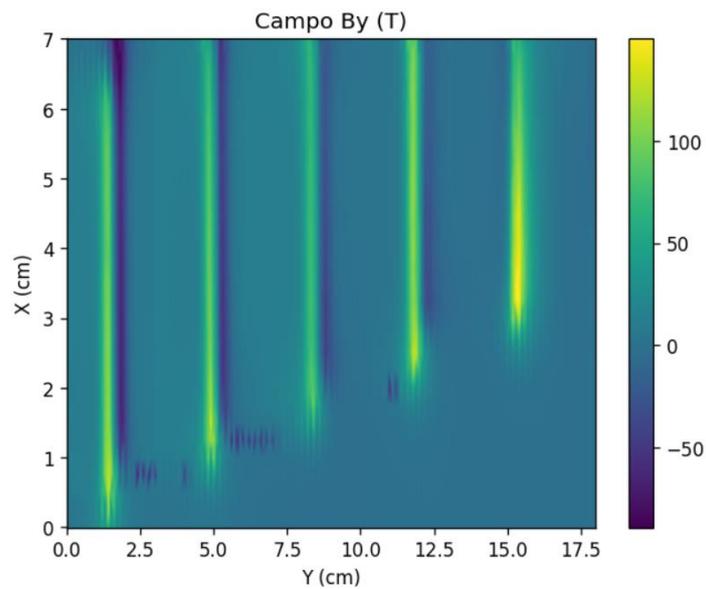


Imagen 47. Campo By medido sobre el segundo bloque según la ruta de la "Tabla 4".
(Fuente: Elaboración propia a partir del código recogido en "2. Mapas magnéticos" del "ANEXO IV")

Adecuación

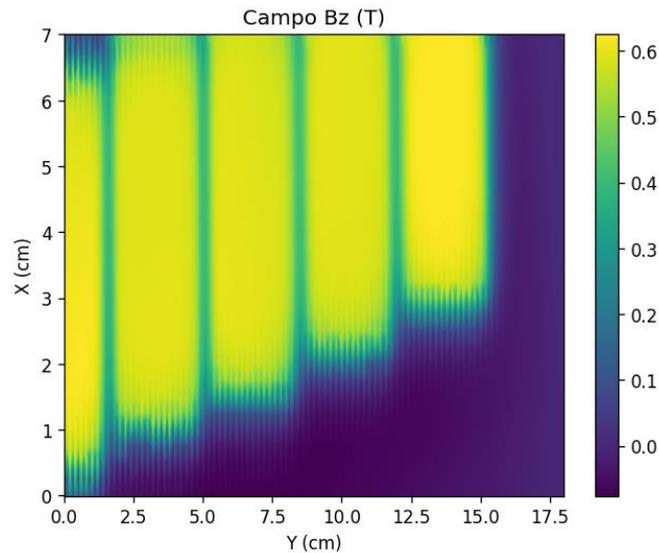


Imagen 48. Campo Bz medido sobre el segundo bloque según la ruta de la “Tabla 4”.
(Fuente: Elaboración propia a partir del código recogido en “2. Mapas magnéticos” del “ANEXO IV”)

Sin embargo, la medida duró 6 horas aproximadamente, así que, para comprobar los resultados, se decidió utilizar la siguiente ruta:

Tabla 7. Ruta de medida con paso ancho.

	MÍNIMO	MÁXIMO	INTERVALO
Y (cm)	0	18	0.1
X (cm)	0	7	1

*Tiempo de Parada = 3 segundos

Con esta ruta se obtuvo una medición igual de precisa en la dirección Y, pero con muy baja resolución en X, aunque es interesante para comprobar que los resultados obtenidos anteriormente estaban bien.

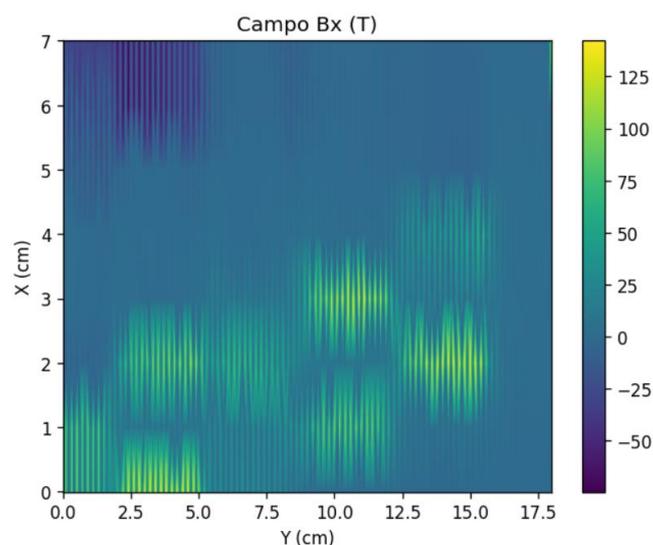


Imagen 49. Campo Bx medido sobre el segundo bloque según la ruta de la “Tabla 5”.
(Fuente: Elaboración propia a partir del código recogido en “2. Mapas magnéticos” del “ANEXO IV”)

Adecuación

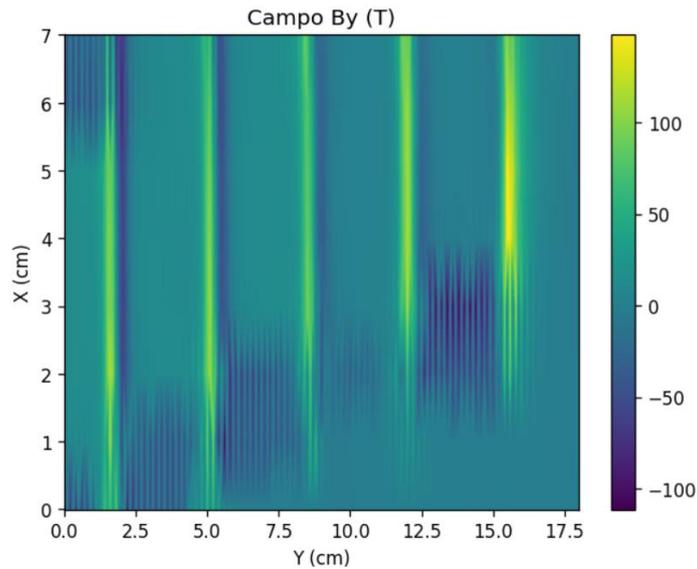


Imagen 50 Campo B_y medido sobre el segundo bloque según la ruta de la "Tabla 5".
(Fuente: Elaboración propia a partir del código recogido en "2. Mapas magnéticos" del "ANEXO IV")

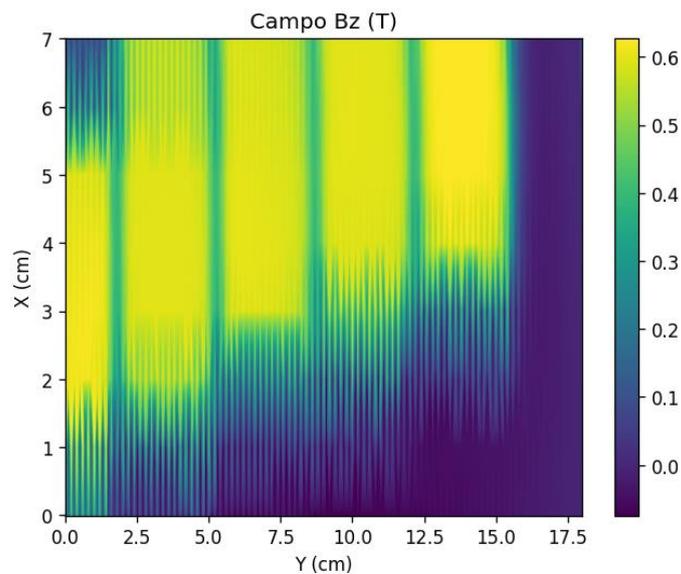


Imagen 51 Campo B_z medido sobre el segundo bloque según la ruta de la "Tabla 5".
(Fuente: Elaboración propia a partir del código recogido en "2. Mapas magnéticos" del "ANEXO IV")

Dando las primeras mediciones por válidas se procedió a medir entonces el primer bloque de imanes, tras nivelar el seleccionador adecuadamente. Como no se dispone indefinidamente del robot y la sonda, para el mapeado del campo magnético en el primer bloque se ha optado por el empleo de la anterior ruta ("Tabla 5"). No obstante, aunque no se requiere de una alta precisión del campo magnético en la dirección del eje X, las medidas que proporciona la ruta con paso ancho en dicho eje pueden trastocar el campo magnético que afecta a las partículas que pasan por los extremos del imán en las simulaciones, al no definir claramente el comienzo y final de los imanes en el eje X a partir de la interpolación de las medidas, tal y como se puede apreciar en la "Imagen 52".

Adecuación

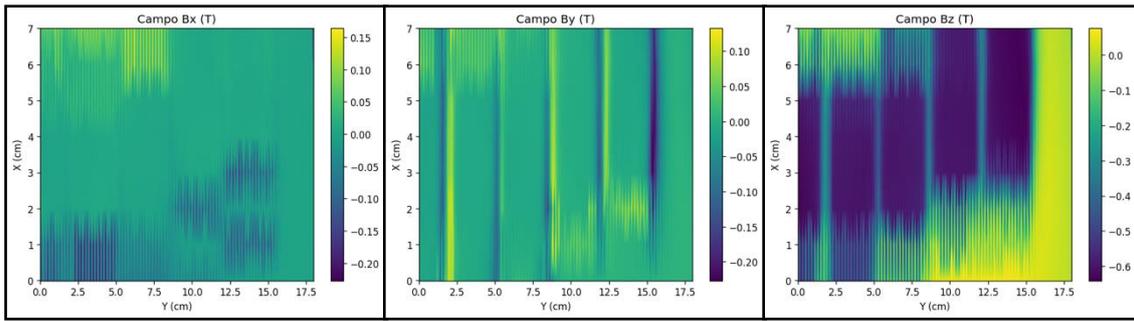


Imagen 52. Medidas del campo magnético sobre el primer bloque según la ruta de la “Tabla 5”.

(Fuente: Elaboración propia a partir del código recogido en “2. Mapas magnéticos” del “ANEXO IV”)

Por ello, se ha vuelto a medir el campo utilizando una ruta con paso estrecho muy similar a la primera, pero con una parada de un segundo en lugar de los tres segundos que se ha utilizado en las anteriores medidas.

La ruta queda (“Tabla 6”):

Tabla 8. Ruta de medida del segundo bloque con paso ancho.

	MÍNIMO	MÁXIMO	INTERVALO
Y (cm)	0	18	0.1
X (cm)	0	7	0.2

*Tiempo de Parada = 3 segundos

Para las anteriores lecturas se obtienen los siguientes mapas magnéticos (“Imagen 53”):

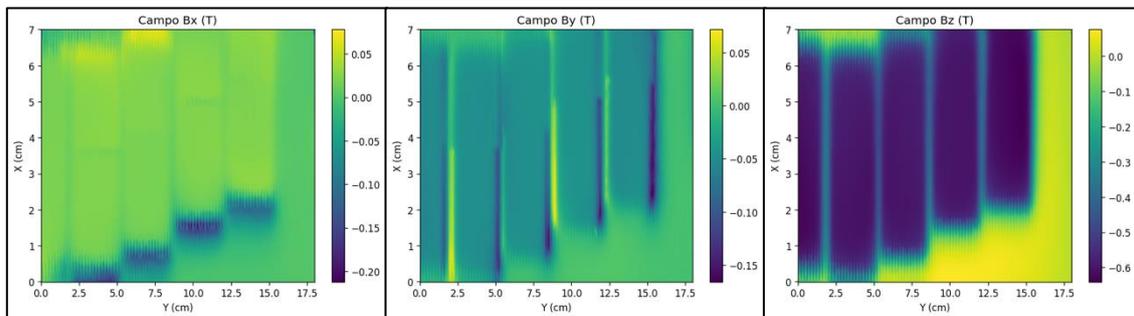


Imagen 53 Medidas del campo magnético sobre el primer bloque según la ruta de la “Tabla 6”.

(Fuente: Elaboración propia a partir del código recogido en “2. Mapas magnéticos” del “ANEXO IV”)

En esta “Imagen 53” es posible apreciar que hay fallos en las lecturas del campo magnético con paso estrecho, pues la intensidad del campo magnético en las direcciones X e Y presente sobre la superficie de los imanes debería ser nula. Esto puede deberse a errores en la medida pues al ser la parada de un segundo, la sonda no está totalmente quieta cuando se efectúa la medida, pues el brazo oscila debido a las inercias generadas por los movimientos de aceleración y deceleración al no tratarse de un cuerpo con elevada rigidez.

Adecuación

La parte que quedaba por medir del primer y segundo imán se ha medido empleando la siguiente ruta que se recoge en la (“Tabla 7”)

Tabla 9. Ruta de medida para la parte de los imanes extremos que queda por medir.

	MÍNIMO	MÁXIMO	INTERVALO
Y (cm)	0	25	0.1
X (cm)	0	85	0.2

Nota. Tiempo de Parada = 1 segundos

Los valores de X e Y estarían intercambiados al introducir la sonda a 90° respecto las medidas anteriores, pero aquí ya aparecen corregidos.

8.4. Simulaciones efectuadas con el campo magnético medido

Tras la toma de medidas, se procede a estudiar el efecto del campo magnético medido en el comportamiento del haz de protones. Para ello, es necesario estudiar por un lado los efectos del campo magnético en la dirección Z, que afecta a la desviación horizontal tal y como se vio anteriormente y, por otro lado, los efectos que producen los campos magnéticos que portan las direcciones de los ejes X e Y, que según la Ley de Lorentz, afectarán a la desviación vertical.

8.4.1. Construcción del mapa magnético del campo BZ

Con las lecturas de la intensidad del campo magnético de los dos bloques de imanes se obtiene la siguiente representación:

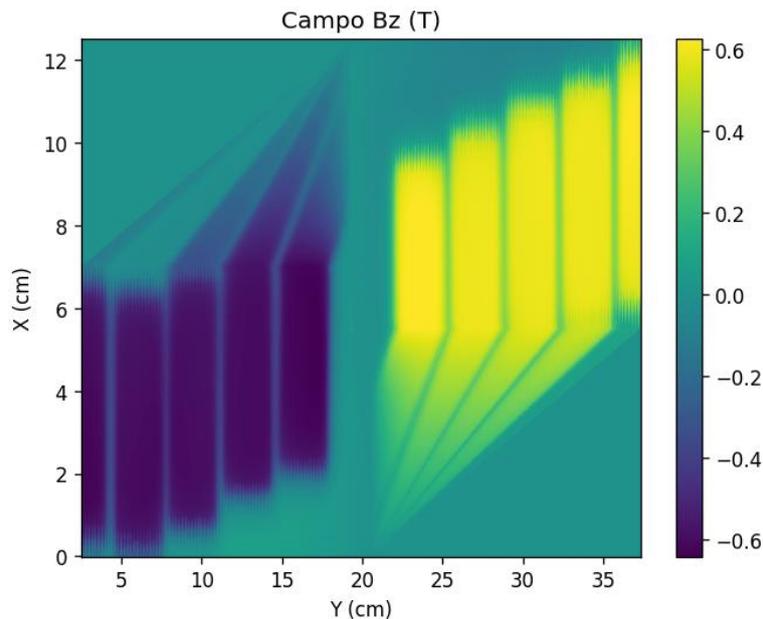


Imagen 54. Mapa magnético construido a partir de las medidas del campo magnético con paso estrecho realizadas para los 2 bloques del seleccionador.
(Fuente: Elaboración propia desde “4.1. Mapas magnéticos” del “ANEXO IV”)

Adecuación

No obstante, es posible añadir a los archivos de datos de ambos bloques los valores del campo magnético que se han medido para las superficies que faltan del primer y último imán y, utilizando el script del apartado “3. Extracción de datos” del “ANEXO IV” es posible copiar los datos del campo magnético que se tienen para una zona de un imán a los lugares donde se necesitan datos tal y como se muestra a continuación “(Imagen 55”):

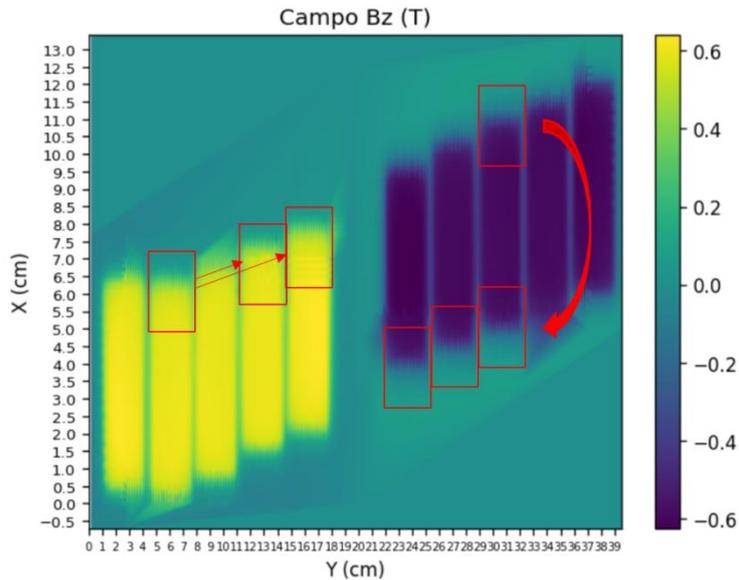


Imagen 55. Campo B_z total del seleccionador obtenido copiando medidas de un único imán en cada bloque.

(Fuente: Elaboración propia a partir de los datos extraídos mediante el código de “3. Extracción de datos” y la representación que permite el código del apartado “4.1. Mapas magnéticos” del “ANEXO IV”)

Nota. El sentido del campo magnético está invertido, pero no afecta al estudio que se está realizando.

Cabe destacar que al leer datos y trasladarlos no se realiza con la suficiente precisión como para no sobrescribir puntos con medidas distintas del campo magnético, pero en aquellos puntos donde existen dos lecturas distintas, el código opera con la media de ambas.

Sin embargo, se aprecian mejores resultados cuando para construir el mapa se copian los valores que se han medido de un imán al extremo sin medir de ese mismo imán:

Adecuación

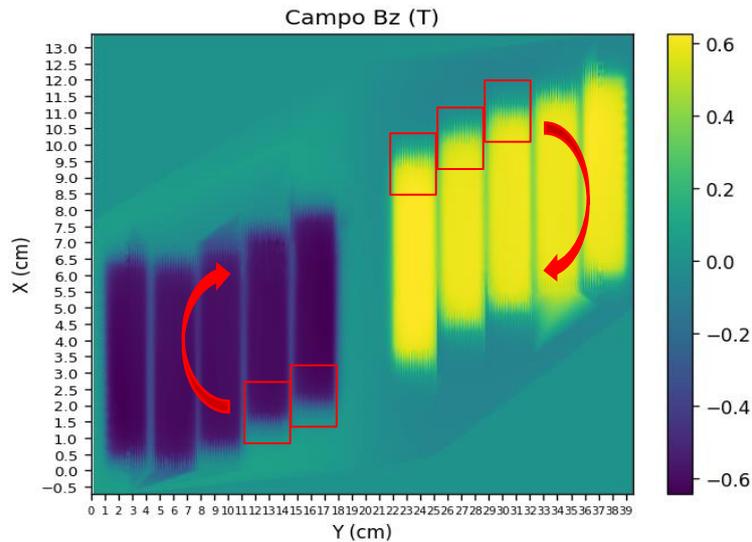


Imagen 56. Campo B_z total del seleccionador obtenido copiando medidas de cada imán a completar.

(Fuente: Elaboración propia a partir de los datos extraídos mediante el código de “3. Extracción de datos” y la representación que permite el código del apartado “4.1. Mapas magnéticos” del “ANEXO IV”)

Estas diferencias obtenidas entre ambos métodos pueden deberse a las pequeñas variaciones del campo magnético que tienen lugar entre distintos imanes o la falta de precisión en la traslación de los datos a lo largo del eje Y que es necesaria si se utiliza el primer método, pero no requiere atención al utilizar el segundo.

Con este mapa del campo magnético es posible obtener finalmente la siguiente simulación:

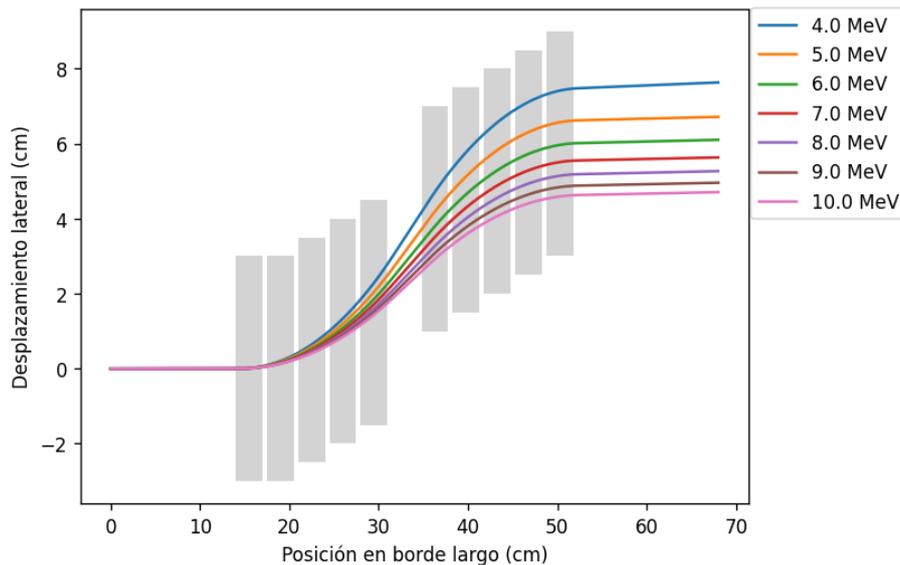


Imagen 57. Trazado de partículas a partir del campo magnético medido.
(Fuente: Elaboración propia desde “4.2. Simulación Plano XY ” del “ANEXO IV”)

Tabla 10. Posición final de los protones en el eje X según simulación de "Imagen 57".

ENERGÍAS [MeV]	4	5	6	7	8	9	10
X_{FINAL} [cm]	7,63	6,72	6,10	5,63	5,27	4,96	4,71
ΔX [cm]	2,95						

Se observa que la simulación que no contempla campo magnético en los huecos está más cerca que la que utiliza la suma de campos magnéticos. No obstante, ninguna de las dos da como resultado una buena aproximación.

8.4.2. Construcción de los mapas magnéticos de los campos B_x Y B_y . Efectos de B_y en el comportamiento del haz.

En primera instancia, es necesario determinar el sentido del campo magnético en los bordes del imán. Para explicarlo se tiene la siguiente imagen:

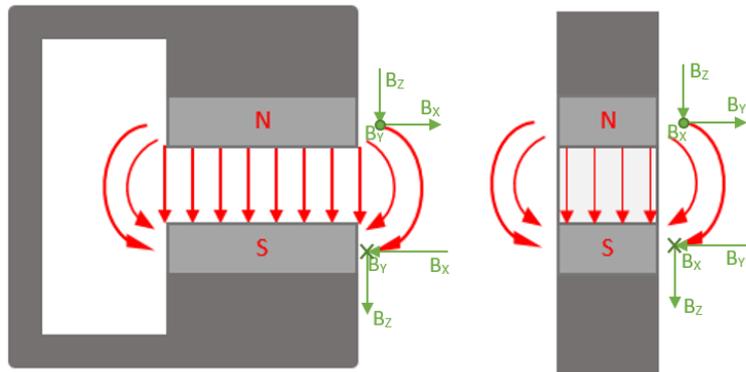


Imagen 58. Descomposición vectorial del campo magnético en los bordes del imán. (Fuente: Elaboración propia)

Atendiendo a la anterior representación, es posible enunciar que el sentido del campo magnético en las direcciones de los ejes X e Y no sólo depende del lado del imán en el que nos encontremos y la dirección del campo magnético en el eje Z, sino que también es dependiente del polo sobre el que se esté efectuando la medida, siendo la medida sobre el centro entre ambos polos igual 0. Por tanto, es posible deducir que la sonda no se ha desplazado justo por el centro entre polos de los imanes, pero para determinar si se ha medido por arriba o por debajo del centro, es necesario conocer que el sensor de la sonda se encuentra en la parte inferior de esta. Es por esto que es más probable que la toma de medidas se haya realizado por debajo del centro del hueco que separa los polos opuestos de los imanes, pero es muy difícil determinarlo con exactitud pues como es posible deducir de las exposiciones realizadas en el apartado de toma de medidas, se carece de la precisión suficiente para lograr que la sonda recorra toda el área a mapear a una distancia dada de cada polo de los imanes.

Teniendo esto en cuenta, para la realización de las simulaciones será necesario efectuar algunas correcciones respecto del campo magnético medido en las direcciones X e Y. En primera instancia, no es posible considerar el campo magnético medido en ambas direcciones para el primer bloque de imanes con paso fino porque, como se estudió

Adecuación

anteriormente, la superficie de los imanes presenta valores de campo magnético distintos de cero, lo cual es incorrecto.

Por tanto, las correcciones a implementar son las siguientes:

- Primeramente, dado que el campo magnético en la dirección X que afectará a los protones que pasan por la superficie del imán es nulo, se considera $B_x = 0$ para toda el área del seleccionador.
- Por otro lado, atendiendo a B_y aunque sería posible utilizar el campo magnético medido para el primer bloque con paso gordo, se replicarán los valores del campo magnético del segundo bloque en el primero. Esto es posible porque todas las partículas pasan por el campo magnético generado por todos los imanes tal y como se muestra a continuación:

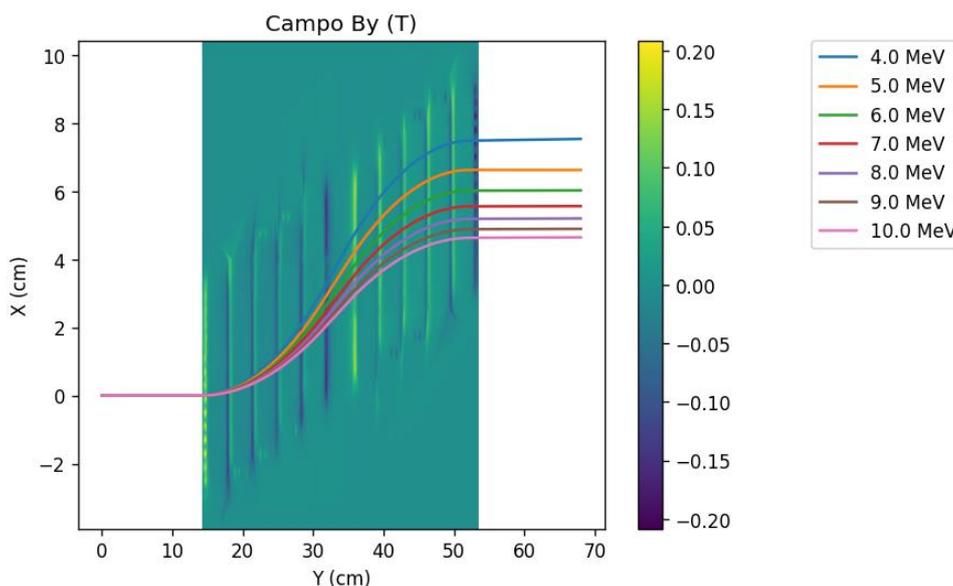


Imagen 59. Superposición (trazado de partículas) – (campo B_y medido).
(Fuente: Elaboración propia desde “5. Superposición trayectorias- B_y ” del “ANEXO IV”)

Se ha escogido este método pues el campo magnético medido con paso estrecho para el segundo bloque es el que mejor resolución tiene.

Una vez escogida la magnitud de la intensidad del campo magnético en cada punto, es necesario pasar a estudiar su sentido. Para ello es necesario tener en cuenta que, si por ejemplo tenemos una partícula que sale de la fuente un poco por encima del centro entre ambos polos de los imanes, en su paso por el primer bloque de imanes atravesará el siguiente campo magnético (“Imagen 60”):

Adecuación

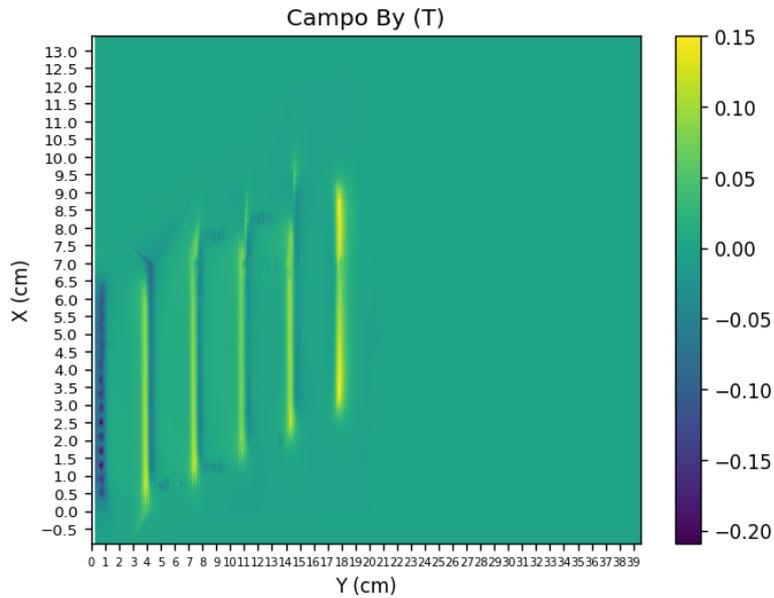


Imagen 60. Campo B_y medido sobre el primer bloque.
(Fuente: Elaboración propia desde “4.1. Mapas magnéticos” del “ANEXO IV”)

Este campo magnético desviaría a los imanes hacia abajo tal y como se muestra en la siguiente imagen:

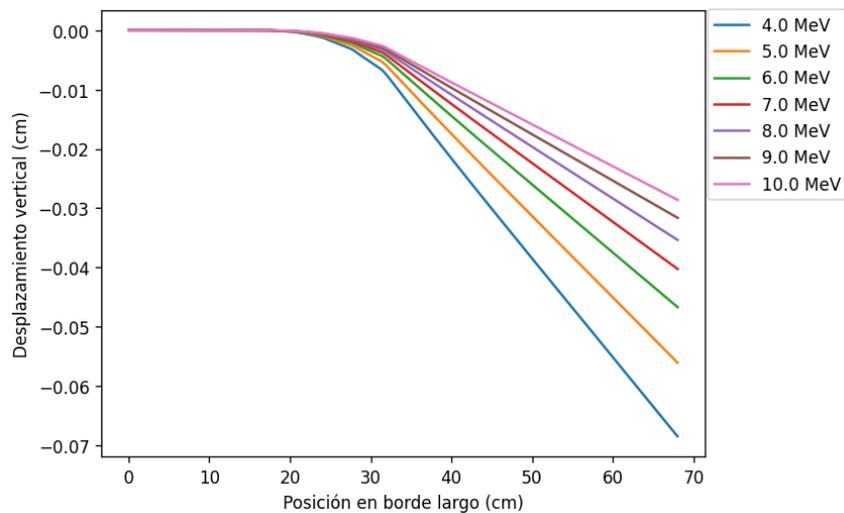


Imagen 61. Trazado de partículas en el plano YZ considerando el campo magnético que aparece en la “Imagen 60”.
(Fuente: Elaboración propia desde “4.3. Simulación Plano YZ” del “ANEXO IV”)

Por lo tanto, en el siguiente bloque de imanes los protones podrían verse afectados 2 campos distintos en función de si pasan a la mitad inferior del hueco o no, pudiéndose distinguir así entre 2 casos distintos:

Adecuación

1. Los protones parten de la mitad superior del hueco entre polos opuestos de los imanes y se desplazan a la mitad inferior en el segundo bloque "Imagen 62"):

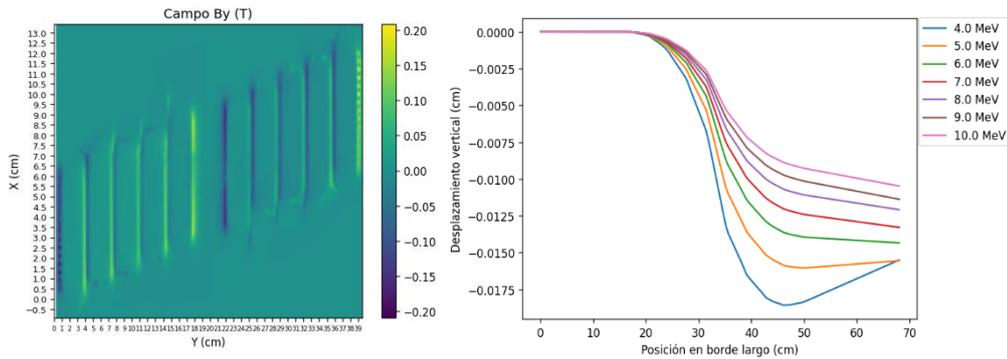


Imagen 62. Campo B_y y trazado de partículas en el plano YZ a partir de este (Resultados 1).

(Fuente: Elaboración propia desde "4.1. Mapas magnéticos" y "4.3. Simulación Plano YZ " del "ANEXO IV")

2. Los protones parten de la mitad superior del hueco entre polos opuestos de los imanes y continúan en la parte superior en el segundo bloque ("Imagen 63"):

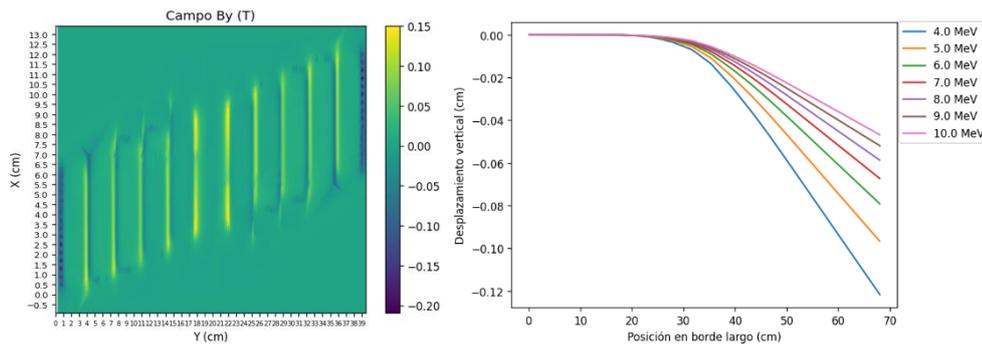


Imagen 63. Campo B_y y trazado de partículas en el plano YZ a partir de este (Resultados 2).

(Fuente: Elaboración propia desde "4.1. Mapas magnéticos" y "4.3. Simulación Plano YZ " del "ANEXO IV")

Si los protones parten de la parte inferior del hueco se obtienen resultados inversos, por lo que, si representamos conjuntamente todos los casos, inicializando los protones en coordenadas lógicas teniendo en cuenta el campo magnético que se les simula, obtenemos ("Imagen 64"):

Adecuación

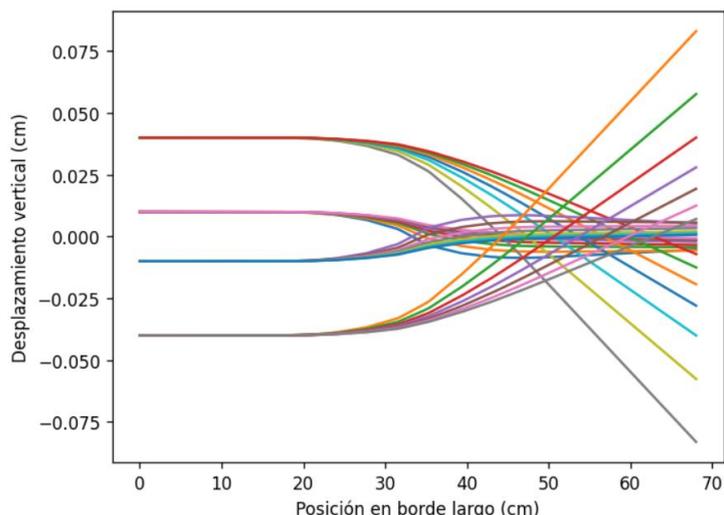


Imagen 64. Superposición del trazado de partículas en el plano YZ con diferentes sentidos de B_y .
(Fuente: Elaboración propia desde “6. Simulación Plano YZ” del “ANEXO IV”)

Estos resultados no son del todo reales puesto que se desconoce la altura exacta a la que se han realizado las medidas, siendo probable que se haya ido modificando a medida que avanza la sonda. Además, cuando los protones pasen de la mitad superior a la inferior se deberán ver afectados por $B_y \approx 0$, y esto puede suceder, no sólo entre bloques, sino también en un mismo bloque.

No obstante, a partir de los resultados anteriores es posible concluir que los imanes del seleccionador focalizan los protones del haz en el eje Z, siendo estos resultados muy interesantes y convenientes para el funcionamiento de la tecnología.

Dados los anteriores resultados y para simplificar las simulaciones, no será necesario tener en cuenta las componentes minoritarias del campo magnético (B_x y B_y). Por tanto, se procede a estudiar la variación del campo magnético con la temperatura.

8.5. Variación del campo magnético con la temperatura

Según (Martínez S. C., 2019) “las variaciones en la temperatura pueden causar pérdidas reversibles en ciertos rangos, sin embargo hay un valor límite, llamado temperatura de Curie, encima del cual los imanes pierden todas sus propiedades”. Por tanto, a partir de la siguiente tabla:

Tabla 11. Datos técnicos sobre los bloques de neodimio (LDA., s.f.).

REMANENCIA B_r		FUERZA DE CAMPO COERCITIVO H_c			PRODUCTO DE ENERGIA (BH) máx.		COEFICIENTE DE TEMP. REVERSIBLE		T° MÁX. TRABAJO
		H_{cb}		H_{ci}			B_r	H_{cj}	
Max.	Min.	Max.	Min.		Max.	Min.	(%/°C)	(%/°C)	
11,2 kG	11,7 kG	11,3 kOe	10,8 kOe	≥ 12 kOe	35,0 MGOe	33,0 MGOe	-0,12	-0,5	≤ 80 °C
1,210 T	1,170 T	899 kA/m	860 kA/m	≥ 955 kA/m	279 kJ/m ³	263 kJ/m ³			

Adecuación

Si tenemos en cuenta el coeficiente de temperatura reversible observamos que, si las medidas se han realizado a 18 °C y tenemos un campo magnético de 0,6 T aproximadamente, teniendo en cuenta un caso muy desfavorable, que sería alcanzar, por ejemplo, los 24 °C durante el experimento, es posible obtener:

$$0,6 [T] - 0,6 [T] \cdot 0,12 \left[\frac{\%}{^{\circ}C} \right] \cdot 6[^{\circ}C] = 0,6 - 0,00432 = 0,59568 [T]$$

Para determinar si la disminución del campo magnético es influyente se han multiplicado los datos de Bz por 0,99 (0,6 [T] · 0,99 = 0,594 [T]) obteniéndose los resultados que se muestran en la "imagen 65".

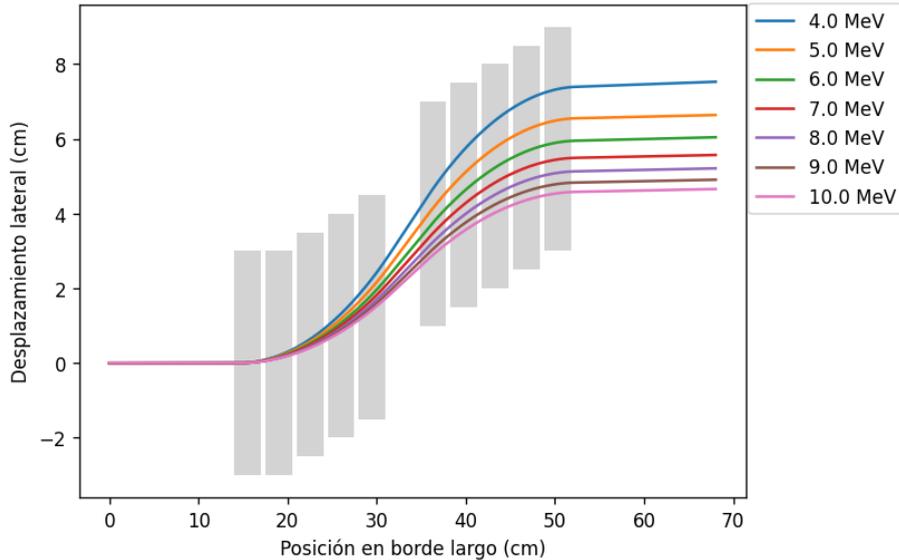


Imagen 65. Superposición del trazado de partículas en el plano XY un campo magnético de intensidad Bz x 0,99.

(Fuente: Elaboración propia desde "6. Simulación Plano YZ" del "ANEXO IV")

Tabla 12. Posición final de los protones de la "Imagen 65".

ENERGÍAS [MeV]	4	5	6	7	8	9	10
X _{FINAL} [cm]	7,53	6,64	6,04	5,57	5,21	4,91	4,66
ΔX [cm]	2,87						

Se obtiene una separación de 2,87 cm frente a los 2,95 cm que se obtienen con el campo medido a 18 °C. Por lo tanto, teniendo en cuenta que la cámara en la que tendrá lugar el experimento se encuentra acondicionada y que la atenuación del campo magnético es mínima incluso si la temperatura fuese de 24 °C, obteniéndose diferencias en el comportamiento del haz poco significativas, las siguientes simulaciones se simplificarán al máximo y no se contemplarán los efectos de la temperatura.

8.6. Efectos del aire en la propagación de los protones

Una vez contempladas todas las variables es necesario comprobar los efectos del aire sobre el comportamiento del haz. Para ello, se parte de una tabla de valores de pérdida de energía de los protones en aire a partir de la energía que portan (apartado “8. Pérdida de energía en aire” del “ANEXO IV”), construida por Michael Seimetz utilizando el software SRIM. Teniendo en cuenta los efectos del aire la simulación queda:

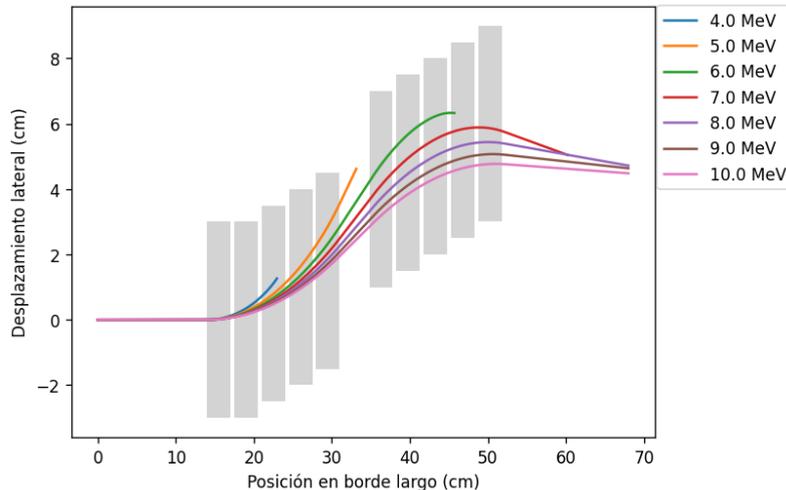


Imagen 66. Trazado de partículas en el Plano XY con pérdida de energía en aire a partir del campo magnético de la “Imagen 56”.

(Fuente: Elaboración propia desde “7. Simulación con aire” del “ANEXO IV”)

Se concluye entonces que los efectos del aire son bastante influyentes, pues las partículas que salen de la fuente con una energía inferior a 8 MeV se frenan totalmente antes de llegar a 68 cm. Así pues, para poder extraer mayores resultados será necesario acercarse lo máximo posible el comienzo del seleccionador a la fuente y situar los detectores lo más cerca posible de su final.

Además, se van a aumentar las energías al máximo por lo que se tendrá en cuenta en las simulaciones una energía máxima de 15 MeV, aunque luego no se consiga en el experimento final.

Como se desconoce la dispersión angular que provocan las placas atenuantes en el haz a la salida de la fuente, y se emplearán colimadores distintos, no se van a tener en cuenta estos aspectos en las simulaciones de momento.

Por otro lado, no se tienen en cuenta los efectos de la gravedad o del scattering o dispersión angular que se produce en el haz al atravesar el aire.

9. Modificaciones

Durante el transcurso del trabajo, se han modificado las condiciones del experimento pues el equipo montará un experimento similar al que tiene por objeto este TFG en el Centro de Láseres Pulsados (CLPU) de Salamanca, pero, para que el experimento de Sevilla sirva de precedente al de Salamanca, se ha decidido modificar el montaje del seleccionador para adaptarlo a las condiciones del experimento de Salamanca. En este sentido se tienen las siguientes condiciones:

- El haz sale aproximadamente a 12° de la fuente con una dispersión angular dada que se limita por el efecto de un colimador o pinhole, que consiste en una placa de grafito (se activa al bombardearlo con protones por lo que no produce radioactividad) con espesor suficiente como para parar todas las partículas del haz que chocan con ella, menos aquellas que atraviesan un orificio de un diámetro que se escogerá más adelante.
- Este experimento se realizará dentro de una cámara de vacío en la que no hay espacio suficiente como para albergar el selector y los detectores de partículas que se utilicen. Por lo tanto, será necesario que el haz salga por uno de los puertos de vacío de los que dispone la cámara.
- La longitud del haz en las simulaciones debe ser de 130 cm aproximadamente para su estudio.
- La distribución energética de los protones es de entre 2 MeV y 8 MeV con una fluencia que sigue un comportamiento similar al de una distribución de Boltzman.
- Se pretende que los protones que alcancen totalmente la ventana sean los de 5 MeV.
- Los imanes deben tener su lado abierto orientado en el mismo sentido para dejar libre un acceso y poder estudiar otros aspectos del haz. En este sentido se debe de posicionar el primer imán sobre uno de los extremos de su recorrido en el eje X, orientándose el lado cerrado del imán hacia fuera de la estructura de seleccionador, para tener más juego con la posición del resto de los imanes.
- La desviación de los protones ha de tener lugar en el sentido negativo del eje X.

Es por esta última restricción que se ha decidido eliminar para los 3 primeros imanes uno de los 2 tornillos que restringen el posicionamiento del imán en el eje X y utilizar los tornillos que sostienen las guías a la estructura en su lugar.

Por tanto, se han vuelto a realizar simulaciones para dar con la configuración correcta, aunque se mantendrá la separación entre imanes de un mismo bloque. Para poder modificar fácilmente la posición de los imanes en la simulación y que el campo magnético resultante sea similar al real, se ha desarrollado la simulación que se recoge en el apartado "4. Simulaciones con aproximación del campo magnético en los huecos" del "ANEXO III". Esta simulación tiene en cuenta el campo magnético medido sobre un imán del que se parte en las primeras simulaciones de este proyecto, pero para asignar valores de intensidad del campo magnético B_z en los espacios entre imanes (0,5 cm) se tienen en cuenta las medidas realizadas sobre los 0,25 cm posteriores a cada imán y anteriores al siguiente imán. De esta manera la simulación con la configuración anterior produce los siguientes resultados ("Imagen 67"):

Modificaciones

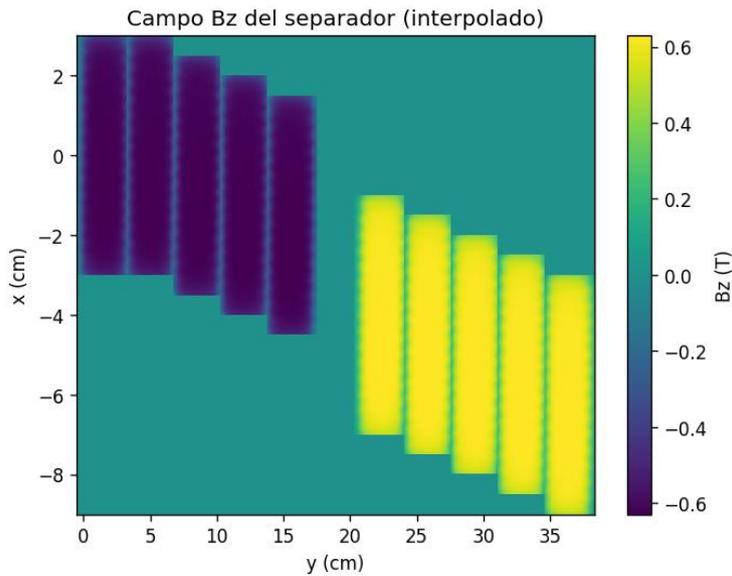


Imagen 67. Mapa magnético más aproximado al medido realmente.
(Fuente: Elaboración propia a partir del código recogido en “4.2. Mapa magnético de la simulación” del “ANEXO III”)

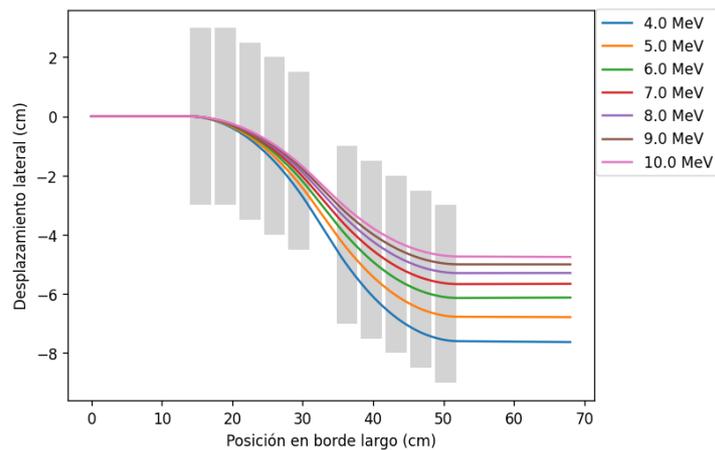


Imagen 68. Trazado de partículas a partir del campo recogido en la “Imagen 67”.
(Fuente: Elaboración propia desde “4.1. Trazado de partículas simple” del “ANEXO III”)

Tabla 13. Posición final de los protones de la “Imagen 68”

ENERGÍAS [MeV]	4	5	6	7	8	9	10
X_{FINAL} [cm]	7,63	6,78	6,12	5,66	5,29	5,00	4,75
ΔX [cm]	2,88						

Se obtiene una separación de 2,88 cm frente a los 2,95 cm que se obtienen con el campo magnético real medido por lo que es posible validar la simulación.

Las principales modificaciones en la configuración a llevar a cabo son la retirada de imanes del segundo bloque para producir un haz capaz de alcanzar uno de los puertos de vacío, y el desplazamiento de los imanes sobre el eje X para posicionarlos, según las nuevas restricciones, de manera que el campo magnético que estos producen afecte a todas las partículas del haz.

Así pues, extrayendo los datos de posición de las partículas del extremo del intervalo (4 MeV y 8 MeV) para diferentes configuraciones del seleccionador, es posible representar el espacio que ocuparía el haz en un plano 3D utilizando SolidWorks. Si se añade la representación de la cámara de vacío con los puertos de vacío es posible dar con la configuración óptima, que ha resultado ser de cinco imanes para el primer bloque y cuatro para el segundo, tal y como se aprecia en la siguiente imagen:

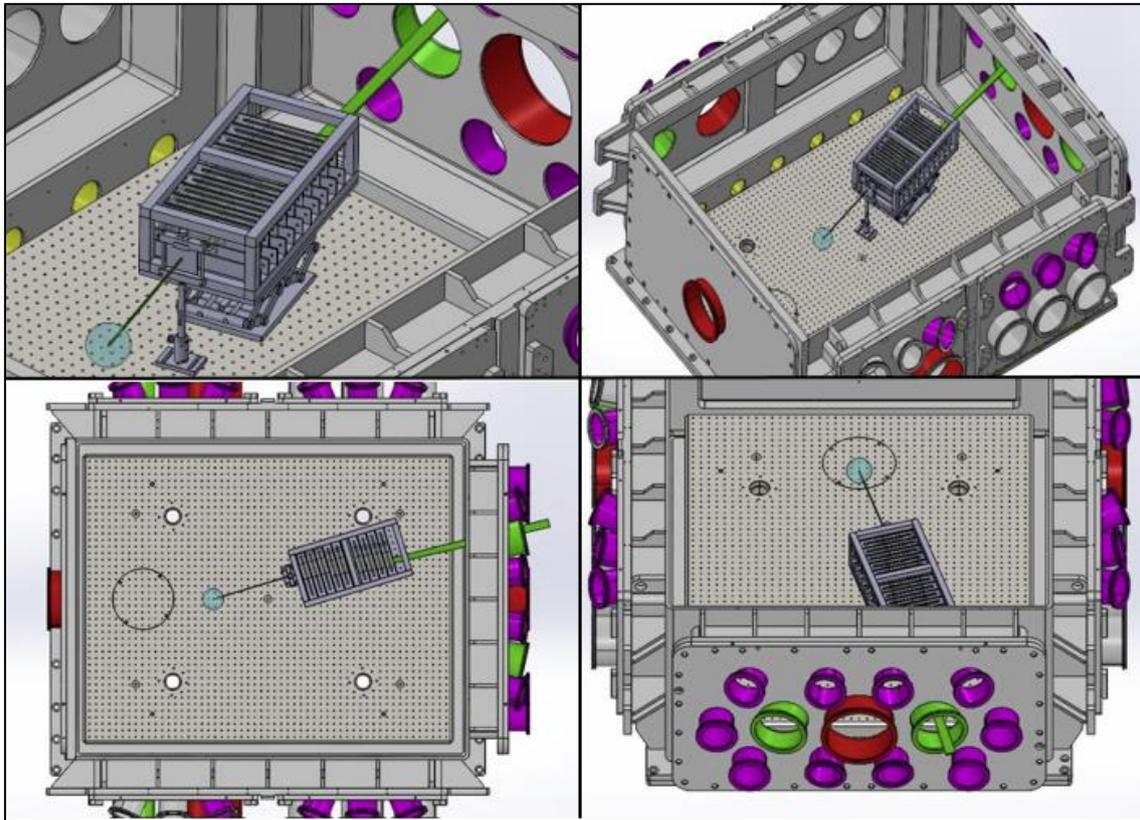


Imagen 69. Montaje del seleccionador en la cámara de vacío.
(Fuente: Elaboración propia)

Estos resultados se han comprobado añadiendo la distribución energética y angular mediante la simulación que se recoge en el “ANEXO III” apartado “4.3. Simulación compleja con nueve imanes”, que además sirve para estudiar la fluencia y la posición de los protones en base a su energía pues se desea que la ventana sea alcanzada por los protones de 5 MeV principalmente.

Además, se ha comprobado que la configuración escogida es también válida para el experimento de Sevilla, utilizando esta vez unas energías de entre 8 MeV y 16 MeV, pues las partículas de poca energía se frenan antes de alcanzar el detector. El código que ha permitido la comprobación no se recoge en ningún anexo para no sobrecargar el trabajo⁷, pero se basa en el que se recoge en el apartado “4.1 Trazado de partículas simple” del “ANEXO III”, que utiliza la aproximación del campo magnético en los huecos comentada anteriormente, pero eliminándose el último imán y añadiendo los efectos del aire.

⁷ Quedo a disposición del tribunal para facilitársela si es necesario

10. Experimentación y toma de resultados

Una vez en el CNA, la primera acción a desarrollar para montar los experimentos es montar el seleccionador dentro del búnker justo delante de la ventana o puerto de vacío del acelerador. Para ello se dispone de una mesa con ruedas, pero ha sido necesario añadir un conjunto de tablas de madera para elevar la base del seleccionador y situar la apertura de los imanes delante de la ventana de salida del acelerador. Además, se ha colocado un colimador de la manera en que se muestra en la imagen inferior y se ha nivelado y orientado el seleccionador de forma correcta utilizando un sistema de referenciación láser para observar la orientación correcta y diferentes calzas de papel para lograrla, tal y como se muestra en las siguientes imágenes:

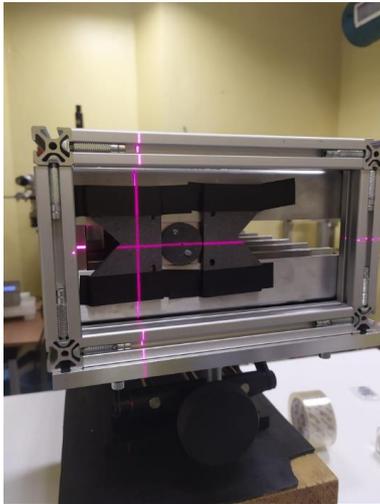


Imagen 70. Montaje del colimador de grafito de 1 mm.
(Fuente: Elaboración propia)



Imagen 71. Seleccionador nivelado y orientado correctamente delante de la fuente de protones.
(Fuente: Elaboración propia)

Siguientemente se ha montado una película radiocrómica justo después del primer imán y se han lanzado partículas a una energía aproximada de 14,5 MeV para observar la posición final, pero no se ha obtenido ninguna marca. Por tanto, se han colocado películas radiocrómicas detrás del primer y cuarto imán para observar la trayectoria de las partículas, obteniéndose que se desvían en el sentido contrario al previsto.

Por tanto, se ha utilizado un imán que tiene caracterizados el polo norte y sur para determinar la polaridad de los imanes. El resultado es que, si bien a priori podría parecer que los imanes tienen la orientación deseada, si acercamos el imán a la parte más central de los imanes observamos que la polaridad es la contraria, pues el campo magnético que producen los imanes no es exactamente igual al de un imán de herradura, sino que es el siguiente (“Imagen 72”):

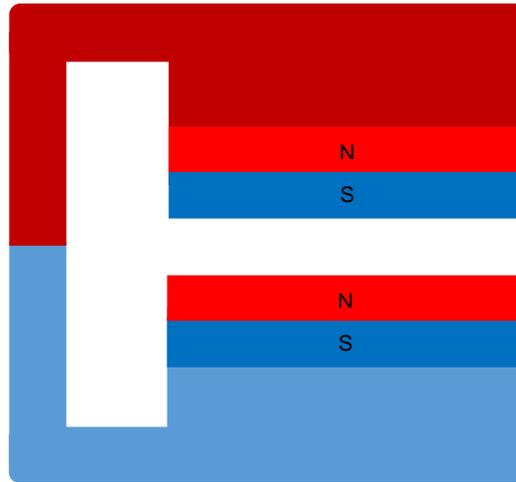


Imagen 72. Campo magnético de los imanes C.
(Fuente: Elaboración propia)

Por lo tanto, se ha procedido a dar la vuelta a los imanes sobre su eje central en dirección X, sin modificar la posición de las guías y tornillos que los sostienen en el plano XY, para modificar la dirección del campo magnético y producir el campo que se esperaba.

Además, en los primeros disparos, se han producido dos marcas más, sobre las películas radiocrómicas, a parte de la que producen los protones que entran por el centro del colimador circular, que tienen lugar debido a que el contacto producido entre el pinhole y las láminas adyacentes es puntual. Por ello, se han añadido dos placas de grafito encima del colimador, tal y como se muestra en la imagen inferior, para garantizar el sellado y que únicamente entren protones al seleccionador por el orificio del colimador (“Imagen 73”).

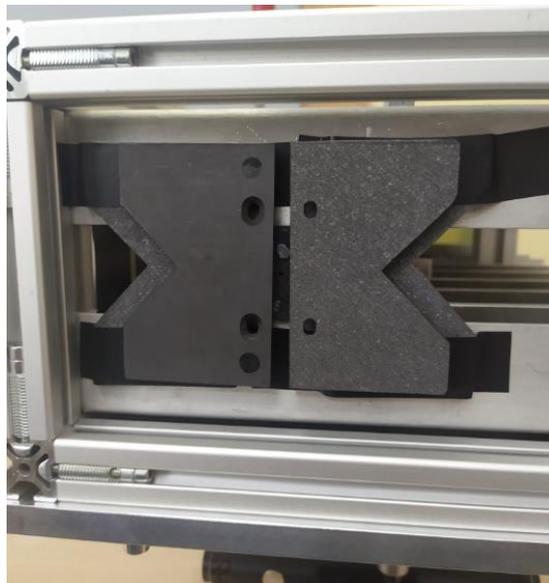


Imagen 73. Modificación del montaje del pinhole circular.
(Fuente: Elaboración propia)

Con estas modificaciones, sí que se han obtenido resultados satisfactorios, observándose distintas posiciones finales del protón para distintas energías de disparo, por lo que se han realizado un total de siete disparos con películas radiocrómicas a

diferentes posiciones para determinar el desvío que sufren los protones que salen de la fuente a 14,5 MeV y poder validar las simulaciones efectuadas.

Los resultados obtenidos se muestran a continuación en la “Imagen 74”:

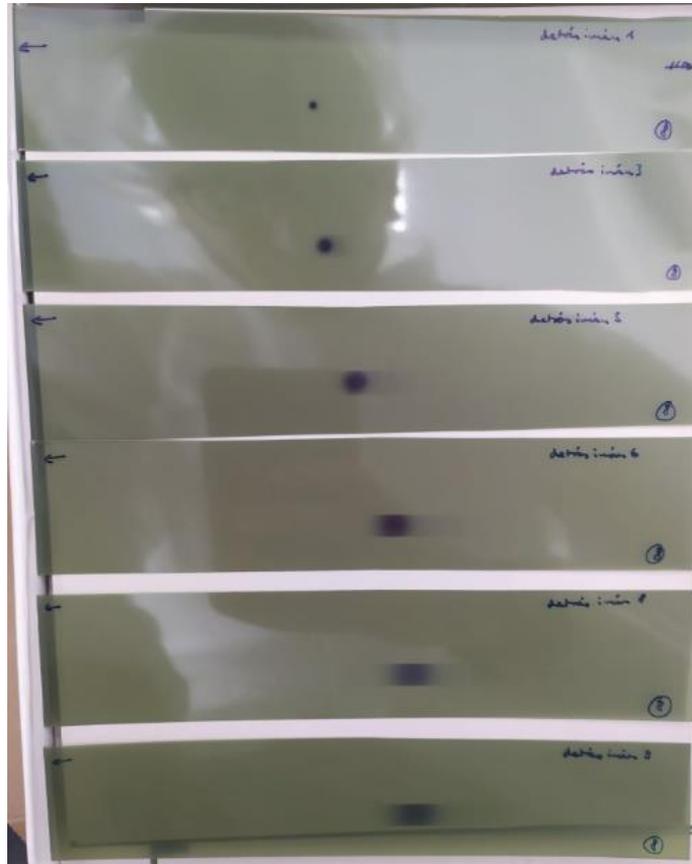


Imagen 74. Resultado de las películas radiocrómicas.
(Fuente: Elaboración propia)

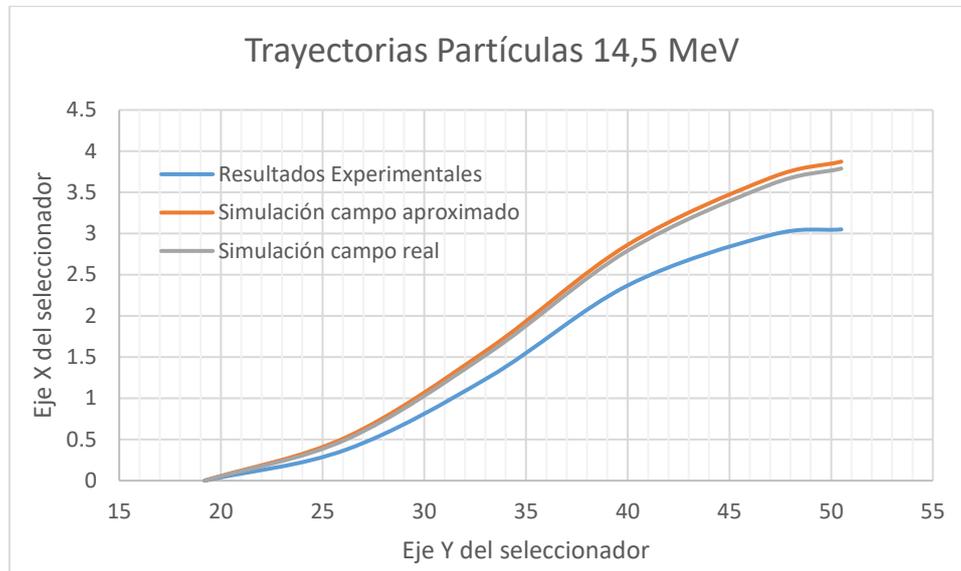
Nota. La distancia medida entre el comienzo de los imanes del seleccionador y la ventana ha sido de aproximadamente 16 cm, pero es difícil de determinar la distancia con exactitud debido a que la ventana de vacío es una superficie cóncava y se encuentra dentro de una arandela bastante gruesa y de difícil acceso.

De esta manera es posible concluir que las simulaciones reflejan de forma correcta lo que sucede en la realidad. No obstante, para poder comparar los resultados obtenidos de manera experimental con los de la simulación es necesario determinar el pico de dosis producido por los protones en las películas, pues el “scattering” o dispersión producido por el aire, dificulta la determinación de un punto que permita la comparación. Para esta función, se ha utilizado un escáner y un programa informático. Por otro lado, para poder determinar la posición exacta del protón en la simulación al pasar por los puntos del seleccionador donde se han colocado películas radiocrómicas, se ha desarrollado el código recogido en “4.4. Extracción de la posición del protón” del “ANEXO III”.

Si comparamos entonces los resultados de la simulación (“ANEXO III” → “4.1. Trazado de partículas simple” añadiendo los efectos del aire) con los que se han obtenido en el seleccionador, observamos que existen diferencias. Para comprobar si las diferencias son resultado de la aproximación del campo magnético en los huecos, y dado que la configuración del seleccionador que se midió (“ANEXO IV” → “4. Simulación con campo

medido”) es válida también para energías de 14,5 MeV, es posible simular el trazado de partículas con el campo magnético medido. Esta operación es posible también dado que el efecto que produzca el último imán será aguas abajo y no afectará a la posición de los protones en su paso por las películas radiocrómicas.

Los resultados, referenciando la posición de los protones a cero tras el primer imán, se recogen a continuación (Gráfica 3):



Gráfica 3. Comparación de resultados.
(Fuente: Elaboración propia)

Cabe destacar que la distancia entre la posición final que se obtiene para un protón de 14,5 MeV experimentalmente y mediante la simulación con campo magnético real es de aproximadamente 8 mm.

Para dar explicación a estas pequeñas diferencias se conjeturan las siguientes hipótesis:

- La distancia entre el seleccionador y la fuente es distinta de 16 cm, por lo tanto, la energía con la que entran los protones al seleccionador es también distinta de la que se tiene en las simulaciones efectuadas con una distancia de 16 cm y la trayectoria descrita por los protones de la simulación tiende hacia la que se ha comprobado experimentalmente.

Para comprobarlo se ha modificado la distancia fuente-seleccionador de manera razonable en las simulaciones, pero no se han apreciado resultados significativos.

- La energía con la que los protones salen de la cámara de vacío es distinta de 14,5 MeV, siendo el resultado el mismo de la anterior hipótesis.

Modificando las energías de salida se obtiene que para que la simulación se acerque a los resultados obtenidos son necesarias unas energías de al menos 20 MeV, por lo que no parece posible que esta hipótesis pueda explicar el comportamiento.

Siendo estos los principales factores que afectan a la trayectoria y posición final de los protones, aunque pueden producirse pequeñas diferencias por pequeñas variaciones entre el montaje y la configuración que se tiene en cuenta en la simulación, queda pendiente el estudio del efecto de los gradientes de campo magnético y la gravedad, pues son capaces de producir una deriva sobre el movimiento de las partículas que no se está teniendo en cuenta en estas simulaciones (Gutiérrez, 2017).

Además, es posible que el efecto del aire tenga bastante influencia también en el experimento porque será también un factor importante a tener en cuenta en futuras revisiones.

11. Conclusiones

A modo de cierre de este trabajo, podemos señalar que ha sido posible comprobar que el diseño del dispositivo es correcto, siendo éste capaz de separar los protones por energías. Además, se ha podido dar con una buena aproximación del campo magnético generado por dos imanes permanentes iguales orientados de forma paralela, pero que no se encuentran conectados en paralelo.

Por otro lado, se ha dado con una configuración del separador óptima para utilizarlo con haces de protones de entre 4 MeV y 16 MeV, aunque no se han podido validar las simulaciones sobre el trazado de partículas efectuadas. En este sentido, quedan pendientes en futuras revisiones estudios sobre otros factores, como el gradiente del campo magnético, que puedan explicar el efecto observado.

Otro gran aporte ha sido el descubrimiento de la focalización del haz en el plano vertical, pues entre las futuras tareas a realizar por el equipo de investigación está el diseño y desarrollo de un focalizador de imanes cuadrupolares, y la capacidad de focalización del separador rema a favor de este objetivo.

12. Bibliografía

- ¿Conoces las normas ISO contra la radiación? (s.f.). Obtenido de ISOTools: <https://www.isotools.us/2013/04/24/conoces-las-normas-iso-contra-la-radiacion/#:~:text=Recientemente%2C%20la%20Internacional%20Organization%20for,y%20riesgos%20en%20la%20salud>
- Acelerador Ciclotrón 18/9 MeV.* (2023). Obtenido de Centro Nacional de Aceleradores: <http://cna.us.es/index.php/es/instalaciones/ciclo>
- Almandoz, T. (2010). *Mas RM sala - imanes - bobinas - medios de contrastes*. Obtenido de Guía práctica para profesionales de Resonancia Manética : <http://dxiparatecnicos.blogspot.com/2010/09/mas-rm-sala-iman-es-bobinas-medios-de.html>
- BOE. (23 de Abril de 1997). *Real Decreto 486/1997, de 14 de abril, por el que se establecen las disposiciones mínimas de seguridad y salud en los lugares de trabajo*. Obtenido de BOE: <https://www.boe.es/buscar/pdf/1997/BOE-A-1997-8669-consolidado.pdf>
- Callejo, J. C., Rodriguez Perez, D., & Santa Marta Pastrana, C. (2013). El experimento de la gota de aceite [Grabado por I. B. Fernandez]. España.
- Caro, R., Delgado, J. M., Fernandez-Flygare, O., Gonzalez Garcia, L., Lizuain Arroyo, M. C., Gustavo, L., . . . Vaño Carruana, E. (1999). *Aceleradores de Partículas*. Madrid : Consejo de Seguridad Nuclear .
- Estatal, A. (21 de diciembre de 2022). Obtenido de Agencia Estatal Boletón Oficial del Estado : <https://www.boe.es/buscar/doc.php?id=BOE-A-2022-21682>
- Fernandez, J. L. (s.f.). *Ley de Lorentz*. Obtenido de Fisicalab: <https://www.fisicalab.com/apartado/ley-de-lorentz>
- García, A. F. (2013). *Movimiento de partículas cargadas*. Obtenido de Curso Interactivo de Física en Internet: http://www.sc.ehu.es/sbweb/fisica/_elecmagnet/movimiento/lineal/lineal.html
- Gutiérrez, C. D. (2017). *Simulación de la dinámica de partículas atrapadas en los cinturones de radiación*. Pereira: Universidad Tecnológica de Pereira.
- History, N. M. (1940). *Fist betatron of Donals Kerst*. *Smithsonian*.
- Infante, C. M. (Mayo de 2012). Estudio de factibilidad de la instalación de un acelerador de partículas en la Universidad de los Andes. Colombia, Colombia, Chile.
- Jimeno, A. R. (2020). Pasado, presente y futuro de los aceleradores de partículas (Parte 2). *Dciencia*.
- Jimeno, A. R. (2020). Presente, pasado y futuro de los aceleradores de partículas (Parte 1). *Dciencia*.
- Jorge. (9 de Junio de 2015). *Breve introducción a los aceleradores de partículas*. Obtenido de Acelerando la ciencia: <https://acelerandolaciencia.wordpress.com/category/historia/>

Bibliografía

- LDA., U. (s.f.). *Borrachas Técnicas MGO*. Obtenido de Unipessoal Lda.: <https://www.borrachasmgo.com/docs/58/pdf>
- Martínez, D. P. (25 de Agosto de 2021). La fuerza de Ampère-Grassmann-Lorentz. Madrid, Madrid, España.
- Martínez, S. C. (2019). *Diselo del campo magnético para imágenes de resonancia magnética de campo bajo*. Bogotá.
- Metrología dimensional*. (2017). Obtenido de Facultad de Ingeniería UNMDP: <http://www3.fi.mdp.edu.ar/tecnologia/archivos/Metrologia/Clase1.0-2017.pdf>
- Millán, P. J. (mayo de 2017). Caracterización de protones acelerados por láser y estudio de aplicaciones médicas. Valencia, Valencia, España. Obtenido de <https://www.educacion.gob.es/teseo/imprimirFicheroTesis.do?idFichero=nuLKZRm7SEI%3D>
- Ministerio de transportes, m. y. (2019). *Anuarios de Geomagnetismo*. Madrid.
- Muñoz, M., Lascano Aimacaña, N., Sebastian, D., Quiles Carrillo, Rojas Lerma, L., Paola, S., . . . Ivorra Martinez, J. (20 de mayo de 2020). *Comportamiento elástico en metales según Ley de Hooke*. Obtenido de Universidad Politécnica de Valencia: <http://hdl.handle.net/10251/143815>
- Poole, P., Obst-Huebl, L., Ginevra, C., Metzkes, J., Schlenvoigt, H.-P., Prencipe, I., . . . Zeil, K. (11 de enero de 2018). Laser-driven ion acceleration via target normal sheath acceleration in the relativistic transparency regime. *New Journal of Physics*, 20.
- Rodriguez, J. P. (6 de marzo de 2022). *Diseño de dispositivo desmagnetizador de componentes de sujeción de chapa metálica en el proceso de fineblanking*. Obtenido de Dirección general de bibliotecas y servicios digitales de información: <https://ri-ng.uaq.mx/handle/123456789/3521>
- Rohlf. (s.f.). *Aceleradores Lineales*. Obtenido de HyperPhysics: <http://hyperphysics.phy-astr.gsu.edu/hbasees/Particles/linac.html>
- Ruiz, L. M. (9 de Abril de 2013). *Ernest Lawrence y los inicios de la Gran Ciencia (I): Nace el ciclotrón*. Obtenido de Los Mundos de Brana: <https://losmundosdebrana.com/2013/04/09/ernest-lawrence-y-los-inicios-de-la-gran-ciencia-i/>
- Ruiz, L. M. (9 de Abril de 2013). *Ernest Lawrence y los inicios de la Gran Ciencia (I): Nace el ciclotrón*. Obtenido de Los Mundos de Brana: <https://losmundosdebrana.com/2013/04/09/ernest-lawrence-y-los-inicios-de-la-gran-ciencia-i/>
- Ruiz, L. M. (9 de Abril de 2013). *Ernest Lawrence y los inicios de la Gran Ciencia (I): Nace el ciclotrón*. Obtenido de Los Mundos de Brana: (Ruiz, Ernest Lawrence y los inicios de la Gran Ciencia (I): Nace el ciclotrón, 2013)
- Ruiz, M. R., Vazquez Serrano, F., Cubero Atienza, A., Salas Morera, L., Jimenez Hornero, J., Arauzo Azofra, A., . . . Garrido Jurado, J. (2010). *Efectos sobre la salud humana de los campos magnéticos y eléctricos de muy baja frecuencia (ELF)*. Andalucía: Junta de Andalucía. Consejería de empleo.

Bibliografía

- Segura, A., Nieto, V., & Segura, E. (21 de Marzo de 2012). Un análisis profundo del fenómeno dualidad onda partícula para la comprensión del mundo cuántico. *Latin-American Journal of Physics Education*, 6.
- Social, M. d. (2014). *Manual de Seguridad y Salud en oficinas*. Obtenido de FREMAP: https://www.usal.es/files/Manual_Seg_ySalud_Oficinas.pdf
- Solans, M., Almazan, C., & Espinas, J. (2014). *La protonterapia en el tratamiento del cáncer*. Barcelona: Ministerio de Sanidad, Servicios Sociales e Igualdad y la Agència de Qualitat i Avaluació Sanitàries de Catalunya.
- Torralba, A., Palenciano, L., Reija, A., Rigla, J., Peñas, J., Llerena, J., . . . Seimetz, M. (21 de febrero de 2022). Experimental Setup for Irradiation of Cell Cultures at L2A2. *Quantum Beam Science*, 6(1).
- UNE. (2022). Obtenido de AENOR: <https://tienda.aenor.com/norma-une-en-iso-13304-2-2022-n0070764>
- UPV. (22 de Marzo de 2022). *News*. Obtenido de Universitat Politècnica de València: <https://innovacion.upv.es/en/mas-ingresos-en-parque-cientifico-upv/>
- Varela, J. (2015). El primer acelerador de partículas; el ciclotrón de Lawrence. *Ciencia y tecnología*.
- Vectores de Campo Magnético*. (s.f.). Obtenido de Freepikcompany: <https://www.freepik.es/vectores/campo-magnetico>
- Wu Chao, A., Moser, H., & Zhao, Z. (2004). *Accelerator Physics, Technology and Applications: Selected Lectures of OCPA International Accelerator School 2002 Singapore*. London: World Scientific.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

ANEXO I

**Importancia de la energía alcanzada por las partículas
en los estudios de física de partículas y relación
energía/velocidad de las mismas**

ÍNDICE DEL ANEXO I

1. Introducción.....	3
2. Simulación simple	3
3. Simulación compleja	22
4. Bibliografía.....	44

1. Interés por el aumento de la energía de las partículas

La relación de De Broglie fue enunciada en 1924, tras el nacimiento de la cuántica (Max Planck, 1900), introduciendo así el concepto de la dualidad onda-corpúsculo, ya que establece que cada partícula en movimiento tiene una naturaleza ondulatoria y una longitud de onda asociada según la siguiente expresión (1):

$$\lambda = \frac{h}{p} = \frac{hc}{E} \quad (1)$$

Donde:

- λ es la longitud de onda
- h es la constante de Planck
- p es el momento lineal de la partícula
- c es la velocidad de la luz en el vacío
- E es la energía de la partícula

Es posible deducir, por tanto, que cuanto mayor sea la energía con la que se acelera una partícula, menor será su longitud de onda asociada y mayor su facilidad para penetrar mejor dentro del núcleo.

Para poder estudiar la estructura de las partículas, estas se aceleran y se hacen pasar a través de un material delgado, como una hoja de metal, cuya estructura afecta a la longitud de onda de la partícula. Esta perturbación en la longitud de onda se puede detectar y se puede usar para estudiar la estructura del material a nivel microscópico y subatómico, funcionando como un microscopio. Al utilizar partículas con longitudes de onda comparables al tamaño del protón, se pueden observar los efectos de su interacción, lo que permite obtener información valiosa sobre su estructura interna y su distribución de carga eléctrica.

Así pues, para poder estudiar la estructura interna del núcleo atómico son necesarias energías de al menos 1 GeV, debido a que la longitud de onda asociada a dicha energía es de 10^{-13} cm, que es el diámetro del protón (Segura, Nieto, & Segura, 2012). Cabe destacar, sin embargo, que G. Gamov predijo el efecto túnel en 1928, por lo que partículas de 500 keV podrían ser suficientes para dividir el núcleo.

Por otro lado, para crear nuevas partículas en una colisión, se debe proporcionar suficiente energía para superar la energía en reposo de las partículas involucradas. Esta energía es fácilmente predecible según la ecuación de Einstein (2):

$$E = m \cdot c^2 \quad (2)$$

Donde:

- m es la masa
- c es la velocidad de la luz en el vacío

Para el caso de los protones, su masa en reposo es de $0.938 \text{ GeV}/c^2$, por lo que para crear nuevas partículas en colisiones protón-protón, será necesaria una energía total de al menos 2 veces la masa del protón en reposo, es decir, 1.876 GeV (Jorge, 2015).

2. Relación energía/velocidad de las partículas

La relación que permite relacionar la velocidad de una partícula en [m/s] con su energía en [MeV], es una expresión relativista, pues las velocidades de las partículas son cercanas a la velocidad de la luz. La expresión tiene la forma (3):

$$v = c \cdot \sqrt{1 - \frac{1}{\left(1 + \frac{E}{m}\right)^2}} \quad (3)$$

Donde:

- v es la velocidad de la partícula
- c es la velocidad de la luz en el vacío
- m es la masa en reposo de la partícula
- E es la energía de la partícula

3. Bibliografía

- Jorge. (9 de Junio de 2015). *Breve introducción a los aceleradores de partículas*. Obtenido de Acelerando la ciencia: <https://acelerandolaciencia.wordpress.com/category/historia/>
- Segura, A., Nieto, V., & Segura, E. (21 de Marzo de 2012). Un análisis profundo del fenómeno dualidad onda partícula para la comprensión del mundo cuántico. *Latin-American Journal of Physics Education*, 6.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

ANEXO II

Simulaciones de partida

ÍNDICE DEL ANEXO II

1. Introducción	2
2. Simulación simple.....	2
3. Simulación compleja.....	21
4. Bibliografía.....	43

ÍNDICE DE IMÁGENES

Imagen 1. Campo Bz medido sobre un imán.....	3
Imagen 2. Campo Bz medido sobre un imán.....	4
Imagen 3. Trayectorias de los protones de diferentes energías en su paso por los imanes.....	19
Imagen 4. Trayectorias de los protones de diferentes energías con condiciones angulares dadas en su paso por los imanes.....	38
Imagen 5. Posición final de los protones.....	39
Imagen 6. Posición final de los protones.....	41
Imagen 7. Posición final de los protones.....	42

ÍNDICE DE TABLAS

Tabla 1. Ruta de la sonda en la medida del campo magnético de un imán.....	2
---	---

1. Introducción

En este anexo se recogen las simulaciones realizadas por Alicia Reija Vecino. Alicia estudió el comportamiento de los haces de protones a través de dos simulaciones principales que desarrolló en el lenguaje de programación “Python” utilizando el entorno “Jupyter Noteboock”, para dar con la configuración óptima del separador teniendo en cuenta sus requisitos. La primera “2. Simulación simple” es la más simple, pero permite estudiar el comportamiento de los protones de manera general, para dar con una configuración óptima que sirva de punto de partida de la segunda simulación “3. Simulación compleja”.

2. Simulación simple

En este anexo se recoge el código empleado en la simulación de la trayectoria descrita por un conjunto de 6 protones acelerados a diferentes energías con un ángulo de partida de 0° a la salida de la fuente. Además, se explican los aspectos más importantes de la simulación.

Las librerías empleadas se muestran a continuación:

```
import numpy as np
from matplotlib import pyplot as plt, patches
plt.rcParams['figure.dpi'] = 120
import pandas as pd
from scipy.interpolate import LinearNDInterpolator
```

El campo magnético que permite simular el comportamiento de las partículas a partir de la Ley de Lorentz es una medida del campo magnético en la dirección Z sobre la superficie proyectada de un imán en el plano central a los dos imanes de neodimio que lo componen, y el área comprendida por la longitud de este en el eje X y los 5 mm anteriores y posteriores en el eje Y. Este campo magnético se ha medido utilizando una sonda de efecto Hall y un robot que permite tomar medidas del magnetismo presente en distintos puntos del plano a mapear, a partir de una ruta introducida previamente en un programa de Matlab que mueve la sonda y crea un archivo de datos. La ruta que se ha empleado para la lectura de los datos se recoge en la siguiente tabla:

Tabla 1. Ruta de la sonda en la medida del campo magnético de un imán.

	MÍNIMO	MÁXIMO	INTERVALO
Y (cm)	-2	2	0.2
X (cm)	-3	3	0.2

El fichero "B_real.txt" que aparece en el siguiente extracto del código, recoge las posiciones del robot en coordenadas relativas en ambos ejes del plano de movimiento y la medida tomada en cada punto.

Por otro lado, el tipo de interpolación que ha permitido llevar a cabo la simulación es la “LinearNDInterpolator” de la librería numpy, que realiza una interpolación lineal en 2D.

ANEXO II: Simulaciones de partida

Las siguientes líneas de código representan el campo magnético medido sobre un imán con una resolución de 1000x1000 puntos.

```
##### Campo magnético #####

data=pd.read_csv("B_real.txt",delim_whitespace=True)
data_np=data.to_numpy()

x = 0.1*np.array(data_np[:,2],dtype='float64') # en MI sistema
y = 0.1*np.array(data_np[:,0],dtype='float64')
Bz = -1*np.array(data_np[:,3],dtype='float64')

precision = 1000 # solo para el plot

plt.xlabel('y (cm)')
plt.ylabel('x (cm)')
X_1 = np.linspace(min(x), max(x),precision)
Y_1 = np.linspace(min(y+2), max(y+2),precision)
X_1,Y_1 = np.meshgrid(X_1,Y_1)
int_Bz = LinearNDInterpolator(list(zip(x, y+2)), Bz,fill_value=0) #zip hace
una tupla de valores de dos arrays fila a fila
Bz_1 = int_Bz(X_1,Y_1)

plt.pcolormesh(Y_1, X_1, Bz_1, shading='auto')
#plt.scatter(y_fin,x_fin,c=Bz_fin,marker='D')

plt.colorbar()
plt.title('Campo Bz de un iman interpolado')
```

Los “outputs” de este fragmento de código se muestran a continuación (Imagen 1):

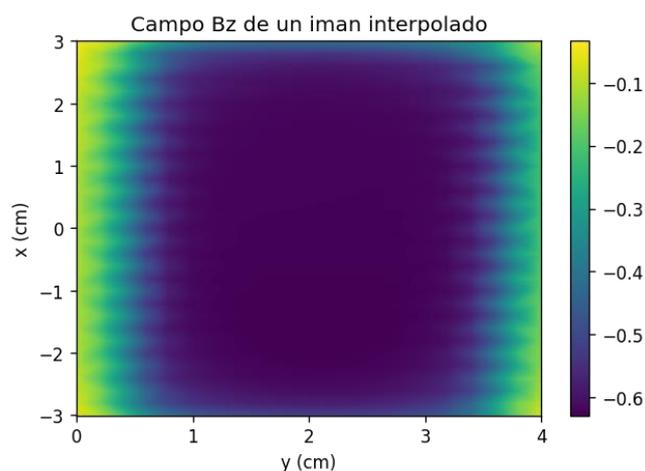


Imagen 1. Campo Bz medido sobre un imán.
(Fuente: Elaboración propia a partir de los resultados del código que se muestra previamente)

Nota. Esta imagen es la misma que la “Imagen 27” de la memoria.

ANEXO II: Simulaciones de partida

Los parámetros que permiten simular el comportamiento del protón a partir de la Ley de Lorentz se recogen a continuación:

```
# Parámetros protón  
  
m = 938.28 # MeV/c2  
m_kg = 1.67E-27 # kg  
q = 1.6E-19 # C  
c = 299792458 # m/s
```

En las siguientes líneas de código se realiza la simulación propiamente dicha. Primero, se establece el intervalo energético de los protones ($[2, 9[$ MeV con un incremento de 1 MeV) y la disposición de los imanes. Es importante destacar que los imanes se encuentran separados 1 cm entre sí en un mismo bloque y 2 cm entre bloques. Este hecho facilita la simulación ya que, como se ha visto anteriormente, el campo magnético medido contempla los 5 mm anteriores y posteriores a cada imán, tendiendo prácticamente a 0 T en los extremos, por lo que no es necesario realizar ninguna operación para determinar el magnetismo entre imanes:

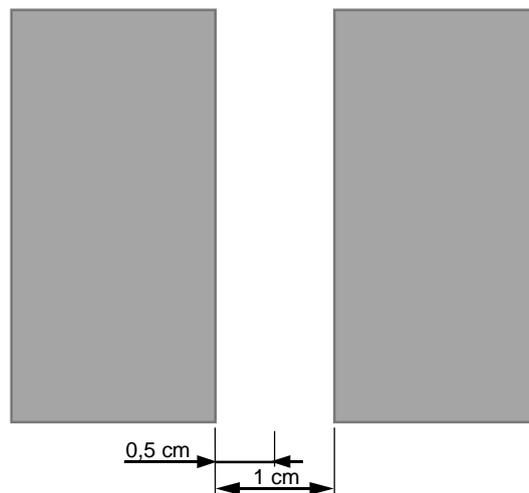


Imagen 2. Campo B_z medido sobre un imán.
(Fuente: Elaboración propia)

A continuación se calcula la velocidad de las partículas a partir de la energía de las mismas, según se explica en el anexo I, y comienza la simulación. Los protones se mueven con una velocidad inicial y a partir de la resolución temporal o timestemp ($dt = 10^{-11}$), se van realizando cálculos sucesivos para determinar la fuerza que se genera sobre ellos, según la ley de Lorentz, teniendo en cuenta el campo magnético interpolado en cada punto según la posición relativa de la partícula sobre cada imán. Las fuerzas resultantes sobre las partículas determinan unas nuevas componentes de la velocidad que volverán a generar una fuerza en la siguiente parada (v [m/s] $\cdot 10^{-11}$ [s]) metros más adelante según el campo magnético en ese punto. Además, en este proceso iterativo la posición absoluta de la partícula se va guardando para generar un mapa de las trayectorias.

ANEXO II: Simulaciones de partida

```
energies = np.arange(2,9,1) # distribución discreta de energías de los
protones

# Posiciones relativas de los imanes
#x_rel_1=0
#x_rel_2=0
#x_rel_3=0
#x_rel_4=3
#x_rel_5=3
#x_rel_6=5
#x_rel_7=6
#x_rel_8=6
#x_rel_9=7.5
#x_rel_10=7.5

x_rel_1=0
x_rel_2=0
x_rel_3=0
x_rel_4=3
x_rel_5=3
x_rel_6=5
x_rel_7=6
x_rel_8=6
x_rel_9=7.5
x_rel_10=7.5

# Figura e imanes

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlabel('Posición en borde largo (cm)')
ax.set_ylabel('Desplazamiento lateral (cm)')

iman1 = patches.Rectangle((14, -3-x_rel_1), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman2 = patches.Rectangle((18, -3-x_rel_2), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman3 = patches.Rectangle((22, -3-x_rel_3), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman4 = patches.Rectangle((26, -3-x_rel_4), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman5 = patches.Rectangle((30, -3-x_rel_5), 3, 6,
edgecolor=None, facecolor="lightgrey")

iman6 = patches.Rectangle((35, -3-x_rel_6), 3, 6,
edgecolor=None, facecolor="lightgrey")
```

ANEXO II: Simulaciones de partida

```
iman7      =      patches.Rectangle((39,-3-x_rel_7),      3,      6,
edgecolor=None,facecolor="lightgrey")
iman8      =      patches.Rectangle((43,-3-x_rel_8),      3,      6,
edgecolor=None,facecolor="lightgrey")
iman9      =      patches.Rectangle((47,-3-x_rel_9),      3,      6,
edgecolor=None,facecolor="lightgrey")
iman10     =      patches.Rectangle((51,-3-x_rel_10),     3,      6,
edgecolor=None,facecolor="lightgrey")

ax.add_patch(iman1)
ax.add_patch(iman2)
ax.add_patch(iman3)
ax.add_patch(iman4)
ax.add_patch(iman5)

ax.add_patch(iman6)
ax.add_patch(iman7)
ax.add_patch(iman8)
ax.add_patch(iman9)
ax.add_patch(iman10)

# Donde nos vamos a guardar las posiciones

data_E=[]

data_x=[]
data_y=[]
data_z=[]

data_Bz=[]

for i in range(len(energies)):

    v_p=c*np.sqrt(1-(1/(1 + energies[i]/m)**2)) # formula relativista para
la velocidad

    x = 0 # parte del punto 1.5 cm. Estas distancias están en m.
    y = 0
    z = 0

    vx = 0
    vy = v_p # incide con velocidad v_y, perpendicular al lado largo de
los imanes
    vz = 0

    # Aquí nos vamos a guardar las posiciones en cada timestep:

    pos_x=[]
    pos_y=[]
```

ANEXO II: Simulaciones de partida

```
pos_z=[]

pos_Bz=[]

t = 0      # inicialización del tiempo
dt = 1E-11 # timestep

# Inicialización de los campos:

Bx=0
By=0
Bz=0

while y<0.135: #L_in. Aquí no hay campo magnético. El protón no se
desvía.

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

# Posición del protón en x justo antes de entrar al primer iman. La
necesitaremos para la función de los campos:

x_start_1=np.array(pos_x)[len(pos_x)-1]

## PRIMER IMAN

x_B=100*x_start_1+x_rel_1
y_B=0

while y>=0.135 and y<0.175:

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1
```

ANEXO II: Simulaciones de partida

```
# Calculamos los campos en cada timestep:

Bz = int_Bz(x_B,y_B)

# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*Bz)-(vz*By))
Fy = q*((vz*Bx)-(vx*Bz))
Fz = q*((vx*By)-(vy*Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

x_start_2=np.array(pos_x)[len(pos_x)-1]

## SEGUNDO IMAN

x_B=100*x_start_2+x_rel_2
y_B=0

while y>=0.175 and y<0.215:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
```

ANEXO II: Simulaciones de partida

```
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

x_start_3=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_3+x_rel_3
y_B=0

## TERCER IMAN

while y>=0.215 and y<0.255:

    # Definimos campos:
```

ANEXO II: Simulaciones de partida

```
x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

Bz = int_Bz(x_B,y_B)

# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*Bz)-(vz*By))
Fy = q*((vz*Bx)-(vx*Bz))
Fz = q*((vx*By)-(vy*Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

x_start_4=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_4+x_rel_4
y_B=0

## CUARTO IMAN

while y>0.255 and y<=0.295:

    # Definimos campos:
```

ANEXO II: Simulaciones de partida

```
x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

Bz = int_Bz(x_B,y_B)

# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*Bz)-(vz*By))
Fy = q*((vz*Bx)-(vx*Bz))
Fz = q*((vx*By)-(vy*Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

x_start_5=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_5+x_rel_5
y_B=0

## QUINTO IMAN

while y>0.295 and y<=0.335:

    # Definimos campos:
```

ANEXO II: Simulaciones de partida

```
x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

Bz = int_Bz(x_B,y_B)

# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*Bz)-(vz*By))
Fy = q*((vz*Bx)-(vx*Bz))
Fz = q*((vx*By)-(vy*Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

Bz = 0

## ESPACIO ENTRE SECTORES

while y>0.335 and y<0.345:

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt
```

ANEXO II: Simulaciones de partida

```
pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

x_start_6=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_6+x_rel_6
y_B=0

#Bz = int_Bz(x_B,y_B)

## INVERTIMOS CAMPO

# SEXTO IMAN

while y>=0.345 and y<0.385:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt
```

ANEXO II: Simulaciones de partida

```
# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_7=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_7+x_rel_7
y_B=0

# SEPTIMO IMAN

while y>=0.385 and y<0.425:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt
```

ANEXO II: Simulaciones de partida

```
# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_8=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_8+x_rel_8
y_B=0

# OCTAVO IMAN

while y>=0.425 and y<0.465:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
```

ANEXO II: Simulaciones de partida

```
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_9=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_9+x_rel_9
y_B=0

# NOVENO IMAN

while y>=0.465 and y<0.505:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
```

ANEXO II: Simulaciones de partida

```
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_10=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_10+x_rel_10
y_B=0

# DECIMO IMAN

while y>=0.505 and y<0.545:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
```

ANEXO II: Simulaciones de partida

```
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_11=np.array(pos_x)[len(pos_x)-1]
Bz=0

while y>=0.545 and y<=0.68:

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

x_start_12=np.array(pos_x)[len(pos_x)-1]

# Guardamos las posiciones en un array para trabajar mejor:

pos_x_arr=np.array(pos_x)
pos_y_arr=np.array(pos_y)
pos_z_arr=np.array(pos_z)

pos_Bz_arr=np.array(pos_Bz)

data_E.append(energies[i])
data_x.append(pos_x)
data_y.append(pos_y)
data_z.append(pos_z)

data_Bz.append(pos_Bz)
```

ANEXO II: Simulaciones de partida

```
ax.plot(pos_y_arr*100,pos_x_arr*100,label=energias[i])
ax.legend(labels=['2 MeV', '3 MeV', '4 MeV', '5 MeV', '6 MeV', '7 MeV', '8
MeV'])
```

Los resultados se muestran a continuación:

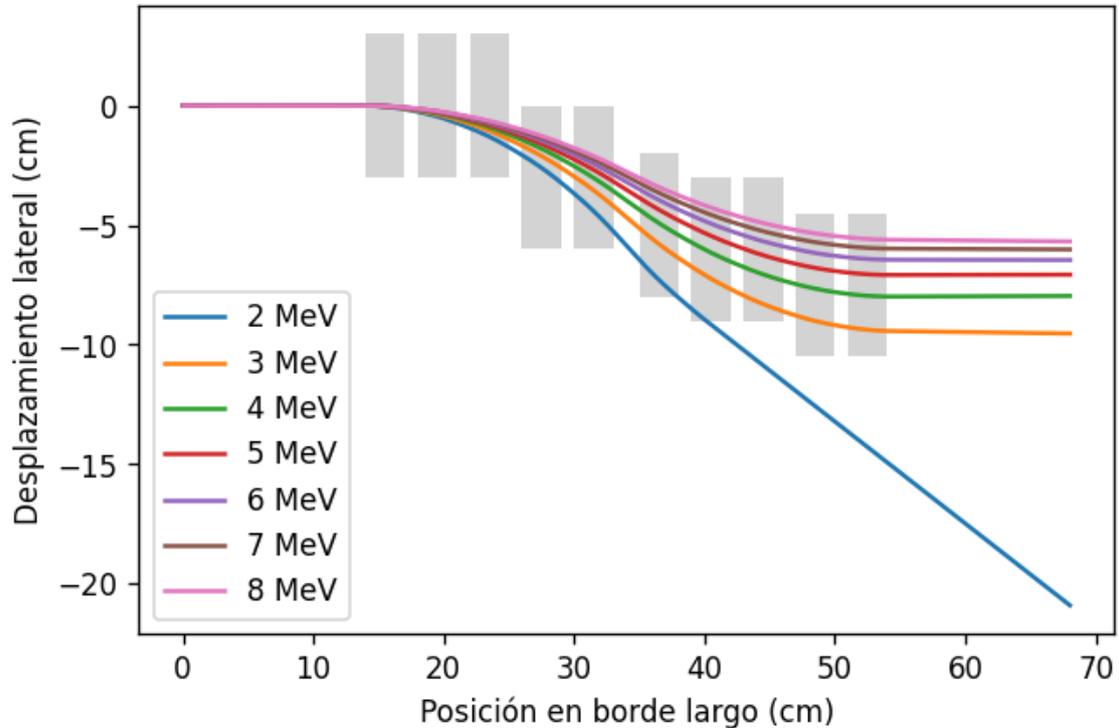


Imagen 3. Trayectorias de los protones de diferentes energías en su paso por los imanes.

(Fuente: Elaboración propia a partir de los resultados del código que se muestra previamente)

Finalmente se muestra por pantalla la posición final de las partículas.

```
## Posiciones finales de las energías centrales

for i in range(len(data_E)):

    if round(data_E[i],1) == 2:
        print('x_final 2 MeV:', np.array(data_x[i])[len(data_x[i])-1]*100)
        print('z_final 2 MeV:', np.array(data_z[i])[len(data_z[i])-1]*100)
        print('-----')

    if round(data_E[i],1) == 3:
        print('x_final 3 MeV:', np.array(data_x[i])[len(data_x[i])-1]*100)
        print('z_final 3 MeV:', np.array(data_z[i])[len(data_z[i])-1]*100)
        print('-----')

    if round(data_E[i],1) == 4:
        print('x_final 4 MeV:', np.array(data_x[i])[len(data_x[i])-1]*100)
```

ANEXO II: Simulaciones de partida

```
print('z_final 4 MeV:', np.array(data_z[i])[len(data_z[i])-1]*100)
print('-----')

if round(data_E[i],1) == 5:
    print('x_final 5 MeV:', np.array(data_x[i])[len(data_x[i])-1]*100)
    print('z_final 5 MeV:', np.array(data_z[i])[len(data_z[i])-1]*100)
    print('-----')

if round(data_E[i],1) == 6:
    print('x_final 6 MeV:', np.array(data_x[i])[len(data_x[i])-1]*100)
    print('z_final 6 MeV:', np.array(data_z[i])[len(data_z[i])-1]*100)
    print('-----')

if round(data_E[i],1) == 7:
    print('x_final 7 MeV:', np.array(data_x[i])[len(data_x[i])-1]*100)
    print('z_final 7 MeV:', np.array(data_z[i])[len(data_z[i])-1]*100)
    print('-----')

if round(data_E[i],1) == 8:
    print('x_final 8 MeV:', np.array(data_x[i])[len(data_x[i])-1]*100)
    print('z_final 8 MeV:', np.array(data_z[i])[len(data_z[i])-1]*100)
```

Los resultados son:

```
x_final 2 MeV: -20.902675840439727
z_final 2 MeV: 0.0
-----
x_final 3 MeV: -9.531810917467894
z_final 3 MeV: 0.0
-----
x_final 4 MeV: -7.9655306459249395
z_final 4 MeV: 0.0
-----
x_final 5 MeV: -7.072954435142939
z_final 5 MeV: 0.0
-----
x_final 6 MeV: -6.457561763899239
z_final 6 MeV: 0.0
-----
x_final 7 MeV: -6.0184425259685685
z_final 7 MeV: 0.0
-----
x_final 8 MeV: -5.682274123893537
z_final 8 MeV: 0.0
```

3. Simulación compleja

Esta simulación parte de la anteriormente comentada, pero introduce ciertas mejoras que permiten dar con una solución mucho más cercana al comportamiento real. Para ello, se añaden la fluencia energética y dispersión angular a las condiciones iniciales del haz de protones. Estas mejoras suponen un aumento del tiempo de cálculo considerable, pasando de unos pocos segundos a más de 10 minutos.

Se requieren de las siguientes librerías:

```
import numpy as np
from matplotlib import pyplot as plt, patches
plt.rcParams['figure.dpi'] = 120
from mpl_toolkits.mplot3d import Axes3D
import pandas as pd
from scipy.interpolate import interp2d
from scipy.interpolate import LinearNDInterpolator
from math import pi
from numpy.random import uniform
```

Además, se parte de los mismos datos que en la simulación anterior (lecturas del campo Bz medido sobre un imán y parámetros del protón), pero se añaden las condiciones del pinhole circular, que es el orificio por el que se hace pasar el haz a la salida del acelerador y determina, a partir de su diámetro y la distancia respecto a la fuente, el ángulo de dispersión con el que entran al separador (1), que define, a su vez, el número de partículas.

$$\theta = \arctg\left(\frac{d_p}{2L_p}\right) \quad (1)$$

Donde:

- Diámetro del pinhole: $d_p = 0.001 \text{ m}$
- Distancia entre pinhole-fuente: $L_p = 0.07 \text{ m}$

```
## Características pinhole circular

l_p = 0.001 # diámetro pinhole (m)
L_p = 0.07 # distancia fuente-pinhole

theta_max = np.arctan((l_p/2)/L_p) # máximo ángulo de observación desde la
fuente hasta el pinhole, en grados
#print('Máximo ángulo de observación fuente-pinhole (rad)', theta_max)
```

ANEXO II: Simulaciones de partida

Atendiendo a la fluencia de los protones, se puede obtener a partir de la siguiente expresión propuesta por Seimetz (2015) para el espectro de Fritzier (2003), ya que la distribución energética de los protones se puede asemejar a una distribución de Boltzman (Millán, 2017) (2):

$$N = \left(\frac{N_0}{E}\right) * e^{-\left(\frac{E}{KBT}\right)} * e^{-\left(\frac{b}{E_{max}^2 - E^2}\right)} \quad (2)$$

Donde:

- N_0 → Número de protones a la salida del acelerador.
- E → Input de la función, es decir, energía para la cual se pretende calcular el número de protones.
- KBT → Constante
- b → Constante
- E_{max} → Energía pico o máxima

```
# Fluencia definida por Seimetz (2015) para el espectro de Fritzier (2003)

def fluence(E):
    N0=10**10
    KBT=10
    b=20
    E_max=10 # ojo, luego las energías de los protones deberían llegar
    hasta menos de 10 para que no haya division por zero

    N = (N0/E) * np.exp(-E/KBT) * np.exp(-b/((E_max**2)-(E**2)))
    return N
```

El resto es ya la simulación propiamente, con la particularidad de que el salto energético que permite dar con una matriz de energías separadas de manera homogénea para un intervalo dado pasa de 1 MeV a 0.01 MeV. Además, para inicializar las partículas se calcula el número de de estas para cada energía del intervalo (fluencia) y se las hace partir de la fuente con una distribución angular aleatoria que se encuentra dentro de un cono en el que el ángulo que forman el eje y la hipotenusa del triángulo de revolución es de $\pi/36$ radianes. Es importante aclarar que la punta del cono apunta hacia a la fuente (sentido negativo del eje Y).

Una vez inicializada cada partícula se comprueba si atravesará el pinhole y si es el caso, se comienza la simulación.

```
delta = 0.01
energias = np.arange(2,9,delta) # distribución discreta de energías de los
protones

# Posiciones relativas de los imanes (respecto a x=0)

x_rel_1=0
x_rel_2=0
x_rel_3=0
```

ANEXO II: Simulaciones de partida

```
x_rel_4=3
x_rel_5=3
x_rel_6=5
x_rel_7=6
x_rel_8=6
x_rel_9=7.5
x_rel_10=7.5

## Nos guardamos los parámetros en listas para poder trabajar más
comodamente fuera de los bucles:

data_E = [] # energía del protón en cada timestep
# Posiciones xyz en cada timestep
data_x = []
data_y = []
data_z = []

data_Bz = []

factor = 100000 # factor de simplificación de la fluencia para reducir
tiempo computacional

for i in range(len(energies)):
    v_p = c*np.sqrt(1-(1/(1 + energies[i]/m)**2)) # formula relativista
para la velocidad

    N = fluence(energies[i])*delta/factor # dividimos para rapidez
computacional

    for j in np.arange(1,N,1):

        # Vectores directores unitarios:

        rho = 1
        u = uniform(0.5*(1+np.cos(pi/18)),1)
        theta = np.arccos(2*u-1) # distribucion uniforme de puntos sobre
una esfera

        v = uniform(0,1)
        phi = 2*pi*v

        if theta < theta_max: # si la dirección del protón cae dentro de
lo que subtiende el pinhole, seguimos

            # Coordenadas esféricas:

            esf_y = rho*np.cos(theta)
            esf_x = rho*np.cos(phi)*np.sin(theta)
```

ANEXO II: Simulaciones de partida

```
esf_z = rho*np.sin(phi)*np.sin(theta)

vx = v_p*esf_x
vy = v_p*esf_y
vz = v_p*esf_z

# Inicializamos las posiciones

x = 0
y = 0
z = 0

pos_x = []
pos_y = []
pos_z = []

pos_Bz = []

t = 0      # inicialización del tiempo
dt = 1E-11 # timestep

# Inicialización de los campos:

Bx = 0
By = 0
Bz = 0

while y<0.135: #L_in. Aquí no hay campo magnético. El protón
no se desvía.

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

# Posición del protón en x justo antes de entrar al primer
iman. La necesitaremos para la función de los campos:

x_start_1=np.array(pos_x)[len(pos_x)-1]

## PRIMER IMAN
```

ANEXO II: Simulaciones de partida

```
x_B=100*x_start_1+x_rel_1
y_B=0

while y>=0.135 and y<0.175:

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

    # Calculamos los campos en cada timestep:

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que
la aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt
```

ANEXO II: Simulaciones de partida

```
x_start_2=np.array(pos_x)[len(pos_x)-1]

## SEGUNDO IMAN

x_B=100*x_start_2+x_rel_2
y_B=0

while y>=0.175 and y<0.215:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
    Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
    la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
    la expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que
    la aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt
```

ANEXO II: Simulaciones de partida

```
x_start_3=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_3+x_rel_3
y_B=0

## TERCER IMAN

while y>=0.215 and y<0.255:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
    Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
    la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
    la expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que
    la aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

x_start_4=np.array(pos_x)[len(pos_x)-1]
```

ANEXO II: Simulaciones de partida

```
x_B=100*x_start_4+x_rel_4
y_B=0

## CUARTO IMAN

while y>0.255 and y<=0.295:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
    Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
    la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
    la expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que
    la aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

x_start_5=np.array(pos_x)[len(pos_x)-1]
```

ANEXO II: Simulaciones de partida

```
x_B=100*x_start_5+x_rel_5
y_B=0

## QUINTO IMAN

while y>0.295 and y<=0.335:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
    Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
    la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
    la expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que
    la aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt
```

ANEXO II: Simulaciones de partida

```
Bz = 0

## ESPACIO ENTRE SECTORES

while y>0.335 and y<0.345:

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

x_start_6=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_6+x_rel_6
y_B=0

## INVERTIMOS CAMPO

# SEXTO IMAN

while y>=0.345 and y<0.385:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
    Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
    la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
    la expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
```

ANEXO II: Simulaciones de partida

```
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que
la aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_7=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_7+x_rel_7
y_B=0

# SEPTIMO IMAN

while y>=0.385 and y<0.425:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
```

ANEXO II: Simulaciones de partida

```
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que
la aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_8=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_8+x_rel_8
y_B=0

# OCTAVO IMAN

while y>=0.425 and y<0.465:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
```

ANEXO II: Simulaciones de partida

```
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que
la aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_9=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_9+x_rel_9
y_B=0

# NOVENO IMAN

while y>=0.465 and y<0.505:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
```

ANEXO II: Simulaciones de partida

```
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que
la aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_10=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_10+x_rel_10
y_B=0

# DECIMO IMAN

while y>=0.505 and y<0.545:

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
```

ANEXO II: Simulaciones de partida

```
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que
# la aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_11=np.array(pos_x)[len(pos_x)-1]
Bz=0

while y>=0.545 and y<=0.68:

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

x_start_12=np.array(pos_x)[len(pos_x)-1]

# Guardamos las posiciones en un array para trabajar mejor:

pos_x_arr=np.array(pos_x)
pos_y_arr=np.array(pos_y)
pos_z_arr=np.array(pos_z)

pos_Bz_arr=np.array(pos_Bz)
```

ANEXO II: Simulaciones de partida

```
data_E.append(energies[i])
data_x.append(pos_x)
data_y.append(pos_y)
data_z.append(pos_z)

data_Bz.append(pos_Bz)

##### TRAYECTORIAS #####

# Figura e imanes

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlabel('Posición en borde largo (cm)')
ax.set_ylabel('Desplazamiento lateral (cm)')

iman1 = patches.Rectangle((14, -3-x_rel_1), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman2 = patches.Rectangle((18, -3-x_rel_2), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman3 = patches.Rectangle((22, -3-x_rel_3), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman4 = patches.Rectangle((26, -3-x_rel_4), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman5 = patches.Rectangle((30, -3-x_rel_5), 3, 6,
edgecolor=None, facecolor="lightgrey")

iman6 = patches.Rectangle((35, -3-x_rel_6), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman7 = patches.Rectangle((39, -3-x_rel_7), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman8 = patches.Rectangle((43, -3-x_rel_8), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman9 = patches.Rectangle((47, -3-x_rel_9), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman10 = patches.Rectangle((51, -3-x_rel_10), 3, 6,
edgecolor=None, facecolor="lightgrey")

ax.add_patch(iman1)
ax.add_patch(iman2)
ax.add_patch(iman3)
ax.add_patch(iman4)
ax.add_patch(iman5)

ax.add_patch(iman6)
ax.add_patch(iman7)
ax.add_patch(iman8)
ax.add_patch(iman9)
```

ANEXO II: Simulaciones de partida

```
ax.add_patch(iman10)

#print(energies)
print(len(data_E))
#print(len(data_x))
#print(len(data_y))
#print(len(data_z))

for i in range(len(data_E)):
    #if round(data_E[i],1) == 1:
    #    plt.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'r',label='3')
    if round(data_E[i],1) == 2:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'g',label='4')
    elif round(data_E[i],1) == 3:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'b',label='5')
    elif round(data_E[i],1) == 4:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'y',label='4')
    elif round(data_E[i],1) == 5:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'k',label='7')
    elif round(data_E[i],1) == 6:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'m',label='8')
    elif round(data_E[i],1) == 7:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'c',label='4')
    elif round(data_E[i],1) == 8:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'r',label='4')
    #elif round(data_E[i],1) == 9:
    #    ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'g',label='4')
    #elif round(data_E[i],1) == 10:
    #    ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'b',label='4')

import matplotlib.lines as mlines

r = mlines.Line2D([], [], color='r', markersize=15)
g = mlines.Line2D([], [], color='g', markersize=15)
b = mlines.Line2D([], [], color='b', markersize=15)
y = mlines.Line2D([], [], color='y', markersize=15)
k = mlines.Line2D([], [], color='k', markersize=15)
m = mlines.Line2D([], [], color='m', markersize=15)
c = mlines.Line2D([], [], color='c', markersize=15)
```

ANEXO II: Simulaciones de partida

```
ax.legend(labels=['2 MeV', '3 MeV', '4 MeV', '5 MeV', '6 MeV', '7 MeV', '8 MeV'], handles=[g, b, y, k, m, c, r])
```

Este código proporciona como resultado el número de protones que traspasan el pinhole y llegan al final de la trayectoria (4811), y la “Imagen 4” que se muestra a continuación:

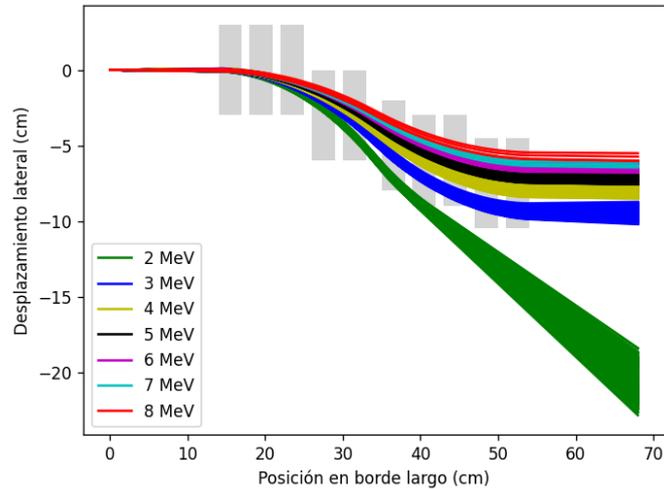


Imagen 4. Trayectorias de los protones de diferentes energías con condiciones angulares dadas en su paso por los imanes.

(Fuente: Elaboración propia a partir de los resultados del código que se muestra previamente)

Para poder estudiar la posición final de los protones se tiene el siguiente fragmento de código:

```
##### PLANO DE SALIDA #####

#plt.axvline(x=-2.606238099732431)
#plt.axhline(y=-0.17585962106528763)

for i in range(len(data_E)):
    #if data_E[i] == 1:
        #
            plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'r+')
            if round(data_E[i],1) == 2:
                plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'g+')
            elif round(data_E[i],1) == 3:
                plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'b+')
            elif round(data_E[i],1) == 4:
                plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'y+')
            elif round(data_E[i],1) == 5:
                plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'k+')
```

ANEXO II: Simulaciones de partida

```
elif round(data_E[i],1) == 6:
    plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'m+')
elif round(data_E[i],1) == 7:
    plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'c+')
elif round(data_E[i],1) == 8:
    plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'r+')
elif data_E[i] == 9:
    plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'g+')
elif data_E[i] == 10:
    plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'b+')

plt.legend(labels=['3 MeV','4 MeV','5 MeV','6 MeV','7 MeV','8 MeV'],handles=[b,y,k,m,c,r],bbox_to_anchor=(1.25, 1))
#plt.grid()
plt.xlabel('x (cm)')
plt.ylabel('y (cm)')
plt.title('Plano de salida ( $L_p = 7$  cm)')
plt.ylim([-1.5,1.5])
plt.xlim([-10.5,-5])
```

El resultado de ejecutar este fragmento de código se muestra en la “Imagen 5”.

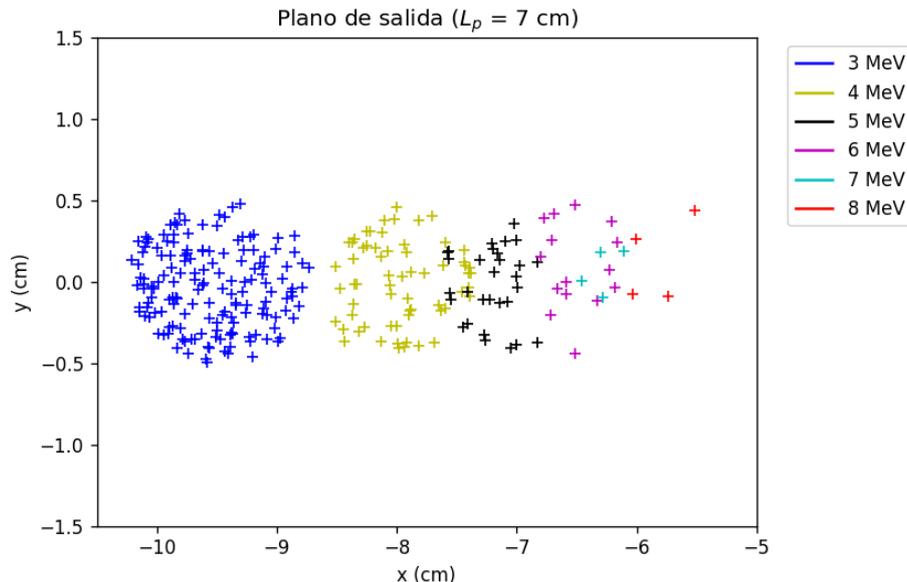


Imagen 5. Posición final de los protones.
(Fuente: Elaboración propia a partir de los resultados del código que se muestra previamente)

ANEXO II: Simulaciones de partida

Seguidamente, es posible estudiar los protones que se tienen en una ventana cuadrada de 5 x 5 mm, a la longitud final (68 cm), con centro la posición final de un protón de 5 MeV que sale a 0° de la fuente. La posición de dicho protón se obtiene mediante la simulación anterior ("2. Simulación simple").

```
## VENTANA DE SALIDA

# Nos ayudamos de la simulación simpl para saber la posición de la energía
central

E_c = 5 # MeV
x_c = -7.072954435142939 # posición en x=68 cm de un protón que sale con
theta=0, para la energía central
z_c = 0 # altura en x=68 de un protón que sale con theta=0

data_E_v=[]
data_x_v=[]
data_z_v=[]

L_v=0.5 # cm
a=0
for i in range(len(data_E)):
    if np.abs(np.array(data_x[i])[len(data_x[i])-1]*100-x_c) <= L_v/2 and
np.abs(np.array(data_z[i])[len(data_z[i])-1]*100-z_c) <= L_v/2:

        # Nos guardamos los datos de la ventana

        data_E_v.append(data_E[i])
        data_x_v.append(np.array(data_x[i])[len(data_x[i])-1]*100)
        data_z_v.append(np.array(data_z[i])[len(data_z[i])-1]*100)

        #plt.axvline(x=x_c)
        #plt.axhline(y=z_c)
        plt.plot((np.array(data_x[i])[len(data_x[i])-
1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100, '+')
        plt.grid()
        a=a+1
print(a)
```

ANEXO II: Simulaciones de partida

A continuación se muestra el resultado (“Imagen 6”):

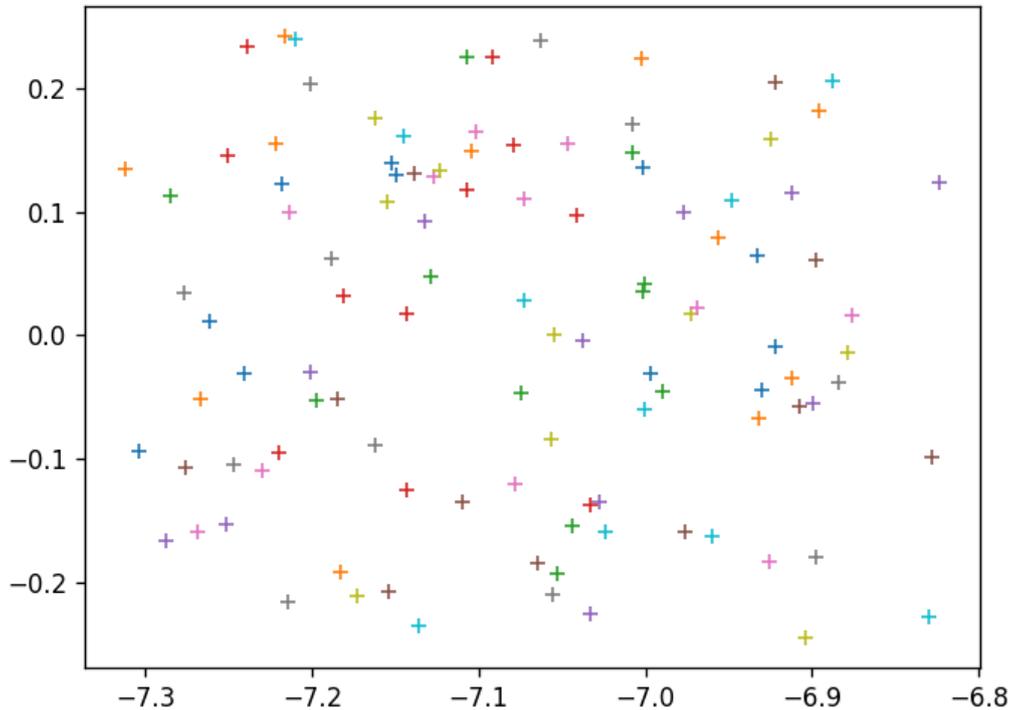


Imagen 6. Posición final de los protones.

(Fuente: Elaboración propia a partir de los resultados del código que se muestra previamente)

Finalmente se obtiene un histograma que representa el número de protones de cada energía en la ventana de salida (“Imagen 7”) y se calcula la fluencia.

```
# Histograma de fluencia de la ventana de salida

bin_size = 0.1

counts,energy=np.histogram(data_E_v,bins=np.arange(2,9,bin_size))

energy_center=np.zeros(len(energy)-1) # valores centrales de los bins

for i in range(len(energy)-1):
    energy_center[i]=(energy[i+1]+energy[i])/2

plt.bar(energy_center,counts,width=bin_size)
plt.xlabel('E (MeV)')
plt.ylabel('N')
plt.title(r'$L_p=$7 cm,   $E_c = $5 MeV')

df=pd.DataFrame([energy_center,counts]).transpose()
df.to_excel("run2.xlsx")

plt.xlim([3,7])
```

ANEXO II: Simulaciones de partida

El histograma queda según se muestra en la “Imagen 7”.

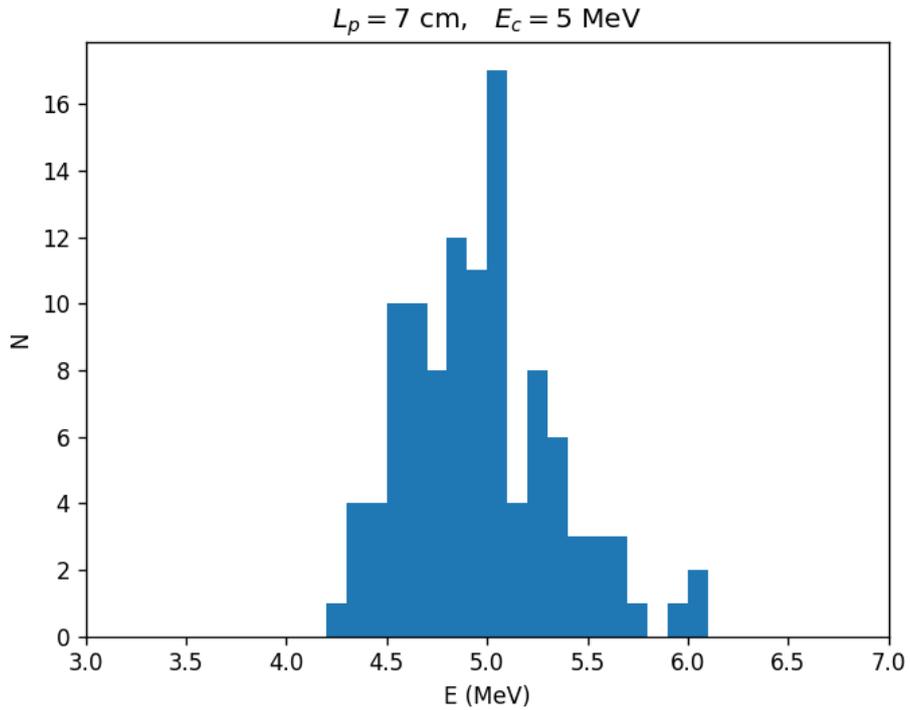


Imagen 7. Posición final de los protones.
(Fuente: Elaboración propia a partir de los resultados del código que se muestra previamente)

```
total_fluence=np.sum(counts)/(0.5**2)
print('Fluencia total en cm^-2', total_fluence)
```

Fluencia total en cm^{-2} 432.0

4. Bibliografía

Millán, P. J. (mayo de 2017). Caracterización de protones acelerados por láser y estudio de aplicaciones médicas. Valencia, Valencia, España. Obtenido de <https://www.educacion.gob.es/teseo/imprimirFicheroTesis.do?idFichero=nuLKZRm7SEI%3D>



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ANEXO III

Recopilación del código de las simulaciones efectuadas con
campo magnético leído sobre un imán

ÍNDICE DEL ANEXO III

1. Introducción.....	2
2. Simulaciones con campo magnético nulo entre imanes.....	3
2.1. Trazado de partículas simple	3
2.2. Mapa magnético de la simulación	27
3. Simulaciones con suma de campo magnético entre imanes.....	32
3.1. Trazado de partículas simple	32
3.2. Mapa magnético de la simulación	57
4. Simulaciones con aproximación del campo magnético en los huecos.....	62
4.1. Trazado de partículas simple	62
4.2. Mapa magnético de la simulación	77
4.3. Simulación compleja con nueve imanes	80
4.4. Extracción de la posición del protón.....	101

ÍNDICE DE IMÁGENES

Imagen 1. Campo Bz medido sobre un imán.....	96
Imagen 2. Campo Bz medido sobre un imán.....	97
Imagen 3. Campo Bz medido sobre un imán.....	100
Imagen 4. Campo Bz medido sobre un imán.....	101

1. Introducción

En este anexo se recoge el código de Python que se ha desarrollado para las distintas simulaciones que permiten obtener los resultados que aparecen en la memoria. Estos códigos comparten muchas similitudes con el que aparece en el apartado “2. Simulación simple” del “ANEXO II”, ya que se parte de los mismos datos de campo magnético para construir el mapa del campo magnético que afectará a los imanes, y utiliza los mismos conceptos físicos para llevar a cabo la simulación. No obstante, se ha facilitado la modificación de la configuración de los imanes y se han introducido cambios en la forma de construir el mapa magnético total del seleccionador.

Todas las simulaciones que se recogen en este anexo requieren de la importación de las mismas librerías y se utiliza la misma lectura del campo magnético sobre un imán:

```
import numpy as np
from matplotlib import pyplot as plt, patches
plt.rcParams['figure.dpi'] = 120
import pandas as pd
from scipy.interpolate import LinearNDInterpolator
```

```
##### Campo magnético #####

data=pd.read_csv("B_real.txt",delim_whitespace=True)
data_np=data.to_numpy()

x = 0.1*np.array(data_np[:,2],dtype='float64') # en MI sistema
y = 0.1*np.array(data_np[:,0],dtype='float64')
Bz = -1*np.array(data_np[:,3],dtype='float64')

precision = 1000 # solo para el plot

X = np.linspace(min(x), max(x), precision)
Y = np.linspace(min(y+2), max(y+2), precision)
X,Y = np.meshgrid(X,Y)
int_Bz = LinearNDInterpolator(list(zip(x, y+2)), Bz,fill_value=0)
```

Además, los parámetros del protón son idénticos y hay otras partes que también se comparten, pero como no todas necesitan de la definición de los parámetros del protón y para simplificar la comprensión, el resto del código que permite la obtención de los resultados deseados aparece copiado en cada uno de los apartados.

2. Simulaciones con campo magnético nulo entre imanes

Los resultados de esta simulación, que únicamente tiene en cuenta el campo magnético medido sobre la superficie de un imán, se recogen en la memoria para distintos valores de posición de los imanes.

2.1. Trazado de partículas simple

En este apartado se muestra el código que permite visualizar la trayectoria de las partículas. Además, se extraen los valores de posición, campo magnético y fuerza que afecta a la partícula durante su recorrido para poder graficarlos en Excel.

```
# Parámetros protón
```

```
m = 938.28 # MeV/c2  
m_kg = 1.67E-27 # kg  
q = 1.6E-19 # C  
c = 299792458 # m/s
```

```
energies = np.round(np.arange(4,11,1),1) # distribución discreta de  
energías de los protones
```

```
# Posiciones relativas de los imanes
```

```
x_rel_1=0  
x_rel_2=0  
x_rel_3=0.5  
x_rel_4=1  
x_rel_5=1.5  
x_rel_6=4  
x_rel_7=4.5  
x_rel_8=5  
x_rel_9=5.5  
x_rel_10=6
```

```
dist_12=0.5  
dist_23=0.5  
dist_34=0.5  
dist_45=0.5  
dist_56=3.8  
dist_67=0.5  
dist_78=0.5  
dist_89=0.5  
dist_910=0.5
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
y_rel_1=14
y_rel_2=y_rel_1+3+dist_12
y_rel_3=y_rel_2+3+dist_23
y_rel_4=y_rel_3+3+dist_34
y_rel_5=y_rel_4+3+dist_45
y_rel_6=y_rel_5+3+dist_56
y_rel_7=y_rel_6+3+dist_67
y_rel_8=y_rel_7+3+dist_78
y_rel_9=y_rel_8+3+dist_89
y_rel_10=y_rel_9+3+dist_910

# Figura e imanes

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlabel('Posición en borde largo (cm)')
ax.set_ylabel('Desplazamiento lateral (cm)')

iman1 = patches.Rectangle((y_rel_1, -3+x_rel_1), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman2 = patches.Rectangle((y_rel_2, -3+x_rel_2), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman3 = patches.Rectangle((y_rel_3, -3+x_rel_3), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman4 = patches.Rectangle((y_rel_4, -3+x_rel_4), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman5 = patches.Rectangle((y_rel_5, -3+x_rel_5), 3, 6,
edgecolor=None, facecolor="lightgrey")

iman6 = patches.Rectangle((y_rel_6, -3+x_rel_6), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman7 = patches.Rectangle((y_rel_7, -3+x_rel_7), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman8 = patches.Rectangle((y_rel_8, -3+x_rel_8), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman9 = patches.Rectangle((y_rel_9, -3+x_rel_9), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman10 = patches.Rectangle((y_rel_10, -3+x_rel_10), 3, 6,
edgecolor=None, facecolor="lightgrey")

ax.add_patch(iman1)
ax.add_patch(iman2)
ax.add_patch(iman3)
ax.add_patch(iman4)
ax.add_patch(iman5)

ax.add_patch(iman6)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
ax.add_patch(iman7)
ax.add_patch(iman8)
ax.add_patch(iman9)
ax.add_patch(iman10)

# Donde nos vamos a guardar las posiciones

data_E=[]

data_x=[]
data_y=[]
data_z=[]

data_Bz=[]

labels=[]

for i in range(len(energies)):

    v_p=c*np.sqrt(1-(1/(1 + energies[i]/m)**2)) # formula relativista para
la velocidad

    x = 0      # parte del punto 1.5 cm. Estas distancias están en m.
    y = 0
    z = 0

    vx = 0
    vy = v_p   # incide con velocidad v_y, perpendicular al lado largo de
los imanes
    vz = 0

    # Aquí nos vamos a guardar las posiciones en cada timestep:

    pos_x=[]
    pos_y=[]
    pos_z=[]

    pos_Bz=[]

    t = 0      # inicialización del tiempo
    dt = 1E-11 # timestep

    # Inicialización de los campos:

    Bx=0
    By=0
    Bz=0
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
Fx = 0
Fy = 0
Fz = 0

# El campo magnético se ha medido para la superficie del imán y los 0.5 cm
adyacentes a cada lado del borde largo

    while y<((y_rel_1-0.5)*0.01): #L_in. Aquí no hay campo magnético. El
protón no se desvía.

        x = x + vx * dt
        y = y + vy * dt
        z = z + vz * dt

        pos_x.append(x)
        pos_y.append(y)
        pos_z.append(z)

        pos_Bz.append(Bz)

        t += dt
        print('Free Antes. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx,
'Fy ', Fy, 'Fz ', Fz)

    # Posición del protón en x justo antes de entrar al primer iman. La
necesitaremos para la función de los campos:

    x_start_1=np.array(pos_x)[len(pos_x)-1] # Se guarda el último
elemento de la lista "pos_x" en la variable "x_start_1"

    ## PRIMER IMAN

    x_B=100*x_start_1-x_rel_1 # "x_start_1" está en m ->
Pasamos a cm
    y_B=0

    while y>=((y_rel_1-0.5)*0.01) and y<=((y_rel_1+3)*0.01):

        # Definimos posiciones para las funciones de los campos:

        x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
        y_B=y_B+100*vy*dt # lo mismo en y.

        # Calculamos los campos en cada timestep:

        Bz = int_Bz(x_B,y_B)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*-Bz)-(vz*-By))
Fy = q*((vz*-Bx)-(vx*-Bz))
Fz = q*((vx*-By)-(vy*-Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt
print('1 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_12=np.array(pos_x)[len(pos_x)-1]

#PRIMER HUECO

x_B=100*x_start_12-x_rel_1
y_B=0

while y>((y_rel_1+3)*0.01) and y<((y_rel_2)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.
    y_desf=y_B+3.5
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Calculamos los campos en cada timestep:

Bz = 0

# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*-Bz)-(vz*-By))
Fy = q*((vz*-Bx)-(vx*-Bz))
Fz = q*((vx*-By)-(vy*-Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt
print('1 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_2=np.array(pos_x)[len(pos_x)-1]

## SEGUNDO IMAN

x_B=100*x_start_2-x_rel_2
y_B=0

while y>=((y_rel_2)*0.01) and y<=((y_rel_2+3)*0.01):

# Definimos campos:
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
y_B=y_B+100*vy*dt # lo mismo en y.

Bz = int_Bz(x_B,y_B+0.5)

# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*-Bz)-(vz*-By))
Fy = q*((vz*-Bx)-(vx*-Bz))
Fz = q*((vx*-By)-(vy*-Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt
print('2 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_23=np.array(pos_x)[len(pos_x)-1]

#SEGUNDO HUECO

x_B=100*x_start_23-x_rel_2
y_B=0

while y>((y_rel_2+3)*0.01) and y<((y_rel_3)*0.01):
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Definimos posiciones para las funciones de los campos:

x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
y_B=y_B+100*vy*dt # lo mismo en y.
y_desf=y_B+3.5
# Calculamos los campos en cada timestep:

Bz = 0

# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*-Bz)-(vz*-By))
Fy = q*((vz*-Bx)-(vx*-Bz))
Fz = q*((vx*-By)-(vy*-Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt
print('2 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_3=np.array(pos_x)[len(pos_x)-1]

## TERCER IMAN
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_B=100*x_start_3-x_rel_3
y_B=0

while y>=((y_rel_3)*0.01) and y<=((y_rel_3+3)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(-1*Bz)

    t += dt
    print('3 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_34=np.array(pos_x)[len(pos_x)-1]

#TERCER HUECO
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_B=100*x_start_34-x_rel_3
y_B=0

while y>((y_rel_3+3)*0.01) and y<((y_rel_4)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.
    y_desf=y_B+3.5
    # Calculamos los campos en cada timestep:

    Bz = 0

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(-1*Bz)

    t += dt
    print('3 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_start_4=np.array(pos_x)[len(pos_x)-1]

## CUARTO IMAN

x_B=100*x_start_4-x_rel_4
y_B=0

while y>=((y_rel_4)*0.01) and y<=((y_rel_4+3)*0.01):

    # Definimos campos:

    x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*v_y*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((v_y*-Bz)-(v_z*-B_y))
    Fy = q*((v_z*-B_x)-(v_x*-B_z))
    Fz = q*((v_x*-B_y)-(v_y*-B_x))

    # Actualizamos posiciones en t+dt

    x = x + v_x * dt
    y = y + v_y * dt
    z = z + v_z * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    v_x = v_x + Fx/m_kg * dt
    v_y = v_y + Fy/m_kg * dt
    v_z = v_z + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(-1*Bz)

    t += dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
    print('4 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ',
', Fy, 'Fz ', Fz)

x_start_45=np.array(pos_x)[len(pos_x)-1]

#CUARTO HUECO

x_B=100*x_start_45-x_rel_4
y_B=0

while y>((y_rel_4+3)*0.01) and y<((y_rel_5)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.
    y_desf=y_B+3.5
    # Calculamos los campos en cada timestep:

    Bz = 0

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt
print('4 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ',
      Fy, 'Fz ', Fz)

x_start_5=np.array(pos_x)[len(pos_x)-1]

## QUINTO IMAN

x_B=100*x_start_5-x_rel_5
y_B=0

while y>=((y_rel_5)*0.01) and y<=((y_rel_5+3.5)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

pos_x.append(x)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt
print('5 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ', Fy, 'Fz ', Fz)

Bz = 0

## ESPACIO ENTRE SECTORES

while y>((y_rel_5+3.5)*0.01) and y<((y_rel_6-0.5)*0.01):

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt
    print('Free EntreSectores. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ', Fy, 'Fz ', Fz)

x_start_6=np.array(pos_x)[len(pos_x)-1]

#Bz = int_Bz(x_B,y_B)

## INVERTIMOS CAMPO

# SEXTO IMAN

x_B=100*x_start_6-x_rel_6
y_B=0
print(x_B)
while y>=((y_rel_6-0.5)*0.01) and y<=((y_rel_6+3)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
Bz = int_Bz(x_B,y_B)

# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*Bz)-(vz*By)) # INVERTIMOS CAMPOS
Fy = q*((vz*Bx)-(vx*Bz))
Fz = q*((vx*By)-(vy*Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('6 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_67=np.array(pos_x)[len(pos_x)-1]

#SEXTO HUECO

x_B=100*x_start_67-x_rel_6
y_B=0

while y>((y_rel_6+3)*0.01) and y<((y_rel_7)*0.01):

# Definimos posiciones para las funciones de los campos:

x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
y_B=y_B+100*vy*dt # lo mismo en y.
y_desf=y_B+3.5
# Calculamos los campos en cada timestep:

Bz = 0

# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*Bz)-(vz*By))
Fy = q*((vz*Bx)-(vx*Bz))
Fz = q*((vx*By)-(vy*Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('6 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_7=np.array(pos_x)[len(pos_x)-1]

# SEPTIMO IMAN

x_B=100*x_start_7-x_rel_7
y_B=0

while y>=((y_rel_7)*0.01) and y<=((y_rel_7+3)*0.01):
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Definimos campos:

x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
y_B=y_B+100*vy*dt # lo mismo en y.

Bz = int_Bz(x_B,y_B+0.5)

# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*Bz)-(vz*By)) # INVERTIMOS CAMPOS
Fy = q*((vz*Bx)-(vx*Bz))
Fz = q*((vx*By)-(vy*Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('7 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_78=np.array(pos_x)[len(pos_x)-1]

#SEPTIMO HUECO

x_B=100*x_start_78-x_rel_7
y_B=0

while y>((y_rel_7+3)*0.01) and y<((y_rel_8)*0.01):
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Definimos posiciones para las funciones de los campos:

x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
y_B=y_B+100*vy*dt # lo mismo en y.
y_desf=y_B+3.5
# Calculamos los campos en cada timestep:

Bz = 0

# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*Bz)-(vz*By))
Fy = q*((vz*Bx)-(vx*Bz))
Fz = q*((vx*By)-(vy*Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('7 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_8=np.array(pos_x)[len(pos_x)-1]

# OCTAVO IMAN
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_B=100*x_start_8-x_rel_8
y_B=0

while y>=((y_rel_8)*0.01) and y<=((y_rel_8+3)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt
    print('8 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_89=np.array(pos_x)[len(pos_x)-1]
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
#OCTAVO HUECO

x_B=100*x_start_89-x_rel_8
y_B=0

while y>((y_rel_8+3)*0.01) and y<((y_rel_9)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.
    y_desf=y_B+3.5
    # Calculamos los campos en cada timestep:

    Bz = 0

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
    print('8 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ', Fy, 'Fz ', Fz)

x_start_9=np.array(pos_x)[len(pos_x)-1]

# NOVENO IMAN

x_B=100*x_start_9-x_rel_9
y_B=0

while y>=((y_rel_9)*0.01) and y<=((y_rel_9+3)*0.01):

    # Definimos campos:

    x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*v_y*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((v_y*Bz)-(v_z*B_y)) # INVERTIMOS CAMPOS
    Fy = q*((v_z*B_x)-(v_x*B_z))
    Fz = q*((v_x*B_y)-(v_y*B_x))

    # Actualizamos posiciones en t+dt

    x = x + v_x * dt
    y = y + v_y * dt
    z = z + v_z * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    v_x = v_x + Fx/m_kg * dt
    v_y = v_y + Fy/m_kg * dt
    v_z = v_z + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
t += dt
print('9 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ', Fy, 'Fz ', Fz)

x_start_910=np.array(pos_x)[len(pos_x)-1]

#NOVENO HUECO

x_B=100*x_start_910-x_rel_9
y_B=0

while y>((y_rel_9+3)*0.01) and y<((y_rel_10)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*v_y*dt # lo mismo en y.
    y_desf=y_B+3.5
    # Calculamos los campos en cada timestep:

    Bz = 0

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((v_y*Bz)-(v_z*B_y))
    Fy = q*((v_z*B_x)-(v_x*B_z))
    Fz = q*((v_x*B_y)-(v_y*B_x))

    # Actualizamos posiciones en t+dt

    x = x + v_x * dt
    y = y + v_y * dt
    z = z + v_z * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    v_x = v_x + Fx/m_kg * dt
    v_y = v_y + Fy/m_kg * dt
    v_z = v_z + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('9 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ',
      Fy, 'Fz ', Fz)

x_start_10=np.array(pos_x)[len(pos_x)-1]

# DECIMO IMAN

x_B=100*x_start_10-x_rel_10
y_B=0

while y>=((y_rel_10)*0.01) and y<=((y_rel_10+3.5)*0.01):

    # Definimos campos:

    x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*v_y*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((v_y*Bz)-(v_z*B_y)) # INVERTIMOS CAMPOS
    Fy = q*((v_z*B_x)-(v_x*B_z))
    Fz = q*((v_x*B_y)-(v_y*B_x))

    # Actualizamos posiciones en t+dt

    x = x + v_x * dt
    y = y + v_y * dt
    z = z + v_z * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    v_x = v_x + Fx/m_kg * dt
    v_y = v_y + Fy/m_kg * dt
    v_z = v_z + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('10 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_11=np.array(pos_x)[len(pos_x)-1]
Bz=0

while y>((y_rel_10+3.5)*0.01) and y<=0.68:

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt
    print('Free Despues. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx,
'Fy ', Fy, 'Fz ', Fz)

x_start_12=np.array(pos_x)[len(pos_x)-1]

# Guardamos las posiciones en un array para trabajar mejor:

pos_x_arr=np.array(pos_x)
pos_y_arr=np.array(pos_y)
pos_z_arr=np.array(pos_z)
pos_Bz_arr=np.array(pos_Bz)

data_E.append(energies[i])
data_x.append(pos_x)
data_y.append(pos_y)
data_z.append(pos_z)

data_Bz.append(pos_Bz)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
line = ax.plot(pos_y_arr*100, pos_x_arr*100, label=energies[i])
labels.append(line[0])

ax.legend(handles=labels, labels=[f'{energy:.1f} MeV' for energy in
energies], loc='upper right', bbox_to_anchor=(1.25, 1.02))

## Posiciones finales de las energías centrales

tolerance = 0.1 # Tolerancia para la comparación de igualdad

for energy in energies:
    for i, data in enumerate(data_E):
        if abs(data - energy) < tolerance:
            print(f'x_final    {energy}    MeV:',
np.array(data_x[i])[len(data_x[i])-1] * 100)
            print(f'z_final    {energy}    MeV:',
np.array(data_z[i])[len(data_z[i])-1] * 100)
            print('-----')
```

2.2. Mapa magnético de la simulación

En este apartado se recoge el código que permite representar el campo magnético que se está considerando en la simulación anterior y que sirve, no sólo para ilustrar sobre el proceso que se lleva a cabo, sino que también como comprobación.

```
# Posiciones relativas de los imanes

x_rel_1=0
x_rel_2=0
x_rel_3=0.5
x_rel_4=1
x_rel_5=1.5
x_rel_6=4
x_rel_7=4.5
x_rel_8=5
x_rel_9=5.5
x_rel_10=6

dist_12=0.5
dist_23=0.5
dist_34=0.5
dist_45=0.5
dist_56=3.8
dist_67=0.5
dist_78=0.5
dist_89=0.5
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
dist_910=0.5

y_rel_1=0
y_rel_2=y_rel_1+3+dist_12
y_rel_3=y_rel_2+3+dist_23
y_rel_4=y_rel_3+3+dist_34
y_rel_5=y_rel_4+3+dist_45
y_rel_6=y_rel_5+3+dist_56
y_rel_7=y_rel_6+3+dist_67
y_rel_8=y_rel_7+3+dist_78
y_rel_9=y_rel_8+3+dist_89
y_rel_10=y_rel_9+3+dist_910

# Figura e imanes

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlabel('Posición en borde largo (cm)')
ax.set_ylabel('Desplazamiento lateral (cm)')

iman1 = patches.Rectangle((y_rel_1, -3+x_rel_1), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman2 = patches.Rectangle((y_rel_2, -3+x_rel_2), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman3 = patches.Rectangle((y_rel_3, -3+x_rel_3), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman4 = patches.Rectangle((y_rel_4, -3+x_rel_4), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman5 = patches.Rectangle((y_rel_5, -3+x_rel_5), 3, 6,
edgecolor=None,facecolor="lightgrey")

iman6 = patches.Rectangle((y_rel_6, -3+x_rel_6), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman7 = patches.Rectangle((y_rel_7, -3+x_rel_7), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman8 = patches.Rectangle((y_rel_8, -3+x_rel_8), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman9 = patches.Rectangle((y_rel_9, -3+x_rel_9), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman10 = patches.Rectangle((y_rel_10, -3+x_rel_10), 3, 6,
edgecolor=None,facecolor="lightgrey")

# Donde nos vamos a guardar las posiciones

nx = 1000 # Número de puntos en x_pix
ny = 1000 # Número de puntos en y_pix

x_pix = np.linspace(x_rel_1-3, x_rel_10+3, nx)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
y_pix = np.linspace(y_rel_1-0.5, y_rel_10+3.5, ny)

# Inicializar las matrices para almacenar los valores de x, y y Bz
data_x = np.empty((ny, nx))
data_y = np.empty((ny, nx))
data_Bz = np.empty((ny, nx))

for a in range(len(y_pix)):
    # PRIMER IMAN
    if y_pix[a] >= (y_rel_1-0.5) and y_pix[a] <= (y_rel_1+3):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b] = int_Bz(x_pix[b]-x_rel_1, y_pix[a]-y_rel_1)

    # PRIMER HUECO
    if y_pix[a] > (y_rel_1+3) and y_pix[a] < (y_rel_2):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b]=0

    ## SEGUNDO IMAN
    if y_pix[a]>=(y_rel_2) and y_pix[a]<=(y_rel_2+3):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b]=int_Bz(x_pix[b]-x_rel_2,y_pix[a]-y_rel_2)

    # SEGUNDO HUECO
    if y_pix[a]>=(y_rel_2+3) and y_pix[a]<=(y_rel_3):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b]=0

    ## TERCER IMAN
    if y_pix[a] >= (y_rel_3) and y_pix[a] <= (y_rel_3 + 3):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b]=int_Bz(x_pix[b]-x_rel_3,y_pix[a]-y_rel_3)

    # TERCER HUECO
    if y_pix[a] > (y_rel_3+3) and y_pix[a] < (y_rel_4):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
        data_Bz[a, b]=0

## CUARTO IMAN
if y_pix[a] >= (y_rel_4) and y_pix[a] <= (y_rel_4 + 3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b]=int_Bz(x_pix[b]-x_rel_4,y_pix[a]-y_rel_4)

# CUARTO HUECO
if y_pix[a] >= (y_rel_4 + 3) and y_pix[a] <= (y_rel_5):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b]=0

## QUINTO IMAN
if y_pix[a] >= (y_rel_5) and y_pix[a]<=(y_rel_5+3.5):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] =int_Bz(x_pix[b]-x_rel_5,y_pix[a]-y_rel_5)

## ESPACIO ENTRE SECTORES
if y_pix[a]>(y_rel_5+3.5) and y_pix[a]<(y_rel_6-0.5):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b]=0

## INVERTIMOS CAMPO

## SEXTO IMAN
if y_pix[a] >= (y_rel_6-0.5) and y_pix[a] <= (y_rel_6 + 3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b] - x_rel_6, y_pix[a] -
y_rel_6)

# SEXTO HUECO
if y_pix[a] >= (y_rel_6 + 3) and y_pix[a] <= (y_rel_7):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = 0

## SÉPTIMO IMAN
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
if y_pix[a] >= (y_rel_7) and y_pix[a] <= (y_rel_7 + 3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b] - x_rel_7, y_pix[a] -
y_rel_7)

# SÉPTIMO HUECO
if y_pix[a] >= (y_rel_7 + 3) and y_pix[a] <= (y_rel_8):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = 0

## OCTAVO IMAN
if y_pix[a] >= (y_rel_8) and y_pix[a] <= (y_rel_8 + 3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b] - x_rel_8, y_pix[a] -
y_rel_8)

# OCTAVO HUECO
if y_pix[a] >= (y_rel_8 + 3) and y_pix[a] <= (y_rel_9):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = 0

## NOVENO IMAN
if y_pix[a] >= (y_rel_9) and y_pix[a] <= (y_rel_9 + 3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b] - x_rel_9, y_pix[a] -
y_rel_9)

# NOVENO HUECO
if y_pix[a] >= (y_rel_9 + 3) and y_pix[a] <= (y_rel_10):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = 0

## DÉCIMO IMAN
if y_pix[a] >= (y_rel_10) and y_pix[a]<=(y_rel_10+3.5):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b]-x_rel_10,y_pix[a]-
y_rel_10)

print(data_x, data_y, data_Bz)
y_pix,x_pix = np.meshgrid(x_pix,y_pix)
plt.pcolormesh(x_pix, y_pix, data_Bz, shading='auto')
plt.xlabel('y (cm)')
plt.ylabel('x (cm)')
plt.colorbar(label='Bz (T)')
plt.title('Campo Bz del separador (interpolado)')
```

3. Simulaciones con suma de campo magnético entre imanes

Esta simulación es un poco más compleja que la anterior pues se suma el campo magnético que se tiene en los 5 mm anteriores y posteriores a cada imán para construir el campo magnético de los huecos.

3.1. Trazado de partículas simple

La trayectoria de las partículas se muestra por pantalla a partir del código aquí recogido. Al igual que en la simulación anterior, se extraen datos de posición, campo magnético y fuerza que afecta a la partícula en cada salto temporal para poder estudiarlas mediante gráficas de Excel.

```
# Parámetros protón
```

```
m = 938.28 # MeV/c2
m_kg = 1.67E-27 # kg
q = 1.6E-19 # C
c = 299792458 # m/s
```

```
energias = np.round(np.arange(4,11,1),1) # distribución discreta de
energías de los protones
```

```
# Posiciones relativas de los imanes
```

```
x_rel_1=0
x_rel_2=0
x_rel_3=0.5
x_rel_4=1
x_rel_5=1.5
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_rel_6=4
x_rel_7=4.5
x_rel_8=5
x_rel_9=5.5
x_rel_10=6

dist_12=0.5
dist_23=0.5
dist_34=0.5
dist_45=0.5
dist_56=3.8
dist_67=0.5
dist_78=0.5
dist_89=0.5
dist_910=0.5

y_rel_1=14
y_rel_2=y_rel_1+3+dist_12
y_rel_3=y_rel_2+3+dist_23
y_rel_4=y_rel_3+3+dist_34
y_rel_5=y_rel_4+3+dist_45
y_rel_6=y_rel_5+3+dist_56
y_rel_7=y_rel_6+3+dist_67
y_rel_8=y_rel_7+3+dist_78
y_rel_9=y_rel_8+3+dist_89
y_rel_10=y_rel_9+3+dist_910

# Figura e imanes

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlabel('Posición en borde largo (cm)')
ax.set_ylabel('Desplazamiento lateral (cm)')

iman1 = patches.Rectangle((y_rel_1, -3+x_rel_1), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman2 = patches.Rectangle((y_rel_2, -3+x_rel_2), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman3 = patches.Rectangle((y_rel_3, -3+x_rel_3), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman4 = patches.Rectangle((y_rel_4, -3+x_rel_4), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman5 = patches.Rectangle((y_rel_5, -3+x_rel_5), 3, 6,
edgecolor=None,facecolor="lightgrey")

iman6 = patches.Rectangle((y_rel_6, -3+x_rel_6), 3, 6,
edgecolor=None,facecolor="lightgrey")
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
iman7      =      patches.Rectangle((y_rel_7, -3+x_rel_7),      3,      6,
edgecolor=None, facecolor="lightgrey")
iman8      =      patches.Rectangle((y_rel_8, -3+x_rel_8),      3,      6,
edgecolor=None, facecolor="lightgrey")
iman9      =      patches.Rectangle((y_rel_9, -3+x_rel_9),      3,      6,
edgecolor=None, facecolor="lightgrey")
iman10     =      patches.Rectangle((y_rel_10, -3+x_rel_10),      3,      6,
edgecolor=None, facecolor="lightgrey")

ax.add_patch(iman1)
ax.add_patch(iman2)
ax.add_patch(iman3)
ax.add_patch(iman4)
ax.add_patch(iman5)

ax.add_patch(iman6)
ax.add_patch(iman7)
ax.add_patch(iman8)
ax.add_patch(iman9)
ax.add_patch(iman10)

# Donde nos vamos a guardar las posiciones

data_E=[]

data_x=[]
data_y=[]
data_z=[]

data_Bz=[]

labels=[]

for i in range(len(energies)):

    v_p=c*np.sqrt(1-(1/(1 + energies[i]/m)**2)) # formula relativista para
la velocidad

    x = 0      # parte del punto 1.5 cm. Estas distancias están en m.
    y = 0
    z = 0

    vx = 0
    vy = v_p   # incide con velocidad v_y, perpendicular al lado largo de
los imanes
    vz = 0

    # Aquí nos vamos a guardar las posiciones en cada timestep:
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
pos_x=[]
pos_y=[]
pos_z=[]

pos_Bz=[]

t = 0      # inicialización del tiempo
dt = 1E-11 # timestep

# Inicialización de los campos:

Bx=0
By=0
Bz=0

Fx = 0
Fy = 0
Fz = 0

# El campo magnético se ha medido para la superficie del imán y los 0.5 cm
adyacentes a cada lado del borde largo

    while y<((y_rel_1-0.5)*0.01): #L_in. Aquí no hay campo magnético. El
protón no se desvía.

        x = x + vx * dt
        y = y + vy * dt
        z = z + vz * dt

        pos_x.append(x)
        pos_y.append(y)
        pos_z.append(z)

        pos_Bz.append(Bz)

        t += dt
        print('Free Antes. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx,
'Fy ', Fy, 'Fz ', Fz)

# Posición del protón en x justo antes de entrar al primer iman. La
necesitaremos para la función de los campos:

x_start_1=np.array(pos_x)[len(pos_x)-1]      # Se guarda el último
elemento de la lista "pos_x" en la variable "x_start_1"

## PRIMER IMAN
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_B=100*x_start_1-x_rel_1 # "x_start_1" está en m ->
Pasamos a cm
y_B=0

while y>=((y_rel_1-0.5)*0.01) and y<=((y_rel_1+3)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    # Calculamos los campos en cada timestep:

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(-1*Bz)

    t += dt
    print('1 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_start_12=np.array(pos_x)[len(pos_x)-1]

#PRIMER HUECO

x_B=100*x_start_12-x_rel_1
y_B=0

while y>((y_rel_1+3)*0.01) and y<((y_rel_2)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.
    y_desf=y_B+3.5
    # Calculamos los campos en cada timestep:

    Bz = int_Bz(x_B,y_desf)+int_Bz(x_B-(x_rel_2-x_rel_1),y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(-1*Bz)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
t += dt
print('1 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ',
Fy, 'Fz ', Fz)

x_start_2=np.array(pos_x)[len(pos_x)-1]

## SEGUNDO IMAN

x_B=100*x_start_2-x_rel_2
y_B=0

while y>=((y_rel_2)*0.01) and y<=((y_rel_2+3)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
pos_Bz.append(-1*Bz)

t += dt
print('2 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ', Fy, 'Fz ', Fz)

x_start_23=np.array(pos_x)[len(pos_x)-1]

#SEGUNDO HUECO

x_B=100*x_start_23-x_rel_2
y_B=0

while y>((y_rel_2+3)*0.01) and y<((y_rel_3)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.
    y_desf=y_B+3.5
    # Calculamos los campos en cada timestep:

    Bz = int_Bz(x_B,y_desf)+int_Bz(x_B-(x_rel_3-x_rel_2),y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt
print('2 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_3=np.array(pos_x)[len(pos_x)-1]

## TERCER IMAN

x_B=100*x_start_3-x_rel_3
y_B=0

while y>=((y_rel_3)*0.01) and y<=((y_rel_3+3)*0.01):

    # Definimos campos:

    x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*v_y*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((v_y*-Bz)-(v_z*-B_y))
    Fy = q*((v_z*-B_x)-(v_x*-B_z))
    Fz = q*((v_x*-B_y)-(v_y*-B_x))

    # Actualizamos posiciones en t+dt

    x = x + v_x * dt
    y = y + v_y * dt
    z = z + v_z * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    v_x = v_x + Fx/m_kg * dt
    v_y = v_y + Fy/m_kg * dt
    v_z = v_z + Fz/m_kg * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt
print('3 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ', Fy, 'Fz ', Fz)

x_start_34=np.array(pos_x)[len(pos_x)-1]

#TERCER HUECO

x_B=100*x_start_34-x_rel_3
y_B=0

while y>((y_rel_3+3)*0.01) and y<((y_rel_4)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*v_y*dt # lo mismo en y.
    y_desf=y_B+3.5
    # Calculamos los campos en cada timestep:

    Bz = int_Bz(x_B,y_desf)+int_Bz(x_B-(x_rel_4-x_rel_3),y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((v_y*-Bz)-(v_z*-B_y))
    Fy = q*((v_z*-B_x)-(v_x*-B_z))
    Fz = q*((v_x*-B_y)-(v_y*-B_x))

    # Actualizamos posiciones en t+dt

    x = x + v_x * dt
    y = y + v_y * dt
    z = z + v_z * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt
print('3 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_4=np.array(pos_x)[len(pos_x)-1]

## CUARTO IMAN

x_B=100*x_start_4-x_rel_4
y_B=0

while y>=((y_rel_4)*0.01) and y<=((y_rel_4+3)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt
print('4 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_45=np.array(pos_x)[len(pos_x)-1]

#CUARTO HUECO

x_B=100*x_start_45-x_rel_4
y_B=0

while y>((y_rel_4+3)*0.01) and y<((y_rel_5)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.
    y_desf=y_B+3.5
    # Calculamos los campos en cada timestep:

    Bz = int_Bz(x_B,y_desf)+int_Bz(x_B-(x_rel_5-x_rel_4),y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

# Actualizamos posiciones en t+dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
# aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt
print('4 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_5=np.array(pos_x)[len(pos_x)-1]

## QUINTO IMAN

x_B=100*x_start_5-x_rel_5
y_B=0

while y>=((y_rel_5)*0.01) and y<=((y_rel_5+3.5)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt
print('5 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

Bz = 0

## ESPACIO ENTRE SECTORES

while y>((y_rel_5+3.5)*0.01) and y<((y_rel_6-0.5)*0.01):

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt
    print('Free EntreSectores. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx
', Fx, 'Fy ', Fy, 'Fz ', Fz)

x_start_6=np.array(pos_x)[len(pos_x)-1]

#Bz = int_Bz(x_B,y_B)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
## INVERTIMOS CAMPO

# SEXTO IMAN

x_B=100*x_start_6-x_rel_6
y_B=0
print(x_B)
while y>=((y_rel_6-0.5)*0.01) and y<=((y_rel_6+3)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
    print('6 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ', Fy, 'Fz ', Fz)

x_start_67=np.array(pos_x)[len(pos_x)-1]

#SEXTO HUECO

x_B=100*x_start_67-x_rel_6
y_B=0

while y>((y_rel_6+3)*0.01) and y<((y_rel_7)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*v_y*dt # lo mismo en y.
    y_desf=y_B+3.5
    # Calculamos los campos en cada timestep:

    Bz = int_Bz(x_B,y_desf)+int_Bz(x_B-(x_rel_7-x_rel_6),y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((v_y*Bz)-(v_z*By))
    Fy = q*((v_z*Bx)-(v_x*Bz))
    Fz = q*((v_x*By)-(v_y*Bx))

    # Actualizamos posiciones en t+dt

    x = x + v_x * dt
    y = y + v_y * dt
    z = z + v_z * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('6 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ',
      Fy, 'Fz ', Fz)

x_start_7=np.array(pos_x)[len(pos_x)-1]

# SEPTIMO IMAN

x_B=100*x_start_7-x_rel_7
y_B=0

while y>=((y_rel_7)*0.01) and y<=((y_rel_7+3)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

pos_x.append(x)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('7 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ', Fy, 'Fz ', Fz)

x_start_78=np.array(pos_x)[len(pos_x)-1]

#SEPTIMO HUECO

x_B=100*x_start_78-x_rel_7
y_B=0

while y>((y_rel_7+3)*0.01) and y<((y_rel_8)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.
    y_desf=y_B+3.5
    # Calculamos los campos en cada timestep:

    Bz = int_Bz(x_B,y_desf)+int_Bz(x_B-(x_rel_8-x_rel_7),y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('7 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ', Fy, 'Fz ', Fz)

x_start_8=np.array(pos_x)[len(pos_x)-1]

# OCTAVO IMAN

x_B=100*x_start_8-x_rel_8
y_B=0

while y>=((y_rel_8)*0.01) and y<=((y_rel_8+3)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('8 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy ',
      Fy, 'Fz ', Fz)

x_start_89=np.array(pos_x)[len(pos_x)-1]

#OCTAVO HUECO

x_B=100*x_start_89-x_rel_8
y_B=0

while y>((y_rel_8+3)*0.01) and y<((y_rel_9)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*v_y*dt # lo mismo en y.
    y_desf=y_B+3.5
    # Calculamos los campos en cada timestep:

    Bz = int_Bz(x_B,y_desf)+int_Bz(x_B-(x_rel_9-x_rel_8),y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((v_y*Bz)-(v_z*B_y))
    Fy = q*((v_z*B_x)-(v_x*B_z))
    Fz = q*((v_x*B_y)-(v_y*B_x))

    # Actualizamos posiciones en t+dt

    x = x + v_x * dt
    y = y + v_y * dt
    z = z + v_z * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('8 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_9=np.array(pos_x)[len(pos_x)-1]

# NOVENO IMAN

x_B=100*x_start_9-x_rel_9
y_B=0

while y>=((y_rel_9)*0.01) and y<=((y_rel_9+3)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
# aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('9 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_910=np.array(pos_x)[len(pos_x)-1]

#NOVENO HUECO

x_B=100*x_start_910-x_rel_9
y_B=0

while y>((y_rel_9+3)*0.01) and y<((y_rel_10)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.
    y_desf=y_B+3.5
    # Calculamos los campos en cada timestep:

    Bz = int_Bz(x_B,y_desf)+int_Bz(x_B-(x_rel_10-x_rel_9),y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('9 hueco. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_10=np.array(pos_x)[len(pos_x)-1]

# DECIMO IMAN

x_B=100*x_start_10-x_rel_10
y_B=0

while y>=((y_rel_10)*0.01) and y<=((y_rel_10+3.5)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    Bz = int_Bz(x_B,y_B+0.5)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*Bx)-(vx*Bz))
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
Fz = q*((vx*By)-(vy*Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt
print('10 iman. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx, 'Fy
', Fy, 'Fz ', Fz)

x_start_11=np.array(pos_x)[len(pos_x)-1]
Bz=0

while y>((y_rel_10+3.5)*0.01) and y<=0.68:

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt
    print('Free Despues. x: ',x*100, 'y ', y*100, 'Bz ', Bz, 'Fx ', Fx,
'Fy ', Fy, 'Fz ', Fz)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_start_12=np.array(pos_x)[len(pos_x)-1]

# Guardamos las posiciones en un array para trabajar mejor:

pos_x_arr=np.array(pos_x)
pos_y_arr=np.array(pos_y)
pos_z_arr=np.array(pos_z)
pos_Bz_arr=np.array(pos_Bz)

data_E.append(energies[i])
data_x.append(pos_x)
data_y.append(pos_y)
data_z.append(pos_z)

data_Bz.append(pos_Bz)

line = ax.plot(pos_y_arr*100, pos_x_arr*100, label=energies[i])
labels.append(line[0])

ax.legend(handles=labels, labels=[f'{energy:.1f} MeV' for energy in
energies], loc='upper right', bbox_to_anchor=(1.25, 1.02))

## Posiciones finales de las energías centrales

tolerance = 0.1 # Tolerancia para la comparación de igualdad

for energy in energies:
    for i, data in enumerate(data_E):
        if abs(data - energy) < tolerance:
            print(f'x_final {energy} MeV:',
np.array(data_x[i])[len(data_x[i])-1] * 100)
            print(f'z_final {energy} MeV:',
np.array(data_z[i])[len(data_z[i])-1] * 100)
            print('-----')
```

3.2. Mapa magnético de la simulación

```
# Posiciones relativas de los imanes

x_rel_1=0
x_rel_2=0
x_rel_3=0.5
x_rel_4=1
x_rel_5=1.5
x_rel_6=4
x_rel_7=4.5
x_rel_8=5
x_rel_9=5.5
x_rel_10=6

dist_12=0.5
dist_23=0.5
dist_34=0.5
dist_45=0.5
dist_56=3.8
dist_67=0.5
dist_78=0.5
dist_89=0.5
dist_910=0.5

y_rel_1=0
y_rel_2=y_rel_1+3+dist_12
y_rel_3=y_rel_2+3+dist_23
y_rel_4=y_rel_3+3+dist_34
y_rel_5=y_rel_4+3+dist_45
y_rel_6=y_rel_5+3+dist_56
y_rel_7=y_rel_6+3+dist_67
y_rel_8=y_rel_7+3+dist_78
y_rel_9=y_rel_8+3+dist_89
y_rel_10=y_rel_9+3+dist_910

# Figura e imanes

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlabel('Posición en borde largo (cm)')
ax.set_ylabel('Desplazamiento lateral (cm)')

iman1 = patches.Rectangle((y_rel_1, -3+x_rel_1), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman2 = patches.Rectangle((y_rel_2, -3+x_rel_2), 3, 6,
edgecolor=None, facecolor="lightgrey")
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
iman3      =      patches.Rectangle((y_rel_3, -3+x_rel_3),      3,      6,
edgecolor=None, facecolor="lightgrey")
iman4      =      patches.Rectangle((y_rel_4, -3+x_rel_4),      3,      6,
edgecolor=None, facecolor="lightgrey")
iman5      =      patches.Rectangle((y_rel_5, -3+x_rel_5),      3,      6,
edgecolor=None, facecolor="lightgrey")

iman6      =      patches.Rectangle((y_rel_6, -3+x_rel_6),      3,      6,
edgecolor=None, facecolor="lightgrey")
iman7      =      patches.Rectangle((y_rel_7, -3+x_rel_7),      3,      6,
edgecolor=None, facecolor="lightgrey")
iman8      =      patches.Rectangle((y_rel_8, -3+x_rel_8),      3,      6,
edgecolor=None, facecolor="lightgrey")
iman9      =      patches.Rectangle((y_rel_9, -3+x_rel_9),      3,      6,
edgecolor=None, facecolor="lightgrey")
iman10     =      patches.Rectangle((y_rel_10, -3+x_rel_10),      3,      6,
edgecolor=None, facecolor="lightgrey")

# Donde nos vamos a guardar las posiciones

nx = 500 # Número de puntos en x_pix
ny = 500 # Número de puntos en y_pix

x_pix = np.linspace(x_rel_1-3, x_rel_10+3, nx)
y_pix = np.linspace(y_rel_1-0.5, y_rel_10+3.5, ny)

# Inicializar las matrices para almacenar los valores de x, y y Bz
data_x = np.empty((ny, nx))
data_y = np.empty((ny, nx))
data_Bz = np.empty((ny, nx))

for a in range(len(y_pix)):
    # PRIMER IMAN
    if y_pix[a] >= (y_rel_1-0.5) and y_pix[a] <= (y_rel_1+3):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b] = int_Bz(x_pix[b]-x_rel_1, y_pix[a]-y_rel_1)

    # PRIMER HUECO
    if y_pix[a] >= (y_rel_1+3) and y_pix[a] <= (y_rel_2):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b]=((int_Bz(x_pix[b]-x_rel_1, y_pix[a]-y_rel_1)) +
(int_Bz(x_pix[b]-x_rel_2, y_pix[a]-y_rel_2)))

    ## SEGUNDO IMAN
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
if y_pix[a]>=(y_rel_2) and y_pix[a]<=(y_rel_2+3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b]=int_Bz(x_pix[b]-x_rel_2,y_pix[a]-y_rel_2)

# SEGUNDO HUECO
if y_pix[a]>=(y_rel_2+3) and y_pix[a]<=(y_rel_3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b]=(int_Bz(x_pix[b]-x_rel_2, y_pix[a]-y_rel_2)) +
(int_Bz(x_pix[b]-x_rel_3, y_pix[a]-y_rel_3))

## TERCER IMAN
if y_pix[a] >= (y_rel_3) and y_pix[a] <= (y_rel_3 + 3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b]=int_Bz(x_pix[b]-x_rel_3,y_pix[a]-y_rel_3)

# TERCER HUECO
if y_pix[a] > (y_rel_3+3) and y_pix[a] < (y_rel_4):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b]=(int_Bz(x_pix[b]-x_rel_3,y_pix[a]-
y_rel_3))+int_Bz(x_pix[b]-x_rel_4,y_pix[a]-y_rel_4))

## CUARTO IMAN
if y_pix[a] >= (y_rel_4) and y_pix[a] <= (y_rel_4 + 3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b]=int_Bz(x_pix[b]-x_rel_4,y_pix[a]-y_rel_4)

# CUARTO HUECO
if y_pix[a] >= (y_rel_4 + 3) and y_pix[a] <= (y_rel_5):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b]=(int_Bz(x_pix[b]-x_rel_4,y_pix[a]-
y_rel_4))+int_Bz(x_pix[b]-x_rel_5,y_pix[a]-y_rel_5))

## QUINTO IMAN
if y_pix[a] >= (y_rel_5) and y_pix[a]<=(y_rel_5+3.5):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] =int_Bz(x_pix[b]-x_rel_5,y_pix[a]-y_rel_5)

## ESPACIO ENTRE SECTORES
if y_pix[a]>(y_rel_5+3.5) and y_pix[a]<(y_rel_6-0.5):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b]=0

## INVERTIMOS CAMPO

## SEXTO IMAN
if y_pix[a] >= (y_rel_6-0.5) and y_pix[a] <= (y_rel_6 + 3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b] - x_rel_6, y_pix[a] -
y_rel_6)

# SEXTO HUECO
if y_pix[a] >= (y_rel_6 + 3) and y_pix[a] <= (y_rel_7):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*((int_Bz(x_pix[b] - x_rel_6, y_pix[a] -
y_rel_6)) + (int_Bz(x_pix[b] - x_rel_7, y_pix[a] - y_rel_7)))

## SÉPTIMO IMAN
if y_pix[a] >= (y_rel_7) and y_pix[a] <= (y_rel_7 + 3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b] - x_rel_7, y_pix[a] -
y_rel_7)

# SÉPTIMO HUECO
if y_pix[a] >= (y_rel_7 + 3) and y_pix[a] <= (y_rel_8):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*((int_Bz(x_pix[b] - x_rel_7, y_pix[a] -
y_rel_7)) + (int_Bz(x_pix[b] - x_rel_8, y_pix[a] - y_rel_8)))

## OCTAVO IMAN
if y_pix[a] >= (y_rel_8) and y_pix[a] <= (y_rel_8 + 3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b] - x_rel_8, y_pix[a] -
y_rel_8)

# OCTAVO HUECO
if y_pix[a] >= (y_rel_8 + 3) and y_pix[a] <= (y_rel_9):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*((int_Bz(x_pix[b] - x_rel_8, y_pix[a] -
y_rel_8)) + (int_Bz(x_pix[b] - x_rel_9, y_pix[a] - y_rel_9)))

## NOVENO IMAN
if y_pix[a] >= (y_rel_9) and y_pix[a] <= (y_rel_9 + 3):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b] - x_rel_9, y_pix[a] -
y_rel_9)

# NOVENO HUECO
if y_pix[a] >= (y_rel_9 + 3) and y_pix[a] <= (y_rel_10):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*((int_Bz(x_pix[b] - x_rel_9, y_pix[a] -
y_rel_9)) + (int_Bz(x_pix[b] - x_rel_10, y_pix[a] - y_rel_10)))

## DÉCIMO IMAN
if y_pix[a] >= (y_rel_10) and y_pix[a]<=(y_rel_10+3.5):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b]-x_rel_10,y_pix[a]-
y_rel_10)

print(data_x, data_y, data_Bz)
y_pix,x_pix = np.meshgrid(x_pix,y_pix)
plt.pcolormesh(x_pix, y_pix, data_Bz, shading='auto')
plt.xlabel('y (cm)')
plt.ylabel('x (cm)')

plt.colorbar(label='Bz (T)')
plt.title('Campo Bz del separador (interpolado)')
```

4. Simulaciones con aproximación del campo magnético en los huecos

En este apartado se recoge el código que ha permitido modificar la configuración de los imanes manteniendo un mapa del campo magnético generado por los imanes del seleccionador en la dirección del eje Z cercano al campo real que se ha podido medir.

Para ello, las simulaciones tienen en cuenta el campo magnético generado por un imán del que se partía al comienzo de este TFG y, para asignar los valores del campo magnético a los huecos entre imanes, que son de 0,5 cm, se utilizan los valores medidos para los 0,25 cm posteriores a cada imán y anteriores al siguiente.

4.1. Trazado de partículas simple

```
# Parámetros protón
```

```
m = 938.28 # MeV/c2  
m_kg = 1.67E-27 # kg  
q = 1.6E-19 # C  
c = 299792458 # m/s
```

```
energias = np.arange(4,11,1) # distribución discreta de energías de los  
protones
```

```
x_rel_1=0  
x_rel_2=0  
x_rel_3=0.5  
x_rel_4=1  
x_rel_5=1.5  
x_rel_6=4  
x_rel_7=4.5  
x_rel_8=5  
x_rel_9=5.5  
x_rel_10=6
```

```
dist_12=0.5  
dist_23=0.5  
dist_34=0.5  
dist_45=0.5  
dist_56=3.8  
dist_67=0.5  
dist_78=0.5  
dist_89=0.5  
dist_910=0.5
```

```
y_rel_1=14  
y_rel_2=y_rel_1+3+dist_12
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
y_rel_3=y_rel_2+3+dist_23
y_rel_4=y_rel_3+3+dist_34
y_rel_5=y_rel_4+3+dist_45
y_rel_6=y_rel_5+3+dist_56
y_rel_7=y_rel_6+3+dist_67
y_rel_8=y_rel_7+3+dist_78
y_rel_9=y_rel_8+3+dist_89
y_rel_10=y_rel_9+3+dist_910

# Figura e imanes

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlabel('Posición en borde largo (cm)')
ax.set_ylabel('Desplazamiento lateral (cm)')

iman1 = patches.Rectangle((y_rel_1,-3-x_rel_1), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman2 = patches.Rectangle((y_rel_2,-3-x_rel_2), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman3 = patches.Rectangle((y_rel_3,-3-x_rel_3), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman4 = patches.Rectangle((y_rel_4,-3-x_rel_4), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman5 = patches.Rectangle((y_rel_5,-3-x_rel_5), 3, 6,
edgecolor=None,facecolor="lightgrey")

iman6 = patches.Rectangle((y_rel_6,-3-x_rel_6), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman7 = patches.Rectangle((y_rel_7,-3-x_rel_7), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman8 = patches.Rectangle((y_rel_8,-3-x_rel_8), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman9 = patches.Rectangle((y_rel_9,-3-x_rel_9), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman10 = patches.Rectangle((y_rel_10,-3-x_rel_10), 3, 6,
edgecolor=None,facecolor="lightgrey")

ax.add_patch(iman1)
ax.add_patch(iman2)
ax.add_patch(iman3)
ax.add_patch(iman4)
ax.add_patch(iman5)

ax.add_patch(iman6)
ax.add_patch(iman7)
ax.add_patch(iman8)
ax.add_patch(iman9)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
ax.add_patch(iman10)

# Donde nos vamos a guardar las posiciones

data_E=[]

data_x=[]
data_y=[]
data_z=[]

data_Bz=[]
labels=[]

for i in range(len(energies)):

    v_p=c*np.sqrt(1-(1/(1 + energies[i]/m)**2)) # formula relativista
para la velocidad

    x = 0 # parte del punto 1.5 cm. Estas distancias están en m.
    y = 0
    z = 0

    vx = 0
    vy = v_p # incide con velocidad v_y, perpendicular al lado largo de
los imanes
    vz = 0

    # Aquí nos vamos a guardar las posiciones en cada timestep:

    pos_x=[]
    pos_y=[]
    pos_z=[]

    pos_Bz=[]

    t = 0 # inicialización del tiempo
    dt = 1E-11 # timestep

    # Inicialización de los campos:

    Bx=0
    By=0
    Bz=0

    while y<((y_rel_1-0.5)*0.01): #L_in. Aquí no hay campo magnético. El
protón no se desvía.

        x = x + vx * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
y = y + vy * dt
z = z + vz * dt

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

# Posición del protón en x justo antes de entrar al primer iman. La
necesitaremos para la función de los campos:

x_start_1=np.array(pos_x)[len(pos_x)-1]

## PRIMER IMAN

x_B=100*x_start_1+x_rel_1
y_B=0

while y>=((y_rel_1-0.5)*0.01) and y<((y_rel_1+3.25)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*v_y*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    # Calculamos los campos en cada timestep:

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((v_y*Bz)-(v_z*By))
    Fy = q*((v_z*Bx)-(v_x*Bz))
    Fz = q*((v_x*By)-(v_y*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
# aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

x_start_2=np.array(pos_x)[len(pos_x)-1]

## SEGUNDO IMAN

x_B=100*x_start_2+x_rel_2
y_B=0.25

while y>=((y_rel_2-0.25)*0.01) and y<((y_rel_2+3.25)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
# aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

x_start_3=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_3+x_rel_3
y_B=0.25

## TERCER IMAN

while y>=((y_rel_3-0.25)*0.01) and y<((y_rel_3+3.25)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
    # de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
    # simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
    # expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
# aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

x_start_4=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_4+x_rel_4
y_B=0.25

## CUARTO IMAN

while y>=((y_rel_4-0.25)*0.01) and y<((y_rel_4+3.25)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
# aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

x_start_5=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_5+x_rel_5
y_B=0.25

## QUINTO IMAN

while y>=((y_rel_5-0.25)*0.01) and y<=((y_rel_5+3.5)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

Bz = 0

## ESPACIO ENTRE SECTORES

while y > ((y_rel_5+3.5)*0.01) and y < ((y_rel_6-0.5)*0.01):

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

x_start_6=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_6+x_rel_6
y_B=0.25

## INVERTIMOS CAMPO
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# SEXTO IMAN

while y>=((y_rel_6-0.5)*0.01) and y<((y_rel_6+3.25)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(-1*Bz)

    t += dt

x_start_7=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_7+x_rel_7
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
y_B=0.25

# SEPTIMO IMAN

while y>=((y_rel_7-0.25)*0.01) and y<((y_rel_7+3.25)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(-1*Bz)

    t += dt

x_start_8=np.array(pos_x)[len(pos_x)-1]
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_B=100*x_start_8+x_rel_8
y_B=0.25

# OCTAVO IMAN

while y>=((y_rel_8-0.25)*0.01) and y<((y_rel_8+3.25)*0.01):

    # Definimos campos:

    x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*v_y*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((v_y*-Bz)-(v_z*-By)) # INVERTIMOS CAMPOS
    Fy = q*((v_z*-Bx)-(v_x*-Bz))
    Fz = q*((v_x*-By)-(v_y*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + v_x * dt
    y = y + v_y * dt
    z = z + v_z * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(-1*Bz)

    t += dt

x_start_9=np.array(pos_x)[len(pos_x)-1]
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_B=100*x_start_9+x_rel_9
y_B=0.25

# NOVENO IMAN

while y>=((y_rel_9-0.25)*0.01) and y<((y_rel_9+3.25)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(-1*Bz)

    t += dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_start_10=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_10+x_rel_10
y_B=0.25

# DECIMO IMAN

while y>=((y_rel_10-0.25)*0.01) and y<=((y_rel_10+3.5)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en la
simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(-1*Bz)

    t += dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x_start_11=np.array(pos_x)[len(pos_x)-1]
Bz=0

while y>((y_rel_10+3.5)*0.01) and y<=0.68:

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

x_start_12=np.array(pos_x)[len(pos_x)-1]

# Guardamos las posiciones en un array para trabajar mejor:

pos_x_arr=np.array(pos_x)
pos_y_arr=np.array(pos_y)
pos_z_arr=np.array(pos_z)

pos_Bz_arr=np.array(pos_Bz)

data_E.append(energies[i])
data_x.append(pos_x)
data_y.append(pos_y)
data_z.append(pos_z)

data_Bz.append(pos_Bz)

line = ax.plot(pos_y_arr*100, pos_x_arr*100, label=energies[i])
labels.append(line[0])

ax.legend(handles=labels, labels=[f'{energy:.1f} MeV' for energy in
energies], loc='upper right', bbox_to_anchor=(1.25, 1.02))
```

4.2. Mapa magnético de la simulación

```
# Posiciones relativas de los imanes

x_rel_1=0
x_rel_2=0
x_rel_3=-0.5
x_rel_4=-1
x_rel_5=-1.5
x_rel_6=-4
x_rel_7=-4.5
x_rel_8=-5
x_rel_9=-5.5
x_rel_10=-6

dist_12=0.5
dist_23=0.5
dist_34=0.5
dist_45=0.5
dist_56=3.8
dist_67=0.5
dist_78=0.5
dist_89=0.5
dist_910=0.5

y_rel_1=0
y_rel_2=y_rel_1+3+dist_12
y_rel_3=y_rel_2+3+dist_23
y_rel_4=y_rel_3+3+dist_34
y_rel_5=y_rel_4+3+dist_45
y_rel_6=y_rel_5+3+dist_56
y_rel_7=y_rel_6+3+dist_67
y_rel_8=y_rel_7+3+dist_78
y_rel_9=y_rel_8+3+dist_89
y_rel_10=y_rel_9+3+dist_910

# Figura e imanes

fig = plt.figure()

# Donde nos vamos a guardar las posiciones

nx = 1000 # Número de puntos en x_pix
ny = 1000 # Número de puntos en y_pix

x_pix = np.linspace(x_rel_1+3, x_rel_10-3, nx)
y_pix = np.linspace(y_rel_1-0.5, y_rel_10+3.5, ny)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Inicializar las matrices para almacenar los valores de x, y y Bz
data_x = np.empty((ny, nx))
data_y = np.empty((ny, nx))
data_Bz = np.empty((ny, nx))

for a in range(len(y_pix)):
    # PRIMER IMAN
    if y_pix[a] >= (y_rel_1-0.5) and y_pix[a] <= (y_rel_1+3.25):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b] = int_Bz(x_pix[b]-x_rel_1, y_pix[a]-y_rel_1)

    ## SEGUNDO IMAN
    if y_pix[a]>=(y_rel_2-0.25) and y_pix[a]<=(y_rel_2+3.25):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b]=int_Bz(x_pix[b]-x_rel_2,y_pix[a]-y_rel_2)

    ## TERCER IMAN
    if y_pix[a] >= (y_rel_3-0.25) and y_pix[a] <= (y_rel_3 + 3.25):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b]=int_Bz(x_pix[b]-x_rel_3,y_pix[a]-y_rel_3)

    ## CUARTO IMAN
    if y_pix[a] >= (y_rel_4-0.25) and y_pix[a] <= (y_rel_4 + 3.25):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b]=int_Bz(x_pix[b]-x_rel_4,y_pix[a]-y_rel_4)

    ## QUINTO IMAN
    if y_pix[a] >= (y_rel_5-0.25) and y_pix[a]<=(y_rel_5+3.5):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b] =int_Bz(x_pix[b]-x_rel_5,y_pix[a]-y_rel_5)

    ## ESPACIO ENTRE SECTORES
    if y_pix[a]>(y_rel_5+3.5) and y_pix[a]<(y_rel_6-0.5):
        for b in range(len(x_pix)):
            data_x[a, b] = x_pix[b]
            data_y[a, b] = y_pix[a]
            data_Bz[a, b]=0
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
## INVERTIMOS CAMPO

## SEXTO IMAN
if y_pix[a] >= (y_rel_6-0.5) and y_pix[a] <= (y_rel_6 + 3.25):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b] - x_rel_6, y_pix[a] -
y_rel_6)

## SÉPTIMO IMAN
if y_pix[a] >= (y_rel_7-0.25) and y_pix[a] <= (y_rel_7 + 3.25):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b] - x_rel_7, y_pix[a] -
y_rel_7)

## OCTAVO IMAN
if y_pix[a] >= (y_rel_8-0.25) and y_pix[a] <= (y_rel_8 + 3.25):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b] - x_rel_8, y_pix[a] -
y_rel_8)

## NOVENO IMAN
if y_pix[a] >= (y_rel_9-0.25) and y_pix[a] <= (y_rel_9 + 3.25):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b] - x_rel_9, y_pix[a] -
y_rel_9)

## DÉCIMO IMAN
if y_pix[a] >= (y_rel_10-0.25) and y_pix[a]<=(y_rel_10+3.5):
    for b in range(len(x_pix)):
        data_x[a, b] = x_pix[b]
        data_y[a, b] = y_pix[a]
        data_Bz[a, b] = -1*int_Bz(x_pix[b]-x_rel_10,y_pix[a]-
y_rel_10)

y_pix,x_pix = np.meshgrid(x_pix,y_pix)
plt.pcolormesh(x_pix, y_pix, data_Bz, shading='auto')
plt.xlabel('y (cm)')
plt.ylabel('x (cm)')
plt.colorbar(label='Bz (T)')
plt.title('Campo Bz del separador (interpolado)')
```

4.3. Simulación compleja con nueve imanes

Esta simulación permite estudiar la fluencia de protones y la separación lograda teniendo en cuenta la dispersión angular que tiene el haz a la entrada y su distribución energética (fluencia). No obstante, debido a los cambios que ha sido necesario introducir para adaptar el seleccionador al experimento de Salamanca se utilizan nueve imanes en lugar de diez, retirándose el último imán del segundo bloque.

Para que el código funcione correctamente, a parte de la lectura del campo magnético y la impartación de las librerías recogidas al principio de este ANEXO V y comunes a todos los apartados del mismo, es necesario importar las siguientes librerías:

```
from math import pi
from numpy.random import uniform
```

A continuación, se determinan los parámetros del protón, el pinhole o colimador circular y la distribución de fluencia. Estos últimos dos factores se han definido según los criterios de Michael Seimetz.

```
# Parámetros protón

m = 938.28 # MeV/c2
m_kg = 1.67E-27 # kg
q = 1.6E-19 # C
c = 299792458 # m/s
```

```
## Características pinhole circular

l_p = 0.002 # diámetro pinhole (m)
L_p = 0.30 # distancia fuente-pinhole

theta_max = np.arctan((l_p/2)/L_p) # máximo ángulo de observación desde
la fuente hasta el pinhole, en grados
#print('Máximo ángulo de observación fuente-pinhole (rad)', theta_max)
```

```
# Fluencia definida por Seimetz (2015) para el espectro de Fritzler
(2003)

def fluence(E):
    N0=1.62*10**12
    KBT=1.1
    E_max=10 # ojo, luego las energías de los protones deberían llegar
hasta menos de 10 para que no haya division por zero

    N = N0*np.exp(-E/KBT)
    return N
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

La simulación se recoge en el siguiente fragmento de código y se basa en la que se explicó en el apartado "3. Simulación compleja" del "ANEXO II". Tarda una hora en proporcionar una solución con un factor 1000 pero proporciona información muy valiosa.

```
delta = 0.01
energies = np.arange(4,8.01,delta) # distribución discreta de energías de
los protones

# Posiciones relativas de los imanes (respecto a x=0)

x_rel_1=-0
x_rel_2=-0
x_rel_3=-0
x_rel_4=-2.5
x_rel_5=-2.5
x_rel_6=-4
x_rel_7=-4.5
x_rel_8=-5
x_rel_9=-5.5
x_rel_10=-6

dist_12=0.5
dist_23=0.5
dist_34=0.5
dist_45=0.5
dist_56=3.8
dist_67=0.5
dist_78=0.5
dist_89=0.5
dist_910=0.5

y_rel_1=35
y_rel_2=y_rel_1+3+dist_12
y_rel_3=y_rel_2+3+dist_23
y_rel_4=y_rel_3+3+dist_34
y_rel_5=y_rel_4+3+dist_45
y_rel_6=y_rel_5+3+dist_56
y_rel_7=y_rel_6+3+dist_67
y_rel_8=y_rel_7+3+dist_78
y_rel_9=y_rel_8+3+dist_89
y_rel_10=y_rel_9+3+dist_910

## Nos guardamos los parámetros en listas para poder trabajar más
comodamente fuera de los bucles:

data_E = [] # energía del protón en cada timestep
# Posiciones xyz en cada timestep
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
data_x = []
data_y = []
data_z = []

data_Bz = []

factor = 1000 # factor de simplificación de la fluencia para reducir
tiempo computacional

for i in range(len(energies)):
    v_p = c*np.sqrt(1-(1/(1 + energies[i]/m)**2)) # formula relativista
para la velocidad

    N = fluence(energies[i])*delta/factor # dividimos para
rapidez computacional

    for j in np.arange(1,N,1): # Inicialización de
cada partícula

        # Vectores directores unitarios:

        rho = 1
        u = uniform(0.5*(1+np.cos(pi/18)),1)
        theta = np.arccos(2*u-1) # distribucion uniforme de puntos sobre
una esfera

        v = uniform(0,1)
        phi = 2*pi*v

        if theta < theta_max: # si la dirección del protón cae dentro de
lo que subtiende el pinhole, seguimos

            # Coordenadas esféricas:

            esf_y = rho*np.cos(theta)
            esf_x = rho*np.cos(phi)*np.sin(theta)
            esf_z = rho*np.sin(phi)*np.sin(theta)

            vx = v_p*esf_x
            vy = v_p*esf_y
            vz = v_p*esf_z

            # Inicializamos las posiciones

            x = 0
            y = 0
            z = 0
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
pos_x = []
pos_y = []
pos_z = []

pos_Bz = []

t = 0      # inicialización del tiempo
dt = 1E-11 # timestep

# Inicialización de los campos:

Bx = 0
By = 0
Bz = 0

while y < ((y_rel_1 - 0.5) * 0.01): # L_in. Aquí no hay campo
magnético. El protón no se desvía.

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

# Posición del protón en x justo antes de entrar al primer
iman. La necesitaremos para la función de los campos:

x_start_1 = np.array(pos_x)[len(pos_x)-1]

## PRIMER IMAN

x_B = 100 * x_start_1 - x_rel_1
y_B = 0

while y >= ((y_rel_1 - 0.5) * 0.01) and y < ((y_rel_1 + 3.25) * 0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B = x_B + 100 * vx * dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
        y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

        # Calculamos los campos en cada timestep:

        Bz = int_Bz(x_B,y_B)

        # Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

        Fx = q*((vy*Bz)-(vz*By))
        Fy = q*((vz*Bx)-(vx*Bz))
        Fz = q*((vx*By)-(vy*Bx))

        # Actualizamos posiciones en t+dt

        x = x + vx * dt
        y = y + vy * dt
        z = z + vz * dt

        # Actualizamos velocidades en t+dt, teniendo en cuenta
que la aceleración es F/m (igualar Lorentz a Newton)

        vx = vx + Fx/m_kg * dt
        vy = vy + Fy/m_kg * dt
        vz = vz + Fz/m_kg * dt

        # Nos guardamos las posiciones en las listas

        pos_x.append(x)
        pos_y.append(y)
        pos_z.append(z)

        pos_Bz.append(Bz)

        t += dt

x_start_2=np.array(pos_x)[len(pos_x)-1]

## SEGUNDO IMAN

x_B=100*x_start_2-x_rel_2
y_B=0.25

while y>=((y_rel_2-0.25)*0.01) and y<((y_rel_2+3.25)*0.01):

        # Definimos campos:
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
        x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
        y_B=y_B+100*v_y*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

        Bz = int_Bz(x_B,y_B)

        # Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

        Fx = q*((v_y*Bz)-(v_z*B_y))
        Fy = q*((v_z*B_x)-(v_x*B_z))
        Fz = q*((v_x*B_y)-(v_y*B_x))

        # Actualizamos posiciones en t+dt

        x = x + v_x * dt
        y = y + v_y * dt
        z = z + v_z * dt

        # Actualizamos velocidades en t+dt, teniendo en cuenta
que la aceleración es F/m (igualar Lorentz a Newton)

        v_x = v_x + Fx/m_kg * dt
        v_y = v_y + Fy/m_kg * dt
        v_z = v_z + Fz/m_kg * dt

        # Nos guardamos las posiciones en las listas

        pos_x.append(x)
        pos_y.append(y)
        pos_z.append(z)

        pos_Bz.append(Bz)

        t += dt

x_start_3=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_3-x_rel_3
y_B=0.25

## TERCER IMAN

while y>=((y_rel_3-0.25)*0.01) and y<((y_rel_3+3.25)*0.01):
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Definimos campos:

x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
y_B=y_B+100*v_y*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

Bz = int_Bz(x_B,y_B)

# Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

Fx = q*((v_y*B_z)-(v_z*B_y))
Fy = q*((v_z*B_x)-(v_x*B_z))
Fz = q*((v_x*B_y)-(v_y*B_x))

# Actualizamos posiciones en t+dt

x = x + v_x * dt
y = y + v_y * dt
z = z + v_z * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta
que la aceleración es F/m (igualar Lorentz a Newton)

v_x = v_x + F_x/m_kg * dt
v_y = v_y + F_y/m_kg * dt
v_z = v_z + F_z/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

x_start_4=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_4-x_rel_4
y_B=0.25

## CUARTO IMAN

while y>=((y_rel_4-0.25)*0.01) and y<((y_rel_4+3.25)*0.01):
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Definimos campos:

x_B=x_B+100*v_x*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
y_B=y_B+100*v_y*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

Bz = int_Bz(x_B,y_B)

# Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

Fx = q*((v_y*B_z)-(v_z*B_y))
Fy = q*((v_z*B_x)-(v_x*B_z))
Fz = q*((v_x*B_y)-(v_y*B_x))

# Actualizamos posiciones en t+dt

x = x + v_x * dt
y = y + v_y * dt
z = z + v_z * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta
que la aceleración es F/m (igualar Lorentz a Newton)

v_x = v_x + F_x/m_kg * dt
v_y = v_y + F_y/m_kg * dt
v_z = v_z + F_z/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

x_start_5=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_5-x_rel_5
y_B=0.25

## QUINTO IMAN
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
while y>=((y_rel_5-0.25)*0.01) and y<=((y_rel_5+3.5)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta
que la aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/mkg * dt
    vy = vy + Fy/mkg * dt
    vz = vz + Fz/mkg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

Bz = 0

## ESPACIO ENTRE SECTORES

while y>((y_rel_5+3.5)*0.01) and y<((y_rel_6-0.5)*0.01):
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

x_start_6=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_6-x_rel_6
y_B=0.25

## INVERTIMOS CAMPO

# SEXTO IMAN

while y>=((y_rel_6-0.5)*0.01) and y<((y_rel_6+3.25)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta
que la aceleración es F/m (igualar Lorentz a Newton)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_7=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_7-x_rel_7
y_B=0.25

# SEPTIMO IMAN

while y>=((y_rel_7-0.25)*0.01) and y<((y_rel_7+3.25)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
# Actualizamos velocidades en t+dt, teniendo en cuenta
que la aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_8=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_8-x_rel_8
y_B=0.25

# OCTAVO IMAN

while y>=((y_rel_8-0.25)*0.01) and y<((y_rel_8+3.25)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By)) # INVERTIMOS CAMPOS
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta
que la aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-1*Bz)

t += dt

x_start_9=np.array(pos_x)[len(pos_x)-1]

x_B=100*x_start_9-x_rel_9
y_B=0.25

# NOVENO IMAN

while y>=((y_rel_9-0.25)*0.01) and y<((y_rel_9+3.5)*0.01):

    # Definimos campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y. Le sumamos 1 porque en
la simulación de alberto el iman empieza en Y=1

    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta
la expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta
que la aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(-1*Bz)

    t += dt

while y>=((y_rel_9+3.5)*0.01) and y<=1.30:

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

#print(math.degrees(math.atan(vx/vy)))

# Guardamos las posiciones en un array para trabajar mejor:

pos_x_arr=np.array(pos_x)
pos_y_arr=np.array(pos_y)
pos_z_arr=np.array(pos_z)
pos_Bz_arr=np.array(pos_Bz)

data_E.append(energies[i])
data_x.append(pos_x)
data_y.append(pos_y)
data_z.append(pos_z)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
data_Bz.append(pos_Bz)
```

```
##### TRAYECTORIAS #####  
  
# Figura e imanes  
  
fig = plt.figure()  
ax = fig.add_subplot(111)  
ax.set_xlabel('Posición en borde largo (cm)')  
ax.set_ylabel('Desplazamiento lateral (cm)')  
  
iman1 = patches.Rectangle((y_rel_1,-3+x_rel_1), 3, 6,  
edgecolor=None,facecolor="lightgrey")  
iman2 = patches.Rectangle((y_rel_2,-3+x_rel_2), 3, 6,  
edgecolor=None,facecolor="lightgrey")  
iman3 = patches.Rectangle((y_rel_3,-3+x_rel_3), 3, 6,  
edgecolor=None,facecolor="lightgrey")  
iman4 = patches.Rectangle((y_rel_4,-3+x_rel_4), 3, 6,  
edgecolor=None,facecolor="lightgrey")  
iman5 = patches.Rectangle((y_rel_5,-3+x_rel_5), 3, 6,  
edgecolor=None,facecolor="lightgrey")  
  
iman6 = patches.Rectangle((y_rel_6,-3+x_rel_6), 3, 6,  
edgecolor=None,facecolor="lightgrey")  
iman7 = patches.Rectangle((y_rel_7,-3+x_rel_7), 3, 6,  
edgecolor=None,facecolor="lightgrey")  
iman8 = patches.Rectangle((y_rel_8,-3+x_rel_8), 3, 6,  
edgecolor=None,facecolor="lightgrey")  
iman9 = patches.Rectangle((y_rel_9,-3+x_rel_9), 3, 6,  
edgecolor=None,facecolor="lightgrey")  
iman10 = patches.Rectangle((y_rel_10,-3+x_rel_10), 3, 6,  
edgecolor=None,facecolor="lightgrey")  
  
ax.add_patch(iman1)  
ax.add_patch(iman2)  
ax.add_patch(iman3)  
ax.add_patch(iman4)  
ax.add_patch(iman5)  
  
ax.add_patch(iman6)  
ax.add_patch(iman7)  
ax.add_patch(iman8)  
ax.add_patch(iman9)  
#ax.add_patch(iman10)  
  
#print(energies)  
print(len(data_E))
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
#print(len(data_x))
#print(len(data_y))
#print(len(data_z))

for i in range(len(data_E)):
    #if round(data_E[i],1) == 1:
    #    plt.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'r',label='3')
    if round(data_E[i],1) == 2:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'g',label='4')
    elif round(data_E[i],1) == 3:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'b',label='5')
    elif round(data_E[i],1) == 4:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'y',label='4')
    elif round(data_E[i],1) == 5:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'k',label='7')
    elif round(data_E[i],1) == 6:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'m',label='8')
    elif round(data_E[i],1) == 7:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'c',label='4')
    elif round(data_E[i],1) == 8:
        ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'r',label='4')
    #elif round(data_E[i],1) == 9:
    #    ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'g',label='4')
    #elif round(data_E[i],1) == 10:
    #    ax.plot(np.array(data_y[i])*100,np.array(data_x[i])*100,'b',label='4')

import matplotlib.lines as mlines

r = mlines.Line2D([], [], color='r', markersize=15)
g = mlines.Line2D([], [], color='g', markersize=15)
b = mlines.Line2D([], [], color='b', markersize=15)
y = mlines.Line2D([], [], color='y', markersize=15)
k = mlines.Line2D([], [], color='k', markersize=15)
m = mlines.Line2D([], [], color='m', markersize=15)
c = mlines.Line2D([], [], color='c', markersize=15)

ax.legend(labels=['2 MeV', '3 MeV', '4 MeV', '5 MeV', '6 MeV', '7 MeV', '8 MeV'], handles=[g,b,y,k,m,c,r])
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

El resultado son 16903 protones y las trayectorias que se recogen en la “Imagen 1”.

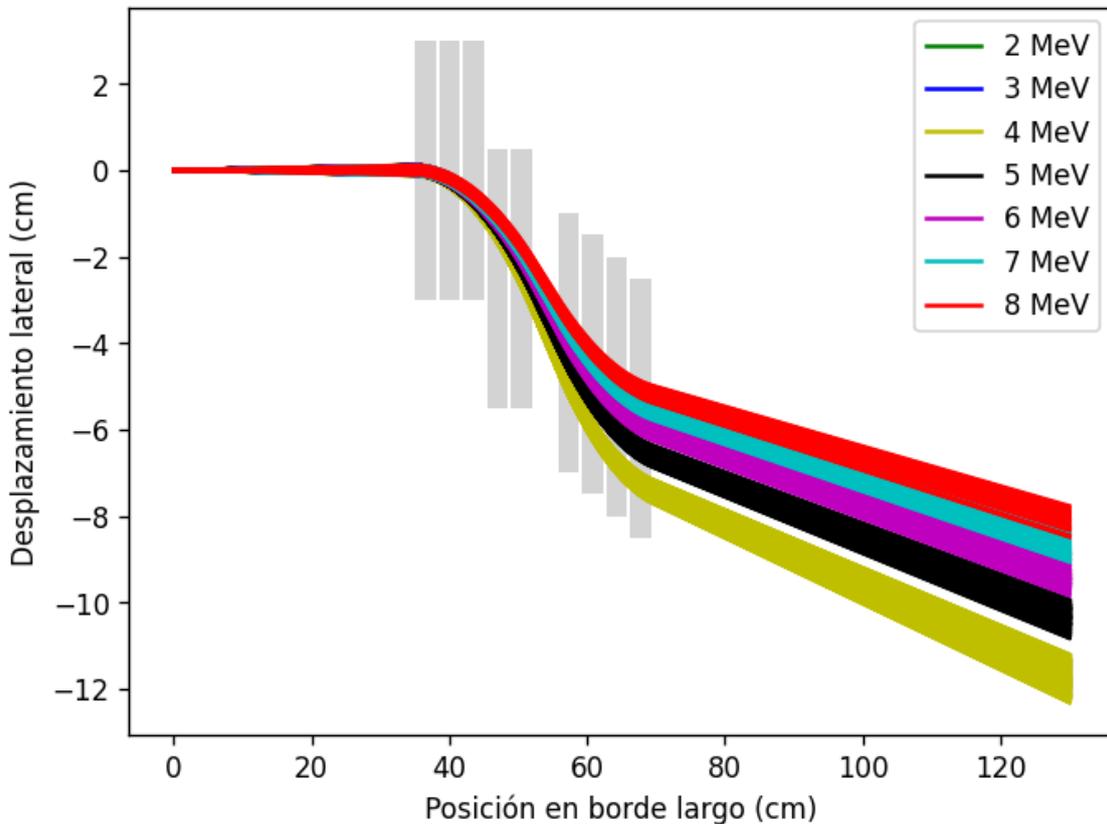


Imagen 1. Trazado de partículas.
(Fuente: Elaboración propia a partir de los resultados del código que se muestra previamente)

La posición final de los protones se grafica:

```
##### PLANO DE SALIDA #####  
  
for i in range(len(data_E)):  
    #if data_E[i] == 1:  
        # plt.plot((np.array(data_x[i])[len(data_x[i])-  
1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'r+')  
        if round(data_E[i],1) == 2:  
            plt.plot((np.array(data_x[i])[len(data_x[i])-  
1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'g+')  
        elif round(data_E[i],1) == 3:  
            plt.plot((np.array(data_x[i])[len(data_x[i])-  
1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'b+')  
        elif round(data_E[i],1) == 4:  
            plt.plot((np.array(data_x[i])[len(data_x[i])-  
1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'y+')  
        elif round(data_E[i],1) == 5:
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'k+')
elif round(data_E[i],1) == 6:
plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'m+')
elif round(data_E[i],1) == 7:
plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'c+')
elif round(data_E[i],1) == 8:
plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'r+')
elif data_E[i] == 9:
plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'g+')
elif data_E[i] == 10:
plt.plot((np.array(data_x[i])[len(data_x[i])-1])*100,(np.array(data_z[i])[len(data_z[i])-1])*100,'b+')

#plt.grid()
plt.xlabel('x (cm)')
plt.ylabel('y (cm)')
plt.title('Plano de salida ($L_p$ = 7 cm)')
plt.ylim([-1.5,1.5])
plt.xlim([-14,-5])
```

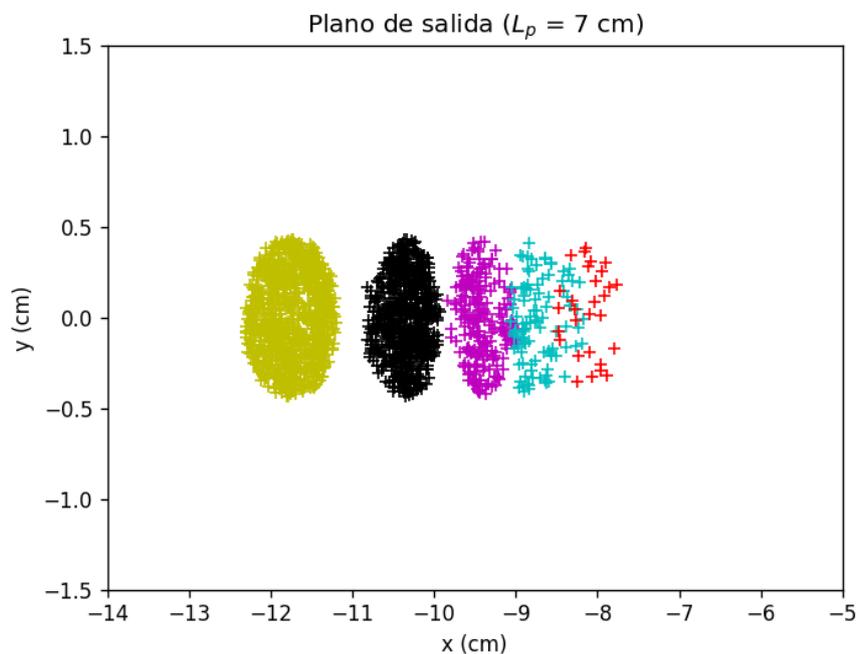


Imagen 2. Posición final de los protones.
(Fuente: Elaboración propia a partir de los resultados del código que se muestra previamente)

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

La siguiente parte del código proporciona la representación de la posición final de los protones en una ventana de 5 x 5 mm con centro la posición final de un protón de 5 MeV que sale de la fuente a 0°.

Esa simulación no está en ningún anexo por no sobrecargarlos ya que es muy parecida a la que se recoge en el apartado “4.1. Trazado de partículas” de este anexo, aunque se elimina el último imán y se colocan todos los imanes en las mismas posiciones que esta simulación. No obstante, quedo a disposición del tribunal para facilitársela si fuese necesario

```
E_c = 5 # MeV
x_c = -10.316757084025367 # POSICIÓN FINAL DE UN PROTÓN DE 5MeV QUE SALE
A 0º DE LA FUENTE
z_c = 0

data_E_v = []
data_x_v = []
data_z_v = []

intervals = np.linspace(4, 7, num=4)
colors = ['g', 'b', 'y', 'k', 'm', 'c', 'r', 'orange', 'purple', 'pink',
'brown', 'olive', 'navy', 'teal', 'lime',
'maroon', 'indigo', 'cyan', 'gold', 'sienna'] # Colores para
cada intervalo

L_v = 0.5 # cm
a = 0
b = 0
c = 0
not_matched = 0 # Variable para contar los puntos que no coinciden con
ningún intervalo
tolerance = (intervals[1]-intervals[0])/2 # Tolerancia para la
comparación de valores de energía

for i in range(len(data_E)):
    if np.abs(np.array(data_x[i])[len(data_x[i]) - 1] * 100 - x_c) <= L_v
/ 2 and np.abs(
        np.array(data_z[i])[len(data_z[i]) - 1] * 100 - z_c) <= L_v /
2:
        a += 1
        # Nos guardamos los datos de la ventana
        data_E_v.append(data_E[i])
        data_x_v.append(np.array(data_x[i])[len(data_x[i]) - 1] * 100)
        data_z_v.append(np.array(data_z[i])[len(data_z[i]) - 1] * 100)

        matched = False
        for j, interval in enumerate(intervals):
            if round(data_E[i], 0) >= interval and round(data_E[i], 0) <=
interval:
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

```
        color = colors[j % len(colors)] # Seleccionar el color
correspondiente al intervalo
        plt.plot((np.array(data_x[i])[len(data_x[i]) - 1]) * 100,
                 (np.array(data_z[i])[len(data_z[i]) - 1]) * 100,
color=color, marker='+')
        b += 1
        matched = True
        break

    if not matched:
        plt.plot((np.array(data_x[i])[len(data_x[i]) - 1]) * 100,
                 (np.array(data_z[i])[len(data_z[i]) - 1]) * 100,
color='r', marker='.')
        print(data_E[i])
        not_matched += 1

# Crear la leyenda con etiquetas correspondientes
legend_labels = [f'{interval:.1f} MeV' for interval in intervals]
plt.legend(
    handles=[plt.Line2D([], [], marker='+', linestyle='None',
color=colors[i % len(colors)]) for i in
              range(len(intervals))],
    labels=legend_labels, bbox_to_anchor=(1.25, 1.025))

# Agregar el centro de la ventana

plt.grid()
plt.show()

print(a)
print(b)
print(not_matched)
```

ANEXO III: Recopilación del código de las simulaciones efectuadas con campo magnético leído sobre un imán

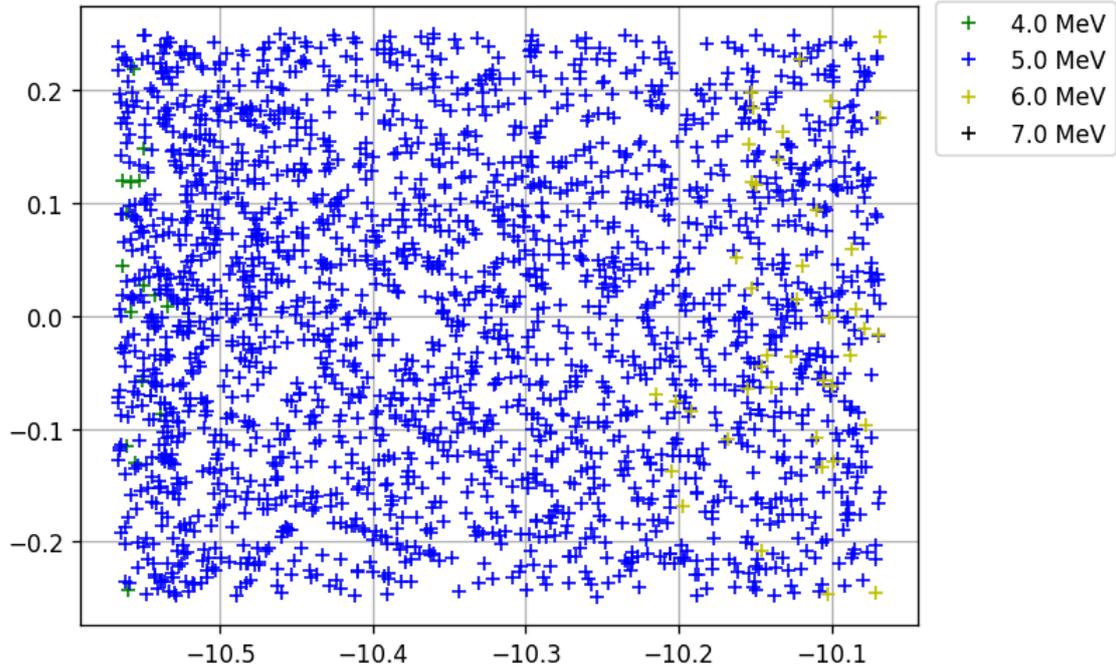


Imagen 3. Ventana final.

(Fuente: Elaboración propia a partir de los resultados del código que se muestra previamente)

El código permite además comprobar que llegan 2205 protones a la ventana.

Finalmente, el histograma de fluencia de la ventana queda:

```
# Histograma de fluencia de la ventana de salida

bin_size = 0.1

counts,energy=np.histogram(data_E_v,bins=np.arange(4,8,bin_size))

energy_center=np.zeros(len(energy)-1) # valores centrales de los bins

for i in range(len(energy)-1):
    energy_center[i]=(energy[i+1]+energy[i])/2

plt.bar(energy_center,counts,width=bin_size)
plt.xlabel('E (MeV)')
plt.ylabel('N')
plt.title(r'$L_p=30$ cm, $E_c =5$ MeV')

df=pd.DataFrame([energy_center,counts]).transpose()
df.to_excel("run2.xlsx")

plt.xlim([4,8])
```

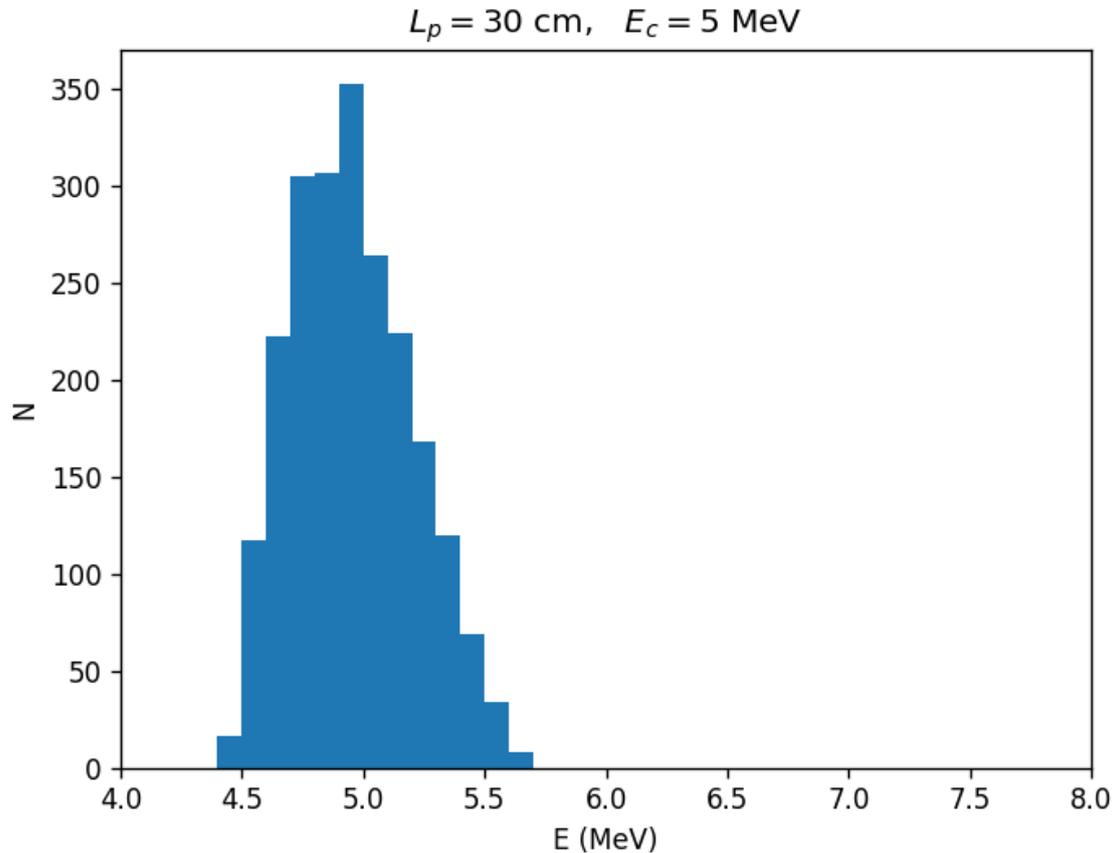


Imagen 4. Histograma de la ventana.
(Fuente: Elaboración propia a partir de los resultados del código que se muestra previamente)

4.4. Extracción de la posición del protón

Este fragmento de código permite extraer la posición del protón en X para valores de Y concretos:

```
y_interp=[y_rel_1+3.2,y_rel_3+3.2,y_rel_5+3.2,y_rel_6+3.2,y_rel_8+3.2,y_r  
el_9+3.2]  
x=[]  
  
for i in range(len(y_interp)):  
    x_interp = np.interp(y_interp[i]*0.01, pos_y_arr, pos_x_arr)  
    print(f'Para y = {y_interp[i]}, x = {x_interp*100}')  
    x.append(x_interp*100)  
  
df=pd.DataFrame([x]).transpose()  
df.to_excel("Temporal.xlsx")
```



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

ANEXO IV

Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

ÍNDICE DEL ANEXO IV

1. Introducción	2
2. Mapas magnéticos.....	2
3. Extracción de datos	4
4. Simulación con campo medido	5
4.1. Mapas magnéticos.....	5
4.2. Simulación Plano XY	8
4.3. Simulación Plano YZ	13
4.4. Datos de posición	13
5. Superposición trayectorias-By	13
6. Superposición trayectorias.....	19
7. Simulación con aire	24
8. Pérdida de energía en aire	32

1. Introducción

En este anexo se recoge el código que ha permitido visualizar las lecturas efectuadas del campo magnético y construir los mapas, así como las simulaciones efectuadas con dichos valores de campo.

En todas las simulaciones será necesario importar las mismas librerías:

```
import numpy as np
from matplotlib import pyplot as plt, patches
plt.rcParams['figure.dpi'] = 120
import pandas as pd
from scipy.interpolate import LinearNDInterpolator
```

El resto del código se recoge más abajo para cada simulación.

2. Mapas magnéticos

Este código permite representar el campo magnético a partir de las medidas que se han realizado y recogido en un Excel.

```
excel="NOMBRE DEL EXCEL QUE RECOGE LOS DATOS.xlsx"
```

```
data=pd.read_excel(excel)
data_np=data.to_numpy()

x = 0.1*np.array(data_np[:,0],dtype='float64') # en MI sistema
y = 0.1*np.array(data_np[:,2],dtype='float64')
Bx = -(1/1000)*np.array(data_np[:,5],dtype='float64')

precision = 1000 # solo para el plot

X = np.linspace(min(x), max(x),precision)
Y = np.linspace(min(y), max(y),precision)
X,Y = np.meshgrid(X,Y)
int_Bx = LinearNDInterpolator(list(zip(x, y)), Bx,fill_value=0)
Bx_1 = int_Bx(X,Y)

plt.pcolormesh(Y, X,Bx_1, shading='auto')
plt.xlabel('Y (cm)')
plt.ylabel('X (cm)')
plt.colorbar()
plt.title('Campo Bx (T)')
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
data=pd.read_excel(excel)
data_np=data.to_numpy()

x = 0.1*np.array(data_np[:,0],dtype='float64') # en MI sistema
y = 0.1*np.array(data_np[:,2],dtype='float64')
By = -(1/1000)*np.array(data_np[:,3],dtype='float64')

precision = 1000 # solo para el plot

X = np.linspace(min(x), max(x),precision)
Y = np.linspace(min(y), max(y),precision)
X,Y = np.meshgrid(X,Y)
int_By = LinearNDInterpolator(list(zip(x, y)), By,fill_value=0)
By_1 = int_By(X,Y)

plt.pcolormesh(Y, X,By_1, shading='auto')
plt.xlabel('Y (cm)')
plt.ylabel('X (cm)')
plt.colorbar()
plt.title('Campo By (T)')
```

```
data=pd.read_excel(excel)
data_np=data.to_numpy()

x = 0.1*np.array(data_np[:,0],dtype='float64') # en MI sistema
y = 0.1*np.array(data_np[:,2],dtype='float64')
Bz = (1/1000)*np.array(data_np[:,4],dtype='float64')

precision = 1000 # solo para el plot

X = np.linspace(min(x), max(x),precision)
Y = np.linspace(min(y), max(y),precision)
X,Y = np.meshgrid(X,Y)
int_Bz = LinearNDInterpolator(list(zip(x, y)), Bz,fill_value=0)
Bz_1 = int_Bz(X,Y)

plt.pcolormesh(Y, X,Bz_1, shading='auto')
plt.xlabel('Y (cm)')
plt.ylabel('X (cm)')
plt.colorbar()
plt.title('Campo Bz (T)')
```

3. Extracción de datos

El siguiente código se encarga de extraer un conjunto de medidas del campo magnético a partir de un intervalo de posición de dichas mediciones. Su propósito es facilitar la incorporación de estos datos al archivo de Excel que registra y representa el mapa magnético del bloque.

El código genera un Excel temporal que contiene los datos extraídos, lo cual permite copiarlos fácilmente y pegarlos en el archivo deseado. Además, de esta manera es posible realizar las correcciones necesarias para ubicar los valores de campo magnético en la posición adecuada a partir de la suma de los valores a la posición original.

Esta funcionalidad es especialmente útil para mantener la integridad y precisión de los datos del mapa magnético, ya que brinda la flexibilidad necesaria para realizar modificaciones y ajustes según sea necesario.

```
data1 = pd.read_excel(NOMBRE DEL ARCHIVO DEL QUE SE QUIEREN EXTRAER LOS
DATOS.xlsx')
data1_np = data1.to_numpy()

x1 = np.array(data1_np[:, 0], dtype='float64')
z1 = np.array(data1_np[:, 1], dtype='float64')
y1 = np.array(data1_np[:, 2], dtype='float64')
Bx1 = np.array(data1_np[:,5],dtype='float64')
By1 = np.array(data1_np[:,3],dtype='float64')
Bz1 = np.array(data1_np[:, 4], dtype='float64')

x = []
y = []
z = []
Bx = []
By = []
Bz = []

for i in range(len(x1)):
    if (x1[i] <= 50) & (y1[i]>=120.5) & (y1[i]<=160):
        x.append(x1[i])
        y.append(y1[i])
        z.append(z1[i])
        Bx.append(Bx1[i])
        By.append(By1[i])
        Bz.append(Bz1[i])

x = np.array(x)
y = np.array(y)
z = np.array(z)
Bx = np.array(Bx)
By = np.array(By)
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
Bz = np.array(Bz)

df=pd.DataFrame([x,z,y,By,Bz,Bx]).transpose()
df.to_excel("AAA_Temporal.xlsx")

precision = 1000

X = np.linspace(np.min(x), np.max(x), precision)
Y = np.linspace(np.min(y), np.max(y), precision)
X, Y = np.meshgrid(X, Y)

points = np.column_stack((x, y))
int_Bz = LinearNDInterpolator(points, Bz, fill_value=0)
Bz_1 = int_Bz(X, Y)

plt.pcolormesh(Y, X, Bz_1, shading='auto')
plt.xlabel('Y (cm)')
plt.ylabel('X (cm)')

plt.xticks(np.arange(np.min(y), np.max(y), step=2), fontsize=6)
plt.yticks(np.arange(np.min(x), np.max(x), step=5), fontsize=8)

plt.colorbar()
plt.title('Campo Bz (T)')
```

4. Simulación con campo medido

A continuación, se recoge el código que permite representar el mapa magnético total del seleccionador a partir de los 2 archivos de Excel creados para recoger las medidas, ya tratadas, del mapa magnético del primer y segundo bloque.

4.1. Mapas magnéticos

```
data1=pd.read_excel('1_Bloque_Grueso.xlsx')
data1_np=data1.to_numpy()

data2=pd.read_excel('2_Bloque_Fino.xlsx')
data2_np=data2.to_numpy()

x1 = 0.1*np.array(data1_np[:,0],dtype='float64')
x2 = 0.1*np.array(data2_np[:,0],dtype='float64')
x = np.concatenate((x1, 12.5-x2))

y1 = 0.1*np.array(data1_np[:,2],dtype='float64')
y2 = 0.1*np.array(data2_np[:,2],dtype='float64')
y = np.concatenate((y1, 34.8-y2))
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
Bx1 = 0*(1/1000)*np.array(data1_np[:,5],dtype='float64')
Bx2 = 0*(1/1000)*np.array(data2_np[:,5],dtype='float64')
Bx = np.concatenate((Bx1, Bx2))

precision = 1000 # solo para el plot

X = np.linspace(min(x), max(x),precision)
Y = np.linspace(min(y), max(y),precision)
X,Y = np.meshgrid(X,Y)
int_Bx = LinearNDInterpolator(list(zip(x, y)), Bx,fill_value=0)
Bx_1 = int_Bx(X,Y)

plt.pcolormesh(Y, X,Bx_1, shading='auto')
plt.xlabel('Y (cm)')
plt.ylabel('X (cm)')
plt.colorbar()
plt.title('Campo Bx (T)')
```

```
data1=pd.read_excel('2_CampoTotal.xlsx')
data1_np=data1.to_numpy()

data2=pd.read_excel('2_CampoTotal.xlsx')
data2_np=data2.to_numpy()

x1 = 0.1*np.array(data1_np[:,0],dtype='float64')
x2 = 0.1*np.array(data2_np[:,0],dtype='float64')
x = np.concatenate((x1, 12.5-x2))

y1 = 0.1*np.array(data1_np[:,2],dtype='float64')
y2 = 0.1*np.array(data2_np[:,2],dtype='float64')
y = np.concatenate((2.5+y1, 34.8+2.5-y2))

By1 = -1*(1/1000)*np.array(data1_np[:,3],dtype='float64')
By2 = 1*(1/1000)*np.array(data2_np[:,3],dtype='float64')
By = np.concatenate((By1, By2))
precision = 1000 # solo para el plot

X = np.linspace(min(x), max(x),precision)
Y = np.linspace(min(y), max(y),precision)
X,Y = np.meshgrid(X,Y)
int_By = LinearNDInterpolator(list(zip(x, y)), By,fill_value=0)
By_1 = int_By(X,Y)

plt.pcolormesh(Y, X, By_1, shading='auto')
plt.xlabel('Y (cm)')
plt.ylabel('X (cm)')
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
plt.xticks(np.arange(0, 40, step=1),fontsize=6)
plt.yticks(np.arange(-0.5, 13.5, step=0.5),fontsize=8)

plt.colorbar()
plt.title('Campo By (T)')
```

```
data1=pd.read_excel('1_CampoTotal.xlsx')
data1_np=data1.to_numpy()

data2=pd.read_excel('2_CampoTotal.xlsx')
data2_np=data2.to_numpy()

x1 = 0.1*np.array(data1_np[:,0],dtype='float64')
x2 = 0.1*np.array(data2_np[:,0],dtype='float64')
x = np.concatenate((x1, 12.5-x2))

y1 = 0.1*np.array(data1_np[:,2],dtype='float64')
y2 = 0.1*np.array(data2_np[:,2],dtype='float64')
y = np.concatenate((2.5+y1, 34.8+2.5-y2))

Bz1 = 0.99*(1/1000)*np.array(data1_np[:,4],dtype='float64')
Bz2 = 0.99*(1/1000)*np.array(data2_np[:,4],dtype='float64')
Bz = np.concatenate((Bz1, Bz2))

precision = 1000 # solo para el plot

X = np.linspace(min(x), max(x),precision)
Y = np.linspace(min(y), max(y),precision)
X,Y = np.meshgrid(X,Y)
int_Bz = LinearNDInterpolator(list(zip(x, y)), Bz,fill_value=0)
Bz_1 = int_Bz(X,Y)

plt.pcolormesh(Y, X,Bz_1, shading='auto')
plt.xlabel('Y (cm)')
plt.ylabel('X (cm)')

plt.xticks(np.arange(0, 40, step=1),fontsize=6)
plt.yticks(np.arange(-0.5, 13.5, step=0.5),fontsize=8)

plt.colorbar()
plt.title('Campo Bz (T)')
```

4.2. Simulación Plano XY

```
# Parámetros protón
```

```
m = 938.28 # MeV/c2  
m_kg = 1.67E-27 # kg  
q = 1.6E-19 # C  
c = 299792458 # m/s
```

```
energias = np.round(np.arange(4,11,1),1) # distribución discreta de  
energías de los protones
```

```
# Posiciones relativas de los imanes
```

```
x_rel_1=0  
x_rel_2=0  
x_rel_3=0.5  
x_rel_4=1  
x_rel_5=1.5  
x_rel_6=4  
x_rel_7=4.5  
x_rel_8=5  
x_rel_9=5.5  
x_rel_10=6
```

```
dist_12=0.5  
dist_23=0.5  
dist_34=0.5  
dist_45=0.5  
dist_56=3.8  
dist_67=0.5  
dist_78=0.5  
dist_89=0.5  
dist_910=0.5
```

```
y_rel_1=14  
y_rel_2=y_rel_1+3+dist_12  
y_rel_3=y_rel_2+3+dist_23  
y_rel_4=y_rel_3+3+dist_34  
y_rel_5=y_rel_4+3+dist_45  
y_rel_6=y_rel_5+3+dist_56  
y_rel_7=y_rel_6+3+dist_67  
y_rel_8=y_rel_7+3+dist_78  
y_rel_9=y_rel_8+3+dist_89  
y_rel_10=y_rel_9+3+dist_910
```

```
fig = plt.figure()
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
ax = fig.add_subplot(111)
ax.set_xlabel('Posición en borde largo (cm)')
ax.set_ylabel('Desplazamiento lateral (cm)')

iman1 = patches.Rectangle((y_rel_1, -3+x_rel_1), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman2 = patches.Rectangle((y_rel_2, -3+x_rel_2), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman3 = patches.Rectangle((y_rel_3, -3+x_rel_3), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman4 = patches.Rectangle((y_rel_4, -3+x_rel_4), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman5 = patches.Rectangle((y_rel_5, -3+x_rel_5), 3, 6,
edgecolor=None, facecolor="lightgrey")

iman6 = patches.Rectangle((y_rel_6, -3+x_rel_6), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman7 = patches.Rectangle((y_rel_7, -3+x_rel_7), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman8 = patches.Rectangle((y_rel_8, -3+x_rel_8), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman9 = patches.Rectangle((y_rel_9, -3+x_rel_9), 3, 6,
edgecolor=None, facecolor="lightgrey")
iman10 = patches.Rectangle((y_rel_10, -3+x_rel_10), 3, 6,
edgecolor=None, facecolor="lightgrey")

ax.add_patch(iman1)
ax.add_patch(iman2)
ax.add_patch(iman3)
ax.add_patch(iman4)
ax.add_patch(iman5)

ax.add_patch(iman6)
ax.add_patch(iman7)
ax.add_patch(iman8)
ax.add_patch(iman9)
ax.add_patch(iman10)

data_E=[]

data_x=[]
data_y=[]
data_z=[]

data_Bz=[]

labels=[]
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
for i in range(len(energies)):

    v_p=c*np.sqrt(1-(1/(1 + energies[i]/m)**2)) # formula relativista para
la velocidad

    x = 0          # parte del punto 1.5 cm. Estas distancias están en m.
    y = 0
    z = 0

    vx = 0
    vy = v_p      # incide con velocidad v_y, perpendicular al lado largo de
los imanes
    vz = 0

    # Aquí nos vamos a guardar las posiciones en cada timestep:

    pos_x=[]
    pos_y=[]
    pos_z=[]

    pos_Bz=[]

    t = 0        # inicialización del tiempo
    dt = 1E-11 # timestep

    # Inicialización de los campos:

    Bx=0
    By=0
    Bz=0

    Fx = 0
    Fy = 0
    Fz = 0

    # El campo magnético se ha medido para la superficie del imán y los 0.5 cm
adyacentes a cada lado del borde largo

    while y<((y_rel_1-0.5)*0.01): #L_in. Aquí no hay campo magnético. El
protón no se desvía.

        x = x + vx * dt
        y = y + vy * dt
        z = z + vz * dt

        pos_x.append(x)
        pos_y.append(y)
        pos_z.append(z)
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
pos_Bz.append(Bz)

t += dt

x_start=np.array(pos_x)[len(pos_x)-1] # Se guarda el último elemento
de la lista "pos_x" en la variable "x_start_1"

## COMIENZA EL PASO POR EL CAMPO MAGNÉTICO

x_B=x_start+3
y_B=0

while y>=((y_rel_1-0.5)*0.01) and y<=((y_rel_10+3.5)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    # Calculamos los campos en cada timestep:

    Bx = int_Bx(x_B,y_B)
    By = int_By(x_B,y_B)
    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-Bz)

t += dt

x_exit=np.array(pos_x)[len(pos_x)-1]

while y>((y_rel_10+3.5)*0.01) and y<=0.68:

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

x_impact=np.array(pos_x)[len(pos_x)-1]

# Guardamos las posiciones en un array para trabajar mejor:

pos_x_arr=np.array(pos_x)
pos_y_arr=np.array(pos_y)
pos_z_arr=np.array(pos_z)
pos_Bz_arr=np.array(pos_Bz)

data_E.append(energies[i])
data_x.append(pos_x)
data_y.append(pos_y)
data_z.append(pos_z)

data_Bz.append(pos_Bz)

line = ax.plot(pos_y_arr*100, pos_x_arr*100, label=energies[i])
labels.append(line[0])

ax.legend(handles=labels, labels=[f'{energy:.1f} MeV' for energy in
energies], loc='upper right', bbox_to_anchor=(1.25, 1.02))
```

4.3. Simulación Plano YZ

Es prácticamente igual que la anterior, pero se modifica el título del eje vertical del gráfico:

```
ax.set_ylabel('Desplazamiento vertical (cm)')
```

Y las últimas líneas de código por:

```
line = ax.plot(pos_y_arr*100, pos_z_arr*100, label=energies[i])
labels.append(line[0])
```

Siendo la última línea idéntica:

```
ax.legend(handles=labels, labels=[f'{energy:.1f} MeV' for energy in
energies], loc='upper right', bbox_to_anchor=(1.25, 1.02))
```

4.4. Datos de posición

Finalmente, el código que permite saber con exactitud la posición final de las partículas queda:

```
## Posiciones finales de las energías centrales

tolerance = 0.1 # Tolerancia para la comparación de igualdad

for energy in energies:
    for i, data in enumerate(data_E):
        if abs(data - energy) < tolerance:
            print(f'x_final      {energy}      MeV:',
np.array(data_x[i])[len(data_x[i])-1] * 100)
            print(f'z_final      {energy}      MeV:',
np.array(data_z[i])[len(data_z[i])-1] * 100)
            print('-----')
```

5. Superposición trayectorias-By

A partir de los mapas magnéticos que se realizaron en el apartado anterior ("4. Simulación con campo medido"), es posible crear una representación conjunta del campo By y el trazado de partículas:

```
# Parámetros protón

m = 938.28 # MeV/c2
m_kg = 1.67E-27 # kg
q = 1.6E-19 # C
c = 299792458 # m/s
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
data1=pd.read_excel('2_CampoTotal.xlsx')
data1_np=data1.to_numpy()

data2=pd.read_excel('2_CampoTotal.xlsx')
data2_np=data2.to_numpy()

x1 = 0.1*np.array(data1_np[:,0],dtype='float64')
x2 = 0.1*np.array(data2_np[:,0],dtype='float64')
x = np.concatenate((x1, 12.5-x2))

y1 = 0.1*np.array(data1_np[:,2],dtype='float64')
y2 = 0.1*np.array(data2_np[:,2],dtype='float64')
y = np.concatenate((2.5+y1, 34.8+2.5-y2))

By1 = (1/1000)*np.array(data1_np[:,3],dtype='float64')
By2 = -(1/1000)*np.array(data2_np[:,3],dtype='float64')
By = np.concatenate((By1, By2))
precision = 1000 # solo para el plot

X = np.linspace(min(x), max(x),precision)
Y = np.linspace(min(y), max(y),precision)
X,Y = np.meshgrid(X-3,Y+14)
int_By = LinearNDInterpolator(list(zip(x-3, y+14)), By,fill_value=0)
By_1 = int_By(X,Y)

plt.pcolormesh(Y, X, By_1, shading='auto')
plt.xlabel('Y (cm)')
plt.ylabel('X (cm)')

plt.colorbar()
plt.title('Campo By (T)')

# Procesar los datos para el segundo gráfico
energies = np.round(np.arange(4, 11, 1), 1)

# Posiciones relativas de los imanes
x_rel_1=0
x_rel_2=0
x_rel_3=0.5
x_rel_4=1
x_rel_5=1.5
x_rel_6=4
x_rel_7=4.5
x_rel_8=5
x_rel_9=5.5
x_rel_10=6
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
dist_12=0.5
dist_23=0.5
dist_34=0.5
dist_45=0.5
dist_56=3.8
dist_67=0.5
dist_78=0.5
dist_89=0.5
dist_910=0.5

y_rel_1=14
y_rel_2=y_rel_1+3+dist_12
y_rel_3=y_rel_2+3+dist_23
y_rel_4=y_rel_3+3+dist_34
y_rel_5=y_rel_4+3+dist_45
y_rel_6=y_rel_5+3+dist_56
y_rel_7=y_rel_6+3+dist_67
y_rel_8=y_rel_7+3+dist_78
y_rel_9=y_rel_8+3+dist_89
y_rel_10=y_rel_9+3+dist_910

iman1 = patches.Rectangle((y_rel_1,-3+x_rel_1), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman2 = patches.Rectangle((y_rel_2,-3+x_rel_2), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman3 = patches.Rectangle((y_rel_3,-3+x_rel_3), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman4 = patches.Rectangle((y_rel_4,-3+x_rel_4), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman5 = patches.Rectangle((y_rel_5,-3+x_rel_5), 3, 6,
edgecolor=None,facecolor="lightgrey")

iman6 = patches.Rectangle((y_rel_6,-3+x_rel_6), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman7 = patches.Rectangle((y_rel_7,-3+x_rel_7), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman8 = patches.Rectangle((y_rel_8,-3+x_rel_8), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman9 = patches.Rectangle((y_rel_9,-3+x_rel_9), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman10 = patches.Rectangle((y_rel_10,-3+x_rel_10), 3, 6,
edgecolor=None,facecolor="lightgrey")

'''ax.add_patch(iman1)
ax.add_patch(iman2)
ax.add_patch(iman3)
ax.add_patch(iman4)
ax.add_patch(iman5)
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
ax.add_patch(iman6)
ax.add_patch(iman7)
ax.add_patch(iman8)
ax.add_patch(iman9)
ax.add_patch(iman10)'''

data_E = []
data_x = []
data_y = []
data_z = []
data_Bz = []
labels = []

for i in range(len(energies)):

    v_p=c*np.sqrt(1-(1/(1 + energies[i]/m)**2)) # formula relativista
para la velocidad

    x = 0      # parte del punto 1.5 cm. Estas distancias están en m.
    y = 0
    z = 0

    vx = 0
    vy = v_p   # incide con velocidad v_y, perpendicular al lado largo
de los imanes
    vz = 0

    # Aquí nos vamos a guardar las posiciones en cada timestep:

    pos_x=[]
    pos_y=[]
    pos_z=[]

    pos_Bz=[]

    t = 0      # inicialización del tiempo
    dt = 1E-11 # timestep

    # Inicialización de los campos:

    Bx=0
    By=0
    Bz=0

    Fx = 0
    Fy = 0
    Fz = 0
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
# El campo magnético se ha medido para la superficie del imán y los 0.5
cm adyacentes a cada lado del borde largo

while y<((y_rel_1-1)*0.01): #L_in. Aquí no hay campo magnético. El
protón no se desvía.

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

x_start=np.array(pos_x)[len(pos_x)-1] # Se guarda el último
elemento de la lista "pos_x" en la variable "x_start_1"

## COMIENZA EL PASO POR EL CAMPO MAGNÉTICO

x_B=x_start+3
y_B=0

while y>=((y_rel_1-1)*0.01) and y<=((y_rel_10+4)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep. Partimos
de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    # Calculamos los campos en cada timestep:

    Bx = int_Bx(x_B,y_B)
    By = int_By(x_B,y_B)
    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*-Bz)-(vz*-By))
    Fy = q*((vz*-Bx)-(vx*-Bz))
    Fz = q*((vx*-By)-(vy*-Bx))
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
# aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-Bz)

t += dt

x_exit=np.array(pos_x)[len(pos_x)-1]

while y>((y_rel_10+4)*0.01) and y<=0.68:

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

x_impact=np.array(pos_x)[len(pos_x)-1]

# Guardamos las posiciones en un array para trabajar mejor:

pos_x_arr=np.array(pos_x)
pos_y_arr=np.array(pos_y)
pos_z_arr=np.array(pos_z)
pos_Bz_arr=np.array(pos_Bz)
```

```
data_E.append(energies[i])
data_x.append(pos_x)
data_y.append(pos_y)
data_z.append(pos_z)

data_Bz.append(pos_Bz)

plt.plot(pos_y_arr * 100, pos_x_arr * 100, label=f'{energies[i]:.1f}
MeV')
plt.legend(loc='upper right', bbox_to_anchor=(1.70, 1.02))

plt.show()
```

6. Superposición trayectorias

A partir de los mapas de campo magnético Bx y Bz construidos en “4. Simulación con campo medido” y construyendo otros 4 mapas con distintos sentidos para By:

```
data1=pd.read_excel('2_CampoTotal.xlsx')
data1_np=data1.to_numpy()

data2=pd.read_excel('2_CampoTotal.xlsx')
data2_np=data2.to_numpy()

x1 = 0.1*np.array(data1_np[:,0],dtype='float64')
x2 = 0.1*np.array(data2_np[:,0],dtype='float64')
x = np.concatenate((x1, 12.5-x2))

y1 = 0.1*np.array(data1_np[:,2],dtype='float64')
y2 = 0.1*np.array(data2_np[:,2],dtype='float64')
y = np.concatenate((2.5+y1, 34.8+2.5-y2))

By1 = -1*(1/1000)*np.array(data1_np[:,3],dtype='float64')
By2 = -1*(1/1000)*np.array(data2_np[:,3],dtype='float64')
By3 = (1/1000)*np.array(data1_np[:,3],dtype='float64')
By4 = (1/1000)*np.array(data2_np[:,3],dtype='float64')
By_case1 = np.concatenate((By1, By4))
By_case2 = np.concatenate((By1, By2))
By_case3 = np.concatenate((By3, By2))
By_case4 = np.concatenate((By3, By4))

precision = 1000 # solo para el plot

X = np.linspace(min(x), max(x),precision)
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
Y = np.linspace(min(y), max(y),precision)
X,Y = np.meshgrid(X,Y)
int_By_1 = LinearNDInterpolator(list(zip(x, y)), By_case1,fill_value=0)
int_By_2 = LinearNDInterpolator(list(zip(x, y)), By_case2,fill_value=0)
int_By_3 = LinearNDInterpolator(list(zip(x, y)), By_case3,fill_value=0)
int_By_4 = LinearNDInterpolator(list(zip(x, y)), By_case4,fill_value=0)
```

Es posible obtener la trayectoria que seguirían las partículas inicializadas en distintas alturas a partir del siguiente código:

```
# Parámetros protón

m = 938.28 # MeV/c2
m_kg = 1.67E-27 # kg
q = 1.6E-19 # C
c = 299792458 # m/s
```

```
energias = np.round(np.arange(4,11,1),1) # distribución discreta de
energías de los protones

# Posiciones relativas de los imanes

x_rel_1=0
x_rel_2=0
x_rel_3=0.5
x_rel_4=1
x_rel_5=1.5
x_rel_6=4
x_rel_7=4.5
x_rel_8=5
x_rel_9=5.5
x_rel_10=6

dist_12=0.5
dist_23=0.5
dist_34=0.5
dist_45=0.5
dist_56=3.8
dist_67=0.5
dist_78=0.5
dist_89=0.5
dist_910=0.5

y_rel_1=14
y_rel_2=y_rel_1+3+dist_12
y_rel_3=y_rel_2+3+dist_23
y_rel_4=y_rel_3+3+dist_34
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
y_rel_5=y_rel_4+3+dist_45
y_rel_6=y_rel_5+3+dist_56
y_rel_7=y_rel_6+3+dist_67
y_rel_8=y_rel_7+3+dist_78
y_rel_9=y_rel_8+3+dist_89
y_rel_10=y_rel_9+3+dist_910

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlabel('Posición en borde largo (cm)')
ax.set_ylabel('Desplazamiento vertical (cm)')

data_E=[]

data_x=[]
data_y=[]
data_z=[]

data_Bz=[]

labels=[]

int_By_list=[int_By_1, int_By_2, int_By_3, int_By_4]
z_list=[0.0001,0.0004,-0.0001,-0.0004]

for j in range(4):
    int_By=int_By_list[j]

    for i in range(len(energies)):

        v_p=c*np.sqrt(1-(1/(1 + energies[i]/m)**2)) # formula relativista
para la velocidad

        x = 0
        y = 0
        z = z_list[j]

        vx = 0
        vy = v_p # incide con velocidad v_y, perpendicular al lado
largo de los imanes
        vz = 0

        # Aquí nos vamos a guardar las posiciones en cada timestep:

        pos_x=[]
        pos_y=[]
        pos_z=[]
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
pos_Bz=[]

t = 0      # inicialización del tiempo
dt = 1E-11 # timestep

# Inicialización de los campos:

Bx=0
By=0
Bz=0

Fx = 0
Fy = 0
Fz = 0

# El campo magnético se ha medido para la superficie del imán y los
0.5 cm adyacentes a cada lado del borde largo

while y<((y_rel_1-0.5)*0.01): #L_in. Aquí no hay campo magnético.
El protón no se desvía.

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    t += dt

x_start=np.array(pos_x)[len(pos_x)-1] # Se guarda el último
elemento de la lista "pos_x" en la variable "x_start_1"

## COMIENZA EL PASO POR EL CAMPO MAGNÉTICO

x_B=x_start+3
y_B=0

while y>=((y_rel_1-0.5)*0.01) and y<=((y_rel_10+3.5)*0.01):

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
# Calculamos los campos en cada timestep:

Bx = int_Bx(x_B,y_B)
By = int_By(x_B,y_B)
Bz = int_Bz(x_B,y_B)

# Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

Fx = q*((vy*-Bz)-(vz*-By))
Fy = q*((vz*-Bx)-(vx*-Bz))
Fz = q*((vx*-By)-(vy*-Bx))

# Actualizamos posiciones en t+dt

x = x + vx * dt
y = y + vy * dt
z = z + vz * dt

# Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

vx = vx + Fx/m_kg * dt
vy = vy + Fy/m_kg * dt
vz = vz + Fz/m_kg * dt

# Nos guardamos las posiciones en las listas

pos_x.append(x)
pos_y.append(y)
pos_z.append(z)

pos_Bz.append(-Bz)

t += dt

x_exit=np.array(pos_x)[len(pos_x)-1]

while y>((y_rel_10+3.5)*0.01) and y<=0.68:

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    pos_x.append(x)
    pos_y.append(y)
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
pos_z.append(z)

pos_Bz.append(Bz)

t += dt

x_impact=np.array(pos_x)[len(pos_x)-1]

# Guardamos las posiciones en un array para trabajar mejor:

pos_x_arr=np.array(pos_x)
pos_y_arr=np.array(pos_y)
pos_z_arr=np.array(pos_z)
pos_Bz_arr=np.array(pos_Bz)

data_E.append(energies[i])
data_x.append(pos_x)
data_y.append(pos_y)
data_z.append(pos_z)

data_Bz.append(pos_Bz)

line = ax.plot(pos_y_arr*100, pos_z_arr*100, label=energies[i])
labels.append(line[0])

ax.legend(handles=labels, labels=[f'{energy:.1f} MeV' for energy in
energies], loc='upper right', bbox_to_anchor=(1.25, 1.02))
```

7. Simulación con aire

Para tener en cuenta los efectos del aire y a partir de las lecturas del campo magnético realizadas en “4. Simulación con campo medido” y de los valores de pérdida de energía en aire que se recoge en el siguiente apartado “8. Pérdida de energía en aire”, es posible utilizar el código que se muestra a continuación para dar con una simulación más real en caso de experimentos fuera de cámara de vacío.

```
# Parámetros protón

m = 938.28 # MeV/c2
m_kg = 1.67E-27 # kg
q = 1.6E-19 # C
c = 299792458 # m/s
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
data=pd.read_csv("H_in_Air.txt",delim_whitespace=True)
data_np=data.to_numpy()

E = np.array(data_np[:,0],dtype='float64') # en MI sistema
unities = np.array(data_np[:,1],dtype='str')
r = np.array(data_np[:,2],dtype='float64')

r=r*1000          #Paso de KeV/micrometro a MeV/m

for i in range(len(unities)):
    if unities[i]=='keV':
        E[i]=E[i]*0.001
        unities[i]='MeV'
```

```
energies = np.round(np.arange(4,11,1),1) # distribución discreta de
energías de los protones

# Posiciones relativas de los imanes

x_rel_1=0
x_rel_2=0
x_rel_3=0.5
x_rel_4=1
x_rel_5=1.5
x_rel_6=4
x_rel_7=4.5
x_rel_8=5
x_rel_9=5.5
x_rel_10=6

dist_12=0.5
dist_23=0.5
dist_34=0.5
dist_45=0.5
dist_56=3.8
dist_67=0.5
dist_78=0.5
dist_89=0.5
dist_910=0.5

y_rel_1=14
y_rel_2=y_rel_1+3+dist_12
y_rel_3=y_rel_2+3+dist_23
y_rel_4=y_rel_3+3+dist_34
y_rel_5=y_rel_4+3+dist_45
y_rel_6=y_rel_5+3+dist_56
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
y_rel_7=y_rel_6+3+dist_67
y_rel_8=y_rel_7+3+dist_78
y_rel_9=y_rel_8+3+dist_89
y_rel_10=y_rel_9+3+dist_910

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlabel('Posición en borde largo (cm)')
ax.set_ylabel('Desplazamiento lateral (cm)')

iman1 = patches.Rectangle((y_rel_1,-3+x_rel_1), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman2 = patches.Rectangle((y_rel_2,-3+x_rel_2), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman3 = patches.Rectangle((y_rel_3,-3+x_rel_3), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman4 = patches.Rectangle((y_rel_4,-3+x_rel_4), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman5 = patches.Rectangle((y_rel_5,-3+x_rel_5), 3, 6,
edgecolor=None,facecolor="lightgrey")

iman6 = patches.Rectangle((y_rel_6,-3+x_rel_6), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman7 = patches.Rectangle((y_rel_7,-3+x_rel_7), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman8 = patches.Rectangle((y_rel_8,-3+x_rel_8), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman9 = patches.Rectangle((y_rel_9,-3+x_rel_9), 3, 6,
edgecolor=None,facecolor="lightgrey")
iman10 = patches.Rectangle((y_rel_10,-3+x_rel_10), 3, 6,
edgecolor=None,facecolor="lightgrey")

ax.add_patch(iman1)
ax.add_patch(iman2)
ax.add_patch(iman3)
ax.add_patch(iman4)
ax.add_patch(iman5)

ax.add_patch(iman6)
ax.add_patch(iman7)
ax.add_patch(iman8)
ax.add_patch(iman9)
ax.add_patch(iman10)

data_E=[]

data_x=[]
data_y=[]
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
data_z=[]

data_Bz=[]

labels=[]

for i in range(len(energies)):

    v_p=c*np.sqrt(1-(1/(1 + energies[i]/m)**2)) # formula relativista
para la velocidad

    x = 0      # parte del punto 1.5 cm. Estas distancias están en m.
    y = 0
    z = 0

    vx = 0
    vy = v_p   # incide con velocidad v_y, perpendicular al lado largo
de los imanes
    vz = 0

    # Aquí nos vamos a guardar las posiciones en cada timestep:

    pos_x=[]
    pos_y=[]
    pos_z=[]

    pos_Bz=[]

    t = 0      # inicialización del tiempo
    dt = 1E-11 # timestep

    # Inicialización de los campos:

    Bx=0
    By=0
    Bz=0

    Fx = 0
    Fy = 0
    Fz = 0

    Ei=energies[i]
    Energy=[]

# El campo magnético se ha medido para la superficie del imán y los 0.5
cm adyacentes a cada lado del borde largo
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
while y < ((y_rel_1)*0.01): #L_in. Aquí no hay campo magnético. El
protón no se desvía.

    r_interp=np.interp(Ei,E,r)
    v_abs = np.sqrt((vx**2)+(vy**2)+(vz**2))
    Ei=Ei-(r_interp*v_abs*dt)

    if Ei <= 0:
        break
    else:
        v = c*np.sqrt(1-(1/(1 + Ei/m)**2))

        vx = (vx/v_abs)*v
        vy = (vy/v_abs)*v
        vz = (vz/v_abs)*v

        x = x + vx * dt
        y = y + vy * dt
        z = z + vz * dt

        pos_x.append(x)
        pos_y.append(y)
        pos_z.append(z)

        pos_Bz.append(Bz)

        Energy.append(Ei)

        t += dt

x_start=np.array(pos_x)[len(pos_x)-1] # Se guarda el último
elemento de la lista "pos_x" en la variable "x_start_1"

## COMIENZA EL PASO POR EL CAMPO MAGNÉTICO

x_B=x_start+3
y_B=0

while y >= ((y_rel_1)*0.01) and y <= ((y_rel_10+4)*0.01):

    r_interp=np.interp(Ei,E,r)

    v_abs = np.sqrt((vx**2)+(vy**2)+(vz**2))

    Ei=Ei-(r_interp*v_abs*dt)

    if Ei <= 0:
        break
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
else:
    v = c*np.sqrt(1-(1/(1 + Ei/m)**2))
    vx = (vx/v_abs)*v
    vy = (vy/v_abs)*v
    vz = (vz/v_abs)*v

    # Definimos posiciones para las funciones de los campos:

    x_B=x_B+100*vx*dt # renovamos el campo en cada timestep.
Partimos de x_start_1.
    y_B=y_B+100*vy*dt # lo mismo en y.

    # Calculamos los campos en cada timestep:

    Bx = int_Bx(x_B,y_B)
    By = int_By(x_B,y_B)
    Bz = int_Bz(x_B,y_B)

    # Fuerza de Lorentz en cada direccion, teniendo en cuenta la
expresion analitica del prod. vectorial

    Fx = q*((vy*Bz)-(vz*By))
    Fy = q*((vz*Bx)-(vx*Bz))
    Fz = q*((vx*By)-(vy*Bx))

    # Actualizamos posiciones en t+dt

    x = x + vx * dt
    y = y + vy * dt
    z = z + vz * dt

    # Actualizamos velocidades en t+dt, teniendo en cuenta que la
aceleración es F/m (igualar Lorentz a Newton)

    vx = vx + Fx/m_kg * dt
    vy = vy + Fy/m_kg * dt
    vz = vz + Fz/m_kg * dt

    # Nos guardamos las posiciones en las listas

    pos_x.append(x)
    pos_y.append(y)
    pos_z.append(z)

    pos_Bz.append(Bz)

    Energy.append(Ei)
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
        t += dt

x_exit=np.array(pos_x)[len(pos_x)-1]

while y>((y_rel_10+3.5)*0.01) and y<=0.68:

    r_interp=np.interp(Ei,E,r)

    v_abs = np.sqrt((vx**2)+(vy**2)+(vz**2))

    Ei=Ei-(r_interp*v_abs*dt)

    if Ei <= 0:
        break
    else:
        v = c*np.sqrt(1-(1/(1 + Ei/m)**2))

        vx = (vx/v_abs)*v
        vy = (vy/v_abs)*v
        vz = (vz/v_abs)*v

        x = x + vx * dt
        y = y + vy * dt
        z = z + vz * dt

        pos_x.append(x)
        pos_y.append(y)
        pos_z.append(z)

        pos_Bz.append(Bz)

        Energy.append(Ei)

    t += dt

print(Ei)

x_impact=np.array(pos_x)[len(pos_x)-1]

# Guardamos las posiciones en un array para trabajar mejor:

pos_x_arr=np.array(pos_x)
pos_y_arr=np.array(pos_y)
pos_z_arr=np.array(pos_z)
pos_Bz_arr=np.array(pos_Bz)
```

ANEXO IV: Construcción de mapas de campo magnético y simulaciones a partir de las medidas efectuadas

```
data_E.append(energies[i])
data_x.append(pos_x)
data_y.append(pos_y)
data_z.append(pos_z)

data_Bz.append(pos_Bz)

line = ax.plot(pos_y_arr*100, pos_x_arr*100, label=energies[i])
labels.append(line[0])

ax.legend(handles=labels, labels=[f'{energy:.1f} MeV' for energy in
energies], loc='upper right', bbox_to_anchor=(1.25, 1.02))
```

8. Pérdida de energía en aire

En este anexo se recogen los datos de la pérdida de energía en aire. Para cada energía E del protón hay un cociente de absorción, dE/dx, que se da en keV/μm. Es decir, si el protón avanza 1 μm perderá justo esta cantidad (dE/dx * 1 μm). En general, la pérdida por micra es menor para altas energías que para bajas.

E	dE/dx [keV/μm]	E	dE/dx [keV/μm]	E	dE/dx [keV/μm]
10.00 keV	4.337E-02	200.00 keV	7.217E-02	4.50 MeV	9.230E-03
11.00 keV	4.540E-02	225.00 keV	6.798E-02	5.00 MeV	8.513E-03
12.00 keV	4.733E-02	250.00 keV	6.423E-02	5.50 MeV	7.910E-03
13.00 keV	4.917E-02	275.00 keV	6.089E-02	6.00 MeV	7.393E-03
14.00 keV	5.093E-02	300.00 keV	5.792E-02	6.50 MeV	6.946E-03
15.00 keV	5.262E-02	325.00 keV	5.526E-02	7.00 MeV	6.554E-03
16.00 keV	5.423E-02	350.00 keV	5.288E-02	8.00 MeV	5.900E-03
17.00 keV	5.579E-02	375.00 keV	5.073E-02	9.00 MeV	5.375E-03
18.00 keV	5.728E-02	400.00 keV	4.878E-02	10.00 MeV	4.943E-03
20.00 keV	6.010E-02	450.00 keV	4.538E-02	11.00 MeV	4.581E-03
22.50 keV	6.335E-02	500.00 keV	4.250E-02	12.00 MeV	4.273E-03
25.00 keV	6.632E-02	550.00 keV	4.002E-02	13.00 MeV	4.007E-03
27.50 keV	6.903E-02	600.00 keV	3.787E-02	14.00 MeV	3.775E-03
30.00 keV	7.152E-02	650.00 keV	3.598E-02	15.00 MeV	3.572E-03
32.50 keV	7.380E-02	700.00 keV	3.429E-02	16.00 MeV	3.391E-03
35.00 keV	7.588E-02	800.00 keV	3.142E-02	17.00 MeV	3.229E-03
37.50 keV	7.779E-02	900.00 keV	2.905E-02	18.00 MeV	3.083E-03
40.00 keV	7.952E-02	1.00 MeV	2.706E-02	20.00 MeV	2.832E-03
45.00 keV	8.252E-02	1.10 MeV	2.552E-02	22.50 MeV	2.575E-03
50.00 keV	8.496E-02	1.20 MeV	2.401E-02	25.00 MeV	2.365E-03
55.00 keV	8.691E-02	1.30 MeV	2.272E-02	27.50 MeV	2.189E-03
60.00 keV	8.843E-02	1.40 MeV	2.159E-02	30.00 MeV	2.041E-03
65.00 keV	8.956E-02	1.50 MeV	2.059E-02	32.50 MeV	1.914E-03
70.00 keV	9.035E-02	1.60 MeV	1.968E-02	35.00 MeV	1.803E-03
80.00 keV	9.110E-02	1.70 MeV	1.887E-02	37.50 MeV	1.706E-03
90.00 keV	9.096E-02	1.80 MeV	1.812E-02	40.00 MeV	1.620E-03
100.00 keV	9.016E-02	2.00 MeV	1.682E-02	45.00 MeV	1.475E-03
110.00 keV	8.891E-02	2.25 MeV	1.545E-02	50.00 MeV	1.357E-03
120.00 keV	8.733E-02	2.50 MeV	1.432E-02	55.00 MeV	1.259E-03
130.00 keV	8.555E-02	2.75 MeV	1.335E-02	60.00 MeV	1.176E-03
140.00 keV	8.365E-02	3.00 MeV	1.252E-02	65.00 MeV	1.104E-03
150.00 keV	8.169E-02	3.25 MeV	1.180E-02	70.00 MeV	1.043E-03
160.00 keV	7.971E-02	3.50 MeV	1.116E-02	80.00 MeV	9.413E-04
170.00 keV	7.776E-02	3.75 MeV	1.060E-02	90.00 MeV	8.611E-04
180.00 keV	7.584E-02	4.00 MeV	1.010E-02	100.00 MeV	7.959E-04



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



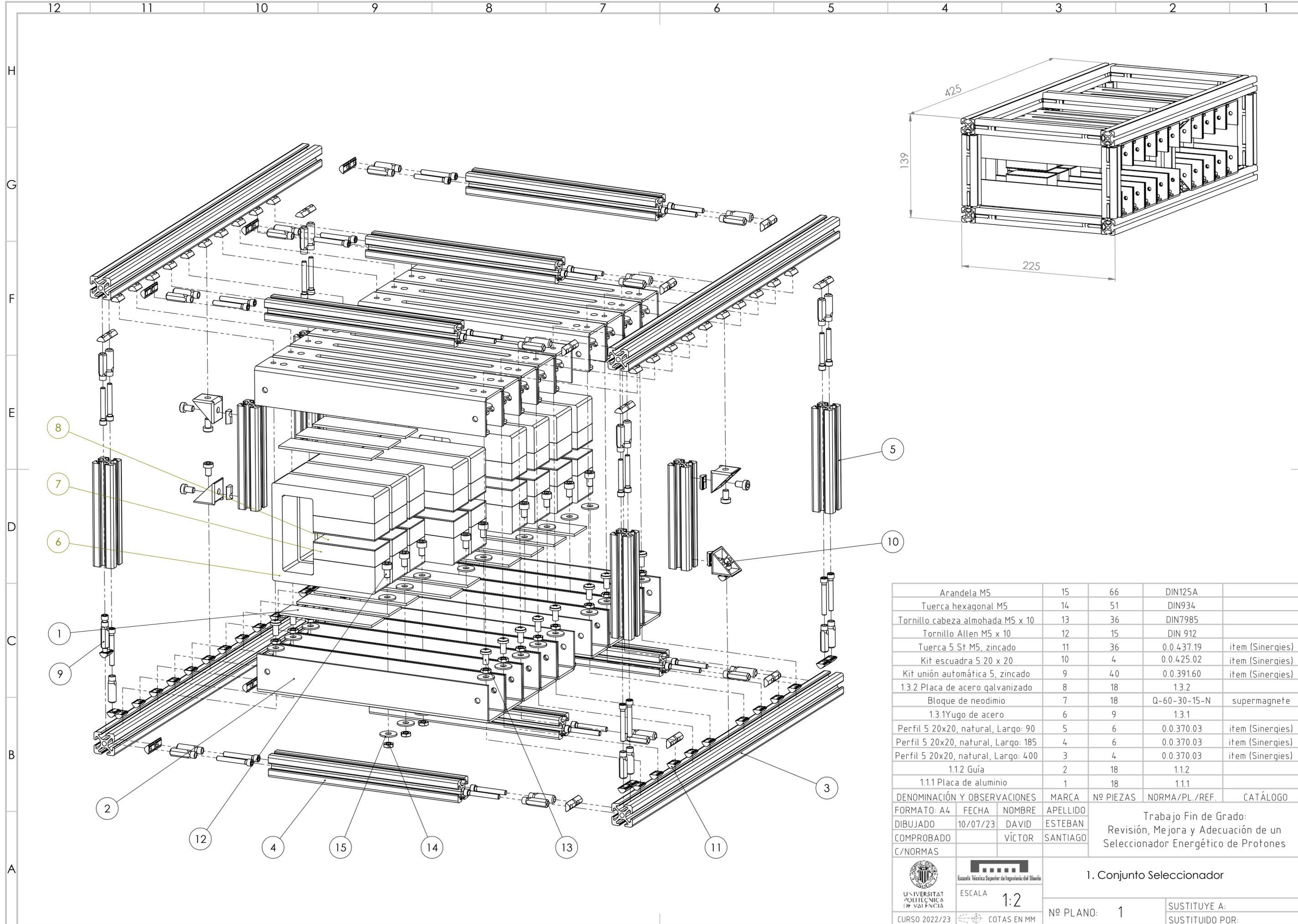
Escuela Técnica Superior de Ingeniería del Diseño

PLANOS

**Análisis mejora y adecuación de un seleccionador
energético de protones**

ÍNDICE DE PLANOS

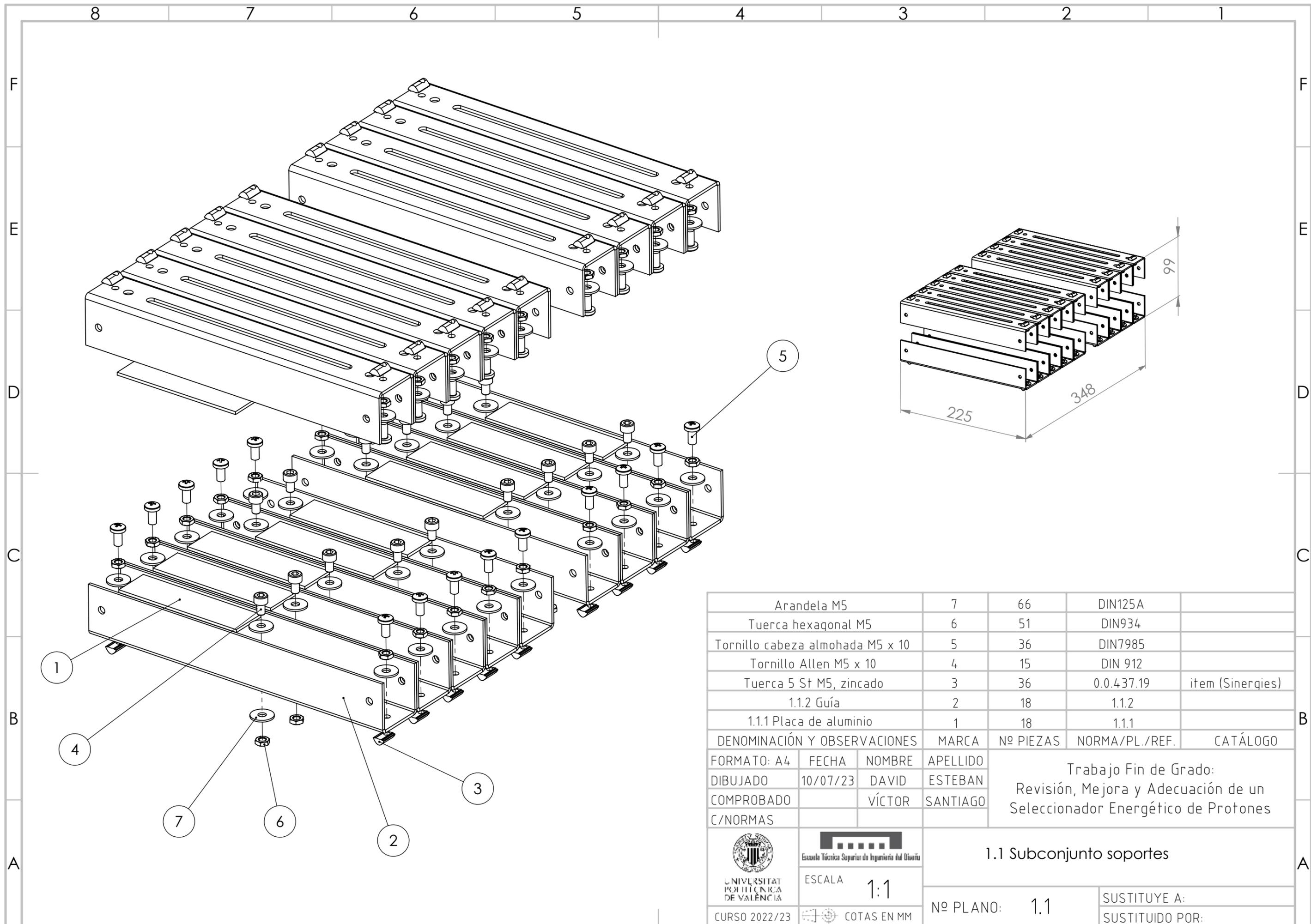
1. Conjunto Seleccionador
 - 1.1. Subconjunto soportes
 - 1.1.1. Placa de aluminio
 - 1.1.2. Guía
 - 1.2. Subconjunto estructura
 - 1.3. Subconjunto Imán
 - 1.3.1. Yugo de acero
 - 1.3.2. Placa de acero galvanizado



Arandela M5	15	66	DIN125A	
Tuerca hexagonal M5	14	51	DIN934	
Tornillo cabeza almohada M5 x 10	13	36	DIN7985	
Tornillo Allen M5 x 10	12	15	DIN 912	
Tuerca 5 St M5, zincado	11	36	0.0.437.19	item (Sinergies)
Kit escuadra 5 20 x 20	10	4	0.0.425.02	item (Sinergies)
Kit unión automática 5, zincado	9	40	0.0.391.60	item (Sinergies)
1.3.2 Placa de acero galvanizado	8	18	1.3.2	
Bloque de neodimio	7	18	Q-60-30-15-N	supermagnete
1.3.1 Yugo de acero	6	9	1.3.1	
Perfil 5 20x20, natural, Largo: 90	5	6	0.0.370.03	item (Sinergies)
Perfil 5 20x20, natural, Largo: 185	4	6	0.0.370.03	item (Sinergies)
Perfil 5 20x20, natural, Largo: 400	3	4	0.0.370.03	item (Sinergies)
1.1.2 Guía	2	18	1.1.2	
1.1.1 Placa de aluminio	1	18	1.1.1	

DENOMINACIÓN Y OBSERVACIONES		MARCA	Nº PIEZAS	NORMA/PL./REF.	CATÁLOGO
FORMATO: A4	FECHA	NOMBRE	APELLIDO	Trabajo Fin de Grado: Revisión, Mejora y Adecuación de un Seleccionador Energético de Protones	
DIBUJADO	10/07/23	DAVID	ESTEBAN		
COMPROBADO		VÍCTOR	SANTIAGO		
C/NORMAS					

 UNIVERSITAT POLITÈCNICA DE VALÈNCIA CURSO 2022/23	 ESCALA 1:2	1. Conjunto Seleccionador	
		Nº PLANO: 1	SUSTITUYE A: SUSTITUIDO POR:



Arandela M5		7	66	DIN125A	
Tuerca hexagonal M5		6	51	DIN934	
Tornillo cabeza almohada M5 x 10		5	36	DIN7985	
Tornillo Allen M5 x 10		4	15	DIN 912	
Tuerca 5 St M5, zincado		3	36	0.0.437.19	item (Sinergies)
1.1.2 Guía		2	18	1.1.2	
1.1.1 Placa de aluminio		1	18	1.1.1	
DENOMINACIÓN Y OBSERVACIONES		MARCA	Nº PIEZAS	NORMA/PL./REF.	CATÁLOGO
FORMATO: A4	FECHA	NOMBRE	APELLIDO	Trabajo Fin de Grado: Revisión, Mejora y Adecuación de un Seleccionador Energético de Protones	
DIBUJADO	10/07/23	DAVID	ESTEBAN		
COMPROBADO		VÍCTOR	SANTIAGO		
C/NORMAS					
UNIVERSITAT POLITÈCNICA DE VALÈNCIA		Escuela Técnica Superior de Ingeniería del Diseño		1.1 Subconjunto soportes	
CURSO 2022/23		ESCALA 1:1		Nº PLANO: 1.1	SUSTITUYE A:
		COTAS EN MM			SUSTITUIDO POR:

4 3 2 1

F

F

E

E

D

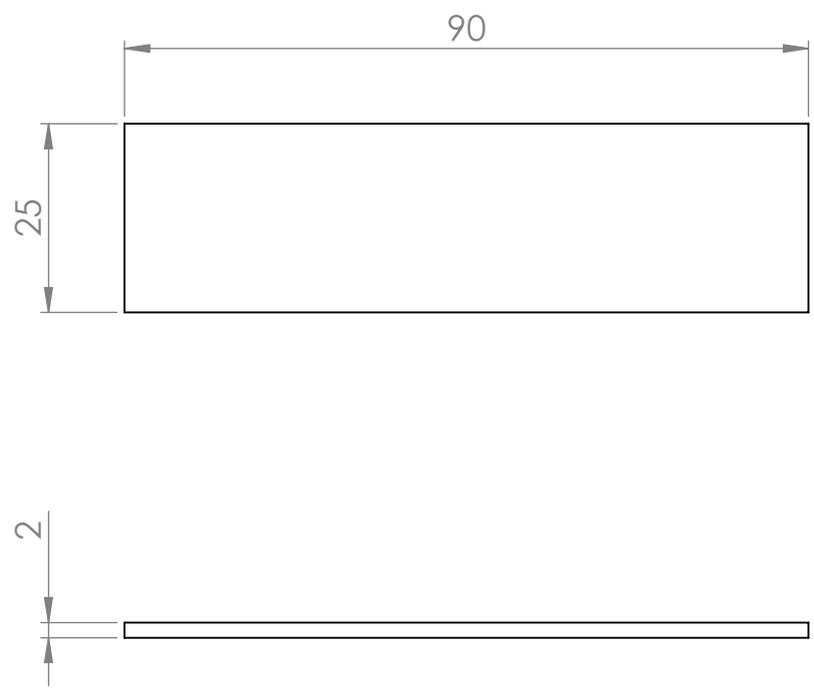
D

C

C

B

B



FORMATO: A4	FECHA	NOMBRE	APELLIDO
DIBUJADO	10/07/23	DAVID	ESTEBAN
COMPROBADO		VÍCTOR	SANTIAGO
C/NORMAS			

Trabajo Fin de Grado:
Revisión, Mejora y Adecuación de un
Seleccionador Energético de Protones



ESCALA
1:1

1.1.1 Placa de aluminio

CURSO 2022/23

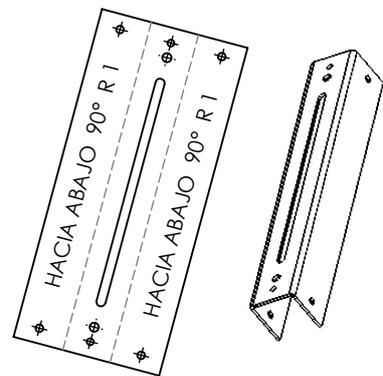
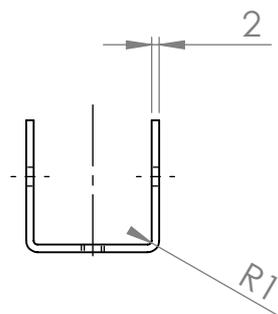
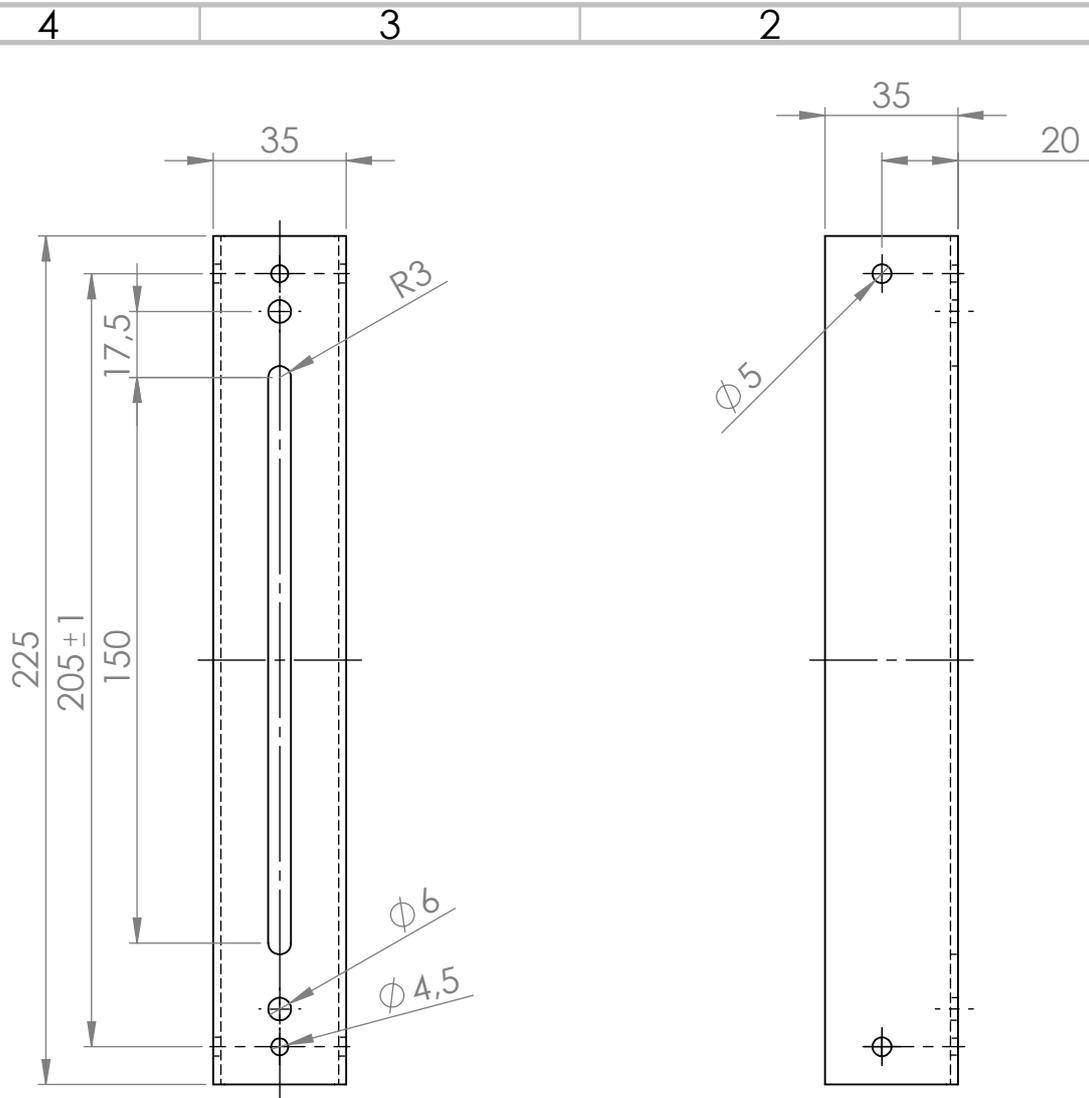
COTAS EN MM

Nº PLANO: 1.1.1

SUSTITUYE A:

SUSTITUIDO POR:

4 3 2 1



PLANO DE DOBLADO
ESCALA 1:5

FORMATO: A4	FECHA	NOMBRE	APELLIDO
DIBUJADO	10/07/23	DAVID	ESTEBAN
COMPROBADO		VÍCTOR	SANTIAGO
C/NORMAS			

Trabajo Fin de Grado:
Revisión, Mejora y Adecuación de un
Seleccionador Energético de Protones



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA

CURSO 2022/23



ESCALA
1:2

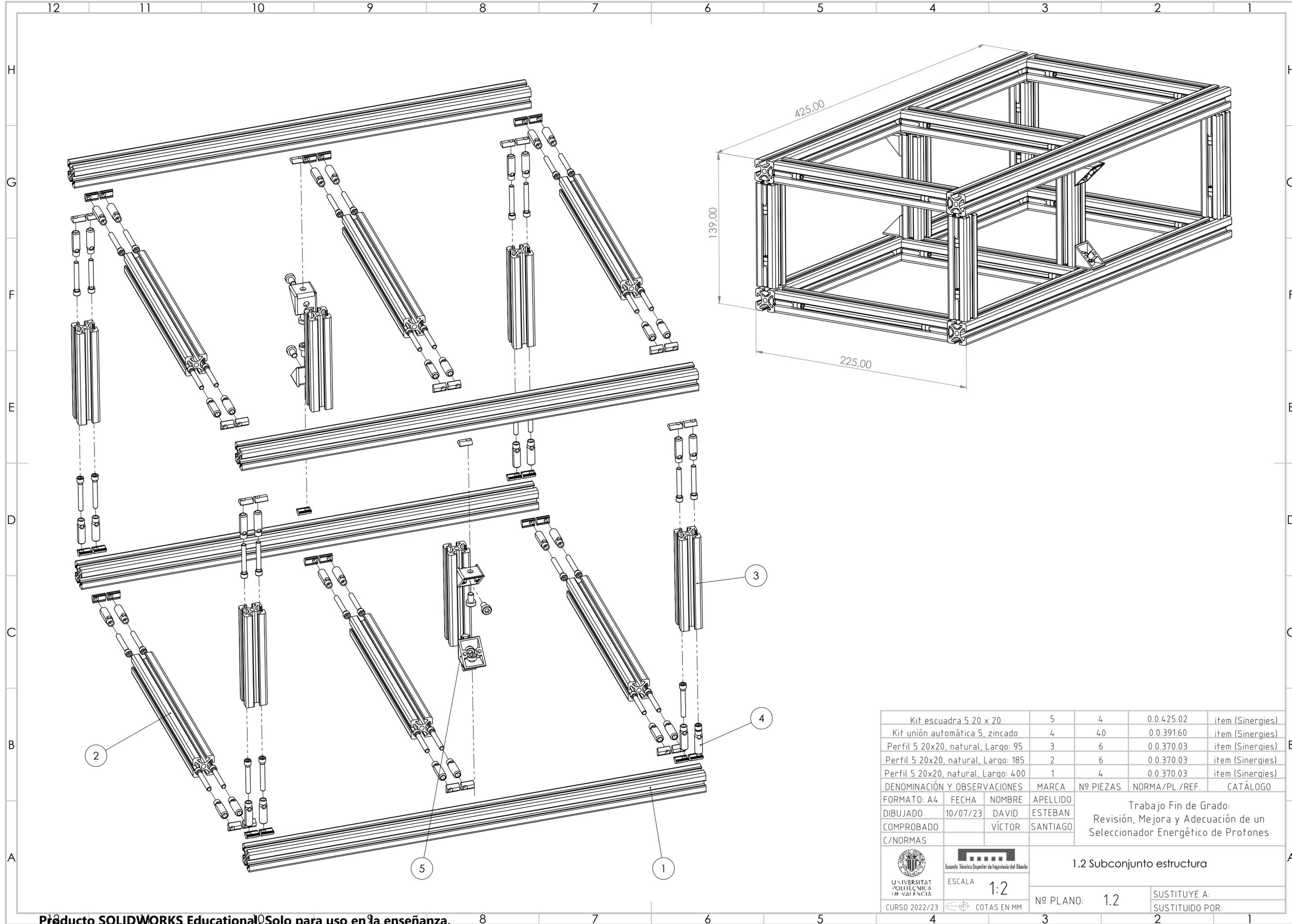
COTAS EN MM

1.1.2 Guía

Nº PLANO: 1.1.2

SUSTITUYE A:

SUSTITUIDO POR:



Kit escuadra 5 20 x 20	5	4	0.0.425.02	item (Sinergies)
Kit unión automática 5, zincado	4	40	0.0.391.60	item (Sinergies)
Perfil 5 20x20, natural, Larqo: 95	3	6	0.0.370.03	item (Sinergies)
Perfil 5 20x20, natural, Larqo: 185	2	6	0.0.370.03	item (Sinergies)
Perfil 5 20x20, natural, Larqo: 400	1	4	0.0.370.03	item (Sinergies)
DENOMINACIÓN Y OBSERVACIONES	MARCA	Nº PIEZAS	NORMA/PL./REF.	CATÁLOGO
FORMATO: A4	FECHA	NOMBRE	APELLIDO	Trabajo Fin de Grado: Revisión, Mejora y Adecuación de un Seleccionador Energético de Protones
DIBUJADO	10/07/23	DAVID	ESTEBAN	
COMPROBADO		VÍCTOR	SANTIAGO	
C/NORMAS				
UNIVERSITAT POLITÈCNICA DE VALÈNCIA		Escuela Técnica Superior de Ingeniería del Diseño		1.2 Subconjunto estructura
CURSO 2022/23	ESCALA	1:2		Nº PLANO: 1.2
COTAS EN MM				SUSTITUYE A:
				SUSTITUIDO POR:

4 3 2 1

F

F

E

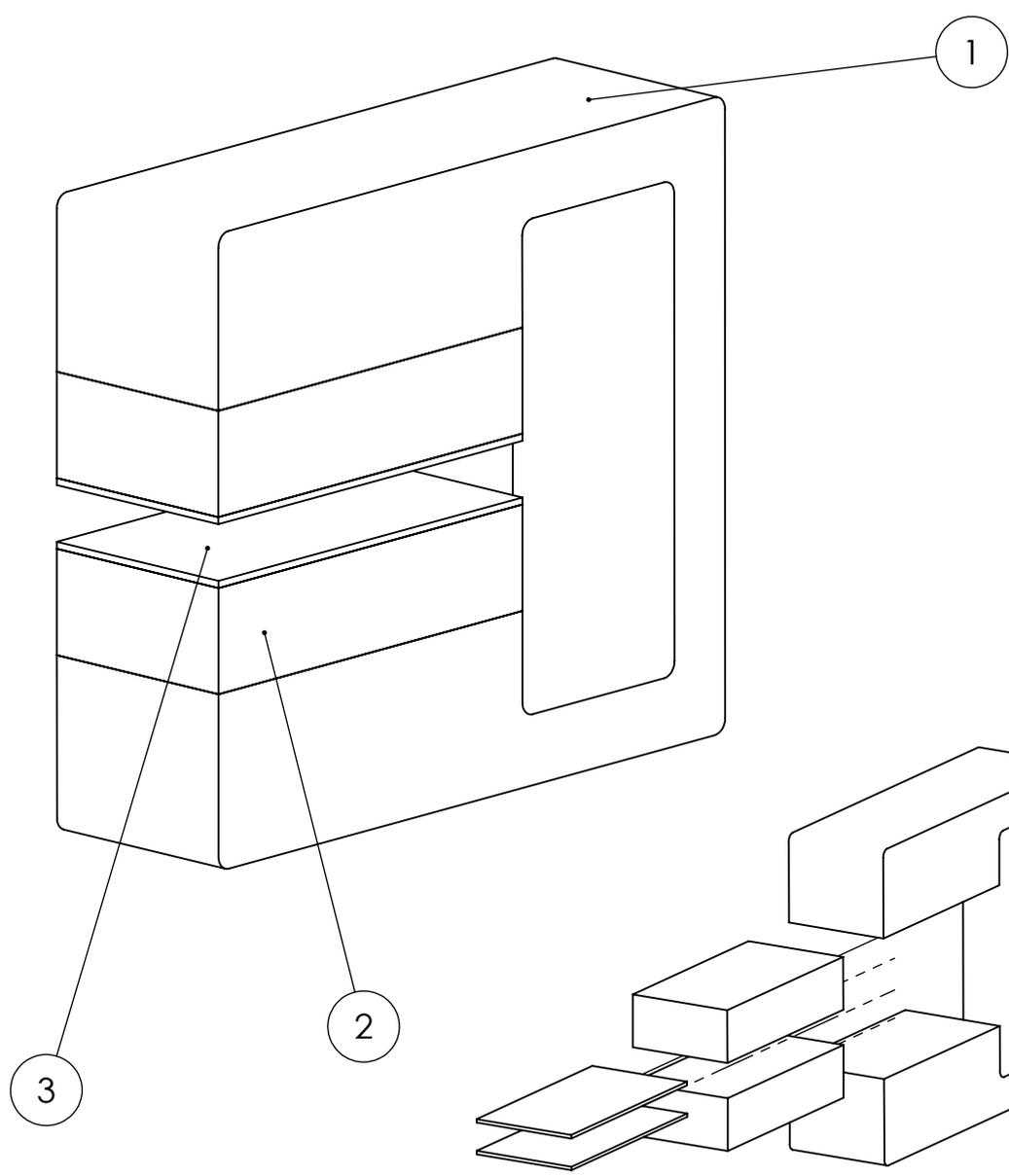
E

D

D

C

C



B

B

1.3.2 Placa de acero galvanizado	3	18	1.3.2	
Bloque de neodimio	2	18	Q-60-30-15-N	supermagnete
1.3.1 Yugo de acero	1	9	1.3.1	

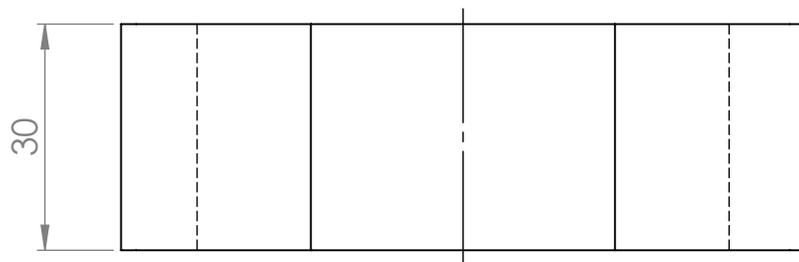
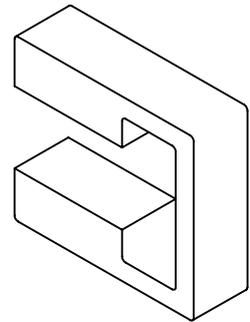
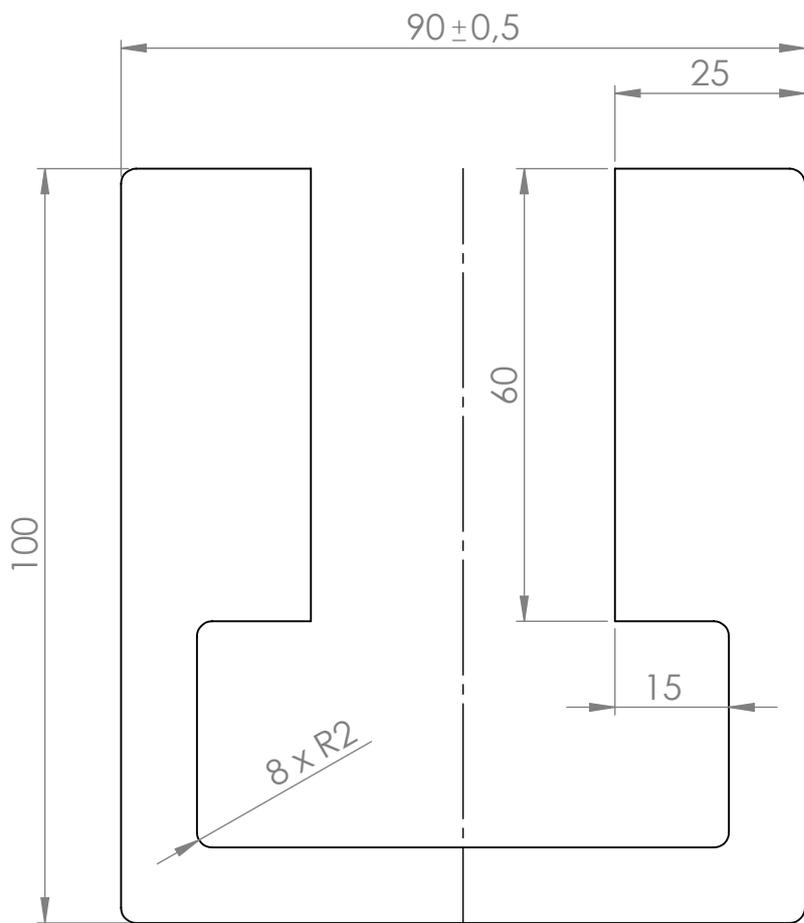
DENOMINACIÓN Y OBSERVACIONES		MARCA	Nº PIEZAS	NORMA/PL./REF.	CATÁLOGO
FORMATO: A4	FECHA	NOMBRE	APELLIDO	Trabajo Fin de Grado: Revisión, Mejora y Adecuación de un Seleccionador Energético de Protones	
DIBUJADO	10/07/23	DAVID	ESTEBAN		
COMPROBADO		VÍCTOR	SANTIAGO		
C/NORMAS					

A

A

<p>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</p>	<p>Escuela Técnica Superior de Ingeniería del Diseño</p>	1.3 Subconjunto Imán			
		ESCALA	1:2	Nº PLANO:	1.3
CURSO 2022/23		COTAS EN MM		SUSTITUIDO POR:	

4 3 2 1



FORMATO: A4	FECHA	NOMBRE	APELLIDO
DIBUJADO	10/07/23	DAVID	ESTEBAN
COMPROBADO		VÍCTOR	SANTIAGO
C/NORMAS			

Trabajo Fin de Grado:
Revisión, Mejora y Adecuación de un
Seleccionador Energético de Protones



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCALA
1:1

CURSO 2022/23

COTAS EN MM

Nº PLANO: 1.3.1

SUSTITUYE A:

SUSTITUIDO POR:

1.3.1 Yugo de acero

4 3 2 1

F

F

E

E

D

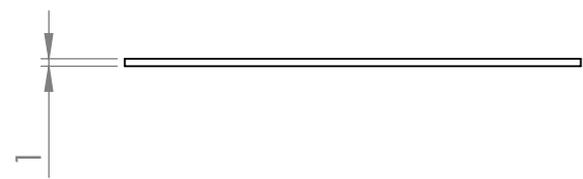
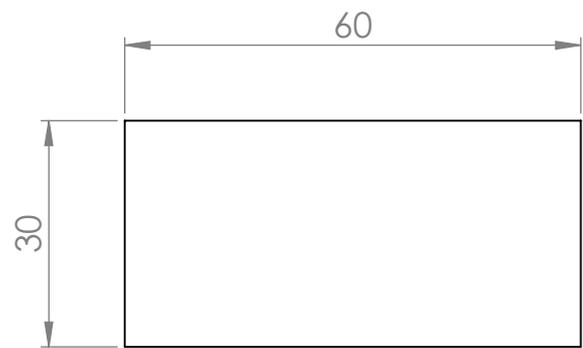
D

C

C

B

B



FORMATO: A4	FECHA	NOMBRE	APELLIDO
DIBUJADO	10/07/23	DAVID	ESTEBAN
COMPROBADO		VÍCTOR	SANTIAGO
C/NORMAS			

Trabajo Fin de Grado:
Revisión, Mejora y Adecuación de un
Seleccionador Energético de Protones



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCALA
1:1

1.3.2 Placa de acero galvanizado

CURSO 2022/23	COTAS EN MM
---------------	-------------

Nº PLANO: 1.3.2

SUSTITUYE A:
SUSTITUIDO POR:

4 3 2 1



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

PLIEGO DE CONDICIONES

Especificaciones Técnicas

ÍNDICE DE ESPECIFICACIONES TÉCNICAS

1. Objeto.....	2
2. Normativa.....	2
3. Especificaciones de diseño.....	3
4. Materiales	3
4.1. Soportes	3
4.1.1. Placas de aluminio.....	4
4.1.2. Guías.....	4
4.2. Estructura	4
4.2.1. Características.....	4
4.2.2. Controles de calidad	5
4.3. Imanes.....	5
4.3.1. Bloques de acero.....	5
4.3.2. Bloques de neodimio	5
4.3.3. Placas de acero galvanizado	6
4.4. Uniones	7
4.4.1. Características.....	7
4.4.2. Controles de calidad	7
5. Ejecución	7
5.1. Aspectos generales	7
5.1.1. Seguridad.....	7
5.1.2. Calidad constructiva	9
5.2. Construcción.....	10
6. Pruebas de servicio	10
7. Condiciones de entrega.....	10
8. Bibliografía	12

ÍNDICE DE TABLAS

Tabla 1. Perfiles de aluminio que conforman la estructura	5
Tabla 2. Uniones de “item (sinerges)”	7
Tabla 3. Uniones normalizadas.	7

1. Objeto

La presente especificación técnica se refiere a la revisión, mejora y adecuación de un seleccionador energético de protones para un equipo de investigación del I3M. Quedan excluidos de la especificación el diseño y ensamblaje de los soportes del seleccionador.

2. Normativa

Cabe mencionar en este apartado la Ley 31/1995, de 8 de noviembre, de Prevención de Riesgos Laborales, que constituye las garantías y responsabilidades para proporcionar cierto nivel de protección de la salud de los trabajadores en relación con los riesgos derivados de las condiciones de trabajo (BOE, 1997).

Compuesto por 12 artículos, los dos primeros establecen las disposiciones generales de esta Ley 31/1995, de 8 de noviembre, mientras que los siguientes artículos, hacen referencia a las obligaciones del empresario. Especial mención al artículo 6, que alude a aspectos más técnicos a cerca de las medidas preventivas, con el objetivo de garantizar la seguridad y salud en los lugares de trabajo, con la pretensión de evitar riesgos para los trabajadores (BOE, 1997).

Además, España ha ratificado Convenios de la Organización Internacional del Trabajo, que mantienen relaciones con la seguridad y salud en los lugares de trabajo, y que constituye nuestro ordenamiento jurídico interno. Hacemos así alusión al Convenio número 155 de la OIT, sobre la seguridad y salud de los trabajadores, de 22 de junio de 1981, ratificado por España el 26 de julio de 1985, así como el Convenio número 148 de la OIT, sobre el medio ambiente de trabajo, de 20 de junio de 1977, ratificado por nuestro país el 24 de noviembre de 1980 (BOE, 1997).

También se han establecido Directivas, concretamente mencionaremos la Directiva 89/654/CEE, de 30 de noviembre, en el ámbito de la Unión Europea diversos criterios en materia de seguridad, salud en los lugares de trabajo, además de medidas de protección contra accidentes y situaciones de riesgo (BOE, 1997).

De la misma manera, cabe mencionar el Área de Prevención de FREMAP, donde se presenta un Manual que pretenda dar a conocer los riesgos relativos a una oficina y las medidas preventivas disponibles para los trabajadores con el objetivo de esquivarlos. Esta información se establece con relación al entorno del puesto de trabajo, a la manipulación manual de cargas, posturas de trabajo, y también en función del tipo de puesto de trabajo (Social, 2014).

Por otro lado, atendiendo a la normativa aplicable en lo referente a la exposición a radiación y campos magnéticos se tiene:

- Directiva 97/43/Euratom del Consejo, de 30 de junio de 1997, en relación con la protección de la salud ante los riesgos procedentes de las radiaciones ionizantes en exposiciones médicas.
- Directiva 90/641/Euratom del Consejo, de 4 de diciembre de 1990, a propósito de la protección operacional de los trabajadores exteriores con amenaza de exposición a radiaciones ionizantes por actuación en zona controlada.

- Directiva 89/618/Euratom del Consejo, de 27 de noviembre de 1988, referente a la información de la comunidad sobre las medidas de protección sanitaria aplicables y acerca de conducta a seguir en caso de emergencia radiológica.
- Directiva 2003/122/Euratom del Consejo, de 22 de diciembre de 2003, en cuanto al control de fuentes radioactivas selladas de actividad elevada y de las fuentes huérfanas.
- La Directiva 97/43/Euratom del Consejo, de 30 de junio de 1997, en el Real Decreto 815/2001, de 13 de julio, acerca del fundamento del uso de las radiaciones ionizantes para la protección radiológica de las personas con ocasión de exposiciones médicas.
- ISO 150 20553:2006, "Protección de la Radiación – Monitoreo de trabajadores ocupacionalmente expuestos a riesgos de contaminación interna con material radiactivo", en relación a aquellos trabajadores bajo amenaza de exposición radioactiva a partir del monitoreo de la absorción potencial y real, además del control de la exposición.
- UNE-EN ISO 14146:2021, sobre la protección radiológica, concretamente haciendo alusión a los criterios y límites para la evaluación periódica de los servicios de dosimetría.
- IEEE Std 1308-1994TM. "IEEE Recommended Practice for Instrumentation: Specifications for Magnetic Flux Density and Electric Field Strength Meters- 10 Hz to 3 kHz".

3. Especificaciones de diseño

Este dispositivo deberá de ser capaz de separar protones por energías a partir de un haz de entre 4 MeV y 8 MeV y deberá poder ubicarse dentro de una cámara de vacío por lo que no puede superar las siguientes dimensiones: Longitud x Ancho x Alto = 50x30x30 cm.

Además, debe ser capaz de funcionar también con energías de hasta 16 MeV sin variar su configuración.

4. Materiales

En este apartado se recogen las características de los materiales que componen el seleccionador para garantizar su reproducibilidad, siendo el aspecto más importante en este sentido la elección de los imanes, por lo que se detallan los requisitos técnicos de los mismos tal y como se verá en el apartado "4.2.2. Bloques de neodimio".

4.1. Soportes

Los materiales que componen la estructura del seleccionador deben ser no magnético o paramagnéticos, para evitar que interactúen con el campo magnético que generan sus imanes.

4.1.1. Placas de aluminio

4.1.1.1. Características

Las placas de aluminio se conformarán a partir del corte, según se recoge en el plano "1.1.1 Placa de aluminio", del "Plano de aluminio gris, Alt.0.2 x An.2 x L.100 cm", con referencia 80109435, suministrado por "LEROY MERLIN". La tolerancia del corte no es importante siempre y cuando se garantice la funcionalidad de las piezas tal y como se recoge en la memoria, pudiéndose limar o recortar más siempre que se cometa un error demasiado grande. Si el error en el corte produce placas más pequeñas no habrá problema alguno.

4.1.1.2. Controles de calidad

Las placas de aluminio no requieren de un control de calidad exhaustivo, pero se deberá comprobar visualmente la planicidad de la materia prima y la inexistencia de golpes ni abolladuras. Además, como no es especialmente relevante que el material de confección sea aluminio, pero sí que no presente propiedades magnéticas, se comprobará la inexistencia de interacción con el campo magnético al acercarse un imán. En caso de disconformidades el suministrador se ocupará de la sustitución de la pieza.

4.1.2. Guías

4.1.2.1. Características

Las guías constan de 20 perfiles en U de aluminio de la serie 6000 cuyas dimensiones y operaciones de mecanizado se recoge en el plano "1.1.2 Guía" de este proyecto. Estas guías han sido mecanizadas por "Werkstat Mecanizados". Cabe destacar que, si bien no es necesario especificar ninguna tolerancia para la longitud de los perfiles, la tolerancia máxima de la distancia entre los taladros exteriores de la cara central es de ± 1 mm, para garantizar la sujeción de los perfiles a la estructura. Para el resto de los procesos de mecanizado se tiene la misma tolerancia dimensional máxima.

4.1.2.2. Controles de calidad

Se verificará que las guías tienen las dimensiones especificadas y que los taladros están correctamente realizados durante su montaje en el dispositivo. Además, se verificará también entonces la carencia de propiedades magnéticas. En caso de disconformidades con los requisitos anteriormente mencionados correrán por cuenta del proveedor.

4.2. Estructura

4.2.1. Características

Las vigas y pilares de la estructura se basan en el Perfil 5 de aluminio de la empresa "item (sinerges)" con número de artículo 0.0370.03 sin mecanizar. La lista de piezas se recoge a continuación ("Tabla 1"):

Tabla 1. *Perfiles de aluminio que conforman la estructura*

Designación del artículo	N.º de artículo	Cantidad
Perfil 5 20x20, natural, Largo: 95mm	0.0.370.03	6
Perfil 5 20x20, natural, Largo: 185mm	0.0.370.03	6
Perfil 5 20x20, natural, Largo: 400mm	0.0.370.03	4

4.2.2. Controles de calidad

Se comprobarán las dimensiones de los perfiles suministrados y se les acercará un imán para comprobar que no presenten propiedades magnéticas. En caso de disconformidades con los requisitos anteriormente mencionados correrán por cuenta del proveedor.

4.3. Imanes

Todos los materiales de los imanes son ferromagnéticos. Dividimos entre:

4.3.1. Bloques de acero

4.3.1.1. Características

Estos bloques estarán compuestos de acero ferrífico AISI 430 por sus altas propiedades magnéticas y se mecanizarán, según la geometría que se detalla en el plano “1.3.1. Yugo de acero”, por la empresa “Werkstat Mecanizados”, con una tolerancia dimensional de $\pm 0,5$ mm en su altura, para garantizar el buen ajuste de los centros de los imanes en el seleccionador.

4.3.1.2. Controles de calidad

Se comprobará que los bloques poseen propiedades magnéticas, utilizando un imán. Además, se verificará la homogeneidad de su altura colocándolos todos en un banco o una mesa plana y dejando caer otro elemento plano, como las vigas de aluminio, en su parte superior. De esta manera, si la luz pasa entre la cara superior de un imán y el elemento plano, será necesario medir su altura para comprobar la tolerancia. En caso de que alguna de las pruebas indique algún fallo, se deberá comunicar al proveedor, que deberá asumir los costes de la sustitución.

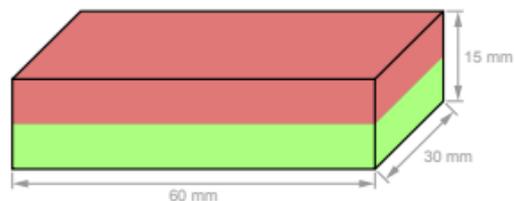
4.3.2. Bloques de neodimio

4.3.2.1. Características

Los bloques de neodimio N40 niquelado de 60 x 30 x 15 mm tienen una tolerancia dimensional de $\pm 0,1$ mm. Se pueden encontrar en la página “supermagnete” y los datos técnicos facilitados por la empresa se detallan a continuación.

Materiales

Cód. artículo	Q-60-30-15-N
EAN	7640155436571
Material	NdFeB
Forma	Bloque
Tamaño	60 x 30 x 15 mm
Página 1	60 mm
Página 2	30 mm
Página 3	15 mm
Superficies polares	60 x 30 mm
Tolerancia	+/- 0,1 mm
Sentido de magnetización	Eje 15 mm
Revestimiento	niquelado (Ni-Cu-Ni)
Tipo de fabricación	sinterizado
Magnetización	N40
Fza. sujec.	aprox. 56 kg (aprox. 549 N)
Fuerza de desplazamiento	aprox. 11 kg (aprox. 110 N)
Temperatura de servicio máx.	80°C
Peso	205,2000 g
Temperatura de Curie	310 °C
Remanencia Br	12600-12900 G, 1.26-1.29 T
Coercitividad bHc	10.5-12.0 kOe, 860-955 kA/m
Coercitividad iHc	≥12 kOe, ≥955 kA/m
Producto energético (BxH)max	38-40 MGOe, 303-318 kJ/m ³



El producto cumple la Directiva europea RoHS.



El producto cumple el Reglamento europeo REACH.

4.3.2.2. Controles de calidad

Para determinar la calidad de los bloques, es necesario comprobar la inexistencia de signos de caídas o golpes como arañazos, golpes o abolladuras. Además, para comprobar la fuerza y homogeneidad del campo y el sentido de magnetización se utilizará una sonda de efecto hall. En caso de disconformidades correrán a cuenta del suministrador.

4.3.3. Placas de acero galvanizado

4.3.3.1. Características

Las placas que protegen los imanes de neodimio han sido suministradas “Werkstat Mecanizados”, y son de acero galvanizado con unas dimensiones de 60 x 30 x 1 mm según se recoge en “1.3.2 Placa de acero galvanizado” y cuyas tolerancias dimensionales no son importantes.

4.3.3.2. Controles de calidad

Se comprobará que no existan signos de caídas ni golpes y las dimensiones son correctas. En caso negativo se hará responsable el proveedor.

4.4. Uniones

4.4.1. Características

Las uniones no requieren de propiedades paramagnéticas. Aquellas que se suministran a partir de la empresa “item (sinerges)” aparecen listadas a continuación:

Tabla 2. *Uniones de “item (sinerges)”.*

Designación del artículo	N.º de artículo	Cantidad
Kit unión automática 5, zincado	0.0.391.60	40
Kit escuadra 5 20x20	0.0.425.02	4
Tuerca 5 St M5, zincado	0.0.437.19	38

Además, los tornillos, tuercas y arandelas de M5 que se necesitan son:

Tabla 3. *Uniones normalizadas.*

Designación del artículo	NORMA	Cantidad
Tornillo Allen M5 de cabeza cilíndrica	DIN 912	15
Tornillo cabeza almohada M5 x 10	DIN7985	40
Tuerca M5 Hexagonal	DIN934	58
Arandela M5 x 7	DIN125A	74

4.4.2. Controles de calidad

Para comprobar la calidad de las uniones se verificarán las métricas de los elementos roscados y que todos los componentes encajan a la perfección. En caso contrario deberán ser sustituidos por el proveedor.

5. Ejecución

Se distingue entre los aspectos generales que aseguran una correcta ejecución y los más específicos, propios del proceso constructivo:

5.1. Aspectos generales

5.1.1. Seguridad

Para garantizar la seguridad del operario al montar el seleccionador, se deben usar guantes gruesos siempre que se trabaje con los imanes para evitar pellizcos y golpes. Además, se debe prestar especial cuidado siempre que se manipule o mueva el seleccionador debido a la fragilidad de los imanes.

A continuación, se detallan algunos aspectos importantes facilitados por “supermagnete”.

Ejecución

5.1.1.1. Advertencias

Advertencia	Contusiones
	<p>Los imanes grandes tienen una fuerza de atracción enorme.</p> <ul style="list-style-type: none">• En caso de uso imprudente, podría pillarse los dedos o la piel entre dos imanes. Esto puede provocar contusiones y hematomas en las áreas afectadas.• Los imanes muy grandes pueden causar fracturas óseas debido a su fuerza. <p>A la hora de manipular imanes grandes, póngase unos guantes protectores gruesos.</p>

Advertencia	Marcapasos
	<p>Los imanes pueden alterar el funcionamiento de marcapasos y de desfibriladores implantados.</p> <ul style="list-style-type: none">• Un marcapasos podría cambiarse al modo de prueba, lo que provocaría una indisposición.• Un desfibrilador podría incluso dejar de funcionar.• Si lleva alguno de estos dispositivos, manténgase a una distancia prudente de los imanes: www.supermagnete.es/faq/distance• Advierta siempre a las personas que lleven en este tipo de dispositivos de su proximidad a los imanes.

Advertencia	Objetos pesados
	<p>Las cargas excesivas o bruscas, los signos de desgaste y los defectos en el material pueden provocar que un imán o un gancho magnético se suelten de su base de fijación.</p> <p>Si un objeto se cae, puede provocar lesiones graves.</p> <ul style="list-style-type: none">• La fuerza de sujeción indicada se alcanza únicamente en condiciones ideales. Actúe siempre garantizando un alto grado de seguridad.• No utilice los imanes en lugares en que puedan causar daños a otras personas en caso de defectos en el material.

Advertencia	Fragmentos metálicos
	<p>Los imanes de neodimio son frágiles. Si dos imanes chocan, pueden saltar fragmentos.</p> <p>Los fragmentos afilados pueden salir despedidos a varios metros de distancia y causar lesiones oculares.</p> <ul style="list-style-type: none">• Evite que los imanes choquen entre sí.• Si va a manipular imanes grandes, póngase unas gafas protectoras.• Asegúrese de que las personas a su alrededor estén protegidas de igual modo o se mantengan a una distancia prudente.

5.1.1.2. Manejo y Almacenamiento

Atención	Campo magnético
	<p>Los imanes generan un campo magnético fuerte y de gran alcance, por lo que algunos dispositivos podrían estropearse, como por ejemplo: televisores, ordenadores portátiles, discos duros, tarjetas de crédito, soportes de datos, relojes mecánicos, audífonos y altavoces.</p> <ul style="list-style-type: none">• Mantenga los imanes alejados de todos aquellos objetos y dispositivos que puedan estropearse debido a campos magnéticos fuertes.• Tenga en cuenta nuestra tabla de distancias recomendadas: www.supermagnete.es/faq/distance

Atención	Inflamabilidad
	<p>Si los imanes se mecanizan, el polvo de perforación se puede inflamar fácilmente.</p> <p>Evite este tipo de mecanizado de los imanes o utilice una herramienta adecuada y agua refrigerante en abundancia.</p>

Atención	Alergia al níquel
	<p>La mayoría de nuestros imanes contiene níquel, incluso los que no llevan revestimiento de níquel.</p> <ul style="list-style-type: none">• Algunas personas tienen reacciones alérgicas al entrar en contacto con el níquel.• Las alergias al níquel se pueden desarrollar debido al contacto continuado con objetos que contienen níquel.• Evite que la piel entre en contacto con imanes de forma continuada.• No haga uso de imanes si ya tiene alergia al níquel.

Ejecución

 Aviso	Efecto sobre las personas <p>Según los conocimientos actuales, los campos magnéticos de imanes permanentes no tienen ningún efecto positivo o negativo apreciable sobre las personas. Es muy improbable que el campo magnético de un imán permanente pueda suponer un riesgo para la salud, pero no se puede excluir del todo.</p> <ul style="list-style-type: none">• Por su seguridad, evite el contacto continuo con imanes.• Mantenga los imanes grandes al menos a un metro de distancia de su cuerpo.
 Aviso	Fragmentación del revestimiento <p>La mayor parte de nuestros imanes de neodimio dispone de un revestimiento fino de níquel-cobre-níquel para protegerlos de la corrosión. Este revestimiento puede fragmentarse o resquebrajarse al ser golpeado o expuesto a grandes presiones. Esto provoca que los imanes se hagan más sensibles ante condiciones ambientales como la humedad, pudiendo llegar a oxidarse.</p> <ul style="list-style-type: none">• Separe los imanes grandes, especialmente las esferas, con ayuda de un trozo de cartón.• Evite que los imanes choquen entre sí, así como las cargas mecánicas continuadas (p. ej. impactos).
 Aviso	Oxidación, corrosión, herrumbre <p>Los imanes de neodimio no tratados se oxidan muy rápidamente y se deshacen. La mayor parte de nuestros imanes dispone de un revestimiento fino de níquel-cobre-níquel para protegerlos de la corrosión. Este revestimiento también ofrece cierta protección frente a la corrosión, pero no resulta lo suficientemente resistente en caso de un uso continuado en espacios exteriores.</p> <ul style="list-style-type: none">• Utilice los imanes tan sólo en espacios interiores secos o protéjalos de las condiciones ambientales.• Evite dañar el revestimiento.
 Aviso	Resistencia a la temperatura <p>Los imanes de neodimio tienen una temperatura de uso máxima de entre 80 y 200 °C. La mayor parte de los imanes de neodimio pierde parte de su fuerza de sujeción de manera permanente a partir de los 80 °C.</p> <ul style="list-style-type: none">• No utilice los imanes en aquellos lugares donde vayan a estar expuestos a temperaturas altas.• Si utiliza pegamento, evite endurecerlo con aire caliente.
 Aviso	Mecanizado <p>Los imanes de neodimio son frágiles, termosensibles y se oxidan fácilmente.</p> <ul style="list-style-type: none">• Los imanes se pueden fragmentar si se utiliza una herramienta inadecuada a la hora de perforarlos o serrarlos.• Los imanes se pueden desmagnetizar como consecuencia del calor producido.• Si el revestimiento está dañado, el imán se oxida y se deshace. <p>Evite la mecanización de imanes si no dispone de las máquinas y la experiencia necesarias. En su lugar, solicite una oferta para una producción a medida: www.supermagnete.es/custom_form.php</p>

5.1.2. Calidad constructiva

El montaje de todas las piezas que componen el selector se regirá por el apartado "Planos" de este proyecto.

Por otro lado, otros aspectos importantes desde el punto de vista del respeto de la función principal del selector es la caracterización de los polos de los imanes, que permite determinar el sentido del campo magnético. Para ello es posible utilizar una brújula magnética o un imán con los polos claramente definidos.

Finalmente, desde el punto de vista de la calidad, se realizarán comprobaciones visuales para asegurar que todos los componentes se encuentran correctamente posicionados y no existen errores en el proceso de montaje.

5.2. Construcción

El proceso constructivo se detalla resumidamente a continuación con los aspectos más destacados, y debe seguir estrictamente este orden salvo para los dos primeros puntos, que pueden intercambiarse:

1. Montaje de la estructura sin apretar los tornillos que unen las partes superior e inferior, aunque deben aproximarse.
2. Montaje de los imanes.
3. Montaje de los imanes y las placas de aluminio en sus guías.
4. Fijación de las guías a la estructura.
5. Fijación de la posición de los imanes.
6. Centrado de placas de aluminio.
7. Apriete de los tornillos de la estructura que unen la parte inferior con la superior.

Es necesario destacar que los tornillos que sostienen las guías a las vigas longitudinales deberán apretarse a partir de la turca que llevan incorporada utilizando una llave inglesa.

6. Pruebas de servicio

Para comprobar el funcionamiento se realizará una prueba de servicio en el ciclotrón del CNA de Sevilla, pero se deberá realizar otra prueba final en el acelerador láser de Salamanca, que es su fin último.

7. Condiciones de entrega

Es importante prestar especial cuidado a la hora de realizar la entrega del dispositivo pues es un equipo pesado y frágil que produce campos magnéticos fuertes. Por tanto, se debe tratar con cuidado y no acercar elementos sensibles al campo magnético tales como dispositivos electrónicos.

Para protegerlo de golpes y vibraciones, se transportará en una caja con material de protección, a la que es conveniente añadir un mensaje de precaución.

Además, es conveniente que no se vea expuesto a elementos corrosivos como el sol y el agua pues no está pensado para ello.

Las instrucciones que da el suministrador de los imanes para su transporte quedan:

<p>Atención</p> 	<p>Envíos postales</p> <p>Los campos magnéticos de los imanes embalados de manera inadecuada pueden provocar daños en los dispositivos de clasificación postal, así como en las mercancías frágiles de otros embalajes.</p> <ul style="list-style-type: none">• Tenga en cuenta nuestros consejos para el envío: www.supermagnete.es/faq/shipping• Utilice una caja con el suficiente espacio y coloque los imanes en el centro del embalaje con ayuda de material de relleno.• Coloque los imanes en el embalaje de manera que los campos magnéticos se neutralicen entre sí.• Utilice placas de acero para proteger del campo magnético, en caso necesario.• Para el transporte aéreo, se aplican normas más estrictas: tenga en cuenta las advertencias para el "transporte aéreo".
--	---

Condiciones de entrega

<p>Atención</p> 	<p>Envíos postales</p> <p>Los campos magnéticos de los imanes embalados de manera inadecuada pueden provocar daños en los dispositivos de clasificación postal, así como en las mercancías frágiles de otros embalajes.</p> <ul style="list-style-type: none">• Tenga en cuenta nuestros consejos para el envío: www.supermagnete.es/faq/shipping• Utilice una caja con el suficiente espacio y coloque los imanes en el centro del embalaje con ayuda de material de relleno.• Coloque los imanes en el embalaje de manera que los campos magnéticos se neutralicen entre sí.• Utilice placas de acero para proteger del campo magnético, en caso necesario.• Para el transporte aéreo, se aplican normas más estrictas: tenga en cuenta las advertencias para el "transporte aéreo".
--	---

8. Bibliografía

BOE. (23 de Abril de 1997). *Real Decreto 486/1997, de 14 de abril, por el que se establecen las disposiciones mínimas de seguridad y salud en los lugares de trabajo*. Obtenido de BOE: <https://www.boe.es/buscar/pdf/1997/BOE-A-1997-8669-consolidado.pdf>

Social, M. d. (2014). *Manual de Seguridad y Salud en oficinas*. Obtenido de FREMAP: https://www.usal.es/files/Manual_Seg_ySalud_Oficinas.pdf



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

PRESUPUESTO

Coste total del proyecto

ÍNDICE DEL PRESUPUESTO

1. Introducción	3
2. Tablas de costes	3
2.1. Mano de obra	3
2.2. IT	3
2.3. Material de construcción	4
2.4. Otros materiales	4
2.5. Alquileres	4
2.6. Procesos	4
3. Metodología y Amortizaciones	5
4. Obtención de Costes	5
4.1. Costes Fijos	6
4.1.1. Construcción	6
4.1.2. Proceso Experimental	6
4.2. Costes variables	6
4.2.1. Construcción	7
4.2.2. Simulaciones	7
4.2.3. Medidas	7
4.2.4. Proceso experimental	8
4.2.5. Redacción de documentos	8
5. Presupuestos finales	9
5.1. Presupuesto parcial	9
5.2. Presupuesto total	9

ÍNDICE DE TABLAS

Tabla 1. Coste de las diferentes unidades de mano de obra necesarias por hora.	3
Tabla 2. Coste de los recursos informáticos necesarios.	3
Tabla 3. <i>Coste de los materiales que componen el seleccionador.</i>	4
Tabla 4. <i>Coste de materiales añadidos que no componen el seleccionador.</i>	4
Tabla 5. <i>Coste de alquiler de instalaciones por hora.</i>	4
Tabla 6. <i>Coste de procesos industriales por hora.</i>	4
Tabla 7. <i>Amortización de recursos IT.</i>	5
Tabla 8. Coste de los elementos físicos que componen el seleccionador según cantidad	6
Tabla 9. Coste de los elementos materiales necesarios para llevar a cabo el proceso experimental.	6
Tabla 10. Coste del tratamiento y ensamblado de los materiales para la construcción del dispositivo.	7
Tabla 11. Coste de los elementos necesarios para llevar a cabo los estudios sobre el trazado de partículas	7
Tabla 12. Coste de los elementos necesarios para la medida del campo magnético....	7
Tabla 13. Coste del proceso experimental.....	8
Tabla 14. Coste monetario de la redacción de los documentos que recogen el trabajo realizado.	8
Tabla 15. Coste parcial del proyecto.....	9
Tabla 16. Gastos generales y beneficio industrial.	9
Tabla 17. Presupuesto total	9

1. Introducción

En el siguiente apartado, observaremos una descripción de los costes relacionados con la realización del proyecto que hemos visto anteriormente. A lo largo de este primer capítulo, tendremos en cuenta los costes directamente relacionados con el presente estudio, como pueden ser los equipos y programas informáticos utilizados durante los ensayos realizados, análisis de datos, y redacción.

2. Tablas de costes

A continuación, se mostrarán diversos cuadros de precios en relación con diferentes aspectos como los materiales utilizados o la mano de obra necesaria. También aparecerán los criterios a partir de los cuales hemos obtenido los costes. Cabe destacar que estos cuadros de precios no incluyen el IVA.

2.1. Mano de obra

En primer lugar, observaremos una tabla en la que se muestra la mano de obra asociada al desarrollo exclusivo del presente proyecto.

Tabla 1. Coste de las diferentes unidades de mano de obra necesarias por hora.

Referencia	Unidad	Concepto	Precio
MO1	Horas [h]	Ingeniero Técnico	14,10 €
MO2	Horas [h]	Investigador	23,61 €
MO3	Horas [h]	Profesor Investigador	25,62 €

2.2. IT

En este apartado, haremos alusión a los costes de los materiales empleados a lo largo del desarrollo del proyecto, englobando tanto las aplicaciones software, como los programas utilizados en los diferentes apartados durante el post-procesado de datos obtenidos y creación de figuras y esquemas ilustrativos. El software Visual Studio Code, utilizado para programar con Python, no supone coste alguno, ya que es gratuito.

Tabla 2. Coste de los recursos informáticos necesarios.

Referencia	Unidad	Concepto	Precio
IT1	Unidad [Ud.]	Ordenador portátil	1.400,00 €
IT2	Unidad [Ud.]	Sistema operativo 2023	220,00 €
IT3	Unidad [Ud.]	Pack Office 365	99,00 €
IT4	Unidad [Ud.]	SolidWorks 2022	4.999,00 €
IT5	Unidad [Ud.]	Visual Studio Code	GRATUITO

2.3. Material de construcción

En la “Tabla 3” se recogen los materiales que componen el seleccionador y su precio.

Tabla 3. Coste de los materiales que componen el seleccionador.

Referencia	Unidad	Concepto	Precio
MAT1	Unidad [Ud.]	Placa de aluminio 0,2 x 2 x 100	4,20 €
MAT2	Unidad [Ud.]	Guías	10,09 €
MAT3	Unidad [Ud.]	Material Estructura	208,72 €
MAT4	Unidad [Ud.]	Yugo de acero	111,67 €
MAT5	Unidad [Ud.]	Bloque de neodimio	24,26 €
MAT6	Unidad [Ud.]	Placa de acero galvanizad	9,18 €
MAT7	Unidad [Ud.]	Tornillo Allen M5 x 10	0,25€
MAT8	Unidad [Ud.]	Tornillo cabeza almohada M5 x 10	0,20 €
MAT9	Unidad [Ud.]	Tuerca hexagonal M5	0,15 €
MAT10	Unidad [Ud.]	Arandela M5	0,05 €

2.4. Otros materiales

En este apartado se detalla el coste de los materiales que han permitido el montaje y ajuste del seleccionador en la cámara de ensayos del ciclotrón de Sevilla:

Tabla 4. Coste de materiales añadidos que no componen el seleccionador.

Referencia	Unidad	Concepto	Precio
ADMAT1	Unidad [Ud.]	Plataforma métrica de laboratorio	907,50 €
ADMAT2	Unidad [Ud.]	Tablero perforado de sujeción	132,64 €

2.5. Alquileres

El coste asociado al alquiler de los equipos que han permitido llevar a cabo el proyecto se detalla a continuación:

Tabla 5. Coste de alquiler de instalaciones por hora.

Referencia	Unidad	Concepto	Precio
A1	Horas [h]	Alquiler ciclotrón	75 €
A2	Horas [h]	Alquiler laboratorio para medidas campo magnético	25 €

2.6. Procesos

Finalmente, el coste de los procesos industriales a llevar a cabo queda (“Tabla 6”):

Tabla 6. Coste de procesos industriales por hora.

Referencia	Unidad	Concepto	Precio
C1	Horas [h]	Corte con sierra manual	10 €
D1	Horas [h]	Desbarbado	10 €

3. Metodología y Amortizaciones

En este apartado, estableceremos una serie de criterios con el objetivo de una correcta obtención de costes sobre el proyecto.

En primer lugar, he de mencionar que nos fijamos en los precios actuales de las licencias anuales de los programas informáticos que aparecen en este presupuesto y que han sido utilizados.

En segundo lugar, en cuanto a la mano de obra, obtenemos sus salarios de la “Tabla 1”, teniendo en cuenta la diferencia entre los distintos perfiles participantes.

Finalmente, se emplea la amortización de los productos realizados durante el propio estudio en referencia a los equipos y programas informáticos (hardware y software), en vez de su coste de adquisición. El cálculo lo realizaremos mediante la siguiente fórmula:

$$A = C \times t_{amort} \times \frac{a}{T_{\acute{u}til}}$$

Donde:

- **A** es la amortización total del recurso informático durante el proyecto.
- **C** es el coste de adquisición
- **t_{amort}** es el tiempo total de uso del recurso durante el proyecto.
- **a** es el porcentaje de la infraestructura tecnológica que se pretende amortizar (en nuestro caso el 100%).
- **T_{útil}** es el tiempo de vida útil del propio software o hardware.

Además, cabe mencionar que, para el cálculo de la amortización, se han tenido en cuenta jornadas de 8 horas y 273 días laborables. Comenzaremos a continuación con la tabla que representa la amortización de los equipos y programas que se han utilizado.

Tabla 7. Amortización de recursos IT.

Producto	Coste de adquisición [€]	Vida útil [años]	Coste [€/h]	Tiempo amortización [h]	Porcentaje utilización [%]	Amortización [€]
Ordenador portátil	1.400,00 €	5	0,1282 €	450	95%	57,69 €
Sistema operativo	220,00 €	5	0,0201 €	450	95%	9,05 €
Pack Office 365	99,00 €	1	0,0453 €	205	43%	9,29 €
SolidWorks 2022	4.999,00 €	1	2,2889 €	35	7%	80,11 €

No obstante, los datos importantes para el resto del documento son los del coste [€/h], pues se separará el número de horas empleadas en cada tarea dentro del proyecto, con los diferentes recursos.

4. Obtención de Costes

En este apartado, observaremos tablas vinculadas a los cuadros de precios descompuestos sobre las tareas referidas con anterioridad, entre las que destacamos

Obtención de Costes

simulaciones, medidas del campo magnético, estudio y análisis de resultados, redacción, planos, montaje de configuraciones del seleccionador, y montaje del seleccionador en bancos de ensayos y medidas. Se engloba la preparación de ensayos, así como la utilización de equipos para el logro de las conclusiones.

4.1. Costes Fijos

En este apartado se recogen los costes fijos del proceso, que son principalmente los asociados al material empleado.

4.1.1. Construcción

Tabla 8. Coste de los elementos físicos que componen el seleccionador según cantidad

Referencia	Unidad [ud.]	Concepto	Precio [€/ud.]	Unidades	Importe [€]
MAT1	[Ud.]	Placa de aluminio 0,2 x 2 x 100	4,20 €	18	75,60 €
MAT2	[Ud.]	Guías	10,09 €	18	181,62 €
MAT3	[Ud.]	Material Estructura	208,72 €	1	208,72 €
MAT4	[Ud.]	Yugo de acero	111,67 €	9	1.005,03 €
MAT5	[Ud.]	Bloque de neodimio	24,26 €	18	436,68 €
MAT6	[Ud.]	Placa de acero galvanizado	9,18 €	18	165,24 €
MAT7	[Ud.]	Tornillo Allen M5 x 10	0,25 €	15	3,75 €
MAT8	[Ud.]	Tornillo cabeza almohada M5 x 10	0,20 €	36	7,20 €
MAT9	[Ud.]	Tuerca hexagonal M5	0,15 €	51	7,65 €
MAT10	[Ud.]	Arandela M5	0,05 €	66	3,30 €
TOTAL					2.094,79 €

4.1.2. Proceso Experimental

Tabla 9. Coste de los elementos materiales necesarios para llevar a cabo el proceso experimental.

Referencia	Unidad [ud.]	Concepto	Precio [€/ud.]	Unidades	Importe [€]
AMAT1	[Ud.]	Plataforma métrica de laboratorio	907,50 €	1	907,50 €
AMAT2	[Ud.]	Tablero perforado de sujeción	132,64 €	1	132,64 €
TOTAL					1.040,14 €

4.2. Costes variables

En este apartado se recogen los costes asociados a la mano de obra, utilización de herramientas y procesos industriales.

4.2.1. Construcción

Tabla 10. Coste del tratamiento y ensamblado de los materiales para la construcción del dispositivo.

Referencia	Unidad [ud.]	Concepto	Precio [€/ud.]	Unidades	Importe [€]
MO1	[h]	Ingeniero Técnico	14,10 €	5,95	83,90 €
C1	[h]	Corte con sierra manual	10 €	2	20 €
D1	[h]	Desbarbado	10 €	0,05	1 €
Costes Directos (CD)					104,40 €
Costes Directos Complementarios (2% CD)					2,09 €
Costes Indirectos (8% CD)					8,35 €
TOTAL					219,23 €

4.2.2. Simulaciones

Tabla 11. Coste de los elementos necesarios para llevar a cabo los estudios sobre el trazado de partículas

Referencia	Unidad [ud.]	Concepto	Precio [€/ud.]	Unidades	Importe [€]
MO1	[h]	Ingeniero Técnico	14,10 €	200	2.820,00 €
IT1	[h]	Ordenador portátil	0,13 €	200	25,64 €
IT2	[h]	Sistema operativo2023	0,02 €	200	4,02 €
IT3	[h]	Pack Office 365	0,05 €	5	0,23 €
IT5	[h]	Visual Studio Code	0 €	200	0,00 €
Costes Directos (CD)					2.849,89 €
Costes Directos Complementarios (2% CD)					57,00 €
Costes Indirectos (8% CD)					57,00 €
TOTAL					2993,77 €

4.2.3. Medidas

Tabla 12. Coste de los elementos necesarios para la medida del campo magnético

Referencia	Unidad [ud.]	Concepto	Precio [€/ud.]	Unidades	Importe [€]
MO1	[h]	Ingeniero Técnico	14,10 €	30	423,00 €
A2	[h]	Alquiler laboratorio para medidas campo magnético	25 €	30	750,00 €
IT1	[h]	Ordenador portátil	0,13 €	5	0,64 €
IT2	[h]	Sistema operativo2023	0,02 €	5	0 €
IT3	[h]	Pack Office 365	0,05 €	5	0 €
Costes Directos (CD)					1.173,97 €
Costes Directos Complementarios (2% CD)					23,48 €
Costes Indirectos (8% CD)					93,92 €
TOTAL					1.291,36 €

4.2.4. Proceso experimental

Tabla 13. Coste del proceso experimental

Referencia	Unidad [ud.]	Concepto	Precio [€/ud.]	Unidades	Importe [€]
MO1	[h]	Ingeniero Técnico	14,10 €	24	338,40 €
MO3	[h]	Investigador	23,61 €	24	566,64 €
A1	[h]	Alquiler ciclotrón	75 €	24	1.800,00 €
IT1	[h]	Ordenador portátil	0,13 €	10	1,28 €
IT2	[h]	Sistema operativo2023	0,02 €	10	0,20 €
IT3	[h]	Pack Office 365	0,05 €	5	0,23 €
IT5	[h]	Visual Studio Code	0,00 €	5	0,00 €
Costes Directos (CD)					2.706,75 €
Costes Directos Complementarios (2% CD)					54,13 €
Costes Indirectos (12% CD)					324,81 €
TOTAL					3.085,69 €

4.2.5. Redacción de documentos

Tabla 14. Coste monetario de la redacción de los documentos que recogen el trabajo realizado.

Referencia	Unidad [ud.]	Concepto	Precio [€/ud.]	Unidades	Importe [€]
MO1	[h]	Ingeniero Técnico	14,10 €	212	2.989,20 €
MO2	[h]	Investigador	23,61 €	2	47,22 €
MO3	[h]	Profesor Investigador	25,62 €	6	153,72 €
IT1	[h]	Ordenador portátil	0,13 €	235	30,13 €
IT2	[h]	Sistema operativo2023	0,02 €	235	4,72 €
IT3	[h]	Pack Office 365	0,05 €	5	0,23 €
IT4	[h]	SolidWorks 2022	2,29 €	35	80,11 €
IT5	[h]	Visual Studio Code	0,00 €	5	0,00 €
Costes Directos (CD)					3.305,33 €
Costes Directos Complementarios (2% CD)					66,11 €
Costes Indirectos (2% CD)					66,11 €
TOTAL					3.437,54 €

5. Presupuestos finales

En este apartado se detalla finalmente el coste total del proyecto

5.1. Presupuesto parcial

En la tabla que se puede ver a continuación, observamos los valores que se han obtenido a lo largo de la realización del presupuesto con el objetivo de realizar el presupuesto parcial. Este no contiene tasas impositivas, beneficios, o gastos generales.

Tabla 15. *Coste parcial del proyecto*

Unidades de obra	Unidad	Coste tipo	Medición [h]	Importe [€]
Construcción	Horas [h]	Fijo	-	2.094,79 €
		Variable	8 h	219,23 €
Simulaciones	Horas [h]	Variable	200 h	2.993,77 €
Medidas	Horas [h]	Variable	30 h	1.291,36 €
Proceso experimental		Fijo	-	1.040,14 €
		Variable	24 h	3.085,69 €
Redacción de documentos		Variable	212 h	3.437,54 €
Total				14.162,53 €

Este presupuesto parcial es directamente el presupuesto de ejecución material. No obstante, para obtener el presupuesto total será necesario añadir el presupuesto de ejecución por contrata y el IVA.

5.2. Presupuesto total

El presupuesto de ejecución por contrata a añadir, teniendo en cuenta gastos generales y beneficio industrial, queda:

Tabla 16. *Gastos generales y beneficio industrial.*

	[%]	Importe [€]
Gastos generales	8%	1.133,00 €
Beneficio Industrial	6%	849,75 €
Total		1.982,75 €

Además, el IVA en España a fecha del proyecto es del 21%, por lo que el presupuesto total se obtiene:

Tabla 17. *Presupuesto total*

Presupuesto de ejecución material	14.162,53 €
Presupuesto de ejecución por contrata	1.982,75 €
IVA (21%)	2.974,13 €
Presupuesto total	19.119,41 €

Con lo expuesto en este documento, es posible afirmar que el presupuesto total asciende a los DIECINUEVE MIL CIENTO DIECINUEVE CON CUARENTA Y UNO euros.