



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

Diseño de una herramienta mediante programación
matemática para la gestión y organización del horario anual
en una escuela de secundaria en Valencia

Trabajo Fin de Máster

Máster Universitario en Ingeniería Industrial

AUTOR/A: García Pérez, Alex

Tutor/a: Maheut, Julien Philippe Dominique

Cotutor/a: Devís Gallego, Sergio

CURSO ACADÉMICO: 2023/2024

AGRADECIMIENTOS

Quiero aprovechar la ocasión para expresar mi agradecimiento a aquellas personas que me han acompañado y apoyado a lo largo de todo este camino, ya que posiblemente sin ellos, esta meta no se hubiera alcanzado.

En primer lugar, a mis padres, quienes han sido mi pilar fundamental, brindándome siempre su amor incondicional, apoyo y sabios consejos. Sin su ayuda en todos los aspectos difícilmente habría conseguido alcanzar lo que he conseguido hasta ahora.

A mi pareja, por su comprensión, por su apoyo, por su insistencia de ayuda en los momentos más desafiantes. También ha sido un componente clave para llegar hasta aquí.

A mis compañeros de la carrera y del máster, con quienes he compartido innumerables horas de biblioteca, trabajo en equipo y momentos inolvidables. Juntos hemos superado muchas adversidades y su amistad ha sido indispensable.

A la Universitat Politècnica de València por brindarme una formación académica sólida y por proporcionar el entorno adecuado para mi desarrollo personal y profesional.

A todos los profesores que he tenido a lo largo de estos años, por transmitir sus conocimientos y enseñanzas.

Finalmente, un especial agradecimiento a mi tutor, Julien, cuyo apoyo, guía y valiosos consejos han sido esenciales para la realización de este trabajo. Gracias por tu paciencia y por haberme orientado a lo largo de este proceso. Mantén siempre esa pasión que te caracteriza en tu profesión, debes servir de faro para quienes seguirán tus pasos.

A todos vosotros, mi más profundo agradecimiento.



RESUMEN

En un contexto educativo en constante evolución, la gestión eficiente del horario anual en una escuela de secundaria en Valencia emerge como un desafío complejo. La necesidad de optimizar el tiempo de los estudiantes y profesores, garantizar una distribución equitativa de las asignaturas y facilitar el proceso de planificación curricular son aspectos críticos que requieren una atención minuciosa. Además, factores como la disponibilidad de recursos y las preferencias de los involucrados plantean retos adicionales que deben abordarse de manera integral.

El presente trabajo tiene como objetivo diseñar una herramienta de programación matemática que permita la gestión y organización eficaz del horario anual en la escuela de secundaria en Valencia. Para lograr este propósito, se realizará un análisis exhaustivo de las restricciones y requisitos específicos de la institución educativa. Se implementarán técnicas avanzadas de programación matemática y se desarrollará un modelo matemático personalizado que tenga en cuenta las preferencias de los estudiantes y profesores, las limitaciones de los recursos y las directrices curriculares. Asimismo, se llevarán a cabo pruebas piloto y se recopilarán datos para evaluar la viabilidad y eficacia de la herramienta propuesta.

El documento académico consistirá en una introducción que establecerá el marco teórico y contextual de la investigación, seguida de una revisión exhaustiva de la literatura en torno a la programación matemática en el ámbito educativo. Posteriormente, se detallará la metodología utilizada para el desarrollo de la herramienta, incluyendo la descripción del modelo matemático y los diferentes desarrollos implementados. Se presentarán los resultados obtenidos de las pruebas piloto y se analizarán en profundidad, destacando las fortalezas y limitaciones del enfoque propuesto. Finalmente, se ofrecerán recomendaciones prácticas para la implementación y mejora continua de la herramienta, junto con posibles direcciones para futuras investigaciones en el campo de la programación matemática aplicada a la gestión escolar.

Palabras Clave: Programación matemática; Herramienta; Profesorado; Grupos; Alumno; Asignatura

RESUM

En un context educatiu en constant evolució, la gestió eficient de l'horari anual en un institut de secundària a València emergeix com un repte complex. La necessitat d'optimitzar el temps dels estudiants i professors, garantir una distribució equitativa de les assignatures i facilitar el procés de planificació curricular són aspectes crítics que requereixen una atenció minuciosa. A més, factors com la disponibilitat de recursos i les preferències dels implicats plantegen reptes addicionals que han de ser abordats de manera integral.

El present treball té com a objectiu dissenyar una eina de programació matemàtica que permeta la gestió i organització eficaça de l'horari anual a l'institut de secundària a València. Per a aconseguir aquest propòsit, es realitzarà una anàlisi exhaustiva de les restriccions i requisits específics de la institució educativa. S'implementaran tècniques avançades de programació matemàtica i es desenvoluparà un model matemàtic personalitzat que tinga en compte les preferències dels estudiants i professors, les limitacions dels recursos i les directrius curriculars. A més, es duran a terme proves pilot i es recopilaran dades per avaluar la viabilitat i eficàcia de l'eina proposada.

El document acadèmic consistirà en una introducció que establisca el marc teòric i contextual de la investigació, seguida d'una revisió exhaustiva de la literatura al voltant de la programació matemàtica en l'àmbit educatiu. Posteriorment, es detallarà la metodologia utilitzada per al desenvolupament de l'eina, incloent la descripció del model matemàtic i els diferents elements implementats. Es presentaran els resultats obtinguts de les proves pilot i s'analitzaran en profunditat, destacant les forces i limitacions de l'abordatge proposat. Finalment, s'oferiran recomanacions pràctiques per a la implementació i millora contínua de l'eina, juntament amb possibles direccions per a futures investigacions en el camp de la programació matemàtica aplicada a la gestió escolar.

Paraules clau: Programació matemàtica; Eina; Professorat; Grups; Alumne; Assignatura



ABSTRACT

In an educational context that is constantly evolving, the efficient management of the annual schedule in a secondary school in Valencia emerges as a complex challenge. The need to optimize the time of students and teachers, ensure an equitable distribution of subjects, and facilitate the curriculum planning process are critical aspects that require careful attention. Additionally, factors such as the availability of resources and the preferences of those involved pose additional challenges that must be addressed comprehensively.

The present work aims to design a mathematical programming tool that allows for effective management and organization of the annual schedule in the secondary school in Valencia. To achieve this purpose, a comprehensive analysis of the specific constraints and requirements of the educational institution will be conducted. Advanced mathematical programming techniques will be implemented, and a customized mathematical model will be developed, taking into account the preferences of students and teachers, resource constraints, and curriculum guidelines. Additionally, pilot tests will be conducted, and data will be collected to assess the feasibility and effectiveness of the proposed tool.

The academic document will consist of an introduction that establishes the theoretical and contextual framework of the research, followed by a comprehensive review of the literature on mathematical programming in the educational field. Subsequently, the methodology used for the development of the tool will be detailed, including the description of the mathematical model and the different elements implemented. The results obtained from the pilot tests will be presented and analyzed in depth, highlighting the strengths and limitations of the proposed approach. Finally, practical recommendations for the implementation and continuous improvement of the tool will be provided, along with possible directions for future research in the field of mathematical programming applied to school management.

Keywords: Mathematical programming; Tool; Faculty; Groups; Student; Subject

ÍNDICE

DOCUMENTOS CONTENIDOS EN EL TFM

- Memoria
- Presupuesto
- Anexos

ÍNDICE DE LA MEMORIA

1.	INTRODUCCIÓN.....	1
1.1.	Objetivo del documento.....	1
1.2.	Estructura del documento.....	1
2.	DESCRIPCIÓN DEL ENTORNO	3
2.1.	Introducción	3
2.2.	Breve descripción del sistema educativo español.....	3
2.3.	Aproximación a la institución educativa.....	6
2.4.	Objeto del problema	9
2.5.	Objetivos del proyecto	10
2.6.	Conclusiones.....	11
3.	ANTECEDENTES TEÓRICOS.....	12
3.1.	Introducción	12
3.2.	Sistemas de información y bases de datos	12
3.3.	Métodos y técnicas de resolución de problemas	15
3.4.	Métodos para resolver problemas de gestión de horarios escolares	19
3.5.	Conclusiones.....	21
4.	ANÁLISIS DE LA SITUACIÓN ACTUAL	22
4.1.	Introducción	22
4.2.	Descripción de incidencias	22
4.3.	Identificación de las causas	23
4.4.	Análisis ES / NO ES.....	24
4.5.	Identificación de causa-raíz.....	24
4.6.	Diseño de requerimientos.....	27
4.7.	Conclusiones.....	28
5.	SELECCIÓN DE ALTERNATIVAS	29
5.1.	Introducción	29



Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

5.2.	Identificación de los criterios de evaluación	29
5.3.	Técnica del Proceso Analítico Jerárquico (AHP)	30
5.4.	Soluciones comerciales	32
5.5.	Sistemas Manuales	38
5.6.	Sistemas de optimización	40
5.7.	Elección final de la alternativa.....	41
5.8.	Conclusiones.....	47
6.	MODELADO	48
6.1.	Introducción	48
6.2.	Índices	48
6.3.	Parámetros y Variables de decisión	50
6.4.	Función Objetivo	52
6.5.	Restricciones	53
6.6.	Dominio de índices y variables de decisión	56
6.7.	Conclusiones.....	58
7.	ANÁLISIS Y ELECCIÓN DE LAS CARACTERÍSTICAS DE LA SOLUCIÓN	59
7.1.	Introducción	59
7.2.	Análisis y elección del lenguaje de programación.....	60
7.3.	Análisis y elección del solver	65
7.4.	Análisis y elección del programa de importación y exportación de datos ...	66
7.5.	Elección final del software.....	66
7.6.	Conclusiones.....	68
8.	DISEÑO DE LA APLICACIÓN Y LA HERRAMIENTA.....	69
8.1.	Introducción	69
8.2.	Metodología empleada en la aplicación y la herramienta	69
8.3.	Diseño de la interfaz de la aplicación	70
8.4.	Herramienta para la creación de horarios.....	74
8.5.	Conclusiones.....	94
9.	VALIDACIÓN DE LA HERRAMIENTA SOBRE EL CASO REAL	95
9.1.	Introducción	95
9.2.	Metodología empleada	95
9.3.	Validación del funcionamiento de la herramienta sobre el caso real de estudio	96

ÍNDICE

9.4.	Análisis de los resultados obtenidos	99
9.5.	Conclusiones.....	100
10.	CONCLUSIONES	101
10.1.	Conclusiones de la memoria	101
10.2.	Lecciones aprendidas	102
10.3.	Futuras líneas de trabajo.....	103
	REFERENCIAS	105

ÍNDICE DEL PRESUPUESTO

1.	CONSIDERACIONES GENERALES	1
2.	PRESUPUESTOS PARCIALES.....	2
3.	RESUMEN DEL PRESUPUESTO	19
4.	ANÁLISIS DE VIABILIDAD ECONÓMICA	20

ÍNDICE DE LOS ANEXOS

	ANEXO 1. CODIFICACIÓN COMPLETA DE LA HERRAMIENTA EN PYTHON.....	1
	ANEXO 2. CODIFICACIÓN DE LA INTERFAZ VISUAL EN VBA.....	21
	ANEXO 3. MANUAL DE USUARIO DE LA APLICACIÓN	26
	ANEXO 3. RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030.....	28

ÍNDICE DE FIGURAS

	Figura 1. Tabla de horario general de cada curso. Fte: Elaboración propia.....	6
	Figura 2. Ejemplo de un Sistema de Información (Sistema de planificación de la producción). Fte: Apuntes de la asignatura Sistemas de Información en la Empresa (SIE).....	12
	Figura 3. Modelado de una BBDD. Fte: Apuntes de la asignatura Sistemas de Información en la Empresa (SIE)	14
	Figura 4. Diagrama de Ishikawa tipo. Fte: Elaboración propia.....	17
	Figura 5. Diagrama de Pareto tipo. Fte: Web Significados.....	18
	Figura 6. Diagrama de causas y efecto (Ishikawa). Fte: Elaboración propia	25
	Figura 7. Inicio de la web Untis. Fte: Web Untis	36
	Figura 8. Plataforma WebUntis. Fte: Web Untis.....	36
	Figura 9. Productos que ofrece la plataforma UNTIS. Fte: Web Untis.....	37
	Figura 10. Esquema de las fases de implementación de la herramienta. Fte: Elaboración propia	67



Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

Figura 11. Menú principal de la interfaz de la aplicación Fte: Elaboración propia	71
Figura 12. Desplegable de los diferentes grupos existentes. Fte: Elaboración propia...	71
Figura 13. Desplegable de la plantilla de docentes. Fte: Elaboración propia	72
Figura 14. Desplegable de las opciones de generación de horarios. Fte: Elaboración propia	72
Figura 15. Ejemplo del horario del grupo 4B en la aplicación. Fte: Elaboración propia	73
Figura 16. Ejemplo de horario del profesor K3 en la aplicación. Fte: Elaboración propia	74
Figura 17. Extracto de la tabla de información del grupo 4A. Fte: Elaboración propia .	75
Figura 18. Extracto de la tabla de las asignaturas específicas totales de la escuela. Fte: Elaboración propia.....	76
Figura 19. Extracto de la tabla de los docentes totales del centro. Fte: Elaboración propia	77
Figura 20. Extracto de la tabla que relaciona Profesor y Departamento. Fte: Elaboración propia	78
Figura 21. Extracto de la tabla del parámetro HS. Fte: Elaboración propia	79
Figura 22. Extracto de la tabla del parámetro HSe. Fte: Elaboración propia	79
Figura 23. Extracto de la tabla del parámetro HSo. Fte: Elaboración propia	80
Figura 24. Extracto de la tabla del parámetro CP. Fte: Elaboración propia	80
Figura 25. Extracto de la tabla del parámetro CPe. Fte: Elaboración propia	81
Figura 26. Extracto de la tabla del parámetro CPo. Fte: Elaboración propia	81
Figura 27. Ejemplo de la hoja de un profesor ejemplo de la base de datos perteneciente las preferencias de los docentes (preferencias del profesorado). Fte: Elaboración propia	82
Figura 28. Definición de los índices I, I_0, D y G del código del modelo 1. Fte: Elaboración propia	83
Figura 29. Definición de los índices E, O y K en Python, importados a través de la base de datos “datos del colegio”. Fte: Elaboración propia	84
Figura 30. Definición del parámetro HSe en Python, importados a través de la base de datos “datos del colegio”. Fte: Elaboración propia	85
Figura 31. Definición del parámetro HSo en Python, importado a través de la base de datos “datos del colegio”. Fte: Elaboración propia	85
Figura 32. Definición del parámetro CPe en Python, importado a través de la base de datos “datos del colegio”. Fte: Elaboración propia	86
Figura 33. Definición del parámetro CPo en Python, importado a través de la base de datos “datos del colegio”. Fte: Elaboración propia	86

Figura 34. Definición del parámetro CS en Python, importado a través de la base de datos “preferencias del profesorado”. Fte: Elaboración propia.....	87
Figura 35. Definición de las variables de decisión creadas para el modelo 1 en Python. Fte: Elaboración propia.....	87
Figura 36. Definición de la Función Objetivo del modelo 1 en Python. Fte: Elaboración propia	88
Figura 37. Definición de las restricciones del modelo 1 en Python 1/4. Fte: Elaboración propia	88
Figura 38. Definición de las restricciones del modelo 1 en Python 2/4. Fte: Elaboración propia	89
Figura 39. Definición de las restricciones del modelo 1 en Python 3/4. Fte: Elaboración propia	89
Figura 40. Definición de las restricciones del modelo 1 en Python 4/4. Fte: Elaboración propia	90
Figura 41. Definición de los índices I, D y G del código del modelo 2. Fte: Elaboración propia	90
Figura 42. Definición de los índices J y K en Python, importados a través de la base de datos "datos del colegio". Fte: Elaboración propia	91
Figura 43. Definición del parámetro HS en Python, importado a través de la base de datos "datos del colegio". Fte: Elaboración propia	91
Figura 44. Definición del parámetro CP en Python, importado a través de la base de datos "datos del colegio". Fte: Elaboración propia	92
Figura 45. Definición de la Función Objetivo del modelo 2 en Python. Fte: Elaboración propia	92
Figura 46. Definición de las restricciones del modelo 2 en Python 1/2. Fte: Elaboración propia	93
Figura 47. Definición de las restricciones del modelo 2 en Python 2/2. Fte: Elaboración propia	93
Figura 48. Horario semanal anual del grupo 3A. Fte: Elaboración propia	96
Figura 49. Horario personalizado del profesor K27. Fte: Elaboración propia	97
Figura 50. Horario personalizado del profesor K2. Fte: Elaboración propia	98



ÍNDICE DE TABLAS

Tabla 1. Tabla relacional Profesor-Departamento. Fte: Elaboración propia.....	7
Tabla 2. Cuadrante del horario tipo de un grupo. Fte: Elaboración propia	8
Tabla 3. Aulas disponibles con su respectiva capacidad. Fte: Elaboración propia.....	9
Tabla 4. Tabla tipo del análisis Es / No Es. Fte: Elaboración propia	15
Tabla 5. Tabla prototipo de los cinco porqués. Fte: Elaboración propia.....	19
Tabla 6. Análisis ES / NO ES. Fte: Elaboración propia	24
Tabla 7. Los 5 porqués. Fte: Elaboración propia	26
Tabla 8. Causa raíz del problema. Fte: Elaboración propia.....	26
Tabla 9. Escala de preferencias del método AHP. Fte: Toskano & Gérard, 2005.....	31
Tabla 10. Nomenclatura utilizada en el método AHP. Fte: Elaboración propia.....	42
Tabla 11. Matriz de comparación del criterio C1. Fte: Elaboración propia.....	42
Tabla 12. Matriz de comparación del criterio C2. Fte: Elaboración propia.....	43
Tabla 13. Matriz de comparación del criterio C3. Fte: Elaboración propia.....	43
Tabla 14. Matriz de comparación del criterio C4. Fte: Elaboración propia.....	43
Tabla 15. Matriz de comparación del criterio C5. Fte: Elaboración propia.....	43
Tabla 16. Matriz de comparación de los criterios. Fte: Elaboración propia.....	44
Tabla 17. Matriz normalizada del criterio C1. Fte: Elaboración propia.....	44
Tabla 18. Matriz normalizada del criterio C2. Fte: Elaboración propia.....	44
Tabla 19. Matriz normalizada del criterio C3. Fte: Elaboración propia.....	44
Tabla 20. Matriz normalizada del criterio C4. Fte: Elaboración propia.....	45
Tabla 21. Matriz normalizada del criterio C5. Fte: Elaboración propia.....	45
Tabla 22. Matriz normalizada de los criterios. Fte: Elaboración propia.....	45
Tabla 23. Vectores promedio de cada criterio respecto de cada alternativa. Fte: Elaboración propia.....	45
Tabla 24. Vector promedio de prioridad de los criterios. Fte: Elaboración propia	46
Tabla 25. Cálculo de la prioridad global de las alternativas. Fte: Elaboración propia....	46
Tabla 26. Descripción de índices del modelo matemático 1. Fte: Elaboración propia ..	49
Tabla 27. Descripción de índices del modelo matemático 2. Fte: Elaboración propia ..	50
Tabla 28. Dominio de los índices del modelo matemático completo. Fte: Elaboración propia	56
Tabla 29. Escala de la matriz CS. Fte: Elaboración propia.....	57

Tabla 30. Matriz multicriterio de la elección del lenguaje de programación. Fte:
Elaboración propia..... 64

ÍNDICE DE GRÁFICOS

Gráfico 1. Resultados finales del AHP. Fte: Elaboración propia 46

Gráfico 2. Comparativa de h/sem esperadas y h/sem reales para cada grupo. Fte:
Elaboración propia..... 96

Gráfico 3. Resultados de los niveles de CS de los profesores con preferencias. Fte:
Elaboración propia..... 98

MEMORIA

1. INTRODUCCIÓ

1.1. OBJETIVO DEL DOCUMENTO

El presente documento tiene como objetivo principal abordar la compleja problemática asociada a la gestión eficiente del horario escolar en instituciones educativas. A través del desarrollo de una herramienta basada en técnicas de optimización combinatoria, se pretende diseñar un sistema eficaz y adaptable que permita una asignación óptima de horarios para los cursos y profesores, teniendo en cuenta las restricciones inherentes a la disponibilidad de recursos y a las preferencias de los involucrados.

Este proyecto se propone facilitar la distribución equitativa de asignaturas, cursos y aulas, maximizando la utilización de los recursos disponibles y minimizando posibles conflictos derivados de superposiciones de horarios o restricciones específicas.

Se busca, en última instancia, mejorar la eficiencia y la calidad de la planificación educativa, optimizando el aprovechamiento del tiempo y de los recursos, y proporcionando un entorno propicio para el desarrollo académico y personal de los estudiantes y profesores involucrados.

1.2. ESTRUCTURA DEL DOCUMENTO

Este documento está estructurado en tres grandes bloques: memoria, presupuesto y anexos. Por lo que respecta a la memoria, esta está dividida en capítulos los cuales se describen brevemente a continuación:

- ◆ **Capítulo 2: Descripción del entorno.**
Se hablará por una parte del sistema educativo español y de la institución académica en cuestión, y por otra se detallarán el objeto del problema y los objetivos del proyecto.
- ◆ **Capítulo 3: Antecedentes teóricos.**
Expondrá las bases teóricas fundamentales que se consideran imprescindibles para poder llevar a cabo la resolución de la problemática planteada.
- ◆ **Capítulo 4: Análisis situacional.**
Se dedicará a realizar una evaluación exhaustiva del estado actual en la institución, con el objetivo de identificar el origen de los problemas que están obstaculizando la consecución incorrecta de resultados.
- ◆ **Capítulo 5: Selección de alternativas.**

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

Se presentarán tres alternativas para la creación del horario, y se evaluarán estas opciones considerando una serie de criterios establecidos, con el fin de determinar cuál de ellas es la más adecuada para su implementación.

◆ **Capítulo 6: Modelado.**

Se abordarán todos los elementos esenciales para la construcción del modelo matemático. Este modelo servirá como base para la creación de la herramienta.

◆ **Capítulo 7: Análisis y elección del software.**

Se realizará un análisis de los diferentes lenguajes de programación, programas de importación y exportación de datos, así como de los diversos solvers existentes en el mercado. La finalidad es evaluar detalladamente cada opción, con el propósito de tomar decisiones informadas para la creación de la herramienta.

◆ **Capítulo 8: Diseño de la aplicación y la herramienta.**

Desarrollará los aspectos relacionados con el diseño tanto de la herramienta como de la aplicación creada. Se prestará atención especial a elementos como la interfaz.

◆ **Capítulo 9: Validación de la herramienta sobre un caso real**

Se validará la herramienta diseñada mediante la aplicación al caso real de estudio, y así ver en qué medida cumple los objetivos del problema.

◆ **Capítulo 10: Conclusiones**

Recogerá las conclusiones, así como las líneas futuras identificadas y la reflexión del autor.

2. DESCRIPCIÓN DEL ENTORNO

2.1. INTRODUCCIÓN

En este segundo capítulo se procede a describir el entorno en el que se encuentra el objeto de estudio, con el fin de obtener una primera aproximación a los elementos que serán más relevantes durante el desarrollo del proyecto.

A continuación, se abordarán los diferentes subcapítulos con el objetivo de poner en contexto el tema que se va a tratar (2.2) y de recopilar los diversos datos de interés (2.3).

Por último, se proporcionará una explicación clara del objeto y objetivos del problema a abordar (2.4).

2.2. BREVE DESCRIPCIÓN DEL SISTEMA EDUCATIVO ESPAÑOL

2.2.1. Introducción y estructura del sistema educativo

El sistema educativo español se ha desarrollado a lo largo de los años con el objetivo de ofrecer una educación integral y equitativa para todos los ciudadanos. Basado en una combinación de tradiciones educativas y marcos legales, el sistema ha experimentado diversas reformas en un esfuerzo continuo por mejorar la calidad y la eficiencia de la enseñanza en España.

Este sistema se organiza en diferentes etapas y niveles, que van desde la educación infantil y la educación primaria hasta la educación secundaria obligatoria, el bachillerato y la educación superior. Cada una de las etapas tiene sus propias características y objetivos específicos, diseñados para satisfacer las necesidades educativas de los estudiantes en diferentes etapas de su desarrollo académico y personal.

Centrando la atención en España, coexisten diferentes tipos de instituciones educativas:

- **Escuela pública**
Reciben financiamiento público y son gestionadas por las administraciones educativas de las comunidades autónomas o entidades locales. La educación es gratuita y siguen el currículo oficial establecido por las autoridades educativas. Suelen ser la opción más común para la mayoría de los estudiantes.
- **Escuelas concertadas**
Reciben fondos públicos, pero también pueden recibir aportaciones económicas de los padres y otros recursos adicionales, además pueden

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

tener ciertas restricciones de acceso. Se suele solicitar una pequeña cuota por alumno y suele seguir el currículum oficial, pero pueden tener ciertas particularidades. Mantienen un equilibrio entre financiamiento público y privado.

- **Escuelas privadas**

Dependen totalmente de recursos privados, como las tarifas pagadas por los padres y otras fuentes privadas. Como las escuelas concertadas, es posible que establezcan unos requisitos específicos para la admisión. La educación es financiada íntegramente por las tarifas pagadas por los padres y pueden seguir el currículum oficial u optar por programas internacionales.

- **Centros educativos especializados**

Se especializan en un área educativa específica, como música, arte, deportes, idiomas, entre otros. Pueden ser públicos, concertados o privados, pero comparten la característica de ofrecer una educación centrada en una disciplina especializada, además pueden variar según el tipo de centro y la especialización.

Estas instituciones ofrecen una variedad de enfoques pedagógicos y programas educativos adaptados a las necesidades de los estudiantes y a las demandas del mercado laboral. Cabe destacar que esta clasificación no es exhaustiva y existen otros tipos de centros educativos con enfoques particulares. Además, la legislación educativa puede variar en las diferentes comunidades autónomas de España.

2.2.2. Leyes y políticas educativas

El sistema educativo español se rige por una serie de leyes y políticas que buscan establecer el marco normativo y las directrices para la educación en el país. A lo largo de las últimas décadas, varias leyes han marcado hitos importantes en el ámbito educativo. A continuación, se destacan algunas de las leyes y políticas más relevantes:

- I. **Ley General de Educación (LGE) de 1970:**

Esta ley estableció las bases del sistema educativo español durante varios años. Introdujo la Educación General Básica (EGC) y dividió la educación en etapas.

- II. **Ley Orgánica del Derecho a la Educación (LODE) de 1985:**

Esta ley introdujo importantes cambios y reconoció el derecho a la educación. Estableció el carácter obligatorio y gratuito de la educación básica y garantizó la participación de los diferentes sectores en la comunidad educativa.

- III. **Ley Orgánica de Ordenación General del Sistema Educativo (LOGSE) de 1990:**

Esta ley fue un intento significativo de modernizar y democratizar el sistema educativo. Introdujo la Educación Infantil, la Educación Primaria, la

Educación Secundaria Obligatoria (ESO) y el Bachillerato. También reforzó la participación de la comunidad educativa en la gestión de los centros.

IV. Ley Orgánica de Mejora de la Calidad de la Educación (LOMCE) de 2013:

Esta ley, buscó mejorar la calidad de la educación y hacer cambios en la estructura curricular. Introdujo evaluaciones externas y modificó el enfoque de asignaturas como la Religión y la Educación para la Ciudadanía.

V. Ley Orgánica de Educación (LOE) de 2006:

La LOE sustituyó a la LOGSE y mantuvo algunas de sus características, aunque realizó modificaciones importantes. Se centró en aspectos como la equidad, la inclusión y la diversificación del currículo.

VI. Ley Orgánica de Educación y Protección de la Infancia (LOEPI) de 2021:

Esta ley tiene como objetivo abordar cuestiones de equidad y protección de la infancia. Busca establecer un marco normativo que garantice el derecho a la educación y promueva la igualdad de oportunidades.

2.2.3. Desafíos y perspectivas futuras

A pesar de los avances logrados, el sistema educativo español se enfrenta a diversos desafíos, como la equidad en el acceso a la educación, la mejora de los estándares de calidad, la formación docente y la integración de las nuevas tecnologías en el aula. En este contexto, se están implementando diversas iniciativas y políticas para abordar estos desafíos y mejorar la educación en España.

Un ejemplo de estas puede ser la mejora de formación en el profesorado de matemáticas. Una de las denuncias permanentes ante la administración educativa es el distanciamiento entre la formación inicial de los profesores con la realidad docente en los centros de infantil y primaria (*Blanco & Climent, 2024*).

Otra demanda que se está viendo solicitada es la de hacer un examen de acceso a la universidad común en toda España. Un 81% de los jóvenes pide un examen de acceso a la universidad común en todo el país. En educación es difícil llegar a acuerdos, pero parece existir un abrumador consenso entre todos los ciudadanos españoles en que el actual modelo de acceso a la universidad es injusto y debería cambiarse. Apenas una minoría de todos los encuestados defiende el actual distrito universitario único, por el cual la nota que obtiene el alumno en su comunidad autónoma le sirve para entrar en cualquier campus de España. Únicamente el 6,7% de los que tienen entre 16 y 17 años defiende el modelo que hay ahora. Los estudiantes de las regiones más exigentes se quedan sin plaza en las carreras con demanda porque les pasan por delante los de otros territorios que han hecho exámenes más fáciles (*Sanmartín, 2024*).

Por último, cabe destacar un artículo que habla sobre la repercusión que tiene el uso de los teléfonos móviles en los menores de 16 años. La profesora Ángela Sánchez-Pérez reunió más de 70.000 firmas para pedir prohibir el uso de los teléfonos móviles por parte de los niños. Esta profesora destaca que es necesario realizar una ley que evite que un menor tenga un teléfono 24 horas en el bolsillo, ya que, si no es imposible vetar su acceso a contenidos muy graves, y que intervienen de manera significativa en la adquisición de aprendizaje. Considera esto como una ayuda a prevenir problemas relacionados con la salud

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

mental y física en los jóvenes y permitirles disfrutar plenamente de su infancia y adolescencia sin las presiones asociadas con estos dispositivos (*Sociedad, 2024*).

Cierto es que el sistema educativo español ha experimentado una evolución significativa en las últimas décadas, con un enfoque renovado en la calidad, la equidad y la modernización. A pesar de los desafíos que persisten, se deben realizar esfuerzos continuos para fortalecer y mejorar el sistema educativo, con la visión de ofrecer una educación de calidad y equitativa para todos los estudiantes en España.

Actualmente el sistema educativo español está controlado por el [Ministerio de Educación, Formación Profesional y Deportes](#). En este enlace, se puede ver en detalle todos aquellos aspectos relacionados con la educación en España.

2.3. APROXIMACIÓN A LA INSTITUCIÓN EDUCATIVA

2.3.1. Cursos Escolares

El colegio se compone de un número específico de cursos, los cuales se subdividen en varios grupos, ya sea debido a una limitación de capacidad o a la selección de las distintas asignaturas. La figura adjunta detalla el número total de cursos que conforman esta escuela secundaria.

Código	Nombre	At. Estudio	Común/Troncal	Modalidad/Específica	Optativa	Horas Semanales Totales (h/sem)
1A	1 ESO Grupo A	1	27	0	2	30
1B	1 ESO Grupo B	1	27	0	2	30
2A	2 ESO Grupo A	1	27	0	2	30
2B	2 ESO Grupo B	1	27	0	2	30
3A	3 ESO Grupo A	1	30	0	2	33
3B	3 ESO Grupo B	1	30	0	2	33
4A	4 ESO Grupo A	1	22	6	3	32
4B	4 ESO Grupo B	1	22	6	3	32
1ART	1 Bachiller Artes	0	15	13	4	32
1HUM	1 Bachiller Humanidades	0	15	13	4	32
1CI	1 Bachiller Ciencias	0	15	13	4	32
2ART	2 Bachiller Artes	0	15	14	4	33
2HUM	2 Bachiller Humanidades	0	15	14	4	33
2CI	2 Bachiller Ciencias	0	15	14	4	33

Figura 1. Tabla de horario general de cada curso. Fte: Elaboración propia

Como bien se observa en la Figura 1, esta representa una tabla que contiene una serie de datos de gran interés para abordar el problema a resolver. En total, consta de siete columnas, cada una correspondiente a un dato específico de cada uno de los grupos presentes en el colegio:

- **Código:** dígitos específicos asignados a cada uno de los grupos para facilitar el trabajo en los capítulos posteriores.
- **Nombre:** indica el curso y el grupo correspondiente.
- **At. Estudio:** horas semanales designadas para atención al estudio.
- **Común/Troncal:** horas semanales asignadas a asignaturas generales.
- **Modalidad/Específica:** horas semanales asignadas a asignaturas modalidad/específicas.
- **Optativa:** horas semanales asignadas a asignaturas optativas.

- **Horas semanales totales:** suma de todas las columnas correspondientes a horas semanales asignadas a cada tipo.

Cabe destacar que, según el curso, las asignaturas se dividen en dos o tres tipos, lo que implica que no todos los cursos cuentan con asignaturas de modalidad/específica. Cada curso funciona de una manera particular y debe ser tratado como tal.

2.3.2. Profesorado

En el marco de esta institución educativa se ha establecido un conjunto definido de profesores, cada uno afiliado a un departamento específico. Dicha plantilla docente se caracteriza por la diversidad de funciones asignadas a los individuos. Algunos profesores se encargan de impartir múltiples asignaturas a un mismo curso, mientras que otros se dedican exclusivamente a una materia concreta para un determinado grupo de estudiantes. Además, existen casos en los que los profesores tienen la capacidad de enseñar diferentes materias a distintos cursos. La Tabla 1 refleja la composición del cuerpo docente de esta institución, donde puede observarse el número de profesores que conforman cada departamento:

Tabla 1. Tabla relacional Profesor-Departamento. Fte: Elaboración propia

N.º de Profesores	Departamento
3	Biología y Geología
3	Inglés
1	Griego
1	Economía
1	Educación Física
4	Educación Plástica
2	Filosofía
3	Física y Química
4	Geografía e Historia
1	Informática
1	Latín
6	Lengua Castellana y Literatura
8	Matemáticas
2	Música
1	Francés
6	Valencià
3	Tecnología

Es de destacar que los profesores desempeñan un papel crucial en la resolución de la problemática planteada, dado que, como se analizará en detalle en próximos capítulos, sus preferencias generan limitaciones dentro del contexto de estudio.

2.3.3. Horario

Para obtener un entendimiento completo del ambiente educativo en esta escuela, es fundamental familiarizarse con el horario de clases. El horario potencial de cada curso se encuentra delimitado dentro del franja horario de 8:00 a 15:05. Además, es importante tener en cuenta la presencia de dos pausas programadas: una de 30 minutos a mitad de la

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

mañana (de 10:45 a 11:15) y otra de 10 minutos después del mediodía (de 13:05 a 13:15). En consecuencia, se establecen siete intervalos de tiempo en los cuales se puede programar las clases. Considerando que la semana consta de cinco días lectivos, cada curso cuenta con un total de 35 tramos horarios semanales disponibles para asignar el total de las correspondientes asignaturas.

Por otro lado, es relevante tener en cuenta que la duración de cada clase es de 55 minutos y varía según el tipo de curso, tal como se menciona en la sección 2.3.1, donde se explica que algunos cursos disponen de una hora adicional de clase. En relación con esto, el cuadrante del horario se organiza de la siguiente manera:

Tabla 2. Cuadrante del horario tipo de un grupo. Fte: Elaboración propia

	Lunes	Martes	Miércoles	Jueves	Viernes
8:00-8:55					
8:55-9:50					
9:50-10:45					
11:15-12:10					
12:10-13:05					
13:15-14:10					
14:10-15:05					

Cabe destacar que los espacios en blanco que aparecen en la Tabla 2 representan los tramos horarios que deben ser asignados a las respectivas asignaturas, omitiendo los períodos de descanso debido a su falta de relevancia en la resolución del problema.

2.3.4. Aulas

Las aulas representan los espacios dentro de la institución donde se llevan a cabo las clases. Generalmente, cada grupo de curso se asigna a una aula específica para la mayoría de sus clases, a excepción de aquellas asignaturas optativas que podrían impartirse en otros espacios designados.

Este factor puede resultar importante en la organización del horario, dado que anualmente el número de estudiantes en cada grupo puede variar y no todas las aulas poseen la misma capacidad. Por lo tanto, esta consideración debe ser tomada en cuenta cuando sea pertinente. En total, se cuenta con 24 aulas disponibles para la impartición de las distintas materias.

A continuación, se detalla una lista que enumera las aulas existentes junto con sus respectivas capacidades.

Tabla 3. Aulas disponibles con su respectiva capacidad. Fte: Elaboración propia

Aulas	Capacidad (alumnos)
APOY01	12
APOY02	12
4ESO	28
4VE	26
4D1	27
D211	18
INF	26
INF02	16
LABOCNC	22
LABOFIS	26
1ESO	28
1ESO01	26
2ESO	27
2ESO01	27
TEC	26
3A	26
3B	27
USMUL	80
1BACH	26
1BACH01	35
1BACH02	35
2BACH	26
2BACH01	26
2BACH02	35

2.4. OBJETO DEL PROBLEMA

A lo largo de este capítulo introductorio sobre la institución educativa objeto de estudio se ha puesto de manifiesto la amplia gama de factores que deben considerarse al operar en este entorno.

Así pues, el objeto del problema va a consistir en el diseño de una herramienta que sea capaz de organizar un horario anual dentro de esta institución. Esta herramienta deberá estar sujeta al cumplimiento de todos los requisitos demandados por los diferentes participantes implicados en el proceso.

Es importante resaltar que este empeño no se antoja sencillo, ya que la satisfacción de todos los requisitos demandará el despliegue de una herramienta capaz de operar con eficacia y eficiencia.

2.5. OBJETIVOS DEL PROYECTO

El objetivo primordial previamente señalado se ha esbozado de manera general en la sección 1.1, donde se ha delineado con precisión la finalidad del documento. En esta instancia, el enfoque se centra en delinear los objetivos con mayor nivel de detalle. Considerando que la meta central consiste en la concepción de un horario escolar eficaz y eficiente en todos los niveles de la institución, se han establecido una serie de objetivos a nivel de proyecto con el propósito de abordar de manera integral la principal problemática en cuestión.

El **primer objetivo** es el de **realizar un análisis situacional que permita obtener una comprensión detallada y objetiva de la situación en la que se encuentra la institución**. Para ello se deberá recopilar, examinar y evaluar información relevante con el fin de tomar decisiones informadas y diseñar estrategias efectivas.

El **segundo objetivo** es el de **desarrollar un modelo matemático que optimice la distribución de los recursos disponibles**, considerando las necesidades grupales y las preferencias individuales, con el fin de maximizar la eficiencia en el uso del tiempo y los espacios escolares a lo largo del año académico.

El **tercero objetivo** a tratar en el proyecto es ser capaz de **implementar un sistema de gestión de horarios escolares que incorpore un mecanismo de adaptación dinámica** capaz de manejar cambios inesperados, como variaciones en la matrícula estudiantil, modificaciones en el personal docente y ajustes en la disponibilidad de aulas en respuestas a las diversas casuísticas surgidas durante el curso.

El **cuarto objetivo** será tratar de **diseñar una interfaz de usuario intuitiva y personalizable para la herramienta de gestión de horarios**, que facilite la interacción entre los usuarios y el sistema, permitiendo la visualización detallada de los horarios de clases, la realización de ajustes y la gestión eficaz de conflictos potenciales, con el objetivo de proporcionar una experiencia amigable y eficiente para los administradores, docentes y estudiantes involucrados en el proceso de programación y seguimiento del horario escolar anual.

Por último, el **quinto objetivo** consistirá en **validar la herramienta diseñada con el caso real de estudio propuesto**. Esto implica verificar la eficacia, precisión y utilidad práctica de la herramienta en un entorno del mundo real. La validación implica poner a prueba la herramienta en condiciones reales para asegurarse de que cumple con los requisitos y objetivos para los cuales fue diseñada.

Por tanto, con el fin de satisfacer las expectativas establecidas, se consideran todos estos objetivos durante el análisis de la situación actual y en todas las fases del proyecto.

2.6. CONCLUSIONES

En el presente capítulo, se ha proporcionado una descripción exhaustiva de la institución educativa objeto de estudio, abarcando su actividad principal, así como los elementos más relevantes, el objeto del problema y los objetivos del proyecto.

Se ha presentado una narrativa detallada de la actividad llevada a cabo, haciendo hincapié en los aspectos destacados como los cursos escolares, el cuerpo docente, el horario y las aulas. A partir de estos elementos, se han establecido de manera precisa los tres objetivos primordiales que deben alcanzarse para garantizar el éxito en la resolución integral del proyecto.

Con base en el análisis preliminar, los apartados siguientes se adentran en una exposición más detallada de la problemática en cuestión, las bases teóricas que guardan una relación directa, así como la metodología aplicada para alcanzar el objetivo propuesto.

3. ANTECEDENTES TEÓRICOS

3.1. INTRODUCCIÓN

En el presente capítulo, se procederá a contextualizar aquellos conceptos teóricos estrechamente vinculados con el desarrollo de la herramienta en cuestión.

En consecuencia, los subcapítulos siguientes se enfocarán en brindar una explicación detallada de ciertos conceptos fundamentales en el ámbito de los sistemas de información y las bases de datos (3.2). A continuación, se detallarán diversas técnicas de resolución de problemas (3.3), y, por último, se expondrán posibles algoritmos heurísticos y modelos matemáticos que revisten una gran utilidad para el propósito planteado (3.4).

3.2. SISTEMAS DE INFORMACIÓN Y BASES DE DATOS

3.2.1. Introducción a los sistemas de información

En la era de la información en la que vivimos, los sistemas de información se han vuelto fundamentales para el funcionamiento eficiente y efectivo de las organizaciones en todos los sectores. Un sistema de información se define como un conjunto formal de procesos/rutinas/tareas que, operando sobre una colección estructurada de datos (según las necesidades de cada organización), recopilan, elaboran y distribuyen la información (o parte de ella) necesaria para las operaciones de dicha institución, y para las actividades de dirección y control correspondientes (toma de decisiones) para desempeñar su actividad de acuerdo con su propósito (Aznar, 1996).

Estos sistemas no solo incluyen hardware y software, sino también personas, procesos, datos y redes. En la Figura 2 se puede ver un ejemplo de un sistema de información.

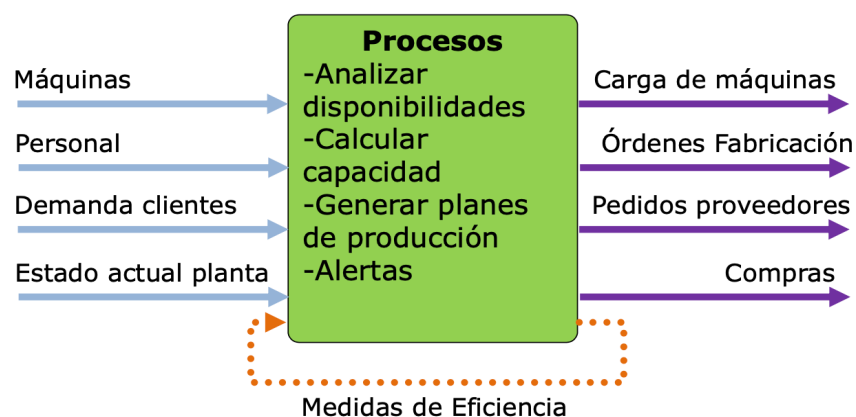


Figura 2. Ejemplo de un Sistema de Información (Sistema de planificación de la producción).
Fte: Apuntes de la asignatura Sistemas de Información en la Empresa (SIE)

La importancia de los sistemas de información radica en su capacidad para recopilar, almacenar, procesar y distribuir información de manera eficiente, lo que facilita la toma de decisiones informada y oportuna. Estos sistemas permiten a las organizaciones mejorar su productividad, coordinar operaciones complejas y adaptarse rápidamente a los cambios del entorno.

3.2.2. Tipos de sistemas de información

Los sistemas de información se pueden clasificar en varias categorías según su función y su contribución a los procesos organizacionales. Algunos de los tipos más comunes incluyen:

- **Sistemas de procesamiento de transacciones (TPS):** Estos sistemas se utilizan para procesar transacciones diarias, como pedidos, facturación, nóminas y registros de inventario.
- **Sistemas de información de gestión (MIS):** Los MIS se centran en el procesamiento de información para ayuda a los gerentes a tomar decisiones informadas y estratégicas.
- **Sistemas de soporte a la toma de decisiones (DSS):** Estos sistemas se diseñan para ayudar a los usuarios a tomar decisiones semiestructuradas o no estructuradas utilizando datos e información interactiva.
- **Sistemas de información ejecutiva (EIS):** Los EIS proporcionan a los altos directivos información relevante y oportuna para ayudar en la toma de decisiones estratégicas.

Además de estos, existen otros tipos de sistemas de información, como sistemas de automatización de oficinas, sistemas expertos y sistemas de información geográfica, cada uno diseñado para satisfacer necesidades específicas dentro de una organización.

3.2.3. Componentes esenciales de los sistemas de información

Tal y como se ha comentado anteriormente en el apartado 3.2, los sistemas de información están compuestos por una serie de componentes interdependientes que trabajan en conjunto para procesar y distribuir información. Estos componentes incluyen:

- **Hardware:** Se refiere a los dispositivos físicos utilizados para la entrada, procesamiento, almacenamiento y salida de datos. Como pueden ser computadoras, servidores, dispositivos de almacenamiento y periféricos.
- **Software:** Incluye los programas informáticos que permiten a los usuarios interactuar con el sistema y procesar los datos de manera efectiva. Esto abarca desde sistemas operativos hasta aplicaciones de software específicas utilizadas para funciones empresariales específicas.
- **Datos:** Son la materia prima de los sistemas de información y representan hechos o cifras sin procesar que se recopilan y almacenan en bases de datos.
- **Procedimientos:** Se refiere a los métodos y reglas establecidos para garantizar que el sistema funcione de manera eficiente y que los usuarios cumplan con los protocolos establecidos.

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

- **Personas:** Los usuarios, administradores y expertos en sistemas de información son esenciales para operar, mantener y utilizar eficazmente los diferentes sistemas de información en una organización.
- **Redes:** Las redes de comunicación permiten la transferencia de datos y la comunicación entre los distintos componentes de un sistema de información, así como entre sistemas de información separados.

3.2.4. Conceptos fundamentales de las bases de datos

Las bases de datos desempeñan un papel crucial en la gestión de grandes volúmenes de datos en organizaciones de todos los tamaños y sectores. Una base de datos se define como un conjunto de datos relacionados entre sí, almacenados en conjunto sin redundancias innecesarias e independientes de los programas que los usan (*De Miguel, A.; Piattini, M.; Marcos, 2000*). Algunos de los conceptos fundamentales relacionados con las bases de datos incluyen:

- **Modelo de datos:** Se refiere a la forma en que se estructuran los datos en la base de datos, lo que determina cómo se organizan, almacenan y manipulan los datos.
- **Lenguajes de bases de datos:** Estos lenguajes permiten a los usuarios interactuar con la base de datos, realizar consultas, extraer información y realizar modificaciones en los datos.
- **Sistemas de gestión de bases de datos (DBMS):** Son programas informáticos que permiten a los usuarios crear y manipular bases de datos, gestionar la seguridad de los datos y garantizar la integridad y la disponibilidad de la información almacenada.
- **Tipos de bases de datos:** Las bases de datos se pueden clasificar en diferentes tipos, como bases de datos relacionales, bases de datos jerárquicas, bases de datos de redes y bases de datos distribuidas, cada una con su propio enfoque y estructura de almacenamiento de datos.

En la Figura 3 se puede apreciar la metodología empleada en el proceso de modelado de un base de datos (BBDD). Como puede observarse, este modelo está compuesto por tres diseños: conceptual, lógico y físico. El conceptual representa la información necesaria que debe contener una BBDD, el lógico describe la estructura de la BBDD, y el físico construye, mediante toda la información recopilada, la BBDD física.

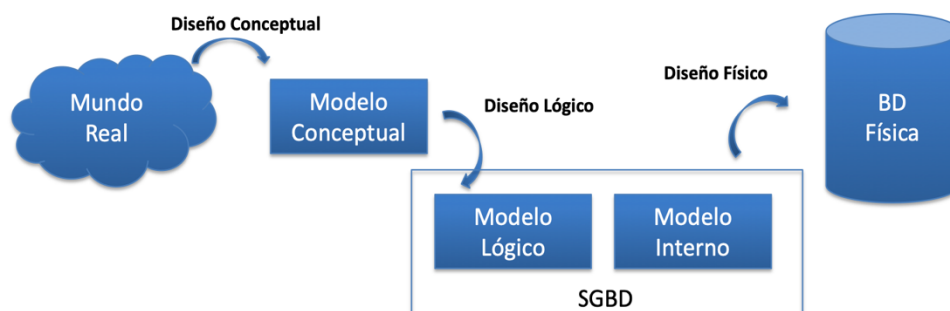


Figura 3. Modelado de una BBDD. Fte: Apuntes de la asignatura Sistemas de Información en la Empresa (SIE)

3.2.5. Importancia de la integración entre sistemas de información y bases de datos

La integración efectiva entre los sistemas de información y las bases de datos es esencial para garantizar que la información se gestione de manera eficiente y esté disponible cuando se necesite. La capacidad de los sistemas de información para acceder a bases de datos actualizadas y precisas permite a las organizaciones mejorar la toma de decisiones, optimizar los procesos y mantener un flujo constante de información entre diferentes departamentos y funciones organizativas.

La integración de sistemas de información y bases de datos también ayuda a reducir la redundancia de datos, mejorar la seguridad de la información y facilitar el análisis y la generación de informes en tiempo real. Al garantizar una integración sólida y coherente, las organizaciones pueden mejorar su eficiencia operativa y su capacidad para adaptarse rápidamente a los cambios imprevistos que puedan acontecer.

En resumen, los sistemas de información y las bases de datos forman el núcleo de la gestión y el uso efectivo de la información en las organizaciones modernas. Comprender la evolución, los tipos, los componentes y los conceptos fundamentales de estos sistemas es esencial para poder optimizar aquellos procedimientos implicados en el funcionamiento de una organización.

3.3. MÉTODOS Y TÉCNICAS DE RESOLUCIÓN DE PROBLEMAS

3.3.1. Análisis Es / No Es

Este análisis se emplea para clarificar y definir con precisión la naturaleza de un problema o situación específica. Se utiliza en varios campos, como la gestión de proyectos, la toma de decisiones, la solución de problemas y la identificación de requisitos.

Consiste en identificar y describir lo que un concepto, objeto o situación es y lo que no es. Esta técnica ayuda a comprender las características esenciales de una situación a través de una comparación en sus atributos o cualidades. Se puede utilizar para definir los límites de un problema, entender sus componentes fundamentales y establecer una base clara para la toma de decisiones.

Tabla 4. Tabla tipo del análisis Es / No Es. Fte: Elaboración propia

	Es	No Es
¿Qué?		
¿Dónde?		
¿Cuándo?		
¿Quién?		
¿Cómo se detecta o produce?		

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

El análisis “Es / No Es” se realiza mediante una serie de preguntas destinadas a explorar y delimitar las características claves de un tema específico. En la Tabla 4 puede observarse el aspecto gráfico que tendría un tabla de este tipo.

Al definir lo que es el objeto de estudio, se establecen las características esenciales que lo identifican. Por otro lado, al determinar lo que no es, se excluyen características que no son inherentes a la situación en cuestión.

Esta técnica ayuda a los individuos y equipos a obtener una comprensión clara y precisa de un problema, lo que facilita la toma de decisiones informadas y la generación de soluciones efectivas. Al identificar con claridad los límites y las características esenciales de un problema, se pueden evitar malentendidos y confusiones, lo que conduce a una resolución más eficiente y efectiva de problemas y desafíos.

3.3.2. Diagrama de Ishikawa

El diagrama de Ishikawa, también conocido diagrama de espina de pescado o diagrama causa-efecto, es una herramienta gráfica utilizada para identificar, explorar y visualizar todas las posibles causas de un problema específico o de un resultado determinado. Fue creado por Kaoru Ishikawa, un profesor de ingeniería industrial en la Universidad de Tokio, y se ha convertido en una herramienta fundamental en la gestión de la calidad y en la resolución de problemas en una variedad de campos, incluyendo la industria manufacturera, la atención médica y la gestión de proyectos.

Este diagrama está considerado como una de las siete herramienta de la calidad. El diagrama causa y efecto ayuda a identificar las posibles causas de un problema de forma ordenada y estructurada (*Edwin Garro, 2017*).

La estructura del diagrama de Ishikawa se asemeja a la estructura de un esqueleto de pescado, con una espina central que representa el problema o el efecto que se desea analizar. Los “huesos” que se ramifican desde la espina. Central representan las diversas categorías de posibles causas que podrían contribuir al problema principal. Estas categorías suelen incluir las 6Ms: método, mano de obra, máquina, material, medida y entorno, aunque en función del contexto del problema, se pueden adaptar, añadir o quitar categorías relevantes. Todo ello puede observarse en la Figura 4.

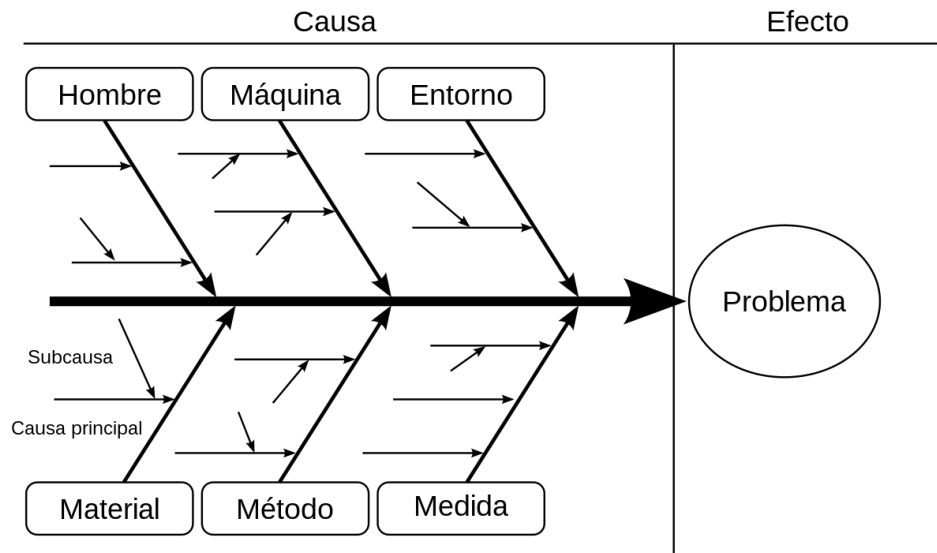


Figura 4. Diagrama de Ishikawa tipo. Fte: Elaboración propia

El proceso de elaboración de un diagrama de Ishikawa implica la realización de una lluvia de ideas con el objetivo de identificar todas las posibles causas que podrían contribuir al problema en cuestión (muchas veces es de gran utilidad que este paso se realice de la mano de un equipo multidisciplinario). Luego, estas causas se organizan y representan gráficamente en el diagrama, lo que permite visualizar de manera clara y sistemática la relación entre las causas potenciales y el efecto observado. Esto facilita la comprensión global del problema y ayuda a priorizar las áreas clave en las que enfocar los esfuerzos de mejora o solución de problemas.

El diagrama de Ishikawa se utiliza, por tanto, como una herramienta de análisis y resolución de problemas para promover una comprensión profunda y sistemática de las causas subyacentes de un problema, lo que permite tomar decisiones informadas y desarrollar estrategias efectivas para abordar y eliminar las causas raíz identificadas.

3.3.3. Diagrama de Pareto

Este diagrama es otra herramienta gráfica utilizada para visualizar y priorizar las causas de un problema o situación en función de su importancia relativa. Esta técnica se basa en el principio de que un pequeño número de causas (también conocidas como “factores vitales”) suele ser responsable de la mayoría de los problemas o efectos observados, mientras que la mayoría de las causas restantes contribuyen en menor medida.

El diagrama de Pareto es un gráfico en donde se organizan diversas clasificaciones de datos por orden descendente, de izquierda a derecha por medio de barras sencillas, después de haber reunido los datos para calificar las causas de modo que se pueda asignar un orden de prioridades (Rincón y Villarreal, s.f). La barra más larga indica la categoría más significativa o la causa principal, mientras que las barras restantes se disponen en orden descendente de importancia. En la Figura 5 puede observarse gráficamente todo lo mencionado anteriormente.

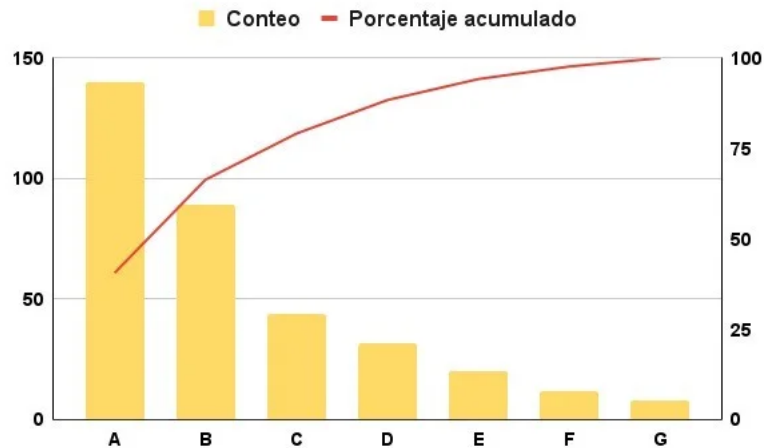


Figura 5. Diagrama de Pareto tipo. Fte: [Web Significados](#)

El gráfico de líneas, que se superpone al gráfico de barras en la Figura 5 muestra la acumulación porcentual de las frecuencias o los impactos de cada categoría. Esto permite identificar el punto en el cual la curva alcanza una pendiente significativa, lo que indica el punto en el que se concentra la mayor parte del problema o el efecto observado.

El objetivo del diagrama de Pareto es ayudar a centrarse en las áreas más críticas o significativas que requieren atención prioritaria. Al identificar y priorizar las causas principales, se pueden asignar recursos de manera más efectiva para abordar y solucionar los problemas más importantes, lo que conduce a una mejora significativa en la eficiencia y la calidad. Típicamente se relaciona al diagrama de Pareto con la regla 80/20, el 80% de los problemas provienen del 20% de las causas (*Edwin Garro, 2017*).

Este diagrama se utiliza en una amplia gama de campos, como la gestión de la calidad, la resolución de problemas, la toma de decisiones y la mejora continua, y es especialmente útil para identificar y abordar de manera proactiva las causas subyacentes de los problemas más relevantes en un proceso o sistema.

3.3.4. Estrategia de los 5 porqués

Esta es una técnica de análisis causa-raíz que busca identificar las causas fundamentales de un problema al realizar sucesivas preguntas “¿Por qué?” de manera iterativa. Este enfoque, originario del Sistema de Producción de Toyota y vinculado al método Kaizan, se utiliza para indagar más allá de las manifestaciones superficiales de un problema y llegar a sus causas subyacentes.

Los pasos que seguir en esta estrategia son los siguientes:

- 1. Identificación del problema:** comienza identificando claramente el problema específico que se desea abordar.
- 2. Pregunta Inicial - ¿Por Qué?:** formula dicha pregunta para indagar sobre la causa inmediata del problema. La respuesta se convierte en la base para la siguiente pregunta.

3. **Iteración de preguntas:** repite el proceso, formulando “¿Por Qué?” sucesivamente para cada respuesta dada. Se realiza este ciclo hasta alcanzar un nivel en el que la respuesta revela la causa raíz del problema.
4. **Análisis y acción correctiva:** una vez identificada. La causa raíz, se puede desarrollar una acción correctiva efectiva para abordar el problema en su origen.

Además, esta estrategia aporta una serie de beneficios:

- **Profundidad de análisis:** permite un análisis que va más allá de las manifestaciones evidentes del problema, explorando las capas más profundas.
- **Identificación de causas subyacentes:** facilita la identificación de las causas subyacentes que podrían no ser evidentes de manera inmediata.
- **Prevención de recurrencias:** al abordar la causa raíz, se contribuye a prevenir la recurrencia del problema en el futuro.

A continuación, puede observarse de que forma quedaría un análisis de este tipo. En la Tabla 5 puede apreciarse un prototipo de tabla que podría ser utilizado para ejecutar esta estrategia de los cinco porqués.

Tabla 5. Tabla prototipo de los cinco porqués. Fte: Elaboración propia

ID	Causa	¿Por Qué? #1	¿Por Qué? #2	¿Por Qué? #3	¿Por Qué? #4	¿Por Qué? #5
C1						
C2						
C3						
Cn						

Esta herramienta revela un interés significativo cuando se integra con un diagrama de causa-raíz (Ishikawa). A través de este diagrama, se identifican causas potenciales vinculadas a categorías específicas, y posteriormente se seleccionan para la aplicación de la técnica de los “cinco ¿por qué?”. Esta metodología implica formular preguntas secuenciales “¿Por qué?” con el propósito de investigar la causa inmediata de la categoría en cuestión. Este proceso iterativo se repite hasta alcanzar la causa raíz, proporcionando así una comprensión más profunda y sistemática de los factores subyacentes asociados al problema identificado.

3.4. MÉTODOS PARA RESOLVER PROBLEMAS DE GESTIÓN DE HORARIOS ESCOLARES

El diseño de una herramienta eficaz para la gestión y organización de un horario anual en un colegio plantea desafíos específicos que requieren enfoques especializados. En los siguientes subapartados se explicarán algunos de los métodos y enfoques efectivos para abordar este problema.

3.4.1. Programación entera

La programación entera se emplea para resolver de manera óptima un problema de asignación. Este método permite asignar asignaturas, aulas y profesores teniendo en cuenta una serie de restricciones y preferencias. Al formular el problema como un modelo matemático de programación entera, se pueden considerar variables binarias para representar la asignación o no de determinadas clases a ciertos horarios. A través de algoritmos de optimización, como el método de la rama y la poda, se pueden encontrar soluciones que cumplan con los requisitos específicos del horario escolar y minimicen los conflictos entre alumnos, asignaturas, aulas y profesores. La definición y la solución del problema de programación lineal va a depender de las restricciones impuestas dentro de cada departamento y los objetivos de este, adaptando su formulación a modelos de gran tamaño (*Dimopoulou & Miliotis, 2001*).

3.4.2. Enfoques metaheurísticos

Este enfoque se fundamenta en soluciones singulares, con el objetivo de desarrollar estrategias que posibiliten la creación de algoritmos de optimización, frecuentemente orientados hacia soluciones locales (*Sörensen & Glover, 2013*). Entre los algoritmos empleados destaca la búsqueda tabú (TS), donde la validación de las soluciones se realiza considerando una lista que evita que el modelo quede estancado en soluciones locales (*Battiti & Tecchiolli, 1994; Nagata, 2018*).

3.4.3. Algoritmos genéticos adaptados

Estos algoritmos se basan en principios evolutivos para encontrar soluciones ante problemas complejos, como puede ser el de la programación de un horario escolar. El algoritmo genético utiliza el concepto de evolución genética biológica integrando tres operadores: selección, cruce y mutación para la creación de la población y la selección de las mejores soluciones. Este proceso se repite por un número de generaciones determinadas (*Alnowaini & Aljomai, 2021; Assi et al., 2018*).

En este contexto, estos algoritmos generan y evalúan posibles horarios considerando la disponibilidad de recursos y las restricciones específicas de asignación de clases y aulas. A través de la selección, reproducción y mutación de horarios candidatos, los algoritmos genéticos adaptados pueden converger hacia soluciones que optimicen la distribución de clases a lo largo del año, minimizando los solapamientos y garantizando la equidad en la asignación de horarios para profesores y estudiantes.

3.4.4. Simulación de escenarios

La simulación de escenarios en el contexto de la gestión de horarios escolares implica modelar y probar diferentes configuraciones de horarios para evaluar su viabilidad y eficacia. Se utilizan datos históricos y preferencias de profesores y estudiantes para simular el desempeño y la viabilidad de los horarios propuestos.

A través de la simulación de múltiples escenarios y la evaluación de resultados en términos de eficiencia, equidad y satisfacción de las partes interesadas, se pueden identificar posibles deficiencias y realizar ajustes en la planificación del horario escolar para

mejorar la experiencia educativa y promover un entorno de aprendizaje equitativo y productivo.

La combinación de estos tres posibles de métodos de resolución permite abordar de manera efectiva la complejidad asociada con la gestión y organización de un horario anual en un colegio, garantizando la optimización de recursos, la equidad en la asignación de clases y la satisfacción de las necesidades de estudiantes y profesores.

3.5. CONCLUSIONES

En el presente tercer capítulo, se ha contextualizado los antecedentes teóricos estrechamente vinculados con el proyecto en cuestión, destacando la significativa relevancia de los sistemas de información y las bases de datos, así como de los métodos de resolución aplicables a esta clase de problemáticas.

Con este propósito, se ha realizado una exposición minuciosa acerca de la naturaleza de los sistemas de información y las bases de datos, exponiendo de manera detallada los diversos componentes inherentes a este ámbito, al tiempo que se han esclarecido la distintas tipologías existentes. Con relación a la metodología, se ha abordado con detalle una serie de enfoques potenciales para la resolución de problemas (Análisis Es / No Es, Diagrama de Ishikawa, Diagrama de Pareto), reservando una sección final para considerar metodologías adicionales que tengan la capacidad de resolver de forma efectiva y eficiente la problemática en cuestión.

A partir de este conjunto de conceptos fundamentales, el próximo capítulo se enfocará en un análisis actualizado de la institución educativa, con el propósito de facilitar una posterior resolución del problema conforme a la metodología detallada.

4. ANÁLISIS DE LA SITUACIÓN ACTUAL

4.1. INTRODUCCIÓN

En el presente capítulo, se procederá a exponer el estado vigente del proceso de confección del horario. En primer término, se detallarán las diversas incidencias que acontecen a lo largo de todo el mencionado proceso (4.2), con el propósito subsiguiente de identificar aquellos obstáculos que están obstruyendo la elaboración de un horario eficiente que propicie un rendimiento óptimo (4.3).

A continuación, se llevará a cabo un análisis de causa-raíz, previamente abordado de manera teórica, con el fin de llegar a conclusiones que permitan comprender cabalmente la situación actual (4.4 y 4.5).

Finalmente, se procederá a la confección de un diseño de requerimientos que aglutine la totalidad de la información previamente expuesta (4.6). Estos requerimientos constituirán el conjunto desde el cual se abordará el diseño integral de la herramienta destinada a la creación de horarios.

El análisis pretende proporcionar una visión esclarecedora sobre los elementos fundamentales que inciden en el estado actual del proceso. Además, es considerado que este proceso se presenta como un paso crucial, consolidando las necesidades y especificaciones identificadas con anterioridad, y sirviendo como la base estructurada y detallada para el desarrollo efectivo de la mencionada herramienta.

4.2. DESCRIPCIÓN DE INCIDENCIAS

En la actualidad, se evidencian dos incidencias primordiales que obstaculizan la consecución de una optimización del horario que resulte satisfactoria para la mayoría de las partes involucradas:

- **Incidencia de Satisfacción:** surge a raíz de las preferencias expresadas por los profesores en relación con los horarios que les serían asignados. No obstante, la mayoría de estas preferencias se enfrenta a limitaciones, ya que deben conciliarse con una variedad de factores adicionales que poseen condiciones de obligatoriedad. Este equilibrio entre las preferencias individuales y las restricciones inherentes al proceso de planificación constituye un desafío significativo.
- **Incidencia de Capacidad:** se manifiesta en la asignación de aulas para las distintas franjas horarias, siendo posible que dichas aulas no cuenten con la capacidad suficiente para albergar el número de alumnos matriculados. Esta falta de correspondencia entre la capacidad de aulas y la demanda

estudiantil contribuye a la generación de inconvenientes logísticos que afectan negativamente la calidad del proceso educativo.

La comprensión detallada de estas incidencias se erige como un paso esencial para abordar con eficacia los desafíos actuales en la optimización del horario, buscando así una solución integral que satisfaga las necesidades y expectativas de las diversas partes involucradas en el proceso.

4.3. IDENTIFICACIÓN DE LAS CAUSAS

En el contexto delineado, la identificación de las causas resulta crucial para delinear estrategias efectivas que permitan superar las dificultades presentes en el proceso de creación del horario. A continuación, se detallan las causas identificadas:

1. Desbalance entre preferencias docentes y restricciones operativas:

- *Descripción de la Causa:* la disparidad entre las preferencias expresadas por los profesores en cuanto a horarios deseados y las restricciones operativas impuestas por los factores obligatorios genera un desafío complejo. La conciliación de las variables se torna difícil, dando lugar a tensiones y a la imposibilidad de satisfacer todas las demandas de manera equitativa.
- *Impacto:* dificultades para lograr una distribución de horarios que concilie las preferencias docentes con los requerimientos logísticos y académicos, afectando la eficiencia general del proceso.

2. Insuficiencia de capacidad en las aulas asignadas:

- *Descripción de la Causa:* la asignación de aulas que carecen de la capacidad necesaria para albergar al número de alumnos matriculados conduce a una falta de correspondencia entre la oferta de espacios y la demanda estudiantil. Esto genera congestiones en el entorno académico y afecta la calidad de la experiencia educativa.
- *Impacto:* dificultades logísticas y académicas debido a la inadecuación entre la capacidad de las aulas y la cantidad de estudiantes, comprometiendo el desarrollo normal de las actividades académicas.

3. Complejidad en la coordinación de factores variables:

- *Descripción de la Causa:* la coordinación efectiva de variables como preferencias docentes, disponibilidad de aulas y requisitos académicos resulta en una tarea intrincada. La complejidad aumenta al intentar satisfacer múltiples demandas y restricciones simultáneamente.
- *Impacto:* retrasos en la elaboración del horario, dificultades en la toma de decisiones y posibles conflictos derivados de la complejidad inherente a la coordinación de factores variables.

La identificación precisa de estas causas proporciona una base sólida para el diseño de estrategias que aborden cada desafío de manera específica, con el objetivo de mejorar la eficacia y eficiencia del proceso de gestión y organización del horario académico.

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

El próximo subapartado se centrará en la distinción entre las causas que surgen como consecuencia de los problemas identificados. Este enfoque tiene como objetivo proporcionar un análisis detallado que permita comprender a fondo cuál es la raíz subyacente del problema que está obstaculizando el logro de los objetivos establecidos. En consecuencia, una de las mejores herramientas para conseguir esto es el análisis ES / NO ES

4.4. ANÁLISIS ES / NO ES

Como se ha evidenciado anteriormente, este tipo de análisis emerge como una herramienta sumamente valiosa para discernir entre las cuestiones que pueden influir o no en la consecución de objetivos o la ejecución de tareas específicas.

En esta instancia, los problemas ya han sido previamente identificados y serán utilizados en este subapartado. Este enfoque permitirá una explicación clara de los problemas y de sus respectivas causas, contribuyendo así a una comprensión más profunda y estructurada de los desafíos afrontados. Todo ello queda claramente reflejado en la Tabla 6.

Tabla 6. Análisis ES / NO ES. Fte: Elaboración propia

	ES	NO ES
¿Qué?	Creación de un horario académico general óptimo	Desequilibrio entre preferencias docentes y restricciones operativas Insuficiencia de capacidad en las aulas asignadas Complejidad en la coordinación de factores variables
¿Dónde?	En la escuela correspondiente	En la jefatura de estudios
¿Cuándo?	Cada año en la época de creación del horario (mes de Julio)	Durante el resto del año
¿Quién?	Los profesores que no ven satisfechas sus necesidades, y aquellos grupos que no caben en el aula asignada	Solamente el jefe de estudios
¿Cómo se detecta o produce?	Con la ejecución del horario anual desde el software actual	De forma muy aislada y ocasional

4.5. IDENTIFICACIÓN DE CAUSA-RAÍZ

El análisis de causa-raíz se erige como una herramienta esencial para comprender las causas fundamentales que generan las incidencias identificadas en el proceso de creación del horario académico.

ANÁLISIS DE LA SITUACIÓN ACTUAL

Tal y como se ha visto en capítulos anteriores, la finalidad es identificar las raíces subyacentes que alimentan los problemas, permitiendo así el desarrollo de soluciones más efectivas y duraderas.

Con el propósito de alcanzar los objetivos delineados, se ha llevado a cabo la elaboración de un diagrama de Ishikawa, con el fin de sintetizar de manera sistemática el problema y sus causas fundamentales, fundamentándose en las incidencias previamente expuestas. Todo ello puede observarse en la Figura 6.

Consecuentemente, se ha ejecutado un análisis mediante la técnica de los “cinco ¿por qué?”, con el propósito de discernir la raíz del problema a partir de las causas detalladas en el diagrama previamente expuesto. Este proceso y sus resultados han sido documentados en la Tabla 7.

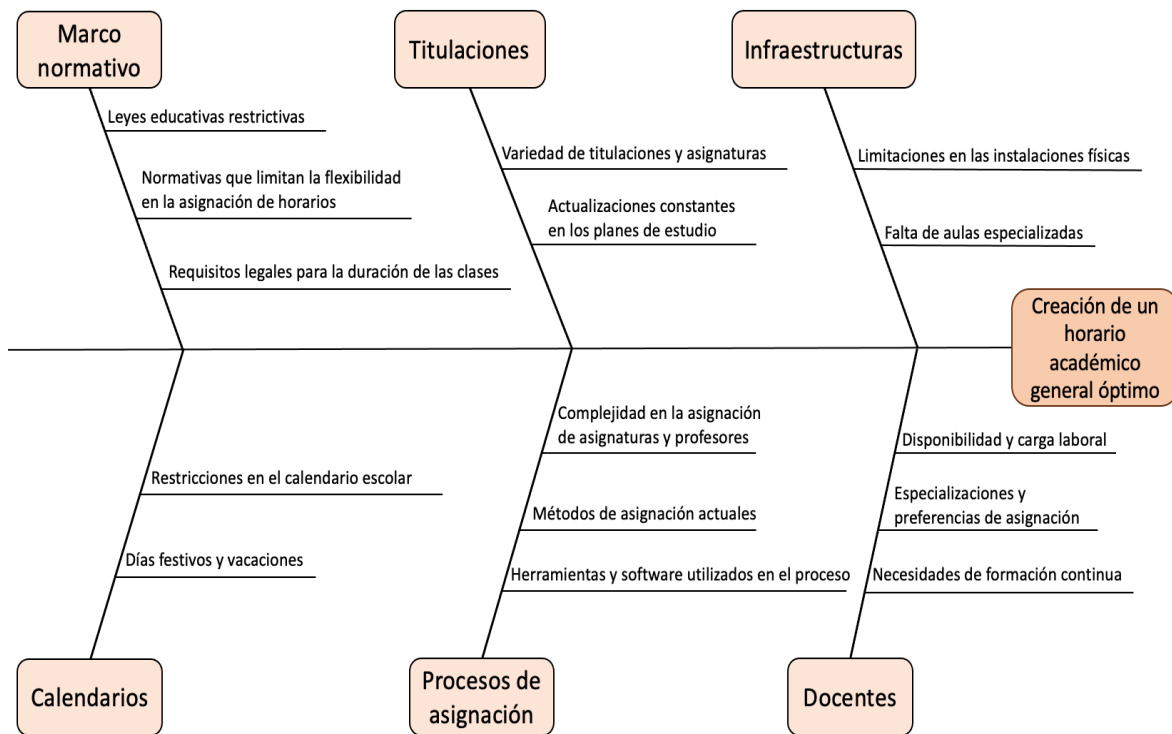


Figura 6. Diagrama de causas y efecto (Ishikawa). Fte: Elaboración propia

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

Tabla 7. Los 5 porqués. Fte: Elaboración propia

ID	Causa	¿Por Qué? #1	¿Por Qué? #2	¿Por Qué? #3
C1	Normativas que limitan la flexibilidad en la asignación de horarios	Abordan cuestiones específicas, no optimizan horarios	Falta de normativa específica de optimización de horarios	
C2	Limitación en las instalaciones físicas	Espacio limitado debido a ubicación geográfica	Zona urbana con restricciones de espacio	Falta de recursos financieros para ampliación de instalaciones
C3	Restricciones en el calendario escolar	Falta de adaptación eficiente a la asignación de horarios		
C4	Métodos de asignación actuales	No se consideran de manera adecuada la carga laboral y las preferencias de los docentes	Falta de revisión de los procesos de asignación en relación con las necesidades cambiantes	
C5	Herramientas y softwares utilizados en el proceso	Falta de conocimiento sobre herramientas de diseño de horarios	Falta de formación	
C6	Disponibilidad y carga laboral	Falta de distribución equitativa entre carga laboral y responsabilidad extracurricular		

A través de la realización de este análisis, se logra una clara visualización de las causas raíz del problema, siendo identificadas de manera destacada en la Tabla 7 mediante el sombreado de las celdas correspondientes. En la Tabla 8 se encuentran representadas de forma individual.

Tabla 8. Causa raíz del problema. Fte: Elaboración propia

ID	Causa raíz
CR11	Falta de recursos normativos y financieros que posibiliten contar con un entorno de trabajo bajo condiciones más eficientes
CR12	Falta de un sistema que recopile toda aquella información que sea necesaria en un mismo sitio.
CR13	Falta de formación de las diferentes partes de interés sobre la finalidad de su trabajo y el proceso global de creación del horario

4.6. DISEÑO DE REQUERIMIENTOS

Luego de completar el análisis de esta situación, el siguiente paso implica abordar la creación de un conjunto de requerimientos que posibilite la resolución del problema. Este proceso tiene como objetivo garantizar que todas las causas raíz sean abordadas mediante la implementación de nuevas funcionalidades que aborden de manera integral los desafíos identificados.

Por lo que respecta a los requerimientos, estos pueden ser de dos tipos: funcionales y no funcionales.

- **Requerimientos Funcionales:** son especificaciones detalladas que describen las funciones y capacidades específicas que debe tener un sistema, software o producto para cumplir con sus objetivos y satisfacer las necesidades del usuario. Se centran en lo que el sistema debe hacer y abarcan las distintas operaciones, servicios y funciones que el sistema debe ofrecer.
- **Requerimientos No Funcionales:** se centran en atributos del sistema que no están relacionados directamente con funciones específicas, sino con características más amplias de su operación. Además, abordan aspectos como la calidad, el rendimiento y las restricciones del sistema.

La elección de desarrollar requerimientos funcionales en este caso específico se fundamenta en la naturaleza de la solución propuesta desde el principio. Desde el inicio del proyecto, se ha determinado que la resolución a este problema consistirá en el diseño de una herramienta que genere un horario óptimo, cumpliendo con una serie de funciones predefinidas. En consecuencia, los requerimientos funcionales se centran en describir detalladamente las operaciones y capacidades específicas que esta herramienta deberá ofrecer para abordar eficazmente los desafíos identificados.

ID	Herramienta	Descripción	Causa raíz afectada
F1	Realización y seguimiento de todos los cursos y profesores	Una base de datos desarrollada en una interfaz donde se albergue las especificaciones más fundamentales y relevantes de cada grupo de estudio y de cada uno de los profesores	CR11 CR12 CR13
F2	Recopilación de toda la información necesaria	Una herramienta con una interfaz friendly que recopile toda la información necesaria en un mismo lugar. Esta misma herramienta puede ser una entrada de la herramienta de creación del horario	CR11 CR12

4.7. CONCLUSIONES

En este cuarto capítulo, se ha llevado a cabo un análisis de la situación con el objetivo de contextualizar las incidencias más significativas que están obstaculizando la consecución del objetivo primordial del presente proyecto. En este contexto, se han descrito e identificado diversas causas, sometiéndolas a un análisis ES / NO ES, con el propósito de esclarecer de manera concluyente cuáles son los problemas fundamentales y cuáles son sus causas subyacentes.

En el siguiente apartado, se ha considerado pertinente realizar un análisis causa-raíz mediante la implementación de un diagrama de Ishikawa. La elección de esta metodología se fundamenta en su eficacia para identificar las raíces subyacentes de los problemas. Asimismo, con el fin de enriquecer el análisis, se ha llevado a cabo la estrategia de los 5 porqués, desentrañando tres causas raíz. Con el propósito de contrarrestar estos déficits identificados, se ha elaborado un diseño de requerimientos, centrándose estos en requerimientos del tipo funcional.

Concluido este análisis situacional, se encuentra ahora la base para la toma de decisiones en relación con la estrategia a seguir en el diseño de la herramienta. En apartados anteriores, se han presentado diversos métodos y diversas metodologías de diseño que podrían ser implementados en la ejecución de dicha herramienta. El objetivo del próximo capítulo será determinar cuál de todos ellos se revela como el más eficaz para lograr una herramienta más efectiva, fiable y óptima.

5. SELECCIÓN DE ALTERNATIVAS

5.1. INTRODUCCIÓN

Este quinto capítulo se centrará, conforme a su título, en la selección de la alternativa más viable para la creación de un horario académico óptimo. Inicialmente, se llevará a cabo una concisa identificación de los criterios considerados más relevantes en la elección de una alternativa para la confección de un horario escolar. Los criterios serán detallados, y mediante la aplicación de la técnica del Proceso Analítico Jerárquico (AHP), se determinará cuál de ellos deberá ser implementado (5.2 y 5.3).

A continuación, se presentan las tres alternativas disponibles para la creación de un horario académico o escolar. Se llevará a cabo un análisis breve de cada una con el propósito de familiarizarse con ellas y comprender cuáles son sus ventajas e inconvenientes fundamentales. Las tres opciones que se han considerado en este caso para lograr un horario escolar adecuado son: soluciones comerciales, sistemas manuales y sistemas de optimización (5.4, 5.5 y 5.6).

Finalmente, después de haber introducido las alternativas y detallado sus características más esenciales, se llevará a cabo la aplicación de la técnica AHP. De esta manera, se podrá llegar a una conclusión sobre cuál de ellas se considera la opción más conveniente, conforme a los criterios de evaluación establecidos para la creación de este horario escolar (5.7).

5.2. IDENTIFICACIÓN DE LOS CRITERIOS DE EVALUACIÓN

Estos los cinco criterios que van a ser considerados en la evaluación de las diferentes alternativas:

1) Eficiencia en la asignación.

Capacidad para asignar eficientemente profesores a asignaturas y franjas horarias, maximizando la utilización de recursos y minimizando posibles conflictos.

2) Flexibilidad y Adaptabilidad.

Capacidad para adaptarse a cambios imprevistos o necesidades específicas, garantizando una gestión eficiente incluso en situaciones dinámicas.

3) Optimización de los recursos.

Eficacia en la utilización óptima de los recursos existentes, considerando factores como la capacidad de las aulas, las preferencias de los profesores y las restricciones horarias.

4) Facilidad en la implementación.

Simplicidad y practicidad en la implementación de la alternativa, considerando la accesibilidad para los usuarios y la viabilidad en términos de recursos y tiempo.

5) Coste total.

Coste total de la creación, ejecución y puesta en marcha del horario, así como posibles costes de mantenimiento, de suscripción a algún software o de licencia de algún programa.

5.3. TÉCNICA DEL PROCESO ANÁLITICO JERÁRQUICO (AHP)

Habiendo identificado los criterios que serán evaluados, se procederá a exponer la técnica se utilizará para determinar cuál es la mejor alternativa. La técnica seleccionada es el Proceso Analítico Jerárquico (AHP), una metodología de toma de decisiones que posibilita el abordaje de problemas complejos al descomponerlos en una estructura jerárquica de criterios, subcriterios y alternativas. El proceso requiere que quien toma las decisiones proporcione evaluaciones subjetivas respecto a la importancia relativa de cada uno de los criterios y que, después, especifique su preferencia con respecto a cada una de las alternativas de decisión y para cada criterio.

El resultado del AHP es una jerarquización con prioridades que muestran la preferencia global para cada una de las alternativas de decisión. El AHP permite de manera eficiente y gráfica organizar la información respecto de un problema, descomponerla y analizarla por partes, visualizar los efectos en los niveles y sintetizar (*Toskano & Gérard, 2005*). Se trata de desmenuzar un problema y luego unir todas las soluciones de los subproblemas en una conclusión (*Saaty, 1980*).

5.3.1. Jerarquización de criterios

Se identifican y jerarquizan los criterios relevantes para la toma de decisiones. En el contexto de la creación de un horario escolar, los criterios mencionados anteriormente son los que formarán parte de esta jerarquía.

Para ello, se utilizan las comparaciones pareadas, es decir, comparaciones de criterios dos a dos. Se presenta una escala con valores de 1 a 9 para calificar las preferencias de los dos elementos. Esta escala queda representada en la Tabla 9.

Tabla 9. Escala de preferencias del método AHP. Fte: Toskano & Gérard, 2005

Planteamiento verbal de la preferencia	Calificación Numérica
Extremadamente preferible	9
Entre muy fuertemente y extremadamente preferible	8
Muy fuertemente preferible	7
Entre fuertemente preferible y muy fuertemente preferible	6
Fuertemente preferible	5
Entre moderadamente preferible y fuertemente preferible	4
Moderadamente preferible	3
Entre igualmente preferible y moderadamente preferible	2
Igualmente preferible	1

Con esta escala de preferencias, se puede ya crear lo que se denomina como matriz de comparaciones pareadas, representada en la siguiente matriz:

$$A = \begin{pmatrix} 1 & a_{12} & \cdots & a_{1n} \\ 1/a_{12} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 1/a_{1n} & 1/a_{2n} & \cdots & 1 \end{pmatrix}$$

Decimos que A representa la matriz de comparaciones pareadas de n alternativas, donde a_{ij} es la medida de la preferencia de la alternativa i cuando se le compara con la alternativa j. Los valores de la diagonal de esta matriz son todos 1, ya que la alternativa en ese caso se está comparando consigo misma. Cabe destacar además que la medida a_{ij} es la inversa de la medida a_{ji} ($a_{ij} = 1/a_{ji}$).

El proceso de creación de esta matriz se realizará para todos los criterios seleccionados en este proceso analítico jerárquico. Además, también se realizará una matriz de comparaciones pareadas para los criterios, con el fin de conocer cuál es la prioridad de cada uno de ellos respecto de los otros.

5.3.2. Asignación de ponderaciones

Una vez elaboradas cada una de las matrices necesarias, se puede calcular lo que se denomina prioridad de cada uno de los elementos que se comparan. A esta parte del AHP se le conoce como **sintetización**. El proceso matemático preciso que se requiere para realizar tal sintetización implica el cálculo de valores y vectores característicos. El siguiente procedimiento de tres pasos proporciona una buena aproximación de las prioridades sintetizadas (Toskano & Gérard, 2005).

- Paso 1. Sumar los valores de cada columna de la matriz**
- Paso 2. Dividir cada elemento de la matriz entre el total su columna.**
Se forma la matriz normalizada.
- Paso 3. Calcular el promedio de los elementos de cada alternativa.**
Se construye el vector promedio.

Igualmente, este proceso habrá que hacerlo para cada uno de los criterios previamente escogidos para este método.

5.3.3. Matriz de prioridades

Esta matriz la forman todos los vectores promedios de las alternativas que se han calculado en la asignación de prioridades. Por un lado, está la matriz de prioridades que resume las prioridades para cada alternativa en términos de cada criterio, y por el otro, tendremos el vector promedio de la prioridad de los criterios. El producto de ambos elementos resulta en la prioridad global de cada alternativa de decisión (*Toskano & Gérard, 2005*).

$$\begin{pmatrix} P_{11} & P_{12} & \dots & P_{1m} \\ P_{21} & P_{22} & \dots & P_{2m} \\ \dots & \dots & \dots & \dots \\ P_{n1} & P_{n2} & \dots & P_{nm} \end{pmatrix} \begin{pmatrix} P'_1 \\ P'_2 \\ \dots \\ P'_m \end{pmatrix} = \begin{pmatrix} Pg_1 \\ Pg_2 \\ \dots \\ Pg_m \end{pmatrix}$$

Donde la matriz P de $n \times m$ representa la matriz de prioridades, siendo P_{ij} la prioridad de la alternativa i respecto al criterio j (cada columna de esta matriz es el vector promedio de cada criterio). Por otra parte, el vector P' representa el vector de prioridad de los criterios, y, por último, el producto de ambos resulta en el vector P_g que indica la prioridad global de cada una de las alternativas.

Por tanto, una vez se llegue a ese vector de prioridades globales, ya se podrá conocer cuál es la mejor alternativa teniendo en cuenta los criterios seleccionados.

En resumen, la técnica AHP proporciona un marco sistemático y analítico para la toma de decisiones, considerando la complejidad inherente de los problemas y permitiendo una evaluación precisa de las alternativas bajo múltiples criterios.

5.4. SOLUCIONES COMERCIALES

Las soluciones comerciales representan una alternativa para la creación eficiente de horarios escolares, ofreciendo una variedad de herramientas y plataformas diseñadas específicamente para abordar los desafíos asociados con la programación académica.

5.4.1. Características de las soluciones comerciales

Al evaluar la opción de soluciones comerciales para la creación de horarios escolares, resulta fundamental examinar detalladamente algunas de las alternativas disponibles en el mercado. Este proceso permitirá adquirir un conocimiento más profundo

sobre las características que estas soluciones pueden ofrecer. A continuación, se presentan dos posibles soluciones que abarcan una variedad de aspectos significativos:

1. Software de gestión académica integrada

Este tipo de soluciones abarcan diversas funcionalidades destinadas a facilitar la planificación y gestión de horarios escolares de manera eficiente. Algunas características notables incluyen:

- *Planificación automatizada:* estos sistemas cuentan con algoritmos de planificación automatizada que tienen en cuenta diversos factores, como la disponibilidad de profesores, la capacidad de las aulas y las preferencias individuales. Esto permite una asignación optimizada de asignaturas y horarios.
- *Integración con plataformas educativas:* facilitan la integración fluida con otras herramientas de gestión académica, como plataformas de gestión de calificaciones y sistemas de información estudiantil. Esto garantiza una colaboración eficiente entre diferentes aspectos del entorno académico.
- *Generación de informes y estadísticas:* ofrecen la capacidad de generar informes detallados y estadísticas sobre la eficiencia del horario, la utilización de los recursos y la equidad en la distribución de asignaturas. Esto proporciona a los administradores una visión completa de la eficiencia del horario.
- *Personalización de horarios:* permiten a los usuarios personalizar los horarios de acuerdo con las políticas y preferencias específicas de la institución, adaptándose a restricciones particulares y requisitos exclusivos.

2. Plataformas especializadas en programación académica

Estas soluciones se centran en ofrecer capacidades avanzadas de optimización y adaptabilidad a restricciones específicas para lograr horarios académicos eficientes y equitativos. Algunas características destacadas incluyen:

- *Optimización de recursos:* utilizan algoritmos avanzados para optimizar la asignación de recursos, como profesores, aulas y franjas horarias. Esto garantiza una distribución equitativa y eficiente de las asignaturas.
- *Adaptabilidad a restricciones específicas:* permiten la incorporación de restricciones específicas, como preferencias de profesores, requisitos de aulas y reglas institucionales. Esto asegura una planificación que se ajusta a las particularidades y necesidades únicas de la institución.

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

- *Interfaz intuitiva y personalizable*: suelen contar con interfaces intuitivas que facilitan la navegación y la personalización según las preferencias y políticas de la institución educativa. Esto mejora la experiencia del usuario y simplifica el proceso de programación.
- *Análisis predictivo*: algunas plataformas incluyen capacidades de análisis predictivo que permiten evaluar posibles escenarios y anticipar desafíos potenciales en la planificación, brindando una mayor capacidad de adaptación.

5.4.2. Inconvenientes de las soluciones comerciales

A pesar de las numerosas características ventajosas que ofrecen, las soluciones comerciales para la creación de horarios también presentan algunos aspectos negativos que deben ser considerados:

- **Costes iniciales y de mantenimiento**
La adquisición de soluciones comerciales a menudo conlleva costes significativos tanto en términos de licencias iniciales como de tarifas recurrentes de mantenimiento. Esto puede representar una carga financiera considerable para las instituciones educativas, especialmente para aquellas con presupuestos ajustados.
- **Curva de aprendizaje**
La implementación de nuevas soluciones puede requerir tiempo y esfuerzo para que el personal educativo se familiarice con la plataforma. La curva de aprendizaje puede generar resistencia y puede llevar algún tiempo antes de que los usuarios aprovechen al máximo todas las funcionalidades.
- **Dependencia de proveedores externos**
Al optar por soluciones comerciales, las instituciones pueden volverse dependientes de proveedores externos. Esto implica que cualquier problema técnico, cambio en las políticas del proveedor o discontinuidad del servicio puede afectar la operatividad de la institución.
- **Personalización limitada**
Aunque muchas soluciones comerciales ofrecen opciones de personalización, estas pueden ser limitadas en comparación con las necesidades específicas de algunas instituciones educativas. Las restricciones en la adaptabilidad pueden conducir a compromisos en términos de requisitos particulares.
- **Actualización y parches**
Las actualizaciones y parches frecuentes pueden generar interrupciones temporales en el servicio y requerir una atención constante para mantener

la compatibilidad y la seguridad. La implementación de estas actualizaciones puede ser un desafío logístico.

- **Problemas de integración**

Integrar la solución comercial con otros sistemas ya existentes en la institución puede ser un proceso complejo. La interoperabilidad puede convertirse en un desafío, especialmente si la solución no es compatible con los sistemas actuales.

- **Privacidad y seguridad**

La gestión de datos sensibles, como la información de los profesores y estudiantes, plantea preocupaciones sobre la privacidad y la seguridad. Las instituciones deben asegurarse de que la solución cumpla con los estándares de seguridad y protección de datos.

Es importante que las instituciones educativas realicen un análisis exhaustivo de estos aspectos negativos y los ponderen frente a los beneficios ofrecidos por las soluciones comerciales antes de tomar decisiones de implementación. La comprensión completa de estos inconvenientes permite una toma de decisiones informada y ayuda a mitigar posibles desafíos a lo largo del tiempo.

5.4.3. Plataforma “Untis”

Esta plataforma es un ejemplo existente en el mercado que proporciona soluciones para la planificación de horarios escolares y la gestión académica. *Untis* se utiliza en numerosas instituciones educativas y ofrece diversas funcionalidades, como la planificación automática de horarios, la asignación eficiente de recursos y la integración con otros sistemas académicos.

Es el líder en el sector de horarios y la planificación de clases. Tiene más de 45 empresas distribuidoras, más de 25.000 instituciones de enseñanza elaboran sus horarios con la ayuda de Untis y sus programas tienen la capacidad de ser traducidos en más de 35 idiomas.

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia



Figura 7. Inicio de la web Untis. Fte: [Web Untis](#)

La configuración visual de la página web se presenta de acuerdo con lo ilustrado en la Figura 7. En la parte izquierda, se destaca la opción para solicitar un presupuesto, proporcionando detalles sobre la oferta deseada. En la parte superior, se encuentran diversas opciones accesibles a través de la interfaz de la página. Además, en la esquina superior derecha, se distingue un rótulo con el nombre "WEBUNTIS", señalando la plataforma que alberga los horarios específicos de cada centro académico. Acceder a la plataforma individual de un centro en particular se logra introduciendo las credenciales correspondientes.

La Figura 8 detalla la página destinada a la búsqueda de un centro específico, donde, después de haber introducido el centro en cuestión, se procede a ingresar las credenciales correspondientes para acceder a la plataforma del centro académico.

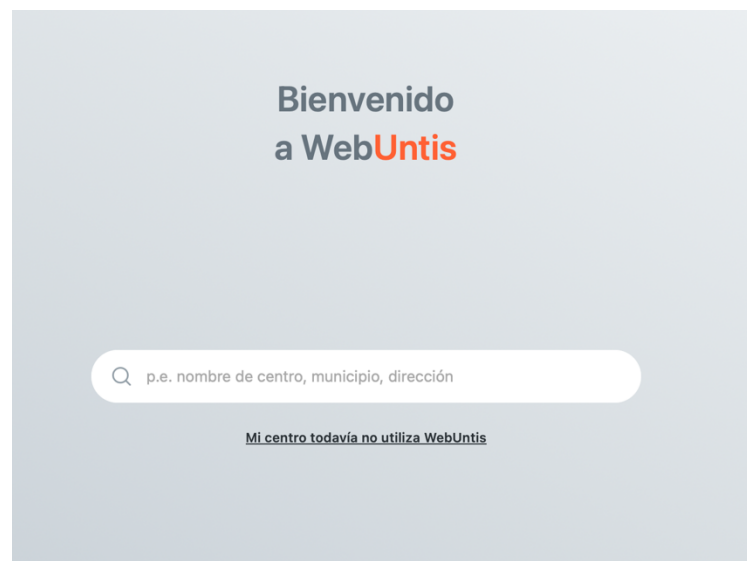


Figura 8. Plataforma WebUntis. Fte: [Web Untis](#)

Es importante resaltar que esta plataforma cuenta con múltiples módulos disponibles. Al acceder al desplegable denominado “PRODUCTOS” ubicado en la barra de herramientas superior, es posible explorar una variedad de módulos o soluciones distintas, adaptadas a los requisitos específicos necesarios para la creación de un horario. Dicho desplegable está ilustrado en la Figura 9.

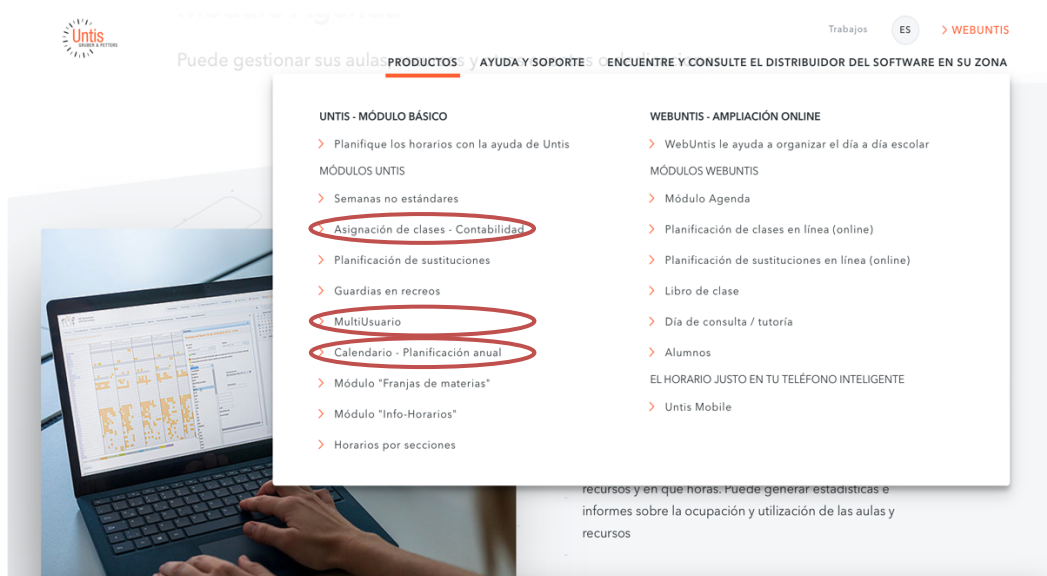


Figura 9. Productos que ofrece la plataforma UNTIS. Fte: [Web Untis](#)

En relación con este proyecto, los módulos siguientes representan algunas de las opciones disponibles que podrían configurar un horario de acuerdo con los requisitos específicos exigidos por este problema:

1) Asignación de clases – Contabilidad

Este módulo desempeña un papel fundamental en las etapas previas a la elaboración del horario, ya que se encarga de asignar las clases a los profesores considerando las características individuales de cada uno de ellos. Además, incluye una funcionalidad de contabilidad que permite comparar en tiempo real las horas asignadas con la jornada contractual de los profesores, asegurando así la conformidad con los acuerdos laborales y optimizando la distribución de recursos. Gracias a esta herramienta, es posible que se pudieran ver satisfechas muchas de las preferencias que pudiesen tener los distintos docentes del centro.

2) MultiUsuario

Los datos de Untis MultiUsuario no se almacenan en un archivo local, sino que se gestionan a través de una base de datos, como MS Access, MS SQL o MySQL. Esta configuración permite que múltiples usuarios accedan y trabajen con los datos de manera simultánea. Además, un sistema detallado de derechos de uso facilita la restricción y especificación del acceso a la información; por ejemplo, los empleados del secretariado pueden consultar los horarios, pero no modificarlos.

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

Esta metodología de planificación de horarios resulta especialmente beneficiosa al permitir que los profesores, quienes pueden cambiar sus preferencias año tras año, tengan la capacidad de modificar ellos mismos sus propios datos. Esto también facilitaría el trabajo de aquellos profesores que pudiesen incorporarse en años siguientes, ya que probablemente sus preferencias serían distintas. Esto facilita significativamente la labor del creador del horario, ya que descentraliza la gestión de las preferencias y reduce la carga administrativa involucrada en la actualización constante de la información. Esta autonomía no solo optimiza el proceso de creación del horario, sino que también asegura que las preferencias de los profesores sean consideradas y actualizadas de manera eficiente.

3) Calendario – Planificación anual

La principal característica de este módulo es la flexibilidad máxima que proporciona en la planificación de las clases. Este módulo ofrece la posibilidad de trabajar con horas anuales, permitiendo a los usuarios planificar estas horas libremente de acuerdo con sus necesidades y distribuir las a lo largo del curso escolar. La ubicación y la cantidad de estas horas pueden variar cada semana, brindando así flexibilidad en la planificación. Además, si se utiliza la función de automatización del módulo "Calendario - Planificación anual", este se encargará automáticamente de distribuir las horas anuales por todo el curso escolar, asegurando una gestión eficiente y coherente del tiempo asignado.

Esta solución, aunque a primera vista parece muy conveniente, puede requerir una preparación y un análisis previos muy exhaustivos. Es fundamental tener un conocimiento profundo y claro de la situación para poder planificar efectivamente todas las semanas del curso escolar de una vez y crear el horario en base a esta planificación. Por otra parte, es importante considerar que este enfoque podría no ser el más habitual en el horario de una institución educativa de secundaria, donde las necesidades y las circunstancias pueden variar significativamente a lo largo del año. Esto implica que, aunque la herramienta ofrece gran flexibilidad y potencial de optimización, su aplicación práctica debe ser cuidadosamente evaluada para asegurar que se ajusta a la dinámica y las necesidades específicas de la institución.

5.5. SISTEMAS MANUALES

En el ámbito de la planificación escolar, los sistemas manuales representan un enfoque arraigado y tradicional para la creación de horarios. Este método implica el diseño meticuloso y la organización de las actividades académicas sin depender de herramientas automatizadas o software especializado.

5.5.1. Características de los sistemas manuales

A continuación, se exploran en mayor profundidad las características más relevantes de estos sistemas:

- **Flexibilidad y control total**

Una de las principales fortalezas de los sistemas manuales radica en la flexibilidad y el control total que brindan a los planificadores. Este enfoque permite ajustar los horarios de manera precisa y personalizada, adaptándolos a las necesidades específicas de la institución educativa y las preferencias individuales de los profesores.

- **Bajo coste inicial**

Este enfoque no requiere la inversión en software especializado, lo que puede resultar en un costo inicial significativamente menor en comparación con las soluciones comerciales. En determinadas instancias, el coste asociado a la creación manual de horarios puede considerarse como potencialmente irrelevante. Este análisis se fundamenta en la premisa de que, si la persona encargada de la elaboración de los horarios cuenta con las competencias necesarias como parte de sus responsabilidades designadas, no se generaría un coste adicional asociado a esta labor. En tal escenario, la creación manual de horarios se percibe como una tarea intrínseca a las funciones asignadas, eliminando así la necesidad de incurrir en gastos adicionales específicos para esta labor.

- **Adaptabilidad a cambios inesperados**

La creación manual de horarios permite una rápida adaptación a cambios imprevistos, como modificaciones en la disponibilidad de profesores o eventos inesperados. Los planificadores pueden ajustar la programación de manera ágil sin depender de actualizaciones de software.

5.5.2. Inconvenientes de los sistemas manuales

Con el objetivo de realizar un análisis exhaustivo de esta alternativa, resulta imperativo considerar detenidamente aquellos aspectos negativos que puedan surgir:

- **Tiempo y esfuerzo**

Aunque ofrece flexibilidad, la creación manual de horarios consume considerable tiempo y esfuerzo, especialmente en instituciones con un elevado número de profesores, asignaturas y restricciones a considerar.

- **Mayor propensión a errores**

La intervención humana conlleva una mayor propensión a errores, desde conflictos en la asignación de recursos hasta problemas en la programación, lo que podría traducirse en una ejecución menos eficiente.

- **Dificultad en la gestión de preferencias**

Manejar las preferencias individuales de los profesores y las restricciones específicas puede resultar más complejo manualmente, lo que podría afectar la optimización del horario en términos de las preferencias de todos los involucrados.

- **Limitaciones en el análisis predictivo**
La capacidad de realizar análisis predictivos y escenarios "¿qué pasaría si?" es limitada en comparación con las soluciones automatizadas. Esto puede afectar la capacidad de anticipar y abordar posibles problemas.
- **Menos eficiente para grandes instituciones**
A medida que la institución crece en tamaño y complejidad, la planificación manual puede volverse impracticable y menos eficiente, ya que la complejidad de la programación aumenta considerablemente.

La elección entre sistemas manuales y automatizados requiere un análisis detenido, considerando la escala de la institución, los recursos disponibles y la importancia atribuida al control y la flexibilidad. Es crucial sopesar cuidadosamente estos aspectos antes de determinar el enfoque óptimo para la creación de horarios escolares.

5.6. SISTEMAS DE OPTIMIZACIÓN

Los sistemas de optimización se presentan como la última alternativa posible para la creación de horarios académicos. Esta solución se puede considerar que es avanzada y tecnológica para abordar la complejidad asociada que conlleva esta práctica.

5.6.1. Características de los sistemas de optimización

A continuación, se detallan las características más relevantes de estos sistemas, así como los aspectos negativos que deben tenerse en cuenta:

- **Automatización y eficiencia**
Los sistemas de optimización permiten la automatización de la creación de horarios, agilizando el proceso y mejorando la eficiencia en comparación con métodos manuales. Esta automatización se traduce en la capacidad de generar horarios optimizados de manera rápida y precisa.
- **Análisis predictivo y escenarios "¿Qué pasaría si?"**
Estos sistemas ofrecen capacidades avanzadas de análisis predictivo, permitiendo a los planificadores explorar escenarios hipotéticos y evaluar el impacto de posibles cambios antes de implementarlos.
- **Gestión de preferencias y restricciones**
Facilitan la gestión de preferencias y restricciones, asegurando la consideración óptima de las necesidades individuales de profesores, aulas y asignaturas. Esto contribuye a la elaboración de horarios que satisfacen tanto a la institución como a los involucrados.
- **Adaptabilidad a grandes instituciones**

Son altamente escalables y eficaces en instituciones grandes, donde la complejidad de la planificación aumenta. Estos sistemas pueden manejar grandes volúmenes de datos y restricciones de manera eficiente.

5.6.2. Inconvenientes de los sistemas de optimización

Asimismo, al igual que las dos anteriores alternativas, la opción de sistemas de optimización para la creación de horarios escolares también exhibe una serie de inconvenientes que deben ser meticulosamente considerados al tomar decisiones respecto a la elección de la alternativa:

- **Posibles costes iniciales**
La implementación de estos sistemas podría implicar un coste inicial, ya que podría ser necesario adquirir licencias de programas para desarrollar esta herramienta. No obstante, este coste difiere sustancialmente en comparación con la adquisición de una solución comercial, dado que este sistema sería desarrollado internamente por la propia institución o algún colaborador.
- **Curva de aprendizaje**
Existe una curva de aprendizaje asociada con la adopción de estos sistemas. La capacitación del personal para utilizar eficazmente el software y aprovechar al máximo sus capacidades puede requerir tiempo y recursos.
- **Posible resistencia al cambio**
La introducción de sistemas de optimización puede encontrarse con resistencia por parte del personal acostumbrado a métodos manuales. La aceptación y la adaptación pueden representar desafíos culturales en la institución.
- **Dependencia de datos precisos**
La eficacia de estos sistemas depende en gran medida de la disponibilidad y precisión de los datos de entrada. Errores en la información pueden afectar la calidad de los horarios generados.

5.7. ELECCIÓN FINAL DE LA ALTERNATIVA

Para determinar la alternativa más adecuada entre las opciones previamente expuestas, es imperativo llevar a cabo un análisis exhaustivo de sus respectivas capacidades y limitaciones. Como se indicó al inicio de este capítulo, se aplicará la técnica del Proceso Analítico Jerárquico (AHP).

Como herramienta para llevar a cabo este análisis, se ha determinado que **Excel** resulta ser la opción más adecuada. Esto se debe a que los cálculos necesarios para la aplicación del Proceso Analítico Jerárquico (AHP) se integran de manera eficiente con las funcionalidades proporcionadas por este programa. La versatilidad y capacidad de Excel

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

para manejar datos y realizar cálculos complejos lo posicionan como una herramienta idónea para llevar a cabo un análisis detallado y preciso.

En los próximos subapartados, se presentarán una serie de tablas del archivo Excel, detallando los cálculos y elementos más relevantes utilizados. Estas imágenes proporcionarán una visión clara y detallada del análisis llevado a cabo mediante esta herramienta. Además, es relevante destacar que las tablas que van a aparecer en los siguientes subapartados tienen su propia nomenclatura la cual se describe en la Tabla 10.

Tabla 10. Nomenclatura utilizada en el método AHP. Fte: Elaboración propia

Nomenclatura	Descripción
C1	Eficiencia en la asignación
C2	Flexibilidad y Adaptabilidad
C3	Optimización de recursos
C4	Facilidad en la implementación
C5	Coste total
A1	Alternativa 1 (Soluciones comerciales)
A2	Alternativa 2 (Sistemas manuales)
A3	Alternativa 3 (Sistemas de optimización)

5.7.1. Jerarquización de los cinco criterios seleccionados

Como se detalló en el apartado 5.3.1, el primer paso para aplicar esta técnica consiste en la jerarquización de cada uno de los criterios. Este proceso implica tener en cuenta la escala de preferencias de la Tabla 9, y en función de esta, construir la matriz de comparaciones de los criterios.

Tabla 11. Matriz de comparación del criterio C1. Fte: Elaboración propia

C1	A1	A2	A3
A1	1	9	3
A2	0,11	1	0,14
A3	0,33	7	1
SUMA	1,44	17	4,14

SELECCIÓN DE ALTERNATIVAS

Tabla 12. Matriz de comparación del criterio C2. Fte: Elaboración propia

C2	A1	A2	A3
A1	1	8	1
A2	0,13	1	0,14
A3	1,00	7	1
SUMA	2,13	16	2,14

Tabla 13. Matriz de comparación del criterio C3. Fte: Elaboración propia

C3	A1	A2	A3
A1	1	8	2
A2	0,13	1	0,13
A3	0,50	8	1
SUMA	1,63	17	3,13

Tabla 14. Matriz de comparación del criterio C4. Fte: Elaboración propia

C4	A1	A2	A3
A1	1	0,14	0,17
A2	7	1	2
A3	6	0,50	1
SUMA	14	1,64	3,17

Tabla 15. Matriz de comparación del criterio C5. Fte: Elaboración propia

C5	A1	A2	A3
A1	1	0,11	0,11
A2	9	1	1
A3	9	1,00	1
SUMA	19	2,11	2,11

Una vez creadas todas las matrices para cada criterio, se procede a construir la matriz de comparación de criterios, de la cual se derivarán las prioridades de cada criterio. La Tabla 16 representa esta matriz, cuyo cálculo sigue el mismo procedimiento que las tablas anteriores, pero teniendo en cuenta únicamente los criterios seleccionados para este estudio.

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

Tabla 16. Matriz de comparación de los criterios. Fte: Elaboración propia

Criterios	C1	C2	C3	C4	C5
C1	1	3	2	5	1
C2	0,33	1	4	6	1
C3	0,5	0,25	1	5	0,2
C4	0,2	0,17	0,2	1	0,11
C5	1	5	7	9	1
SUMA	3,03	9,42	14,2	26	3,31

5.7.2. Asignación de ponderaciones

Conforme se detallada en el apartado 5.3.2, este fase se conoce como sintetización y consta de tres pasos hasta la obtención del vector promedio de cada criterio para cada alternativa, así como el vector promedio de la prioridad de criterios.

El primer paso, como se mencionó previamente, implica la suma de cada columna de la matriz de comparación. Posteriormente, cada elemento de la matriz se divide por el total de su columna, generando así una matriz normalizada.

La Tabla 17, Tabla 18, Tabla 19, Tabla 20, Tabla 21 y Tabla 22 muestran cómo quedan las matrices normalizadas en comparación con las tablas anteriores.

Tabla 17. Matriz normalizada del criterio C1. Fte: Elaboración propia

Matriz normalizada del C1			
C1	A1	A2	A3
A1	0,69	0,53	0,72
A2	0,08	0,06	0,03
A3	0,23	0,41	0,24

Tabla 18. Matriz normalizada del criterio C2. Fte: Elaboración propia

Matriz normalizada del C2			
C2	A1	A2	A3
A1	0,47	0,50	0,47
A2	0,06	0,06	0,07
A3	0,47	0,44	0,47

Tabla 19. Matriz normalizada del criterio C3. Fte: Elaboración propia

Matriz normalizada del C3			
C3	A1	A2	A3
A1	0,62	0,47	0,64
A2	0,08	0,06	0,04
A3	0,31	0,47	0,32

Tabla 20. Matriz normalizada del criterio C4. Fte: Elaboración propia

Matriz normalizada del C4			
C4	A1	A2	A3
A1	0,07	0,09	0,05
A2	0,50	0,61	0,63
A3	0,43	0,30	0,32

Tabla 21. Matriz normalizada del criterio C5. Fte: Elaboración propia

Matriz normalizada del C5			
C5	A1	A2	A3
A1	0,05	0,05	0,05
A2	0,47	0,47	0,47
A3	0,47	0,47	0,47

Tabla 22. Matriz normalizada de los criterios. Fte: Elaboración propia

Matriz normalizada de criterios					
Criterios	C1	C2	C3	C4	C5
C1	0,33	0,32	0,14	0,19	0,30
C2	0,11	0,11	0,28	0,23	0,30
C3	0,16	0,03	0,07	0,19	0,06
C4	0,07	0,02	0,01	0,04	0,03
C5	0,33	0,53	0,49	0,35	0,30

Una vez obtenidas estas matrices, es posible calcular los vectores promedio. Se toma cada fila de cada una de las matrices normalizadas y se calcula su valor promedio. De esta manera, se obtienen las ponderaciones correspondientes. A continuación, se representan dichos vectores, donde VP y Cs significan respectivamente Vector Promedio y Criterios.

La Tabla 23 aglutina los vectores promedio de cada criterio para cada alternativa y la Tabla 24 representa el vector promedio de prioridad de los criterios.

Tabla 23. Vectores promedio de cada criterio respecto de cada alternativa. Fte: Elaboración propia

	VP de C1	VP de C2	VP de C3	VP de C4	VP de C5
A1	0,72	0,47	0,64	0,05	0,05
A2	0,03	0,07	0,04	0,63	0,47
A3	0,24	0,47	0,32	0,32	0,47

Tabla 24. Vector promedio de prioridad de los criterios. Fte: Elaboración propia

	VP de Cs
C1	0,24
C2	0,23
C3	0,09
C4	0,03
C5	0,42

5.7.3. Matriz de prioridades y prioridad global

Por último, tal y como se indica en el apartado 5.3.3, se tiene que construir la matriz de prioridades de cada una de las alternativas. Esta tarea consiste simplemente en construir una matriz donde las columnas estén formadas por los vectores promedio de cada criterio para cada alternativa.

Esta matriz se multiplica por el vector promedio de prioridad de los criterios, con el fin de obtener una prioridad global, que será la que indique cuál es la mejor alternativa a seleccionar según el estudio realizado.

Tabla 25. Cálculo de la prioridad global de las alternativas. Fte: Elaboración propia

	C1	C2	C3	C4	C5	
A1	0,65	0,48	0,58	0,07	0,05	0,35
A2	0,06	0,06	0,06	0,58	0,47	0,24
A3	0,29	0,46	0,37	0,35	0,47	0,41
PONDERACIÓN	0,26	0,21	0,10	0,03	0,40	

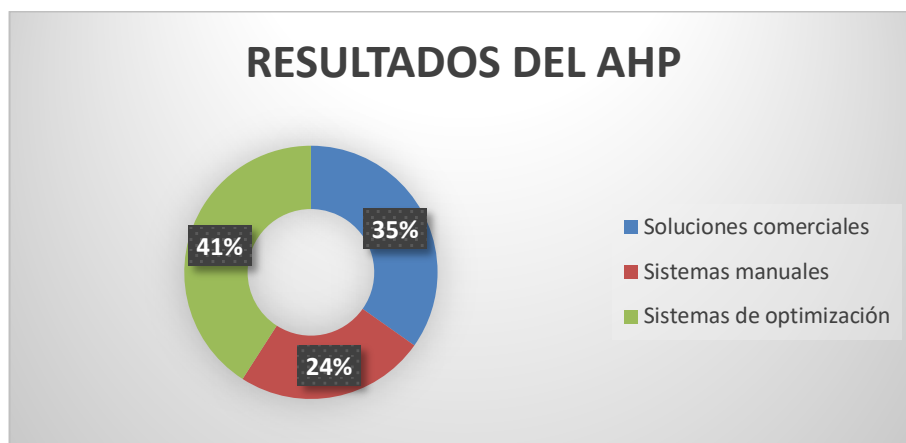


Gráfico 1. Resultados finales del AHP. Fte: Elaboración propia

Por lo tanto, según la aplicación de la técnica de Proceso Analítico Jerárquico (AHP), la mejor opción para la creación de un horario académico es la de realizarlo mediante un **sistema de optimización**.

5.8. CONCLUSIONES

Este extenso capítulo ha sido dedicado a realizar un análisis detallado de las diversas alternativas disponibles para la elaboración de un horario académico. Para ello, se inició introduciendo los criterios predominantes que deben considerarse en la selección de la alternativa. Este primer paso resultaba esencial para llevar a cabo un análisis correcto y exhaustivo de las diferentes opciones seleccionadas. El segundo paso consistió en presentar el método utilizado para el análisis de las alternativas, que, como se comentó anteriormente, fue la técnica del Proceso Analítico Jerárquico (AHP).

Para llevar a cabo este método de manera efectiva, el primer paso fue describir cada una de las tres alternativas: soluciones comerciales, sistemas manuales y sistemas de optimización. Se detallaron las características más relevantes de cada una, así como los aspectos negativos e inconvenientes más destacados a tener en cuenta al aplicar la técnica AHP. El último apartado, por consiguiente, se centró en la aplicación de este método al caso en cuestión. Gracias a este análisis, se pudo llegar a la conclusión de que la opción más recomendable para la creación de este horario escolar es mediante la elaboración de una herramienta utilizando un sistema de optimización.

En el próximo capítulo, después de haber seleccionado la metodología para la elaboración de la herramienta, se llevará a cabo la formulación de un modelo matemático mediante programación matemática. El objetivo principal de este paso será construir un modelo que, posteriormente, posibilite el correcto funcionamiento de la herramienta. Se tendrán en cuenta todas las condiciones y restricciones pertinentes, asegurando así la eficacia y precisión en la creación del horario académico. Este proceso implica la traducción de las reglas y requerimientos del sistema en ecuaciones matemáticas, lo que permitirá automatizar y optimizar la generación del horario, considerando la complejidad y las diversas variables involucradas en el proceso educativo. Este enfoque matemático proporcionará una base sólida para la implementación práctica de la herramienta, asegurando su eficiencia y capacidad para abordar las complejidades inherentes a la programación de horarios académicos.

6. MODELADO

6.1. INTRODUCCIÓN

En el presente capítulo, se llevará a cabo el modelado de la herramienta, atendiendo a las condiciones y requerimientos inherentes a su implementación.

En primer lugar, se abordará el estudio de los índices que son necesarios para una correcta creación de la herramienta, detallando en qué consistirá cada uno de ellos (6.2).

A continuación, se procederá al desglose detallado del diseño intrínseco de la presente herramienta. En este sentido, se trazarán todos los pasos necesarios. En primer lugar, se describirán todos los parámetros que compondrán los modelos, además de la creación de las variables de decisión que se necesiten para llevar a cabo una correcta ejecución de los dos modelos existentes (6.3).

Una vez descritos estos pasos, se procederá a la descripción de las funciones objetivo que marcará el devenir de la función de esta herramienta (6.4).

Asimismo, se llevará a cabo la transformación de todas aquellas condiciones y necesidades inherentes a la creación del horario, expresándolas en términos de diseño específicos para una herramienta de estas características (6.5).

En tercer y último lugar, se llevará a cabo la creación del dominio de todas los índices y todas las variables que habrán sido creados para el modelo en cuestión (6.6).

Una vez llegado este punto, los dos modelos tendrán la consistencia necesaria para poder ser implementado dentro de la herramienta de creación del horario. Cabe destacar que en primer lugar se van a asignar todas aquellas asignaturas específicas y optativas, con el objetivo de facilitar el trabajo de la herramienta, ya que estas asignaturas al impartirse al mismo tiempo son las que pueden crear conflictos dentro de la programación.

6.2. ÍNDICES

La diferenciación de los modelos se hará mediante la numeración de cada uno de ellos. El modelo 1 será el que asigne las asignaturas específicas y optativas, y el modelo 2 será el que asigne las asignaturas generales.

6.2.1. Modelo 1

En este caso, el modelo matemático que se ha escogido diseñar es el basado en la programación matemática con variables del tipo binario.

A continuación, tal y como se ha introducido en el anterior apartado, se empieza con una descripción de cada uno de los índices que componen este diseño matemático.

MODELADO

El inicio del proceso para la creación de un modelo matemático basado en programación matemática implica la descripción de todos los índices necesarios para llevar a cabo dicho diseño.

$I = \text{Franja horaria}$

- *Franja horaria*: indica la hora en que se puede impartir una asignatura a un determinado grupo de un curso.

$E = \text{Bloque específicas}$

- *Bloque específicas*: indica el bloque de asignaturas específicas correspondiente a un determinado grupo de un curso.

$O = \text{Bloque optativas}$

- *Bloque optativas*: indica el bloque de asignaturas optativas correspondiente a un determinado grupo de un curso.

$K = \text{Profesor}$

- *Profesor*: indica la persona que puede impartir las diferentes asignaturas.

$G = \text{Grupo}$

- *Grupo*: indica el grupo de curso correspondiente dentro de todos los posibles grupos existentes.

$D = \text{Día de la semana}$

- *Día*: indica el día de la semana.

Tabla 26. Descripción de índices del modelo matemático 1. Fte: Elaboración propia

Índice	Descripción
i	<i>Franja horaria</i>
e	<i>Bloque específicas</i>
o	<i>Bloque optativas</i>
k	<i>Profesor</i>
d	<i>Día de la semana</i>
g	<i>Grupo</i>

En la Tabla 26 se presenta la disposición de los índices que serán empleados en el diseño de la herramienta para el modelo 1.

6.2.2. Modelo 2

Respecto al modelo 2, los índices que conforman este modelo ya están definidos en el anterior, a excepción de uno de ellos.

$$J = \text{Asignatura general}$$

- *Asignatura general*: indica la asignatura general correspondiente a un determinado grupo de un curso.

Tabla 27. Descripción de índices del modelo matemático 2. Fte: Elaboración propia

Índice	Descripción
<i>i</i>	<i>Franja horaria</i>
<i>j</i>	<i>Asignatura general</i>
<i>k</i>	<i>Profesor</i>
<i>d</i>	<i>Día de la semana</i>
<i>g</i>	<i>Grupo</i>

Por lo tanto, en la Tabla 27 se presenta la disposición de los índices que completan el modelo 2.

6.3. PARÁMETROS Y VARIABLES DE DECISIÓN

Los parámetros se definen en base a las políticas y características de la escuela y el programa con respecto a los cursos, los profesores y la capacidad instalada.

6.3.1. Modelo 1

A continuación, quedan definidos los datos del modelo 1:

$$HSe_{g,e}$$

[Número de franjas horarias (horas semanales) que se deben impartir semanalmente para el grupo *g* del bloque de específicas *e*]

$$HSO_{g,o}$$

[Número de franjas horarias (horas semanales) que se deben impartir semanalmente para el grupo *g* del bloque de optativas *o*]

$$CPe_{k,e}$$

[Parámetro binario que toma valor 1 en caso de que el profesor *k* pueda impartir el bloque de específicas *e*, toma valor 0 si no está capacitado]

MODELADO

$$CP_{o,k,o}$$

[Parámetro binario que toma valor 1 en caso de que el profesor k pueda impartir el bloque de optativas o , toma valor 0 si no está capacitado]

$$CS_{k,i,d}$$

[Satisfacción del profesor k en dar clase en la franja horaria i del día d]

Por lo que respecta a las variables de decisión, como ya se ha comentado en el apartado anterior, son del tipo binario, por lo tanto, solo podrán tomar como valor 0 ó 1.

Para la resolución de este problema en concreto, se ha considerado necesario definir las siguientes variables de decisión:

$$Z_{g,i,e,k,d}$$

[Variable binaria que toma valor 1 en caso de que para el grupo g en la franja horaria i se asigna el bloque de específicas e impartida por el profesor k en el día d , y en caso contrario toma valor 0]

$$T_{g,i,o,k,d}$$

[Variable binaria que toma valor 1 en caso de que para el grupo g en la franja horaria i se asigna el bloque de optativas o impartida por el profesor k en el día d , y en caso contrario toma valor 0]

$$V_{g,e,k}$$

[Variable binaria que toma valor 1 en caso de que para el grupo g se imparte el bloque de específicas e por el profesor k , y en caso contrario toma valor 0]

$$W_{g,o,k}$$

[Variable binaria que toma valor 1 en caso de que para el grupo g se imparte el bloque de optativas o por el profesor k , y en caso contrario toma valor 0]

6.3.2. Modelo 2

De este modelo hay que destacar los siguientes parámetros:

$$HS_{g,j}$$

[Número de franjas horarias (horas semanales) que se deben impartir semanalmente para el grupo g de la asignatura j]

$$CP_{k,j}$$

[Parámetro binario que toma valor 1 en caso de que el profesor k pueda impartir la asignatura j , toma valor 0 si no está capacitado]

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

$$CS_{k,i,d}$$

[Satisfacción del profesor k en dar clase en la franja horaria i del día d]

Las variables de decisión en el caso de este modelo son las siguientes:

$$X_{g,i,j,k,d}$$

[Variable binaria que toma valor 1 en caso de que para el grupo g en la franja horaria i se asigna la asignatura j impartida por el profesor k en el día d , y en caso contrario toma valor 0]

$$Y_{g,j,k}$$

[Variable binaria que toma valor 1 en caso de que para el grupo g se imparte la asignatura j por el profesor k , y en caso contrario toma valor 0]

En este caso particular, dado que se trata de una escuela de secundaria, se establece que cada profesor es el único responsable de impartir una asignatura, es decir, no existe la posibilidad de que una asignatura sea enseñada por más de un profesor. Exceptuando las asignaturas específicas y optativas, conformadas en bloques, que sí podrán ser impartidas por uno o más profesores, ya que, al tratarse de un bloque de asignaturas, pueden existir en él más de una asignatura diferente, y, por tanto, impartida por un profesor distinto.

A pesar de esta restricción, se ha decidido definir las variables de decisión de este modelo de la misma forma que en el modelo 1, con el propósito de ampliar las características de la herramienta. Esta elección anticipa posibles aplicaciones futuras en otros centros académicos que podrían presentar mayores complejidades en la asignación de profesores a asignaturas.

6.4. FUNCIÓN OBJETIVO

Después de haber definido los índices, las variables y los datos de ambos modelos, se puede ya describir matemáticamente cuál es la función objetivo de este proyecto. Esta será la suma de ambas funciones objetivos que se definirán a continuación.

6.4.1. Modelo 1

Esta primera función es la que se encuentra representada en la siguiente ecuación:

$$\begin{aligned} \text{Máx} \left(\sum_{g \in G} \sum_{i \in I} \sum_{e \in E} \sum_{k \in K} \sum_{d \in D} CS_{k,i,d} \times Z_{g,i,e,k,d} \right. \\ \left. + \sum_{g \in G} \sum_{i \in I} \sum_{o \in O} \sum_{k \in K} \sum_{d \in D} CS_{k,i,d} \times T_{g,i,o,k,d} \right) \end{aligned} \quad (1)$$

Como se evidencia en la ecuación (1), el objetivo planteado consistirá en la maximización del nivel estimado de satisfacción de los profesores para la asignación de las asignaturas específicas y optativas.

En este contexto, la función objetivo también estará condicionada por si la asignatura en cuestión ha sido asignada al profesor correspondiente en el día adecuado y dentro de la franja horaria designada.

6.4.2. Modelo 2

La función objetivo del modelo 2 se define de la misma forma que la anterior cambiando la variable correspondiente a las asignaturas generales:

$$\text{Máx} \left(\sum_{g \in G} \sum_{i \in I} \sum_{e \in E} \sum_{k \in K} \sum_{d \in D} CS_{k,i,d} \times X_{g,i,j,k,d} \right) \quad (2)$$

Tal y como se puede observar, la ecuación (2) plantea la maximización del nivel estimado de satisfacción de los profesores para la asignación de las asignaturas generales.

Por tanto, la suma de las ecuaciones (1) y (2) representará la función objetivo global que proporciona la herramienta.

6.5. RESTRICCIONES

6.5.1. Modelo 1

En lo que respecta a las restricciones del modelo 1, es posible distinguir los siguientes tipos:

$$\sum_{k \in K} V_{g,e,k} \geq 1 \quad \forall g, \forall e \quad (3)$$

La ecuación (3) indica que uno o más de un profesor k puede impartir el bloque de específicas e al grupo g .

$$\sum_{k \in K} W_{g,o,k} \geq 1 \quad \forall g, \forall o \quad (4)$$

La ecuación (4) indica que uno o más de un profesor k puede impartir el bloque de optativas o al grupo g .

$$CPe_{k,e} = 0 \rightarrow V_{g,e,k} = 0 \quad \forall g, \forall e, \forall k \quad (5)$$

La ecuación (5) limita para el grupo g la asignación de bloque de específicas e a los profesores k capacitados.

$$CPO_{k,o} = 0 \rightarrow W_{g,o,k} = 0 \quad \forall g, \forall o, \forall k \quad (6)$$

La ecuación (6) limita para el grupo g la asignación de bloque de optativas o a los profesores k capacitados.

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

$$\sum_{i \in I} \sum_{d \in D} Z_{g,i,e,k,d} \leq HSe_{g,e} \times V_{g,e,k} \quad (7)$$

La ecuación (7) requiere que en caso de que un profesor k esté asignado al bloque de específicas e del grupo g en cualquier momento, es el responsable.

$$\sum_{i \in I} \sum_{d \in D} T_{g,i,o,k,d} \leq HSo_{g,o} \times W_{g,o,k} \quad (8)$$

La ecuación (8) requiere que en caso de que un profesor k esté asignado al bloque de optativas o del grupo g en cualquier momento, es el responsable.

$$\sum_{i \in I} \sum_{k \in K} Z_{g,i,e,k,d} \leq 1 \quad \forall g, \forall e, \forall d \quad (9)$$

La ecuación (9) limita que para un grupo g el bloque de específicas e sea impartido como máximo una vez en un mismo día d .

$$\sum_{i \in I} \sum_{k \in K} T_{g,i,o,k,d} \leq 1 \quad \forall g, \forall o, \forall d \quad (10)$$

La ecuación (10) limita que para un grupo g el bloque de optativas o sea impartido como máximo una vez en un mismo día d .

$$\sum_{i \in I} \sum_{k \in K} \sum_{d \in D} Z_{g,i,e,k,d} = HS_{g,e} \quad \forall g, \forall e \quad (11)$$

La ecuación (11) restringe que para el grupo g todas las horas semanales de los bloques de específicas e deben impartirse.

$$\sum_{i \in I} \sum_{k \in K} \sum_{d \in D} T_{g,i,o,k,d} = HS_{g,o} \quad \forall g, \forall o \quad (12)$$

La ecuación (12) restringe que para el grupo g todas las horas semanales de los bloques de optativas o deben impartirse.

$$\left(\sum_{g \in G} \sum_{e \in E} Z_{g,i,e,k,d} + \sum_{g \in G} \sum_{o \in O} T_{g,i,o,k,d} \right) \leq 1 \quad \forall i, \forall k, \forall d \quad (13)$$

La ecuación (13) limita que un profesor k solo puede impartir una específica e o una optativa o en una franja horaria i de un día d .

$$\left(\sum_{e \in E} \sum_{k \in K} Z_{g,i,e,k,d} + \sum_{o \in O} \sum_{k \in K} T_{g,i,o,k,d} \right) \leq 1 \quad \forall g, \forall i, \forall d \quad (14)$$

La ecuación (14) limita que un grupo g solo puede tener un bloque de específicas e o un bloque de optativas o en una franja horaria i de un día d .

$$T_{g,i,o,k,d} \leq T_{g,1,o,k,d} + T_{g,7,o,k,d} \quad \forall g, \forall o, \forall k, \forall d \quad (15)$$

La ecuación (15) limita que cualquier bloque de optativas o debe asignarse en la primera o en la última franja horaria i .

Este bloque de restricciones forma todas aquellas condiciones que deben estar implementadas dentro del modelo 1. De esta forma estarían cumpliéndose todos los requisitos demandados.

6.5.2. Modelo 2

Por su parte, el modelo 2 consta de las siguientes restricciones:

$$\sum_{k \in K} Y_{g,j,k} = 1 \quad \forall g, \forall j \quad (16)$$

La ecuación (16) restringe que solamente un profesor único k puede impartir la asignatura j al grupo g .

$$Y_{g,j,k} = 0 \quad \forall g, \forall j, \forall k \quad (17)$$

La ecuación (17) limita para el grupo g la asignación de asignaturas generales j a los profesores k capacitados.

$$\sum_{i \in I} \sum_{d \in D} X_{g,i,j,k,d} = HS_{g,j} \times Y_{g,j,k} \quad (18)$$

La ecuación (18) requiere que en caso de que un profesor k esté asignado a la asignatura j del grupo g en cualquier momento, es el responsable.

$$\sum_{i \in I} \sum_{k \in K} X_{g,i,j,k,d} \leq 1 \quad \forall g, \forall j, \forall d \quad (19)$$

La ecuación (19) limita que para un grupo g la asignatura j sea impartida como máximo una vez en un mismo día d .

$$\sum_{i \in I} \sum_{k \in K} \sum_{d \in D} X_{g,i,j,k,d} = HS_{g,j} \quad \forall g, \forall j \quad (20)$$

La ecuación (20) restringe que para el grupo g todas las horas semanales de las asignaturas generales j deben impartirse.

$$\sum_{j \in J} \sum_{k \in K} X_{g,i,j,k,d} \leq 1 \quad \forall g, \forall i, \forall d \quad (21)$$

La ecuación (21) restringe que para el grupo g solo puede haber asignada una asignatura j en una misma franja horaria i de un mismo día d .

$$\sum_{g \in G} \sum_{j \in J} X_{g,i,j,k,d} \leq 1 \quad \forall i, \forall k, \forall d \quad (22)$$

La ecuación (22) restringe que el profesor k solo puede impartir una asignatura general j en una misma franja horaria i de un mismo día d .

$$\begin{aligned} &((i, d) \in ocupacion_grupos(g) \cup (i, d) \\ &\in ocupacion_profesores(k)) \rightarrow X_{g,i,j,k,d} = 0 \end{aligned} \quad (23)$$

La ecuación (23) limita la asignación de ejecutado por el modelo 2 a aquellas franjas horarias que estén libres tanto en el horario creado para los grupos como para los profesores. Para ello son creadas las dos variables que aparecen en esa ecuación, cuyo código quedará reflejado en los Anexos del trabajo.

Con este pequeño bloque de restricciones están controladas todas las condiciones para que la asignación de las generales sea la correcta, y, por tanto, para que el modelo 2 funcione de la forma adecuada.

6.6. DOMINIO DE ÍNDICES Y VARIABLES DE DECISIÓN

Cabe destacar en este punto los dominios que van a tener tanto los diferentes índices como las variables de decisión para el caso concreto de estudio.

Por ello, en primer lugar, se describe en la Tabla 28 donde quedarían definidos los dominios de todos los índices:

Tabla 28. Dominio de los índices del modelo matemático completo. Fte: Elaboración propia

Índice	Descripción	Dominio
i	<i>Franja horaria</i>	$i=1,2,\dots,7$
j	<i>Asignatura general</i>	$j=1,2,\dots,75$
e	<i>Bloque específicas</i>	$e=1,2,\dots,23$
o	<i>Bloque optativas</i>	$o=1,2,\dots,12$
k	<i>Profesor</i>	$k=1,2,\dots,39$
d	<i>Día de la semana</i>	$d=1,2,\dots,5$

Asimismo, en caso de que hubiese cualquier modificación en cualquiera de los índices, esta debería verse reflejada en el índice correspondiente.

Por lo que respecta al dominio de las variables de decisión, al tratarse de variables binarias es sencillo describir su dominio. Se detallan mediante las siguientes ecuaciones:

$$X_{g,i,j,k,d} = \{0,1\} \quad \forall g, \forall i, \forall j, \forall k, \forall d \quad (24)$$

$$Y_{g,j,k} = \{0,1\} \quad \forall g, \forall j, \forall k \quad (25)$$

$$Z_{g,i,e,k,d} = \{0,1\} \quad \forall g, \forall i, \forall e, \forall k, \forall d \quad (26)$$

$$T_{g,i,o,k,d} = \{0,1\} \quad \forall g, \forall i, \forall o, \forall k, \forall d \quad (27)$$

$$V_{g,e,k} = \{0,1\} \quad \forall g, \forall e, \forall k \quad (28)$$

$$W_{g,o,k} = \{0,1\} \quad \forall g, \forall o, \forall k \quad (29)$$

A parte de estos dominios, es necesario representar una matriz que pondere los niveles de satisfacción de los docentes. Esta ponderación se divide en cinco escalas que se representan en la Tabla 29:

Tabla 29. Escala de la matriz CS. Fte: Elaboración propia

Nivel	Criterio
<i>10</i>	<i>Muy deseado</i>
<i>5</i>	<i>Deseado</i>
<i>0</i>	<i>Sin Preferencia</i>
<i>-5</i>	<i>Poco deseado</i>
<i>-10</i>	<i>Altamente no deseado</i>

Como bien se puede apreciar el 10 representaría el nivel máximo que puede tener un profesor respecto a una franja horaria de un día en concreto, y el -10 el nivel mínimo que puede tener un profesor respecto a un franja de un día en concreto. Entre este máximo y este mínimo cabría destacar tres niveles intermedios de satisfacción.

6.7. CONCLUSIONES

Este sexto capítulo se ha dedicado exclusivamente al modelado para la creación de la herramienta, con el fin de que queden descritas todas las condiciones que vienen impuestas por parte de la escuela. Se ha querido dividir todo el capítulo diferenciando los elementos que conforman los dos modelos, para que quede constancia de la importancia y relevancia que tienen cada uno de ellos. Esto se debe a que el mal funcionamiento de un índice, parámetro, variable o restricción puede entorpecer de una forma muy significativa el desarrollo de la herramienta en cuestión.

Al haberse definido o separado el problema en dos modelos, se ha considerado que la forma de redactar, para que quedase claro, ha sido la opción escogida. Dentro de cada uno de los subcapítulos se ha diferenciado el modelo 1 y el modelo 2. De esta forma no dejamos de lado la globalidad del problema, ni tampoco juntamos información de un modelo en otro.

Se puede considerar que este capítulo está dividido en dos grandes partes: la creación de elementos y la transformación de las condiciones.

En la primera parte, se han creado todos estos elementos necesarios para la creación de los modelos de programación matemática operativos, que tuvieran consistencia y efectividad. En la segunda parte, se han transformado las restricciones impuestas por parte de la escuela en términos de enseñanza, capacidad y asignación en ecuaciones matemáticas que han permitido que estas condiciones y requisitos se cumplan, siempre sin dejar de lado el objetivo por el que se ha decidido crear este modelo global (la función objetivo).

En el siguiente capítulo, se tratarán las características que va a tener esta herramienta. Para ello, se desglosará el análisis y la elección de diversos aspectos: el lenguaje de programación, el Solver y el programa de importación y exportación de datos a utilizar. Se realizarán los diferentes análisis de las diversas opciones que existen en el mercado, y aquellas que se adecúen mejor a las características del problema serán las escogidas. De esta forma quedará elaborado un software propio mediante el cual la herramienta será capaz de crear y planificar el horario demandado.

7. ANÁLISIS Y ELECCIÓN DE LAS CARACTERÍSTICAS DE LA SOLUCIÓN

7.1. INTRODUCCIÓN

Como se ha presentado en el capítulo anterior, se ha definido el modelo matemático correspondiente que será capaz de dar forma a la herramienta de creación del horario académico. Este modelo incorpora las condiciones y restricciones, tanto duras como blandas, que son esenciales tener en cuenta. Además, incluye el objetivo principal que se buscará minimizar dentro de la propia herramienta. Este enfoque matemático proporcionará una base sólida para el desarrollo de la herramienta, permitiendo la consideración precisa de todos los factores y requisitos necesarios para la elaboración de un horario académico óptimo. La inclusión de condiciones duras asegura el cumplimiento de los requisitos fundamentales, mientras que las restricciones blandas ofrecen flexibilidad y adaptabilidad para mejorar la calidad del horario. Este modelo matemático sienta las bases para la implementación práctica y funcional de la herramienta.

En el presente capítulo, se llevará a cabo un análisis exhaustivo de los diversos programas informáticos disponibles en el mercado. El propósito es seleccionar la solución más idónea que permita establecer el sistema informático integral para abarcar todas las partes de este proyecto. Este proceso implica la evaluación detallada de las características y funcionalidades de varias opciones de software, con el fin de determinar cuál se adapta de manera eficiente a los requisitos del modelo matemático previamente definido. Se considerarán aspectos cruciales como la capacidad para la programación entera, la flexibilidad para incorporar restricciones y condiciones específicas del horario académico, así como la eficacia en términos de optimización. La elección del software adecuado es esencial para la implementación exitosa de la herramienta, garantizando su funcionalidad y rendimiento óptimos en la creación del horario académico.

En la primera sección de este capítulo, se llevará a cabo el análisis y elección de los diversos elementos que formarán parte del software. Este análisis abarcará la evaluación de los lenguajes de programación más adecuados para la implementación del modelo matemático, considerando su compatibilidad con la lógica y las operaciones requeridas (7.2). Además, se examinarán los solvers existentes en el mercado y las herramientas y programas que posibilitan la transferencia fluida de datos entre diferentes plataformas (7.3 y 7.4).

En la segunda y última sección, se realizará la elección del software teniendo en cuenta tanto el lenguaje de programación como el programa de datos que previamente se han escogido (7.5). Este elección deberá garantizar una sinergia efectiva entre el lenguaje de programación, los programas de exportación e importación de datos, y el modelo

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

matemático, contribuyendo así al éxito de la herramienta de creación de horarios académicos.

7.2. ANÁLISIS Y ELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN

En esta sección, se abordarán los lenguajes de programación más relevantes en la actualidad, entre los cuales se incluyen: Python, Java y C++. Se llevará a cabo un análisis exhaustivo de las características clave de cada uno de estos lenguajes, con el objetivo de evaluar su idoneidad para el diseño de la herramienta que se pretende desarrollar. Se examinarán aspectos como la facilidad de sintaxis, la versatilidad en la manipulación de datos, el soporte para operaciones matemáticas y la capacidad de integración con otros componentes del sistema.

Este análisis permitirá tomar una decisión fundamentada sobre el lenguaje de programación más adecuado para implementar el modelo matemático y lograr una ejecución eficiente y precisa de la herramienta de creación de horarios académicos.

7.2.1. Python

Python es un lenguaje de programación interpretado, de alto nivel y de propósito general, que se destaca por su sintaxis clara y legible. A continuación, se describen algunas de las características más importantes de este lenguaje:

- **Legibilidad y sencillez**
Python se diseñó para ser fácil de leer y escribir. Su sintaxis simple y clara permite a los programadores expresar conceptos en menos líneas de código que en otros lenguajes, lo que facilita la comprensión y mantenimiento del código.
- **Multiparadigma**
Admite múltiples paradigmas de programación, incluyendo programación orientada a objetos, programación imperativa y programación funcional. Esto brinda flexibilidad a los desarrolladores para elegir el enfoque que mejor se adapte a sus necesidades.
- **Interpretado e interactivo**
Python es un lenguaje interpretado, lo que significa que no se compila antes de ejecutarse. Esto facilita el desarrollo y la prueba interactiva del código, permitiendo a los programadores probar y depurar segmentos de código de manera rápida.
- **Amplia biblioteca estándar**
Incluye una extensa biblioteca estándar que abarca desde operaciones básicas hasta módulos especializados. Esta biblioteca facilita el desarrollo de una variedad de aplicaciones sin la necesidad de escribir código desde cero.

- **Portabilidad**
Es compatible con varias plataformas, lo que significa que el mismo código puede ejecutarse en diferentes sistemas operativos sin necesidad de modificaciones significativas.
- **Comunicación activa y ecosistema**
Cuenta con una comunidad de desarrolladores activa y dedicada. Además, dispone de un amplio ecosistema de bibliotecas y frameworks que facilitan la creación de una variedad de aplicaciones, desde desarrollo web hasta análisis de datos y aprendizaje automático.
- **Dinámico y tipado dinámico**
Python es un lenguaje de tipado dinámico, lo que significa que no es necesario declarar el tipo de variable al definirla. Esto simplifica el proceso de desarrollo, pero también requiere atención al tratar con tipos de datos.
- **Open source y gratuito**
Es de código abierto, lo que significa que su código fuente está disponible para el público y puede ser modificado y distribuido libremente.

Estas características hacen de Python una opción popular para una amplia gama de aplicaciones, desde desarrollo de software hasta ciencia de datos y aprendizaje automático.

7.2.2. Java

Java es un lenguaje de programación versátil y robusto que se utiliza ampliamente en diversas aplicaciones, desde desarrollo web hasta aplicaciones empresariales. Aquí se presentan algunas de las características más importantes de Java:

- **Orientado a objetos**
Java sigue el paradigma de programación orientada a objetos, lo que significa que todo en Java es un objeto. Esto promueve la reutilización de código, la modularidad y la organización estructurada del software.
- **Portabilidad**
El lema principal de Java es "Write Once, Run Anywhere" (Escribe una vez, ejecuta en cualquier lugar). Esto se debe a la capacidad de Java de compilar el código fuente en un formato intermedio llamado bytecode, que puede ejecutarse en cualquier máquina virtual Java (JVM), independientemente del sistema operativo subyacente.
- **Máquina Virtual Java**
Más conocido como JVM, la cual este lenguaje la utiliza para ejecutar el bytecode. Esto proporciona una capa de abstracción entre el código Java y el hardware subyacente, mejorando la portabilidad y permitiendo que las aplicaciones Java se ejecuten en diversos entornos.

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

- **Seguro y robusto**
Incorpora características de seguridad y robustez en su diseño. La JVM realiza verificaciones de seguridad durante la ejecución para prevenir posibles amenazas y errores.
- **Manejo automático de memoria**
Cuenta con un recolector de basura que gestiona automáticamente la asignación y liberación de memoria, evitando la necesidad de que los programadores gestionen la memoria manualmente. Esto ayuda a prevenir problemas comunes como fugas de memoria.
- **Multihilo y concurrencia**
Facilita la programación concurrente mediante el soporte integrado para hilos. Los programadores pueden desarrollar aplicaciones que ejecutan tareas simultáneamente, lo que mejora el rendimiento y la capacidad de respuesta.
- **Amplia biblioteca estándar**
Incluye una rica biblioteca estándar que proporciona funcionalidades listas para usar para diversas tareas, desde manipulación de cadenas hasta redes y acceso a bases de datos.
- **Comunidad activa y documentación abundante**
Cuenta con una comunidad de desarrolladores activa y una abundancia de recursos de documentación en línea. Esto facilita el aprendizaje y el desarrollo en Java.
- **Empresarial y escalable**
Java es ampliamente utilizado en aplicaciones empresariales debido a su fiabilidad y escalabilidad. Es especialmente común en el desarrollo de sistemas de backend para grandes empresas.
- **Open source**
Aunque Java tiene implementaciones propietarias, como la de Oracle, también cuenta con implementaciones de código abierto, como OpenJDK, lo que promueve la transparencia y la contribución comunitaria.

Estas características hacen que Java sea una elección popular para el desarrollo de una amplia variedad de aplicaciones, desde pequeñas aplicaciones hasta sistemas empresariales críticos.

7.2.3. C++

C++ es un lenguaje de programación de propósito general que ha sido ampliamente utilizado en una variedad de aplicaciones, desde sistemas embebidos hasta desarrollo de software de alto rendimiento. Aquí están algunas de las características clave de C++:

- **Orientado a objetos**
C++ es un lenguaje de programación orientado a objetos que permite la encapsulación, la herencia y el polimorfismo. Esto facilita la organización y estructuración del código, así como la reutilización de componentes.
- **Eficiencia y rendimiento**
Está diseñado para proporcionar un alto rendimiento y eficiencia en el uso de recursos. Permite un control directo sobre la memoria y proporciona características como punteros, que pueden ser beneficiosos para la optimización de código.
- **Sistemas de tipos fuerte**
Tiene un sistema de tipos fuerte que permite una gestión precisa de los datos. Proporciona tipos de datos básicos y permite la definición de tipos de datos personalizados.
- **Programación genérica**
Soporta programación genérica, permitiendo a los programadores escribir algoritmos y estructuras de datos que son independientes del tipo de datos específico. Esto se logra mediante plantillas.
- **Manipulación de memoria**
Proporciona control directo sobre la asignación y liberación de memoria, lo que permite la gestión manual de la memoria. Esto puede ser beneficioso en situaciones donde se necesita un control preciso sobre los recursos.
- **Portabilidad**
Aunque no es tan portátil como Java, C++ sigue siendo bastante portátil entre diferentes plataformas. Sin embargo, la portabilidad puede requerir ajustes en el código fuente dependiendo del sistema operativo y el compilador utilizado.
- **Biblioteca estándar rica**
Incluye una biblioteca estándar que proporciona una amplia gama de funciones y algoritmos listos para usar, desde manipulación de cadenas hasta operaciones de entrada/salida y estructuras de datos.
- **Multihilo y concurrencia**
Aunque el soporte para multihilo en C++ no es tan integrado como en algunos otros lenguajes, la norma C++11 y versiones posteriores han introducido funcionalidades para la programación concurrente.
- **Compilación estática**
Utiliza un modelo de compilación estática, lo que significa que el código fuente se compila antes de la ejecución. Esto puede resultar en un mejor rendimiento en comparación con lenguajes interpretados.

- **Herencia múltiple**

C++ permite la herencia múltiple, lo que significa que una clase puede heredar de varias clases base. Aunque puede ser poderoso, también requiere una gestión cuidadosa para evitar problemas de ambigüedad.

C++ es conocido por su versatilidad y flexibilidad, lo que lo convierte en una elección popular para aplicaciones que van desde desarrollo de juegos hasta sistemas embebidos y desarrollo de software de sistemas. Sin embargo, su mayor control sobre la memoria y otras características avanzadas también requieren una mayor atención a los detalles por parte del programador.

7.2.4. Elección del lenguaje de programación

En la selección del lenguaje de programación para el desarrollo del proyecto, se ha llevado a cabo un análisis comparativo utilizando una tabla multicriterio. El objetivo de esta evaluación es identificar el programa que mejor se ajuste a las propiedades fundamentales que caracterizan el proyecto académico.

La tabla multicriterio se ha estructurado considerando diversos aspectos clave, todo este análisis se encuentra en la tabla

Tabla 30. Matriz multicriterio de la elección del lenguaje de programación. Fte: Elaboración propia

	Python	Java	C++
Fácil aprendizaje	X		
Múltiples bibliotecas	X	X	X
Extensibilidad	X	X	X
Sintaxis sencilla	X		
Extensibilidad	X	X	X
Gratuito	X		X

En virtud de las consideraciones realizadas, se llega a la conclusión de que la elección adoptada para el lenguaje de programación es **Python**. Este dictamen se fundamenta principalmente en su carácter gratuito, su accesibilidad para personas no especializadas en informática debido a su sintaxis amigable, y su capacidad para integrar todas las funcionalidades esenciales requeridas para el desarrollo de la herramienta previamente definida. La decisión de optar por Python se alinea con la premisa de optimizar la accesibilidad, el aprendizaje eficiente y la plena cobertura de las capacidades necesarias para la implementación exitosa del proyecto.

7.3. ANÁLISIS Y ELECCIÓN DEL SOLVER

La tarea de la creación de un horario escolar implica la resolución de un problema de optimización complejo, donde se deben tener en cuenta múltiples restricciones y preferencias, las cuáles se explicaron durante los capítulos anteriores. Para abordar este desafío, se requiere la selección de un Solver eficaz que se integre sin problemas con el entorno de desarrollo en Python, que es lenguaje que ha sido escogido para resolución del proyecto.

7.3.1. Solvers considerados

Se van a evaluar tres solvers diferentes para determinar cuál de ellos se considera el más adecuado en estas circunstancias:

1) PuLP

Se trata de una biblioteca de optimización en Python que proporciona una interfaz sencilla para definir problemas de programación lineal y entera. Es de código abierto y se integra bien con varios solvers. Puede no ser tan potente para problemas extremadamente grandes o complejos.

2) Google OR-Tools

Ofrece una amplia gama de herramientas para resolver problemas de optimización, incluyendo programación lineal, programación entera, enrutamiento, etc. Es gratuito y de código abierto. Su interfaz puede ser más compleja para usuarios principiantes.

3) Gurobi Optimization

Gurobi es un Solver comercial de alto rendimiento que ofrece eficiencia y escalabilidad. Tiene una interfaz Python robusta y es adecuado para problemas grandes y complejos. Puede requerir una licencia comercial para el uso completo de todas las características.

7.3.2. Elección del Solver

Una vez realizado este pequeño análisis, y en este contexto, se puede dictaminar que la mejor opción a escoger es la de la biblioteca **PuLP**. Esto se debe a sus características beneficiosas, especialmente su gratuidad y su adecuación para problemas de nivel escolar con una complejidad moderada.

A continuación, se presentan las principales justificaciones de esta elección:

- **Gratuito y de código abierto:** no implica costes adicionales, lo cual es esencial en entornos educativos con presupuestos ajustados.
- **Interfaz amigable:** la interfaz de PuLP es accesible y fácil de entender, lo que facilita su implementación por parte de personal educativo y sin experiencia avanzada en programación

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

- **Adecuado para problema de nivel escolar:** dado que la creación del horario es a nivel escolar, la capacidad proporcionada de manejar restricciones y preferencias será suficiente y se saldará sin complicaciones innecesarias.

En resumen, la elección de PuLP como Solver se alinea con las necesidades específicas del proyecto, ofreciendo una solución efectiva y económica para la creación del horario escolar a nivel de una escuela secundaria. Su gratuidad y simplicidad hacen de PuLP una opción ideal para abordar este desafío sin comprometer la calidad y eficiencia del resultado final.

7.4. ANÁLISIS Y ELECCIÓN DEL PROGRAMA DE IMPORTACIÓN Y EXPORTACIÓN DE DATOS

En cuanto a la selección del software para la importación de datos al lenguaje Python y la exportación de los resultados generados por los modelos, se ha evaluado diversas opciones. A pesar de la variedad de herramientas disponibles en el mercado, se ha tomado la decisión de optar por **Microsoft Excel** como la plataforma preferida.

La elección de Microsoft Excel se fundamenta en su amplia presencia y aceptación tanto en entornos académicos como profesionales. Este software ofrece funcionalidades robustas que facilitan la manipulación y organización de datos, así como la presentación de resultados de manera efectiva. Su interfaz intuitiva y la familiaridad que tienen muchos usuarios con esta herramienta contribuyen a su elección.

Adicionalmente, la universidad cuenta con licencias renovables anualmente para Microsoft Excel, lo que garantiza un acceso continuo y legal a esta herramienta. Esta disponibilidad de licencias facilita la implementación y uso del software en el contexto académico de manera consistente.

En resumen, la elección de Microsoft Excel como herramienta para la manipulación de datos y la presentación de resultados se basa en su versatilidad, aceptación generalizada y la conveniencia de contar con licencias institucionales que respaldan su uso continuo en el ámbito académico de la universidad.

7.5. ELECCIÓN FINAL DEL SOFTWARE

Una vez establecidos los programas esenciales para llevar a cabo el proyecto, se procederá a describir la estructura secuencial que se implementará. Esta explicación busca proporcionar una visión detallada del proceso de trabajo, permitiendo una comprensión más profunda y facilitando el entendimiento de las soluciones planteadas.

La estructura secuencial delimitará las fases clave del proyecto, desde la recopilación y preparación de datos hasta la presentación de resultados. Cada etapa será abordada de manera sistemática, siguiendo un orden lógico que garantice la coherencia y eficacia del proyecto en su conjunto.

En la Figura 10, se detallan las fases para llevar a cabo este proyecto.

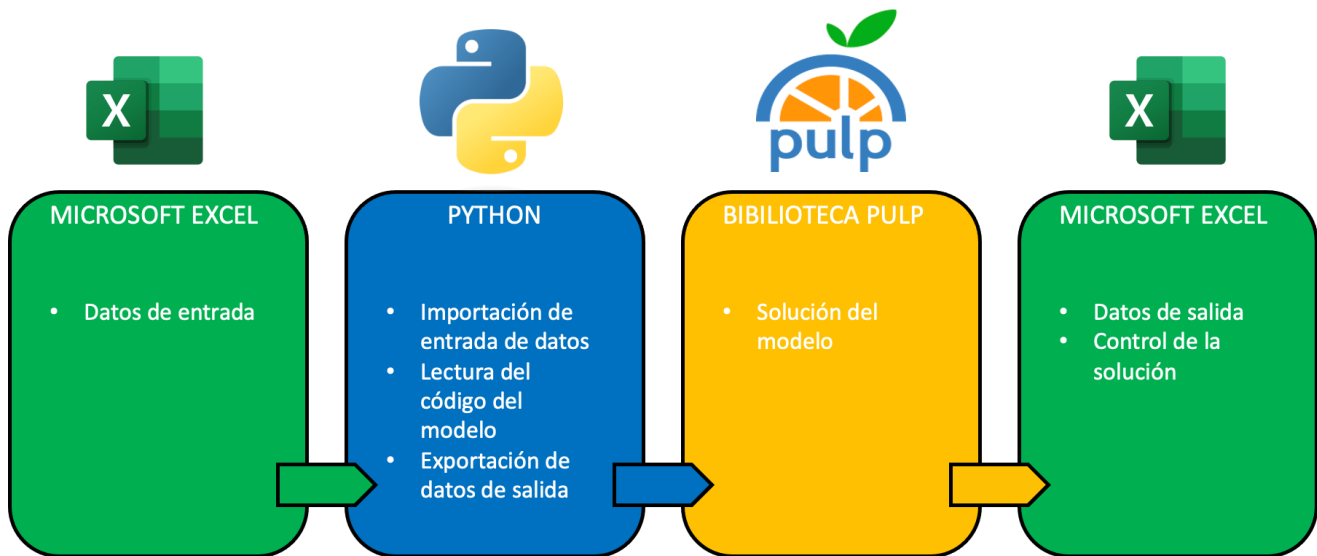


Figura 10. Esquema de las fases de implementación de la herramienta. Fte: Elaboración propia

El proceso inicial de implementación se inicia con la obtención de datos desde el programa Microsoft Excel. En esta fase, se recopilarán los diversos conjuntos de datos o bases de datos necesarios para el desarrollo del proyecto. Es imperativo que estos datos estén meticulosamente definidos y estructurados para facilitar su posterior integración en el entorno de programación de Python.

A continuación, el lenguaje de programación Python asumirá un papel central en el proceso. Mediante el uso de funciones específicas, Python llevará a cabo la importación eficiente de los datos previamente recopilados en Excel. Este paso implica la aplicación de técnicas de codificación para garantizar la coherencia y compatibilidad de los datos entre el entorno de Excel y el entorno de Python. Además, Python se encargará de la codificación de las herramientas esenciales para la creación del horario académico, así como de la exportación de los resultados generados durante el proceso.

Una vez definida la codificación y ejecutado el código modelado en Python, se resolverán los modelos matemáticos utilizando la biblioteca PuLP y su Solver predeterminado correspondiente.

Por último, se exportarán los datos y las soluciones pertinentes de nuevo a Microsoft Excel, con el fin de obtener una nueva base de datos legible y óptima. Cabe destacar además que, la relevancia de esta herramienta va más allá de simplificar el análisis de datos; también desempeñará una función fundamental en el proceso de toma de decisiones estratégicas dentro de la institución académica.

7.6. CONCLUSIONES

El séptimo capítulo ha abordado un análisis exhaustivo de los diversos lenguajes de programación, solvers y programas disponibles para la creación de la herramienta. El objetivo principal ha sido comprender a fondo las características de cada opción y realizar una elección informada que garantice la eficacia y eficiencia del proyecto.

En la fase inicial de este capítulo, se llevó a cabo un análisis detallado de los diferentes lenguajes de programación que podrían ser aplicados en el desarrollo de la herramienta. La elección final recayó en Python, destacando por su accesibilidad y versatilidad, lo que lo convierte en una opción sólida para personas no especializadas en informática. Posteriormente, se abordó la selección del Solver, siendo PuLP la elección preferida dada su especialización en modelos matemáticos y su disponibilidad gratuita. Finalmente, se optó por Microsoft Excel como herramienta para la importación y exportación de datos, aprovechando sus funcionalidades robustas y las licencias universitarias disponibles para su utilización.

El próximo capítulo, se enfocará en la construcción de la herramienta, así como en el desarrollo de la aplicación creada a partir de esta. Se explorarán los pasos necesarios para llevar a cabo este proceso, detallando la implementación práctica de la solución diseñada en los capítulos anteriores. Además, se abordarán las funcionalidades específicas de la herramienta y cómo estas se traducirán en la aplicación final destinada a facilitar la creación de un horario académico eficiente.

8. DISEÑO DE LA APLICACIÓN Y LA HERRAMIENTA

8.1. INTRODUCCIÓN

En el presente capítulo, se llevará a cabo el modelado de la herramienta, atendiendo a las condiciones y requerimientos inherentes a su implementación.

En primer lugar, se abordará el estudio de los criterios de selección de la herramienta, detallando cuál de las metodologías previamente expuestas se considera más apropiada para la concepción adecuada de la misma (8.2).

A continuación, se procederá al desglose detallado del diseño intrínseco de la aplicación y de la herramienta, imprimiendo la importancia que debe tener cada una.

En este sentido, primero se realizará el diseño de la aplicación visual, que mostrará el desarrollo de la misma, sus funcionalidades y los elementos que componen la interfaz (8.3).

En el siguiente subcapítulo, se llevará a cabo la transformación de todas aquellas condiciones y necesidades inherentes a la creación del horario. Estas estarán expresadas en términos de diseño específicos para una herramienta de estas características. Se podrán observar cada uno de los elementos externos que se han escogido en el capítulo anterior para la composición de esta herramienta de programación matemática. Además, todo ello se podrá ver por medio de pantallazos de los diferentes archivos creados para la consecución del proyecto, con el fin de facilitar el funcionamiento y los pasos seguidos dentro todo el proceso realizado (8.4).

8.2. METODOLOGÍA EMPLEADA EN LA APLICACIÓN Y LA HERRAMIENTA

En este pequeño subcapítulo se pretende explicar la metodología utilizada con la cual se pretende llevar a cabo la consecución de la aplicación y de la herramienta, detallando la información necesaria para asegurar el correcto funcionamiento de ambas.

8.2.1. Metodología empleada en la aplicación

Por lo que respecta a la aplicación, en primer lugar, se explicará en que consiste el diseño y la estructura de la aplicación, seguido de cada una de las ventanas emergentes que pueden ir apareciendo a lo largo de esta. Esta información podrá aparecer tanto como un dato esquematizado como un dato gráfico.

Por otra parte, resulta interesante detallar aquellos aspectos más relevantes tanto de las funcionalidades como de los datos visuales de las ventanas.

Al mismo tiempo, será importante destacar algunos ejemplos de datos y gráficos para así entender de una forma más ejemplarizada el funcionamiento de la aplicación.

8.2.2. Metodología empleada en la herramienta

En primer lugar, se presentarán los datos de entrada importados al lenguaje de programación Python desde los archivos de Microsoft Excel creados como base de datos. A continuación, se mostrarán las diferentes informaciones en forma de tabla, incluyendo los datos necesarios para la implementación y ejecución del modelo. Además, se definirán algunos de los parámetros del modelo a partir de estas bases de datos relevantes. Es importante destacar que, en lo que respecta a los datos de entrada, la relevancia de la información expuesta residirá principalmente en su estructura, más que en su contenido específico.

Por otro lado, se presentará la codificación principal del modelo matemático implementado en Python, que se utilizarán para la toma de decisiones en la asignación de recursos humanos y la activación o desactivación de grupos. Estos modelos incluirán la racionalización de la oferta académica, la racionalización basada en la limitación de la saturación departamental, y la gestión de la plantilla docente en relación con las variaciones en la oferta académica. Es importante resaltar que los códigos secundarios se adjuntarán directamente en los anexos con la finalidad de no sobrecargar este capítulo, y permitir de esta forma una correcta comprensión del código.

Por último, se mostrarán datos de salida junto con sus resultados, y la monitorización de aquellos que tengan una mayor importancia dentro del modelo completo para así poder concluir sobre cuál es el desarrollo de esta herramienta.

8.3. DISEÑO DE LA INTERFAZ DE LA APLICACIÓN

8.3.1. Marco de desarrollo de la aplicación

La aplicación comprenderá un marco de desarrollo de tres ventanas denominadas de la siguiente forma: grupos, profesores y generación de horarios. Esta última ventana es la que permitirá la creación del horario, además de la depuración de este en caso de que se quieran hacer pruebas con la creación de varios horarios con diferentes condiciones.

La programación de esta aplicación se ha realizado en un archivo Excel mediante VBA. Se ha escogido esta opción, ya que resulta sencillo crear macros y botones dentro de este entorno, y para un prototipo como este lo mejor era iniciar una interfaz de estas características. En la Figura 11 se puede observar el menú principal de la aplicación.

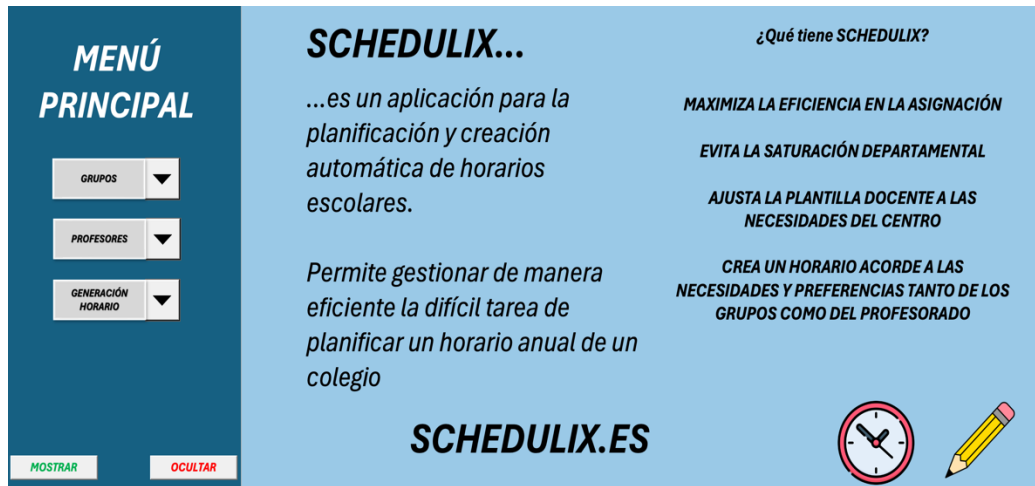


Figura 11. Menú principal de la interfaz de la aplicación Fte: Elaboración propia

En la parte de la izquierda se encuentran las tres ventanas, las cuales funcionan como tres menús desplegables con diferentes opciones para abrir. En la parte baja existen dos botones creados únicamente para facilitar el trabajo a la hora de programar las diferentes macros utilizados en la interfaz.

Respecto a los menús desplegables, el primero de ellos está compuesto por cada uno de los grupos existentes de la escuela, el de profesores está compuesto por todos los profesores del centro, y, por último, el desplegable de generación de horarios contiene únicamente dos opciones, generar y depurar horario. En la Figura 12, Figura 13, y Figura 14 se detalla la visualización de estos desplegables.

En la siguiente sección se pasará al detalle de cada una de ellas, explicando las funcionalidades que pueden encontrarse dentro de cada ventana.



Figura 12. Desplegable de los diferentes grupos existentes. Fte: Elaboración propia

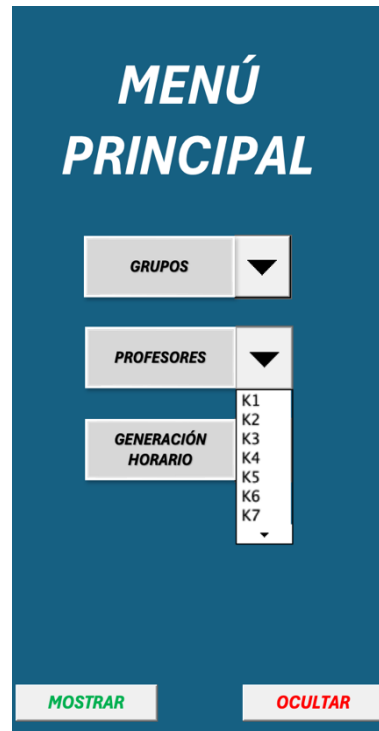


Figura 13. Desplegable de la plantilla de docentes. Fte: Elaboración propia



Figura 14. Desplegable de las opciones de generación de horarios. Fte: Elaboración propia

8.3.2. Funcionalidades y elementos de la interfaz

Respecto a las funcionalidades y elementos de la interfaz, se va a proceder explicando cada una de las ventanas por separado. De esta forma se podrá comprender la base del funcionamiento de esta aplicación.

En primer lugar, se encuentra el desplegable de **GRUPOS**. Esta ventana alberga, como se ha comentado en la anterior sección, cada uno de los grupos que conforman esta escuela. Cada una de estas ventanas corresponde a un grupo único, y la información que se va a almacenar en estas hojas es la correspondiente a los horarios de cada grupo. Una vez se haya creado un horario, este aparecerá en cada uno de los grupos, para que así los profesores y/o alumnos puedan conocer de primera mano esta información.

Además de esto, también se puede observar un botón denominado “**ATRÁS**”. Este botón aparece en todas y cada una de las ventanas de los desplegables, y lo que permite es navegar a lo largo de toda la interfaz con la posibilidad de volver al menú principal desde cualquier hoja con un único click sobre él.

En la Figura 15, queda detallado como sería la hoja de un grupo una vez el horario ha sido creado. La información necesaria para esta interfaz se ha considerado que debe ser esta, esta aplicación no funciona como una base de datos, para esa función existen otros ficheros que podrán estudiarse y analizarse más adelante.

	L	M	X	J	V
I1	3E4B	104E	3E4B	104E	104E
I2		LCL4E (Prof: K29)	T4E (Prof: K13)	LCL4E (Prof: K29)	AE4E (Prof: K2)
I3	GH4E (Prof: K12)	GH4E (Prof: K12)	EF34E (Prof: K39)		V4B (Prof: K27)
I4	2E4B	2E4B	1E4E	I4E (Prof: K22)	I4E (Prof: K22)
I5	V4B (Prof: K27)		V4B (Prof: K27)	3E4B	3E4B
I6	I4E (Prof: K22)	1E4E	2E4B	1E4E	LCL4E (Prof: K29)
I7	EF34E (Prof: K39)	204E	204E	204E	GH4E (Prof: K12)

H/SEM ESPERADAS 32
H/SEM REALES 32
ATRÁS

Figura 15. Ejemplo del horario del grupo 4B en la aplicación. Fte: Elaboración propia

Referente al segundo desplegable de **PROFESORES**, almacena la misma información que en el desplegable de los grupos, pero en este caso la información que tiene es completamente individual. Cada ventana es para un profesor diferente, y en estas ventanas aparecerá el horario anua correspondiente que tendrá cada profesor. A su vez también aparece un horario personalizado que lo que alberga son las preferencias que tiene cada profesor dependiendo de la franja horaria y el día de la semana.

De la misma forma que en el desplegable anterior, en cada una de las opciones de este existe el mismo botón para poder acceder al menú principal en el momento que se requiera.

Por tanto, la Figura 16 muestra cómo sería la hoja individual de un profesor cualquiera dentro de esta interfaz. Cabe destacar que esta opción también podría servir para que los profesores al inicio de cada curso pudieran rellenar su horario de preferencias con el fin de facilitar el trabajo en la creación del horario anual. Para ello, podría ser interesante plantearse la opción de solicitar unas credenciales a la hora de abrir una ventana de este desplegable.

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

	I1	I2	I3	I4	I5	I6	I7
L	10	0	0	0	0	-5	-10
M	10	0	0	0	0	-5	-10
X	10	0	0	0	0	-5	-10
J	10	0	0	0	0	-5	-10
V	10	0	0	0	0	-5	-10

	L	M	X	J	V
I1	1E1ART (1ART)	1E1ART (1ART)	V2BA (2ART)	1E1ART (1ART)	1E1ART (1ART)
I2	T2HUM (2HUM)				
I3	V2BA (2ART)	1E2ART (2ART)			1E2ART (2ART)
I4	V2BA (2CI)	V2BA (2ART)	1E2ART (2ART)		T1HUM (1HUM)
I5	1E2ART (2ART)	T2E (2B)	V2BA (2HUM)		
I6		T2E (2A)			
I7					

H/sem	17
-------	----

CS	40
----	----

ATRÁS

Figura 16. Ejemplo de horario del profesor K3 en la aplicación. Fte: Elaboración propia

Por último, se halla el desplegable de **GENERACIÓN HORARIO**. Tal y como se ha comentado al inicio de esta sección, este menú desplegable solo tiene dos posibles opciones a elegir, que además están bastante relacionadas.

La opción de **GENERAR** consiste en la generación del horario. Una vez todos los datos estén correctamente almacenados y se hayan considerado todas las condiciones y requisitos para la creación del horario, es posible proceder a crear el horario. A pesar de que la herramienta es ejecutada mediante Python, en esta hoja Excel se podrán ver los resultados definidos por la herramienta. La opción que no es posible es directamente ejecutar desde el Excel el código Python.

Por su parte, la opción de **DEPURAR** sirve como una forma rápida de eliminar un horario completo. Esta opción se ha considerado crearla debido a que puede resultar interesante a la hora de la creación de este prototipo.

Teniendo en cuenta que pueden ir apareciendo imprevistos a lo largo de la creación de un horario anual, es muy posible que el primer horario creado no sea el definitivo. Por ello, esta rápida vía de eliminación del horario puede ahorrar mucho tiempo a la hora de encontrar el horario óptimo posible.

8.4. HERRAMIENTA PARA LA CREACIÓN DE HORARIOS

8.4.1. Datos de entrada

Los datos requeridos para el funcionamiento de la herramienta se han definido de forma que faciliten su lectura dentro del lenguaje de programación Python.

Cabe diferenciar dos bases de datos desde las cuáles se puede recopilar toda la información necesaria. Estos dos bases se han denominad de la siguiente forma: **datos del colegio y preferencias del profesorado**. La primera de ellas consiste en los datos más

imprescindibles que puede tener un colegio (asignaturas, profesores, horas semanales, capacidad de los profesores, departamentos, etc.). En la Figura 17, se puede observar el ejemplo de la estructura que tiene en la base de datos un grupo cualquiera. En este caso cada grupo tiene una hoja exclusiva donde puede hallarse parte de la información más relevante de ese grupo.

GENERAL	Código	Nombre	Responsable	h/sem
	AE4E	Atención al Estudio 4E	K2	1
	EF34E	Educación Física 3y4E	K21	2
	GH4E	Geografía e Historia 4E	K12	3
	LCL4E	Lengua castellana y Literatura 4E	K29	3
	I4E	Inglés 4E	K22	3
	T4E	Tutoría 4E	K13	1
	V4A	Valenciano: Lengua y Literatura 4A	K9	3
ESPECÍFICA	Código	Nombre	Responsable	h/sem
	1E4E	Biología y Geología / Economía / Tecnología	K1 / K23 / K20	3
	2E4A	Latín / Física y Química 4A / Iniciación Actividad Empresarial	K19 / K7 / K23	3
	3E4A	Matemáticas Acad 4A / Matemáticas Apli	K33 / K20	4
OPTATIVA	Código	Nombre	Responsable	h/sem
	1O4E	Filosofía / Oral Inglés	K31 / K22	3
	2O4E	Francés / Informática / Educación Plástica	K4 / K28 / K36	3
				HORAS TOTALES: 32

Figura 17. Extracto de la tabla de información del grupo 4A. Fte: Elaboración propia

Por lo que respecta a esta tabla, simplemente sirve de ayuda para la construcción de la base de datos, gracias a la cual se leen los mismos. De esta hoja tipo no se lee ningún dato, ha servido de base para la creación de las demás hojas que se podrán ver a continuación.

Cabe destacar, sin embargo, la composición de la misma:

- **Código:** representa el código de la asignatura correspondiente.
- **Celdas en amarillo:** representan el tipo de asignatura que se puede encontrar a la derecha de estas mismas celdas.
- **Nombre:** es la columna donde se encuentra el nombre las diferentes asignaturas generales, bloques de específicas o bloques de optativas.
- **Responsable:** es la columna correspondiente a los profesores que imparten las diferentes asignaturas o los diferentes bloques.
- **h/SEM:** corresponde a las horas semanales de cada asignatura o de cada bloque.
- **HORAS TOTALES:** corresponde a las horas totales semanales que debe recibir ese grupo.

Una vez visto y entendido esta primera hoja de la base de datos, es posible adentrarse en aquellos datos que sí van a ser leídos dentro de los modelos de programación matemática.

Al elaborarse todas las hojas para cada uno de los grupos existentes dentro de esta escuela, se pueden ya aglutinar el conjunto total de asignaturas. En la Figura 18, se observa

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

como quedaría estructurada la hoja donde se hayan, este caso particular, todas las asignaturas optativas existentes en esta escuela de secundaria, también se encuentra en dicha hoja una columna correspondiente a los bloques de los cuales son pertenecientes las diferentes asignaturas optativas (índice O del modelo 1 de programación matemática). Las hojas correspondientes a las asignaturas generales y específicas no se han insertado debido al tamaño tan grande que tienen, y, en consecuencia, a la poca visibilidad que tendrían en el documento.

Código	Bloque	Descripción
1.1O1E	1O1E	Francés
1.2O1E	1O1E	Laboratorio de Creación Audiovisual
1.3O1E	1O1E	Oral Inglés
2.1O1E	2O1E	Taller de Refuerzo Lingüístico
2.2O1E	2O1E	Taller de Refuerzo Matemático
2.3O1E	2O1E	Taller de Relaciones Digitales Responsables
1.1O2E	1O2E	Francés
1.2O2E	1O2E	Programación Robótica I
2.1O2E	2O2E	Taller de Refuerzo Lingüístico
2.2O2E	2O2E	Taller de Refuerzo Matemático
2.3O2E	2O2E	Taller Profundización
1.1O3E	1O3E	Cultura Clásica
1.2O3E	1O3E	Programación Robótica II
2.1O3E	2O3E	Francés
2.2O3E	2O3E	Taller de Refuerzo Lingüístico
2.3O3E	2O3E	Taller de Refuerzo Matemático
1.1O4E	1O4E	Filosofía
1.2O4E	1O4E	Oral Inglés
2.1O4E	2O4E	Francés
2.2O4E	2O4E	Educación Plástica
2.3O4E	2O4E	Informática
1.1O1BA	1O1BA	Biología Humana
1.2O1BA	1O1BA	Descubriendo nuestras raíces clásicas
1.3O1BA	1O1BA	Imagen y Sonido
2.1O1BA	2O1BA	Francés
2.2O1BA	2O1BA	Programación y Redes I
2.3O1BA	2O1BA	Tecnología e Ingeniería I
1.1O2BA	1O2BA	Francés
1.2O2BA	1O2BA	Psicología
1.3O2BA	1O2BA	Tecnología e Ingeniería II
2.1O2BA	2O2BA	Dibujo Artístico
2.2O2BA	2O2BA	Historia de la Filosofía
2.3O2BA	2O2BA	Historia de la Música y la Danza
2.4O2BA	2O2BA	Tecnología, Información y Comunicación

Figura 18. Extracto de la tabla de las asignaturas específicas totales de la escuela. Fte: Elaboración propia

Por otra parte, también se puede conocer el total de docentes que componen la escuela. Para ello, se ha elaborado un tabla parecida a la anterior que alberga el nombre de los distintos docentes que imparten clase en el centro (de esta hoja se lee el índice K, que aparece en ambos modelos). Todo esto se queda reflejado en la Figura 19.

De igual forma que ocurre con la lectura de asignaturas, en la hoja de profesores se leerá únicamente la columna de la izquierda correspondiente al ID de cada docente.

ID	Descripción
K1	ALEJANDRO LUAÑA
K2	ANA NIETO
K3	CARME LÓPEZ
K4	CARMEN GALINDO
K5	CARMEN ORS
K6	CARMEN TOLOSA
K7	CHARO SORIANO
K8	CLARA DÍEZ
K9	CRISTINA CEBRIÁN
K10	CRISTÓBAL ALCAIDE
K11	GRACIA GIL
K12	IRENE CASTELLANOS
K13	ISABEL HUERTA
K14	JOAN ROVIRA
K15	JOSEP LLUÍS VILAGELIU
K16	JOSEP M ^a CALATAYUD
K17	JUAN J. POMERMON
K18	JUAN M. VERA
K19	JUAN RIVERA
K20	MARGA MONSELL
K21	MARÍA GAUDES
K22	MARTA BORRULL
K23	MARTA MAGRO
K24	NIEVES CALATAYUD
K25	PAU FENOLLOSA
K26	PEDRO PÉREZ
K27	PEPA PALAU
K28	PERE MORALAR
K29	PILAR GÓMEZ
K30	RAFA CARABELLA
K31	ROSANA SOLER
K32	ROSARIO GARRIDO
K33	ROSER CEBRIÀ
K34	SANTI TELLO
K35	SERGIO DEVIS
K36	SONIA ALZOLA
K37	VICENTE CEREZO
K38	VICENTE SOLER
K39	XAVIER PLANES

Figura 19. Extracto de la tabla de los docentes totales del centro. Fte: Elaboración propia

A continuación, cabe resaltar una de las tablas que une la información recopilada de la hoja de cada grupo junto con la anterior tabla de profesores. Esta tabla queda representada en la Figura 20, y permite conocer cuáles son los departamentos existentes dentro de la escuela y a qué departamento pertenece cada uno de los profesores. Lo que hay representado es únicamente un extracto de la misma.

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

	Biología y Geología	Economía	Educación Física	Educación Plástica	Filosofía	Física y Química
K1	1	0	0	0	0	0
K2	0	0	0	0	0	0
K3	0	0	0	1	0	0
K4	0	0	0	0	0	0
K5	0	0	0	0	1	0
K6	1	0	0	0	0	0
K7	0	0	0	0	0	1
K8	0	0	0	0	0	0
K9	0	0	0	0	0	0
K10	0	0	0	0	0	1
K11	0	0	0	1	0	0
K12	0	0	0	0	0	0
K13	0	0	0	0	0	0
K14	0	0	0	0	0	0
K15	0	0	0	0	0	0

Figura 20. Extracto de la tabla que relaciona Profesor y Departamento. Fte: Elaboración propia

El funcionamiento es muy simple, los 1s que aparecen en la fila correspondiente a cada profesor corresponde a los departamentos a los que este pertenece. Puede ocurrir que un mismo profesor pertenezca a más de un departamento, y que un departamento pueda tener más de un profesor, como ya se vio en capítulos anteriores del proyecto.

Las siguientes figuras (Figura 21, Figura 22 y Figura 23) se corresponden con extractos de las tablas que relacionan las asignaturas con sus horas semanales respectivas para cada grupo. Tal y como se observa, las columnas corresponden con las asignaturas y las filas con los grupos existentes. Gracias a estas tablas se importan los datos de los parámetros de horas semanales al modelo de programación matemática (parámetros HS perteneciente al modelo 2, y, HSe y HSo pertenecientes al modelo 1).

DISEÑO DE LA APLICACIÓN Y LA HERRAMIENTA

HS	1A	1B	2A	2B	3A	3B
AE1A	1	0	0	0	0	0
AE1B	0	1	0	0	0	0
AE2E	0	0	1	1	0	0
AE3E	0	0	0	0	1	1
AE4E	0	0	0	0	0	0
BG1A	3	0	0	0	0	0
BG1B	0	3	0	0	0	0
BG3E	0	0	0	0	2	2
EF12E	2	2	2	2	0	0
EF34E	0	0	0	0	2	2
EF1ART	0	0	0	0	0	0
EF1BA	0	0	0	0	0	0
EPA23E	0	0	2	2	2	2
F1ART	0	0	0	0	0	0
F1BA	0	0	0	0	0	0
FQ2A	0	0	2	0	0	0
FQ2B	0	0	0	2	0	0
FQ3A	0	0	0	0	2	0
FQ3B	0	0	0	0	0	2
GH1A	3	0	0	0	0	0

Figura 21. Extracto de la tabla del parámetro HS. Fte: Elaboración propia

HSe	4A	4B	1ART	1CI	1HUM	2ART	2CI	2HUM
1E4E	3	3	0	0	0	0	0	0
2E4A	3	0	0	0	0	0	0	0
2E4B	0	3	0	0	0	0	0	0
3E4A	4	0	0	0	0	0	0	0
3E4B	0	4	0	0	0	0	0	0
1E1ART	0	0	4	0	0	0	0	0
2E1ART	0	0	4	0	0	0	0	0
3E1ART	0	0	4	0	0	0	0	0
1E1CI	0	0	0	4	0	0	0	0
2E1CI	0	0	0	4	0	0	0	0
3E1CI	0	0	0	4	0	0	0	0
1E1HUM	0	0	0	0	4	0	0	0
2E1HUM	0	0	0	0	4	0	0	0
3E1HUM	0	0	0	0	4	0	0	0
1E2ART	0	0	0	0	0	4	0	0
2E2ART	0	0	0	0	0	4	0	0
3E2ART	0	0	0	0	0	4	0	0
1E2CI	0	0	0	0	0	0	4	0
2E2CI	0	0	0	0	0	0	4	0
3E2CI	0	0	0	0	0	0	4	0
1E2HUM	0	0	0	0	0	0	0	4
2E2HUM	0	0	0	0	0	0	0	4
3E2HUM	0	0	0	0	0	0	0	4

Figura 22. Extracto de la tabla del parámetro HSe. Fte: Elaboración propia

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

HSbo	1A	1B	2A	2B	3A	3B
101E	1	1	0	0	0	0
201E	1	1	0	0	0	0
102E	0	0	1	1	0	0
202E	0	0	1	1	0	0
103E	0	0	0	0	1	1
203E	0	0	0	0	1	1
104E	0	0	0	0	0	0
204E	0	0	0	0	0	0
101BA	0	0	0	0	0	0
201BA	0	0	0	0	0	0
102BA	0	0	0	0	0	0
202BA	0	0	0	0	0	0

Figura 23. Extracto de la tabla del parámetro HSo. Fte: Elaboración propia

El siguiente parámetro que destacar es el de las capacidades que tienen los profesores de dar los diferentes tipos de asignaturas. En el caso del modelo 1, la gran mayoría de los bloques de específicas y optativas son impartidas por diferentes profesores, ya que, como se ha visto anteriormente, pueden estar compuestos por más de una asignatura. Respecto al modelo 2, las asignaturas generales únicamente pueden ser impartidas por un profesor exclusivo. Dicha asignatura es única, ya sea para un grupo en concreto o para un curso total. Todo ello queda definido en las siguientes figuras donde se observan diferentes extractos de estas tablas relacionales entre profesor y asignatura (Figura 24, Figura 25 y Figura 26).

CAPACIDAD GENERAL (CP)	F1BA	FQ2A	FQ2B	FQ3A	FQ3B	GH1A	GH1B	GH2E	GH3E	GH4E	HE
K1	0	0	0	0	0	0	0	0	0	0	0
K2	0	0	0	0	0	0	0	0	0	0	0
K3	0	0	0	0	0	0	0	0	0	0	0
K4	0	0	0	0	0	0	0	0	0	0	0
K5	1	0	0	0	0	0	0	0	0	0	0
K6	0	0	0	0	0	0	0	0	0	0	0
K7	0	1	0	1	0	0	0	0	0	0	0
K8	0	0	0	0	0	0	0	0	0	0	0
K9	0	0	0	0	0	0	1	0	0	0	0
K10	0	0	1	0	0	0	0	0	0	0	0
K11	0	0	0	0	0	0	0	0	0	0	0
K12	0	0	0	0	0	1	0	0	0	1	0
K13	0	0	0	0	0	0	0	0	0	0	0
K14	0	0	0	0	0	0	0	0	0	0	0
K15	0	0	0	0	0	0	0	1	0	0	0
K16	0	0	0	0	0	0	0	0	1	0	1
K17	0	0	0	0	0	0	0	0	0	0	0
K18	0	0	0	0	0	0	0	0	0	0	0
K19	0	0	0	0	0	0	0	0	0	0	0
K20	0	0	0	0	0	0	0	0	0	0	0
K21	0	0	0	0	0	0	0	0	0	0	0
K22	0	0	0	0	0	0	0	0	0	0	0
K23	0	0	0	0	0	0	0	0	0	0	0

Figura 24. Extracto de la tabla del parámetro CP. Fte: Elaboración propia

CAPACIDAD ESPECÍFICA (CPe)	1E4E	2E4A	2E4B	3E4A	3E4B	1E1ART	2E1ART	3E1ART	1E1CI	2E1CI	3E1CI
K1	1	0	0	0	0	0	0	0	0	0	0
K2	0	0	0	0	0	0	0	0	0	0	0
K3	0	0	0	0	0	1	0	0	0	0	0
K4	0	0	0	0	0	0	0	0	0	0	0
K5	0	0	0	0	0	0	0	0	0	0	0
K6	0	0	0	0	0	0	0	0	1	0	0
K7	0	1	0	0	0	0	0	0	0	1	0
K8	0	0	0	0	0	0	0	0	0	0	0
K9	0	0	0	0	0	0	0	0	0	0	0
K10	0	0	0	0	0	0	0	0	0	0	0
K11	0	0	0	0	0	0	1	0	1	0	0
K12	0	0	0	0	0	0	0	0	0	0	0
K13	0	0	0	0	0	0	0	0	0	0	0
K14	0	0	0	0	1	0	0	0	0	0	1
K15	0	0	0	0	0	0	0	0	0	0	0
K16	0	0	0	0	0	0	0	0	0	0	0
K17	0	0	0	0	0	0	0	0	0	0	0
K18	0	0	0	0	0	0	0	0	0	0	0
K19	0	1	1	0	0	0	0	0	0	0	0
K20	1	0	0	1	1	0	0	0	0	0	0
K21	0	0	0	0	0	0	0	0	0	0	0

Figura 25. Extracto de la tabla del parámetro CPe. Fte: Elaboración propia

CAPACIDAD OPTATIVA (CPo)	101E	201E	102E	202E	103E	203E	104E	204E	101BA	201BA	102BA	202BA
K21	0	0	0	0	0	0	0	0	0	0	0	0
K22	0	0	0	0	0	0	1	0	0	0	0	0
K23	0	0	0	0	0	0	0	0	0	0	0	0
K24	1	0	0	0	0	0	0	0	0	0	1	1
K25	0	0	0	0	0	0	0	0	0	0	0	0
K26	0	0	0	0	0	0	0	0	0	0	0	0
K27	0	0	0	0	0	0	0	0	0	0	0	0
K28	0	1	0	0	1	0	0	1	0	1	0	1
K29	0	0	0	0	0	0	0	0	0	0	0	0
K30	0	0	0	0	0	0	0	0	0	0	0	1
K31	0	0	0	0	0	0	1	0	0	0	1	0
K32	0	1	0	1	0	1	0	0	0	0	0	0
K33	0	1	0	1	0	1	0	0	0	0	0	0
K34	0	0	0	0	0	0	0	0	0	0	0	0
K35	0	0	0	0	0	0	0	0	1	0	0	0
K36	0	0	0	0	0	0	0	1	0	0	0	0
K37	0	0	0	0	0	0	0	0	0	0	0	0
K38	0	0	0	0	0	0	0	0	0	0	0	0
K39	0	0	0	0	0	0	0	0	0	0	0	0

Figura 26. Extracto de la tabla del parámetro CPo. Fte: Elaboración propia

Es importante resaltar que todas estas figuras son extractos, tal y como se ha comentado anteriormente, por lo tanto, no todas ellas se ven al completo. Esto no es necesario debido a que los datos que faltan son fáciles de entender de qué forma se encuentran dentro de esta hoja Excel.

Una vez expuestos todos estos extractos, queda por destacar la tabla que forma parte de la otra base de datos. Esta base de datos consiste en una hoja Excel donde cada profesor tiene su propia hoja en la cual puede modificar, a su gusto, las preferencias que tiene dentro de cada franja horaria de cada día de la semana. Esto permite recopilar la información que luego se guarda en el parámetro CS (perteneciente tanto al modelo 1 como

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

al modelo 2). Los niveles de preferencia de los profesores se miden según la escala de satisfacción vista en la Tabla 29. En la Figura 27, queda detallado lo que se acaba de explicar.

	A	B	C	D	E	F	G	H
1		I1	I2	I3	I4	I5	I6	I7
2	L	10	0	0	0	0	-5	-10
3	M	10	0	0	0	0	-5	-10
4	X	10	0	0	0	0	-5	-10
5	J	10	0	0	0	0	-5	-10
6	V	10	0	0	0	0	-5	-10
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								
34								
35								
36								

Figura 27. Ejemplo de la hoja de un profesor ejemplo de la base de datos perteneciente las preferencias de los docentes (preferencias del profesorado). Fte: Elaboración propia

El ejemplo del profesor que puede verse en la anterior figura tiene la preferencia de trabajar las primeras horas, es decir, sus niveles de satisfacción serán elevados en caso de que se le asignen horarios donde dé clase a primera hora de la mañana. Por otra parte, tiene la preferencia de acabar su jornada cuánto antes, por tanto, si se le asignan clases en las últimas franjas, su nivel de satisfacción bajará considerablemente.

Este es el parámetro CS que recoge los horarios individualizados de cada profesor. Le da un valor correspondiente a cada franja horaria de cada día de la semana, dependiendo de las diferentes preferencias de los distintos docentes. Es por ello, que el parámetro CS al completo será de una gran extensión, ya que se cuenta actualmente con 39 profesores en total.

Así pues, estas dos bases de datos recopilan toda esta información que se ha visto. Gracias a ellas se puede conocer:

- Información propia de cada grupo
- Asignaturas totales de cada tipo

- Docentes existentes
- Composición de los diferentes departamentos
- Parámetros de horas semanales para cada tipo de asignatura
- Parámetros de capacidad de profesor para cada tipo de asignatura
- Parámetros de niveles de preferencia de cada profesor respecto al horario que se le pueda ser asignado.

8.4.2. Implementación del modelo 1 en Python

En este apartado se va a mostrar la codificación desarrollada en Python que ha permitido el diseño del modelo 1 de la herramienta. Tal y como se expone en el capítulo del modelado, este modelo podría llegar a ser conflictivo a la hora del desarrollo de la herramienta, por ello, se decide diseñar este en primer lugar.

A lo largo de esta sección se presentarán diferentes screenshots del código escrito detallando cada uno de ellos que representa dentro del código y dentro de este modelo.

Todo este código ha sido creado en base al modelo 1 de programación matemática anteriormente diseñado, exceptuando aquellas partes del código que forman parte tanto de la importación como de la exportación de los diferentes datos necesarios.

En primer lugar, se detalla el código que permite leer los diferentes índices del problema. Algunos de ellos, como las franjas horarias, los días de la semana o los grupos, han sido directamente implementados en el código, ya que su extensión es pequeña respecto a los índices que se importan desde Microsoft Excel. En la Figura 28, se exponen los índices que han sido directamente escritos dentro del código.

Figura 28

```
# ÍNDICES
I = ["I1", "I2", "I3", "I4", "I5", "I6", "I7"]

I_0 = ["I1", "I7"]

D = ["L", "M", "X", "J", "V"]

G = ["1A", "1B", "2A", "2B", "3A", "3B", "4A", "4B", "1ART", "1CI", "1HUM", "2ART", "2CI", "2HUM"]
```

Figura 28. Definición de los índices I, I_0, D y G del código del modelo 1. Fte: Elaboración propia

En esta primera figura cabe destacar una diferencia del código respecto del modelo matemático. La ecuación **¡Error! No se encuentra el origen de la referencia.**), correspondiente a la restricción de que las asignaturas optativas solo puedan ser asignadas en la primera o última franja horaria de cualquier día de la semana, no se encuentra dentro del código Python de este modelo 1. En el modelado matemático no hizo falta crear un índice nuevo para que esta restricción quedase bien definida, pero a la hora de escribirlo en Python, resultó más sencillo crear un pequeño índice que comprendiera únicamente la primera y la última franja horaria del día (I1 e I7).

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

A continuación, se expone en Figura 29 en las cuales podemos encontrar el resto de los índices que han sido necesarios de definir para la construcción del modelo.

```
# Función para leer y cargar los datos de los índices E (Específicas), O (Optativas) y K (Profesores)
def leer_listado_especificas(BaseDatos):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='E', header=None)
    # Buscar el índice de la fila donde están las asignaturas
    indice_especificas = df[df[0] == 'Código'].index[0] + 1
    # Leer la columna de asignaturas desde ese índice hacia abajo
    especificas = df.loc[indice_especificas:, 0].tolist()
    return especificas

def leer_listado_optativas(BaseDatos):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='O', header=None)
    # Buscar el índice de la fila donde están las asignaturas
    indice_optativas = df[df[0] == 'Código'].index[0] + 1
    # Leer la columna de asignaturas desde ese índice hacia abajo
    optativas = df.loc[indice_optativas:, 0].tolist()
    return optativas

def leer_listado_profesores(BaseDatos):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='Profesores', header=None)
    # Buscar el índice de la fila donde están los profesores
    indice_profesores = df[df[0] == 'ID'].index[0] + 1
    # Leer la columna de profesores desde ese índice hacia abajo
    profesores = df.loc[indice_profesores:, 0].tolist()
    return profesores
```

Figura 29. Definición de los índices E, O y K en Python, importados a través de la base de datos “datos del colegio”. Fte: Elaboración propia

En la siguiente parte del código se pueden encontrar definidos los parámetros. Todos estos son importados a través de Microsoft Excel, desde la base de datos expuesta en la anterior sección. Esta parte del código se puede comprobar en la Figura 30, Figura 31, Figura 32, Figura 33 y Figura 34.

```
# PARÁMETROS
# Parámetro HSe. Número de horas semanales que deben impartirse para el grupo g del bloque de asignaturas específicas e
def leer_horas_semanales_especificas(BaseDatos, G, E):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='HSbe', index_col=0, header=0)
    # Limpiar espacios en blanco de columnas e índice
    df.columns = df.columns.str.strip()
    df.index = df.index.str.strip()
    HSe = {}
    for e in E:
        for g in G:
            if g in df.columns and e in df.index:
                valor_horas = df.at[e, g]
                HSe[(e, g)] = valor_horas
            else:
                print(f"Advertencia: Grupo '{g}' o bloque '{e}' no encontrado en el DataFrame HSe")
    return HSe
```

Figura 30. Definición del parámetro HSe en Python, importados a través de la base de datos “datos del colegio”. Fte: Elaboración propia

```
# Parámetro HSo. Número de horas semanales que deben impartirse para el grupo g de optativas o
def leer_horas_semanales_optativas(BaseDatos, G, O):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='HSbo', index_col=0, header=0)

    # Limpiar espacios en blanco de columnas e índice
    df.columns = df.columns.str.strip()
    df.index = df.index.str.strip()

    HSo = {}
    for o in O:
        for g in G:
            if g in df.columns and o in df.index:
                valor_horas = df.at[o, g]
                HSo[(o, g)] = valor_horas
            else:
                print(f"Advertencia: Grupo '{g}' o optativa '{o}' no encontrado en el DataFrame HSo")
    return HSo
```

Figura 31. Definición del parámetro HSo en Python, importado a través de la base de datos “datos del colegio”. Fte: Elaboración propia

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

```
# Parámetro CPe. Parámetro binario que toma valor 1 en caso de que el profesor k pueda impartir
# el bloque de asignaturas específicas e, toma valor 0 si no está capacitado
def leer_datos_CPe(BaseDatos, E, K):
    xls = pd.ExcelFile(BaseDatos)

    # Leer los datos de la hoja CPe
    df_cpe = pd.read_excel(xls, sheet_name='CPbe', index_col=0, header=0)

    # Inicializar el diccionario CPe
    CPe = {}

    # Iterar sobre cada profesor y bloque de específicas para obtener el valor binario correspondiente
    for profesor in K:
        for especifica in E:
            # Obtener el valor de la celda correspondiente
            valor_celda = df_cpe.at[profesor, especifica]
            # Asignar el valor binario al diccionario CPe
            CPe[(profesor, especifica)] = valor_celda

    return CPe
```

Figura 32. Definición del parámetro CPe en Python, importado a través de la base de datos “datos del colegio”. Fte: Elaboración propia

```
# Parámetro CPo. Parámetro binario que toma valor 1 en caso de que el profesor k pueda impartir
# el bloque de asignaturas optativas o, toma valor 0 si no está capacitado
def leer_datos_CPo(BaseDatos, O, K):
    xls = pd.ExcelFile(BaseDatos)

    # Leer los datos de la hoja CPo
    df_cpo = pd.read_excel(xls, sheet_name='CPbo', index_col=0, header=0)

    # Inicializar el diccionario CPo
    CPo = {}

    # Iterar sobre cada profesor y bloque de optativas para obtener el valor binario correspondiente
    for profesor in K:
        for optativa in O:
            # Obtener el valor de la celda correspondiente
            valor_celda = df_cpo.at[profesor, optativa]
            # Asignar el valor binario al diccionario CPo
            CPo[(profesor, optativa)] = valor_celda

    return CPo
```

Figura 33. Definición del parámetro CPo en Python, importado a través de la base de datos “datos del colegio”. Fte: Elaboración propia

```
# Parámetro CS. Coste de satisfacción del profesor k en la franja horaria i del día de la semana d
def procesar_excel_a_diccionarios(Horario_Docentes):
    # Leer el archivo de Excel
    xl = pd.ExcelFile(Horario_Docentes, engine='openpyxl')
    # Diccionario que contendrá todos los horarios por persona
    CS = {}
    # Procesar cada hoja del archivo de Excel
    for hoja in xl.sheet_names:
        # Leer la hoja actual en un DataFrame
        df = xl.parse(hoja)
        # Eliminar filas donde 'día de la semana' es NaN
        df.dropna(subset=[df.columns[0]], inplace=True)
        coste_profesor = {} # Diccionario para el profesor actual
        # Obtener los nombres de las columnas (intervalos de horas)
        intervalos_horas = df.columns[1:] # Excluir la primera columna de días
        # Iterar sobre cada fila del DataFrame
        for index, row in df.iterrows():
            dia_semana = row[0] # El día de la semana está en la primera columna
            if pd.isna(dia_semana):
                continue # Saltar filas donde el día de la semana es NaN
            # Iterar sobre cada intervalo de horas
            for hora in intervalos_horas:
                valor_entero = row[hora]
                if pd.isna(valor_entero):
                    continue # Saltar valores binarios NaN
                # Crear la clave del diccionario
                clave = (hoja, hora, dia_semana)
                # Asignar el valor correspondiente
                coste_profesor[clave] = valor_entero
        # Agregar el diccionario de la persona al diccionario general
        CS.update(coste_profesor)

    return CS
```

Figura 34. Definición del parámetro CS en Python, importado a través de la base de datos “preferencias del profesorado”. Fte: Elaboración propia

Por lo que respecta a las variables de decisión, la Figura 35 muestra la definición de todas ellas dentro del código.

```
# VARIABLES DE DECISIÓN
# Variable Z. Variable binaria que toma valor 1 en caso de que para el grupo g en la franja horaria i
# se asigna el bloque de específicas e impartida por el profesor k en el día d, 0 en caso contrario
Z = LpVariable.dicts("Z", (G, I, E, K, D), cat='Binary')

# Variable T. Variable binaria que toma valor 1 en caso de que para el grupo g en la franja horaria i
# se asigna el bloque de optativas o impartida por el profesor k en el día d, 0 en caso contrario
T = LpVariable.dicts("T", (G, I, O, K, D), cat='Binary')

# Variable V. Variable binaria que toma valor 1 en caso de que para el grupo g
# se imparte el bloque de específicas e por el profesor k, 0 en caso contrario
V = LpVariable.dicts("V", (G, E, K), cat='Binary')

# Variable W. Variable binaria que toma valor 1 en caso de que para el grupo g
# se imparte el bloque de optativas o por el profesor k, 0 en caso contrario
W = LpVariable.dicts("W", (G, O, K), cat='Binary')
```

Figura 35. Definición de las variables de decisión creadas para el modelo 1 en Python. Fte: Elaboración propia

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

Por último, cabe destacar la parte más condicionante de la resolución de este primer modelo: la función objetivo y las restricciones. En las siguientes X figuras se encuentran definida la función objetivo y todas las restricciones que ha sido necesario añadir al modelo.

```
# F0: Maximizar la satisfacción de las preferencias horarias de los profesores
prob += (
    lpSum(Z[g][i][e][k][d] * CS.get((k, i, d), 0) for g in G for i in I for e in E for k in K for d in D) +
    lpSum(T[g][i][o][k][d] * CS.get((k, i, d), 0) for g in G for i in I for o in O for k in K for d in D)
)
```

Figura 36. Definición de la Función Objetivo del modelo 1 en Python. Fte: Elaboración propia

```
# RESTRICCIONES
# 1. Solamente un profesor único k puede impartir la específica e al grupo g
for g in G:
    for e in E:
        prob += lpSum(V[g][e][k] for k in K) == 1

# 2. Solamente un profesor único k puede impartir la optativa o al grupo g
for g in G:
    for o in O:
        prob += lpSum(W[g][o][k] for k in K) == 1

# 3. Asignación específica limitada a profesores capacitados
for g in G:
    for e in E:
        for k in K:
            if CPe[(k, e)] == 0:
                prob += V[g][e][k] == 0

# 4. Asignación optativa limitada a profesores capacitados
for g in G:
    for o in O:
        for k in K:
            if CPo[(k, o)] == 0:
                prob += W[g][o][k] == 0
```

Figura 37. Definición de las restricciones del modelo 1 en Python 1/4. Fte: Elaboración propia

```
# 5. En caso de que un profesor  $k$  esté asignado a la específica  $e$  del grupo  $g$  en cualquier momento, es el responsable
for g in G:
    for e in E:
        for k in K:
            prob += lpSum(Z[g][i][e][k][d] for i in I for d in D) == HSe.get((e, g), 0) * V[g][e][k]

# 6. En caso de que un profesor  $k$  esté asignado a la optativa  $o$  del grupo  $g$  en cualquier momento, es el responsable
for g in G:
    for o in O:
        for k in K:
            prob += lpSum(T[g][i][o][k][d] for i in I for d in D) == HSo.get((o, g), 0) * W[g][o][k]

# 7. Para un grupo  $g$  la específica  $e$  en una franja horaria  $i$ 
# puede impartirse como máximo una vez en un día  $d$ 
for g in G:
    for e in E:
        for d in D:
            prob += lpSum(Z[g][i][e][k][d] for i in I for k in K) <= 1

# 8. Para un grupo  $g$  la optativa  $o$  en una franja horaria  $i$  puede impartirse como máximo una vez en un día  $d$ 
for g in G:
    for o in O:
        for d in D:
            prob += lpSum(T[g][i][o][k][d] for i in I for k in K) <= 1
```

Figura 38. Definición de las restricciones del modelo 1 en Python 2/4. Fte: Elaboración propia

```
# 9. Todas las horas semanales específicas deben impartirse
for g in G:
    for e in E:
        prob += lpSum(Z[g][i][e][k][d] for i in I for k in K for d in D) == HSe.get((e, g), 0)

# 10. Todas las horas semanales optativas deben impartirse
for g in G:
    for o in O:
        prob += lpSum(T[g][i][o][k][d] for i in I for k in K for d in D) == HSo.get((o, g), 0)

# 11. Cada grupo  $g$  solo puede tener una específica  $e$  o una optativa  $o$ 
# en una franja horaria  $i$ 
for g in G:
    for i in I:
        for d in D:
            prob += (
                lpSum(Z[g][i][e][k][d] for e in E for k in K) +
                lpSum(T[g][i][o][k][d] for o in O for k in K)
            ) <= 1
```

Figura 39. Definición de las restricciones del modelo 1 en Python 3/4. Fte: Elaboración propia

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

```
# 12. Un profesor k solo puede impartir una asignatura j, una específica e o una optativa o
# en una franja horaria i de un día d
for i in I:
    for k in K:
        for d in D:
            prob += (
                lpSum(Z[g][i][e][k][d] for g in G for e in E) +
                lpSum(T[g][i][o][k][d] for g in G for o in O)
            ) <= 1

# 13. Las optativas o solo pueden impartirse en la primera o en la última franja horaria i
for g in G:
    for o in O:
        for k in K:
            for d in D:
                for i in I:
                    if i not in I_0:
                        prob += T[g][i][o][k][d] == 0
```

Figura 40. Definición de las restricciones del modelo 1 en Python 4/4. Fte: Elaboración propia

Las restantes partes del código que no aparecen en este capítulo, se añadirá al capítulo de Anexos del proyecto.

8.4.3. Implementación del modelo 2 en Python

Al igual que en la anterior sección, se va a mostrar la codificación desarrollada en Python que ha permitido el diseño del modelo 2 de la herramienta.

Se volverán a presentar diferentes screenshots del código escrito explicando en qué consisten cada uno de los elementos que aparecen.

Todo este código ha sido creado en base al modelo 2 de programación matemática anteriormente diseñado, exceptuando aquellas partes del código que forman parte tanto de la importación como de la exportación de los diferentes datos necesarios.

En este caso los índices que componen este modelo pueden encontrarse en la Figura 41, que se adjunta a continuación:

```
# ÍNDICES

I = ["I1", "I2", "I3", "I4", "I5", "I6", "I7"]

D = ["L", "M", "X", "J", "V"]

G = ["1A", "1B", "2A", "2B", "3A", "3B", "4A", "4B", "1ART", "1CI", "1HUM", "2ART", "2CI", "2HUM"]
```

Figura 41. Definición de los índices I, D y G del código del modelo 2. Fte: Elaboración propia

Estos tres índices ya están definidos en el modelo 1, pero como también van a ser utilizados dentro de este es necesario que quede reflejado.

En la Figura 42 se pueden observar el resto de los índices que son utilizados para la elaboración de este modelo.

```

# Función para leer y cargar los datos de los índices J (Asignaturas) y K (Profesores)
def leer_listado_asignaturas(BaseDatos):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='Generales', header=None)

    # Buscar el índice de la fila donde están las asignaturas
    indice_asignaturas = df[df[0] == 'Código'].index[0] + 1

    # Leer la columna de asignaturas desde ese índice hacia abajo
    asignaturas = df.loc[indice_asignaturas:, 0].tolist()

    return asignaturas

def leer_listado_profesores(BaseDatos):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='Profesores', header=None)

    # Buscar el índice de la fila donde están los profesores
    indice_profesores = df[df[0] == 'ID'].index[0] + 1

    # Leer la columna de profesores desde ese índice hacia abajo
    profesores = df.loc[indice_profesores:, 0].tolist()

    return profesores
    
```

Figura 42. Definición de los índices J y K en Python, importados a través de la base de datos "datos del colegio". Fte: Elaboración propia

Los siguientes elementos a definir son los parámetros, tal y como se ha explicado en el modelo 1, estos se leen a través del archivo Excel que trabajo como base de datos (datos del colegio). Quedan reflejados en la Figura 43 y Figura 44.

```

# PARÁMETROS
# Parámetro HS. Número de horas semanales que deben impartirse para el grupo g de la asignatura j
def leer_horas_semanales(BaseDatos, G, J):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='HS', index_col=0, header=0)
    # Limpiar espacios en blanco de columnas e índice
    df.columns = df.columns.str.strip()
    df.index = df.index.str.strip()
    HS = {}
    for j in J:
        for g in G:
            if g in df.columns and j in df.index:
                valor_horas = df.at[j, g]
                HS[(j, g)] = valor_horas
            else:
                print(f"Advertencia: Grupo '{g}' o asignatura '{j}' no encontrado en el DataFrame HS")
    return HS
    
```

Figura 43. Definición del parámetro HS en Python, importado a través de la base de datos "datos del colegio". Fte: Elaboración propia

Diseño de una herramienta mediante programación matemática
para la gestión y organización del horario anual en una escuela de secundaria en Valencia

```
# Parámetro CP. Parámetro binario que toma valor 1 en caso de que el profesor k pueda impartir la asignatura j
# Toma valor 0 si no está capacitado
def leer_datos_CP(BaseDatos, J, K):
    xls = pd.ExcelFile(BaseDatos)

    # Leer los datos de la hoja CP
    df_cp = pd.read_excel(xls, sheet_name='CP', index_col=0, header=0)

    # Inicializar el diccionario CP
    CP = {}

    # Iterar sobre cada profesor y asignatura para obtener el valor binario correspondiente
    for profesor in K:
        for asignatura in J:
            # Obtener el valor de la celda correspondiente
            valor_celda = df_cp.at[profesor, asignatura]
            # Asignar el valor binario al diccionario CP
            CP[(profesor, asignatura)] = valor_celda

    return CP
```

Figura 44. Definición del parámetro CP en Python, importado a través de la base de datos "datos del colegio". Fte: Elaboración propia

La definición del parámetro CS sería exactamente igual que en el modelo 1, por tanto, no es necesario desglosar su escritura en Python. Esta información ya está reflejada en la Figura 34.

Para acabar, cabe destacar la definición tanto de la función objetivo como de las restricciones. Para ello, se han implementado las siguientes tres figuras.

```
# F0: Maximizar la satisfacción de las preferencias horarias de los profesores
prob += (
    lpSum(X[g][i][j][k][d] * CS.get((k, i, d), 0) for g in G for i in I for j in J for k in K for d in D)
)
```

Figura 45. Definición de la Función Objetivo del modelo 2 en Python. Fte: Elaboración propia

```
# RESTRICCIONES
# 1. Solamente un profesor único k puede impartir la asignatura j al grupo g
for g in G:
    for j in J:
        prob += lpSum(Y[g][j][k] for k in K) == 1

# 2. Asignación general limitada a profesores capacitados
for g in G:
    for j in J:
        for k in K:
            if CP[(k, j)] == 0:
                prob += Y[g][j][k] == 0

# 3. En caso de que un profesor k esté asignado a la asignatura j del grupo g en cualquier momento, es el responsable
for g in G:
    for j in J:
        for k in K:
            prob += lpSum(X[g][i][j][k][d] for i in I for d in D) == HS.get((j, g), 0) * Y[g][j][k]

# 4. Para un grupo g la asignatura j en una franja horaria i puede impartirse como máximo una vez en un día d
for g in G:
    for j in J:
        for d in D:
            prob += lpSum(X[g][i][j][k][d] for i in I for k in K) <= 1
```

Figura 46. Definición de las restricciones del modelo 2 en Python 1/2. Fte: Elaboración propia

```
# 5. Todas las horas semanales deben impartirse
for g in G:
    for j in J:
        prob += lpSum(X[g][i][j][k][d] for i in I for k in K for d in D) == HS.get((j, g),

# 6. Cada grupo g solo puede tener una asignatura j en una franja horaria i de un día d
for g in G:
    for i in I:
        for d in D:
            prob += lpSum(X[g][i][j][k][d] for j in J for k in K) <= 1

# 7. Cada profesor k solo puede tener una asignatura j en una franja horaria i de un día d
for i in I:
    for k in K:
        for d in D:
            prob += lpSum(X[g][i][j][k][d] for g in G for j in J) <= 1

# 8. No se pueden realizar asignaciones en franjas ocupadas
for g in G:
    for i in I:
        for d in D:
            for j in J:
                for k in K:
                    if (i, d) in ocupacion_grupos[g] or (i, d) in ocupacion_profesores[k]:
                        prob += X[g][i][j][k][d] == 0 # No asignar en franjas ocupadas
```

Figura 47. Definición de las restricciones del modelo 2 en Python 2/2. Fte: Elaboración propia

De la misma forma que se ha detallado para el modelo 1, en este segundo modelo hay partes del código que no se han expuesto, pero quedarán detalladas en los anexos del proyecto.

8.5. CONCLUSIONES

Este capítulo ha estado enfocado en el diseño de una aplicación funcional con una interfaz visual que permite implementar el diseño de la herramienta previamente elaborada. Además, se han detallado y explicado los dos modelos matemáticos desarrollados, mostrando cómo fueron plasmados en código Python.

Gracias a este trabajo, se podido seleccionar y desarrollar la metodología empleada, tanto en la aplicación como en la herramienta en general. Se han descrito las funcionalidades de la aplicación y los diferentes elementos que componen la interfaz visual, brindando una visión del marco de desarrollo potencial de esta herramienta para la creación de horarios escolares.

En la parte final del capítulo, se ha abordado el desarrollo y explicación tanto de la base de datos utilizada por el código como del propio código. Al igual que en el capítulo de modelado, se han descrito ambos modelos de manera separada, priorizando aquel que podría presentar conflictos en la ejecución de la herramienta.

En el futuro cercano, se procederá a la validación de esta herramienta consolidada aplicándola a un caso de estudio real, con el fin de evaluar de manera precisa su eficacia en la creación de horarios escolares reales.

9. VALIDACIÓN DE LA HERRAMIENTA SOBRE EL CASO REAL

9.1. INTRODUCCIÓN

Este último capítulo va a consistir en realizar una validación de la herramienta sobre el caso real por el que se empezó a trabajar en este proyecto.

La primera tarea consistirá en comentar la metodología empleada para la validación de la herramienta, tratando de ver cómo se puede plantear el análisis de los resultados que se proporcionan (9.2).

En segundo lugar, se llevará a cabo la necesaria validación del diseño elaborado, permitiendo así una evaluación experimental para determinar la eficacia del diseño con respecto al objetivo establecido (9.3).

Finalmente, tras la validación del prototipo de la herramienta desarrollada, se procederá al análisis de los resultados obtenidos. Esta sección detallará aquellos puntos favorables y desfavorables que tiene la herramienta creada, con el fin de entender las cosas que pueden controlarse, y aquellas que no se pueden controlar, y, por tanto, será interesante seguir trabajando en ellas en el futuro (9.4).

9.2. METODOLOGÍA EMPLEADA

La validación de un modelo es un paso fundamental en cualquier proceso de desarrollo, ya que permite comprobar no solo su correcto desempeño, sino también la coherencia y precisión de los resultados obtenidos. Es crucial verificar que los resultados sean confiables y estén alineados con los objetivos planteados dentro del contexto de análisis. Esta etapa asegura que el modelo esté adecuadamente ajustado a las características del problema y que responda de manera efectiva a las necesidades para las cuales fue diseñado.

En los apartados siguientes, se presentarán los resultados generados tras la lectura y procesamiento completo de los datos por parte del modelo de programación matemática. Se procederá a analizar en detalle las diferentes metas alcanzadas, destacando aquellos aspectos en los que el modelo ha cumplido con los objetivos propuestos. Además, se realizará un análisis exhaustivo de los posibles elementos que puedan generar dificultades o conflictos en la consecución de un resultado plenamente satisfactorio, identificando así las áreas que podrían requerir ajustes o mejoras para optimizar el desempeño del modelo en situaciones reales.

Como se puede observar en el gráfico anterior, cada uno de los grupos está recibiendo las horas semanales correspondientes respecto de su programación escolar. Este dato refleja uno de los objetivos marcados en el diseño de la herramienta, en el que todos los grupos debían recibir el número de horas semanales correspondientes referentes al curso que pertenecen.

Además de esto, todos los grupos están recibiendo las asignaturas correspondientes por cada profesor capacitado, otro punto favorable a destacar, ya que no se están asignando clases a profesores incapaces de impartir dichas asignaturas.

Por lo que respecta a los profesores, la correcta asignación de estos se mide con la observación de los horarios de los mismos. En la Figura 49, se haya el ejemplo del horario de un profesor (K27) donde primero aparece el horario referente al parámetro CS (nivel de satisfacción del profesor respecto a una asignación en una franja horaria concreta de un día de la semana concreto), debajo de este se encuentra el horario personalizado de cada una de las asignaturas a impartir por el mismo, y por último el parámetro CS de cada uno obtenido respecto a la asignación realizada por la herramienta.

	I1	I2	I3	I4	I5	I6	I7
L	10	0	0	0	0	-5	-10
M	10	0	0	0	0	-5	-10
X	10	0	0	0	0	-5	-10
J	10	0	0	0	0	-5	-10
V	10	0	0	0	0	-5	-10
	L	M	X	J	V		
I1	V1BA (1CI)	PI1B (1B)		V1BA (1HUM)	PI1B (1B)		
I2	V1BA (1ART)						
I3					V4B (4B)		
I4		V1BA (1CI)	V1BA (1HUM)		V1BA (1ART)		
I5	V4B (4B)	V1BA (1HUM)	V4B (4B)	V1BA (1ART)	V1BA (1CI)		
I6							
I7							
	CS	40					

Figura 49. Horario personalizado del profesor K27. Fte: Elaboración propia

Tal y como se detalla en la figura, este docente concreto prefiere siempre dar el máximo de clases a primera hora e intentar acabar lo más pronto posible de darlas. Respecto a este profesor en concreto, se puede observar que se están satisfaciendo todas sus necesidades, ya que está impartiendo 4 de 5 clases posibles a primera hora y ninguna en las últimas dos franjas horarias de todos los días de la semana. Por todo ello, su nivel de satisfacción es de 40.

Para destacar también la parte menos positiva, en la Figura 50 se puede ver como este profesor en concreto no está viendo satisfechas sus necesidades.

Así pues, el número total de profesores que tienen preferencia son 27, y como puede observarse en el gráfico anterior, 11 de estos se están sintiendo insatisfechos con la asignación de alguna clase en su horario. Esto representa un 41% del total de profesores. Sin embargo, si no consideramos insatisfechos aquellos profesores que tienen un -10 o un -15 en su nivel individual de satisfacción, ya que únicamente están teniendo una o dos horas asignadas insatisfechas del total que imparten (una media de 14,15 clases por profesor), el nivel de insatisfacción representaría un 14,81% respecto de los profesores que tienen preferencias, y un 10,26% del total de profesores.

Además, el nivel de satisfacción CS total es de **70**. Este valor es positivo y relevante, ya que indica que hay mayor satisfacción que insatisfacción.

En conclusión, se puede afirmar que esta herramienta está generando un horario con un rendimiento de satisfacción de casi un **90%**, respecto a la función objetivo del modelo matemático global. El detrimento que se experimenta en algunos profesores es debido a la asignación realizada en los grupos, que como se ha comentado ya, esta asignación es condición más primordial que la satisfacción de los profesores.

9.4. ANÁLISIS DE LOS RESULTADOS OBTENIDOS

Respecto a los resultados obtenidos por parte de la herramienta, es importante entrar al detalle y observar cómo se comporta el modelo, tanto en la parte de la asignación de los profesores como en la parte de asignación de los grupos.

- **Aparición de huecos entre clases de los grupos**

Tal y como se observa en los resultados importados de la Figura 48, existen huecos en algunos días. Es decir, el alumnado tendrá horas sin asignar algunos días de la semana, esto se debe a que ningún grupo tiene las 35 horas que completan un calendario semanal. Sin embargo, resultaría interesante que los huecos libres quedarán a primera o última hora del día, para que así los alumnos puedan entrar más tarde o salir antes del colegio. Este punto podría ser un aspecto que mejorar, aunque también es cierto que esos huecos le pueden permitir al alumnado un tiempo interesante para adelantar trabajo de la semana. Posiblemente, si entrarán más tarde a clase o salieran antes dedicarían esas horas para otras cosas externas al trabajo realizado en el colegio.

- **Imposibilidad de impartir algunas asignaturas optativas**

Los resultados que proporciona el modelo 1 impide que se puedan impartir la totalidad de las asignaturas optativas disponibles. Esto es debido al gran volumen de clases que soportan algunos profesores, y a la restricción de que las asignaturas optativas solo pueden impartirse a primera o última hora del día. Estas dos condiciones dificultan que pueda existir la posibilidad de impartir todas las asignaturas disponibles en un bloque de optativas. Este es claramente un aspecto que mejorar, pero también es cierto que habiendo hasta cuatro asignaturas en un mismo bloque de optativas es difícil que haya varios alumnos que escojan la totalidad de estas, ya que

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

también hay que destacar que algunos de ellos tienen estas asignaturas convalidadas.

Se puede concluir, por tanto, que existen tres aspectos negativos detallados en estas dos últimas secciones. El primero de ellos referencia que no todos los profesores van a verse satisfechos con la asignación de sus clases, pero únicamente se verán insatisfechos unos pocos más del 10%. Es un dato que puede mejorarse, pero viendo la dificultad en la asignación, habría que hacer una reestructuración de la capacidad de los profesores para poder mejorar el rendimiento de satisfacción.

Respecto a la aparición de huecos entre clases y la imposibilidad de ofrecer todas las asignaturas optativas de un mismo bloque, sería más destacable el hecho de que no aparecieran huecos en los horarios de los grupos en franjas horarias como I3 o I4, porque son horas en las que ya están programadas pausas dentro del propio programa escolar. El inconveniente de las optativas es mejorable, pero no es un elemento diferencial, ya que no son asignaturas troncales (generales) que son de vital importancia para la formación del alumno. Al fin y al cabo, son horas añadidas con diferentes disciplinas creadas para despertar el interés del alumnado en ciertos ámbitos poco cotidianos.

9.5. CONCLUSIONES

En este capítulo, se ha planteado un proceso de validación para la herramienta matemática, aplicando la misma metodología empleada en el desarrollo de un modelo de programación matemática.

En consecuencia, se han presentado inicialmente los resultados obtenidos, poniendo especial atención en aquellos aspectos clave que permiten evaluar de manera precisa tanto la efectividad como el rendimiento de la herramienta. El análisis se ha realizado de manera detallada con el objetivo de poder valorar de forma fundamentada cada uno de los elementos que influyen en la validez del modelo programado. De este modo, se ha buscado identificar posibles puntos fuertes y debilidades que puedan impactar en el desempeño general de la herramienta, asegurando que los resultados sean coherentes y confiables dentro del contexto de su aplicación.

En cuanto al análisis de los resultados, se han resaltado los puntos de mejora de la herramienta, subrayando su importancia y relevancia en el contexto del modelo. Estos aspectos representan áreas clave que podrían mejorar la oferta de asignaturas y los aspectos de los horarios creados. El énfasis en estos puntos permite una evaluación más profunda y un plan de acción para fortalecer la funcionalidad global, asegurando que la herramienta esté alineada con los objetivos y necesidades planteadas.

Una vez finalizado el proceso de validación, se expondrán las conclusiones obtenidas a lo largo del proyecto, junto con un resumen de los aprendizajes adquiridos y las posibles direcciones para trabajos futuros.

10. CONCLUSIONES

10.1. CONCLUSIONES DE LA MEMORIA

El presente documento para la ejecución del proyecto escogido ha seguido un enfoque exhaustivo metodológico. A lo largo del desarrollo, se abordaron diversas facetas, desde la comprensión del entorno educativo hasta la validación práctica de la herramienta final.

Inicialmente, el estudio del sistema educativo español y la evaluación de la institución académica proporcionaron un marco claro sobre las dificultades actuales en la gestión de horarios. Se identificaron problemas como la falta de optimización y la rigidez de los sistemas existentes, los cuales limitaban la capacidad de adaptarse a cambios y optimizar el uso de recursos. Este entendimiento fue fundamental para definir los objetivos del proyecto: crear una herramienta que facilitara una gestión de horarios más eficiente y flexible.

En la fase de antecedentes teóricos, se estableció la base conceptual necesaria para desarrollar la solución propuesta. La investigación en optimización matemática y programación lineal resultó esencial, ya que proporcionó los principios y técnicas necesarios para modelar el problema de manera eficaz. Estas teorías no solo fundamentaron el modelo matemático, sino que también guiaron la elección de los conceptos y las técnicas adecuadas para la implementación.

El análisis situacional permitió una evaluación detallada del estado actual en la institución, revelando las causas subyacentes de los problemas en la organización de horarios. Se identificaron deficiencias en la gestión manual y la falta de herramientas automatizadas, lo que reforzó la necesidad de una solución tecnológica que pudiera abordar estos desafíos de manera sistemática y eficiente.

En cuanto a la selección de alternativas, se presentaron y evaluaron tres enfoques distintos para la creación del horario. La evaluación se centró en criterios como la viabilidad económica, la eficiencia en la asignación o la flexibilidad a las necesidades institucionales. La alternativa del sistema de optimización, en este caso, basad en la programación matemática emergió como la más adecuada, dada su capacidad para ofrecer soluciones optimizadas y flexibles, alineándose perfectamente con los requisitos del proyecto.

El desarrollo de los dos modelos matemáticos fue un componente clave en la creación de la herramienta. Ambos modelos integraron todas las variables relevantes y restricciones del problema, permitiendo una resolución efectiva del mismo. La formulación de estos aseguró que la herramienta pudiera generar horarios que cumplieran con todas las exigencias de la institución, optimizando tanto el tiempo como los recursos disponibles.

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

Además, trató maximizar en la medida de lo posible las preferencias de los diferentes docente, siendo este el principal objetivo al inicio del proyecto.

El análisis y la elección del software implicaron una revisión de diferentes lenguajes de programación, herramientas de importación y exportación de datos, y solvers disponibles. Esta evaluación detallada permitió seleccionar las opciones más adecuadas para el desarrollo de la herramienta, garantizando una implementación eficiente y efectiva.

El diseño de la aplicación se centró en la creación de una interfaz intuitiva y accesible para los usuarios. La herramienta final no solo incluye funcionalidades avanzadas para la gestión de horarios, sino que también presenta una interfaz amigable que facilita su uso por parte de los responsables de la creación de horarios en la escuela. La atención al diseño y la experiencia del usuario fueron aspectos importantes para asegurar que la herramienta fuera práctica y fácil de manejar.

Finalmente, la validación de la herramienta en el caso real valoró su efectividad en la creación y gestión de horarios. La herramienta demostró su capacidad para cumplir con los objetivos establecidos, facilitando la adaptación a cambios y optimizando la planificación horaria. Los resultados obtenidos validaron la solución propuesta como una herramienta eficaz para resolver los problemas identificados y mejorar la eficiencia en la gestión de horarios escolares.

En conclusión, el proyecto ha logrado desarrollar una solución integral que aborda los desafíos de la gestión de horarios en esta escuela secundaria. La combinación de teoría matemática, análisis situacional, y una implementación práctica ha resultado en una herramienta que no solo cumple con los objetivos del proyecto, sino que también ofrece una mejora significativa en la eficiencia y flexibilidad en la organización de horarios.

10.2. LECCIONES APRENDIDAS

El proceso de elaboración de este proyecto ha sido un proceso enriquecedor que ha proporcionado valiosas lecciones tanto a nivel técnico como personal. A causa de la realización del trabajo, he podido desarrollar competencias y habilidades que seguro en un futuro cercano agradeceré haberlas adquirido.

Una de las lecciones aprendidas, ha sido la crucial importancia que tiene un buen análisis inicial y una buena planificación. Entender el contexto educativo y las necesidades específicas de la institución ha permitido diseñar una solución que realmente abordara los problemas existentes. Esta fase inicial es esencial para definir claramente los objetivos y establecer una base sólida para todo el proyecto.

Por otra parte, el trabajo ha subrayado cómo la integración de teorías matemáticas y conceptos prácticos puede ofrecer soluciones efectivas a problemas complejos. Aplicar principios de optimización matemática y programación lineal al problema de la creación de horarios no solo ha facilitado la resolución eficiente del problema, sino que también ha mejorado la comprensión del proceso y la calidad de la herramienta final.

También cabe destacar la relevancia de aprender a adaptarse a los cambios y ser flexible en la implementación, para así poder superar obstáculos y avanzar hacia el objetivo final de manera eficiente.

Por último, resaltar la comunicación y la colaboración. Trabajar con diferentes partes interesadas, incluyendo personal de la institución, facilitó la obtención de información clave y la validación de la solución. La comunicación clara y constante es esencial para alinear expectativas y garantizar que el proyecto se mantenga en el camino correcto.

En resumen, el proyecto ha proporcionado valiosas lecciones en áreas clave como la planificación, la integración de teoría y práctica o la comunicación y colaboración con las otras partes interesadas. Estas lecciones no solo han contribuido al éxito del proyecto, sino que también ofrecen aprendizajes aplicables a futuros proyectos y desafíos profesionales.

10.3. FUTURAS LÍNEAS DE TRABAJO

Una vez completado el diseño de la herramienta es importante destacar una serie de futuras líneas de trabajo en las que centrarse para mejorar el funcionamiento de esta herramienta, y conseguir así resultados más optimizados y con una mayor amplitud. Estas mejoras se han redactado de mayor a menor importancia en la mejora de la herramienta.

- **Unificar los dos modelos de programación matemática en uno**

La implementación de esta línea de trabajo tiene el potencial de mejorar significativamente los resultados de la herramienta, ya que permitiría asignar todas las asignaturas simultáneamente. Este enfoque podría satisfacer las necesidades de un mayor número de profesores en comparación con la situación actual. Sin embargo, esta mejora no fue viable desde el inicio debido a la complejidad inherente en la asignación de asignaturas específicas y optativas. A pesar de los esfuerzos realizados en este sentido, no se lograron resultados satisfactorios en la asignación de estas asignaturas, lo que impidió su inclusión en la versión inicial de la herramienta.

- **Desarrollar el problema de las aulas**

Como se mencionó al inicio del documento, la asignación de aulas para cada grupo de estudiantes representaba un desafío significativo en la programación de horarios. El problema surgía porque una aula podía no tener la capacidad suficiente para acomodar a todos los estudiantes de una clase determinada. Aunque a corto plazo este problema se ha mitigado mediante el desdoble de clases —ya que hay profesores suficientes y capacitados para enseñar la misma asignatura a un grupo dividido en dos—, esta solución temporal podría convertirse en un problema a largo plazo si no se encuentran dos profesores capacitados para impartir la clase en cuestión.

Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual en una escuela de secundaria en Valencia

Por lo tanto, es fundamental explorar soluciones alternativas que aborden este problema de manera más definitiva. Investigar y desarrollar una solución más robusta podría ayudar a evitar problemas futuros relacionados con la capacidad de las aulas y la disponibilidad de profesores, asegurando así una asignación de horarios más eficiente y sostenible en el tiempo.

- **Implementar más funcionalidades y visualizaciones a la interfaz del usuario**

La interfaz actual de la herramienta se ha desarrollado como un prototipo inicial y presenta áreas significativas para mejorar tanto en términos de funcionalidad como de visualización. En su estado actual, la aplicación permite visualizar los horarios de cada grupo y de cada profesor, así como generar y depurar horarios. Esta implementación se realizó en Excel utilizando VBA debido a su simplicidad y facilidad de uso.

No obstante, es sabido que existen opciones más avanzadas que podrían ofrecer una mayor eficiencia en el desarrollo de la herramienta. La exploración de otros lenguajes de programación y librerías podría permitir una optimización de la funcionalidad y una mejora en la interfaz de usuario. Entre las posibles mejoras, se podría considerar la introducción de sistemas de autenticación para acceder a horarios personalizados, lo que añadiría un nivel adicional de seguridad y personalización, además de indagar en la creación de una interfaz en Python mediante la librería Tkinter.

La investigación de alternativas tecnológicas y la mejora de los elementos actuales, como la integración de credenciales para el acceso a horarios personalizados, deberían ser las prioridades para el desarrollo futuro de la herramienta. Estas mejoras no solo optimizarán el rendimiento de la aplicación, sino que también ofrecerán una experiencia de usuario más completa y adaptada a las necesidades específicas de la institución.

REFERENCIAS

- Alnowaini, G., & Aljomai, A. A. (2021). *Genetic Algoritim For Solving University Course Timetabling Problem Using Dynamic Chromosomes*. Yemen.
- Assi, M., Halawi, B., & Haraty, R. A. (2018). *Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetabling Problem*. Beirut.
- Aznar, R. A. (1996). *Estrategias y sistemas de información*. Barcelona.
- Battiti, R., & Tecchiolli, G. (1994). *The Reactive Tabu Search*.
- Blanco Nieto, L. J., & Climent Rodríguez, N. (16 de Febrero de 2024). Para mejorar la formación del profesorado de matemáticas en primaria. *EL PAÍS*.
- Daskalaki, S., Birbas, T., & Housos, E. C. (2004). *An integer programming formulation for a case study in university timetabling*. Patras.
- Dimopoulou, M., & Miliotis, P. A. (2001). *Implementation of a university course and examination timetabling system*. Atenas.
- Miguel Castaño, A. d., Piattini Velthuis, M., & Marcos Martínez, E. (2000). *Diseño de bases de datos relacionales*. Madrid: Ra-Ma.
- Nagata, Y. (2018). *Random partial neighborhood search for the post-enrollment course timetabling problem*. Tokushima.
- Oliveira, R. (2021). *5 porqués: herramienta de análisis y solución de problemas*. Independently Published.
- Rincón, C. A., & Villarreal Vásquez, F. (2009). *Costos. Decisiones empresariales*. Ecoe Ediciones.
- Sörensen, K., & Glover, F. W. (2013). *Metaheuristics*.
- Saaty, T. L. (1980). *The Analytic Hierarchy Process*. Nueva York.
- Sanmartín, O. R. (6 de Febrero de 2024). El 81% de los jóvenes en edad de hacer Selectividad pide un examen de acceso a la universidad común en toda España. *EL MUNDO*.
- Sociedad, E. P. (19 de Febrero de 2024). La profesora que reunió 70.000 firmas para prohibir los móviles en menores se reúne mañana en el Congreso con 5 partidos. *Europa Press*.
- Toskano, H., & Gérard, B. (2005). *El Proceso de análisis jerárquico (AHP) como herramienta para la toma de decisiones en la selección de proveedores*. Lima.

PRESUPUESTO

1. CONSIDERACIONES GENERALES

Este documento integra el presupuesto del proyecto “Diseño de una herramienta mediante programación matemática para la gestión y organización del horario anual de una escuela de secundaria en Valencia”. Es por ello que, lo que englobará la mayor parte de este presupuesto serán los costes asociados al personal participante en la ejecución de este trabajo.

Por lo que respecta al presupuesto, para la ejecución del mismo se han tenido en cuenta los siguientes puntos:

- **Gastos Generales (GG):** significarán el 13% del Presupuesto de Ejecución Material (P.E.M). Dentro de estos gastos estarán contabilizados los siguientes términos:
 - Material de mantenimiento y oficina
 - Electricidad
 - Teléfono
 - Licencias de softwares
 - Equipos informáticos
 - Encuadernaciones
- **Costes Indirectos:** estos costes son los que no pueden ser contabilizados en una o varias unidades de obra específicas, ya que pueden afectar a muchas partidas o incluso al proyecto en su conjunto total. Es por ello que, este coste se representará en cada unidad de obra como el 1% de la misma.
- **Beneficio Industrial (BI):** este representa, según el Real Decreto 1098/2001, el 6% respecto del total del P.E.M.
- **IVA:** en España está establecido como el 21% del Presupuesto de Ejecución por Contrata (P.E.C), y este está denominado como el Impuesto sobre el Valor Añadido.

Además de esto, es necesario contabilizar el coste de mano de obra que ha sido participante de esto proyecto:

Personal Participante	Coste Mano de Obra (€/h)
Ingeniero Industrial Junior	30 €/h
Ingeniero Industrial Senior	60 €/h

2. PRESUPUESTOS PARCIALES

2.1. DESGLOSE POR CAPÍTULOS

N.º	Código Ud.	Descripción			
Total					
1. Descripción del entorno					
1.1	P01.01	Introducción			
	1,00	h	Ingeniero Industrial Junior	30,00 €	30,00 €
			1,000 % Costes indirectos	30,00 €	0,30 €
			Coste total		<u>30,30 €</u>
1.2	P01.02	Estructura del sistema educativo			
	1,00	h	Ingeniero Industrial Junior	30,00 €	30,00 €
			1,000 % Costes indirectos	30,00 €	0,30 €
			Coste total		<u>30,30 €</u>
1.3	P01.03	Leyes educativas			
	1,00	h	Ingeniero Industrial Junior	30,00 €	30,00 €
			1,000 % Costes indirectos	30,00 €	0,30 €
			Coste total		<u>30,30 €</u>
1.4	P01.04	Estructura de la institución educativa			
	1,00	h	Ingeniero Industrial Junior	30,00 €	30,00 €
			1,000 % Costes indirectos	30,00 €	0,30 €
			Coste total		<u>30,30 €</u>
1.5	P01.05	Cursos escolares			
	1,00	h	Ingeniero Industrial Junior	30,00 €	30,00 €
			1,000 % Costes indirectos	30,00 €	0,30 €
			Coste total		<u>30,30 €</u>
1.6	P01.06	Profesorado			
	1,00	h	Ingeniero Industrial Junior	30,00 €	30,00 €
			1,000 % Costes indirectos	30,00 €	0,30 €
			Coste total		<u>30,30 €</u>

2. PRESUPUESTOS PARCIALES

N.º	Código Ud.	Descripción		
Total				
2. Antecedentes históricos				
2.1	P02.01	Sistemas de información y bases de datos		
	2,00	h Ingeniero Industrial Junior	30,00 €	60,00 €
		1,000 % Costes indirectos	60,00 €	0,60 €
		Coste total		<u>60,60 €</u>
2.2	P02.02	Métodos y técnicas de resolución de problemas		
	3,00	h Ingeniero Industrial Junior	30,00 €	90,00 €
		1,000 % Costes indirectos	90,00 €	0,45 €
		Coste total		<u>90,90 €</u>
2.3	P02.03	Métodos para resolver la gestión de horarios escolares		
	1,00	h Ingeniero Industrial Junior	30,00 €	30,00 €
		1,000 % Costes indirectos	30,00 €	0,30 €
		Coste total		<u>30,30 €</u>



Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

N.º	Código Ud.	Descripción			
Total					
3. Análisis de la situación actual					
3.1	P03.01	Descripción de incidencias			
	1,00	h Ingeniero Industrial Junior		30,00 €	30,00 €
		1,000 % Costes indirectos		30,00 €	0,30 €
		Coste total			<u>30,30 €</u>
3.2	P03.02	Identificación de las causas			
	2,00	h Ingeniero Industrial Junior		30,00 €	60,00 €
		1,000 % Costes indirectos		60,00 €	0,60 €
		Coste total			<u>60,60 €</u>
3.3	P03.03	Análisis ES / NO ES y causa-raíz			
	3,00	h Ingeniero Industrial Junior		30,00 €	90,00 €
		1,000 % Costes indirectos		90,00 €	0,90 €
		Coste total			<u>90,90 €</u>
3.4	P03.04	Diseño de requerimientos			
	1,00	h Ingeniero Industrial Junior		30,00 €	30,00 €
		1,000 % Costes indirectos		30,00 €	0,30 €
		Coste total			<u>30,30 €</u>

N.º	Código Ud.	Descripción			
Total					
4. Selección de alternativas					
4.1	P04.01	Criterios de evaluación			
	1,00	h	Ingeniero Industrial Junior	30,00 €	30,00 €
			1,000 % Costes indirectos	30,00 €	0,30 €
			Coste total		30,30 €
4.2	P04.02	Método AHP			
	2,00	h	Ingeniero Industrial Junior	30,00 €	60,00 €
			1,000 % Costes indirectos	60,00 €	0,60 €
			Coste total		60,60 €
4.3	P04.03	Introducción de posibles alternativas			
	3,00	h	Ingeniero Industrial Junior	30,00 €	90,00 €
			1,000 % Costes indirectos	90,00 €	0,90 €
			Coste total		90,90 €
4.4	P04.04	Elección final de la alternativa			
	5,00	h	Ingeniero Industrial Junior	30,00 €	150,00 €
			1,000 % Costes indirectos	150,00 €	1,50 €
			Coste total		151,50 €



Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

N.º	Código Ud.	Descripción			
Total					
5. Modelado					
5.1	P05.01	Índices			
	2,00	h	Ingeniero Industrial Junior	30,00 €	60,00 €
			1,000 % Costes indirectos	60,00 €	0,60 €
			Coste total		<u>60,60 €</u>
5.2	P05.02	Parámetros y variables de decisión			
	10,00	h	Ingeniero Industrial Junior	30,00 €	300,00 €
	2,00	h	Ingeniero Industrial Senior	60,00 €	120,00 €
			1,000 % Costes indirectos	410,00 €	4,20 €
			Coste total		<u>414,20 €</u>
5.3	P05.03	Función objetivo			
	1,00	h	Ingeniero Industrial Junior	30,00 €	30,00 €
			1,000 % Costes indirectos	30,00 €	0,30 €
			Coste total		<u>30,30 €</u>
5.4	P05.04	Restricciones y dominios			
	20,00	h	Ingeniero Industrial Junior	30,00 €	600,00 €
	5,00	h	Ingeniero Industrial Senior	60,00 €	300,00 €
			1,000 % Costes indirectos	900,00 €	9,00 €
			Coste total		<u>909,00 €</u>

2. PRESUPUESTOS PARCIALES

N.º	Código Ud.	Descripción			
Total					
6. Análisis y elección de las características de la solución					
6.1	P06.01	Análisis y elección del lenguaje de programación			
	3,00	h	Ingeniero Industrial Junior	30,00 €	90,00 €
			1,000 % Costes indirectos	90,00 €	0,90 €
			Coste total		<u>90,90 €</u>
6.2	P06.02	Análisis y elección del Solver			
	6,00	h	Ingeniero Industrial Junior	30,00 €	180,00 €
			1,000 % Costes indirectos	180,00 €	1,80 €
			Coste total		<u>181,80 €</u>
6.3	P06.03	Elección del programa de importación y exportación de datos			
	2,00	h	Ingeniero Industrial Junior	30,00 €	60,00 €
			1,000 % Costes indirectos	60,00 €	0,60 €
			Coste total		<u>60,60 €</u>
6.4	P06.04	Elección final del software			
	1,00	h	Ingeniero Industrial Junior	30,00 €	30,00 €
			1,000 % Costes indirectos	30,00 €	0,30 €
			Coste total		<u>30,30 €</u>



Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

N.º	Código Ud.	Descripción			
Total					
7. Diseño de la aplicación y la herramienta					
7.1	P07.01	Metodología empleada			
	5,00	h	Ingeniero Industrial Junior	30,00 €	150,00 €
			1,000 % Costes indirectos	150,00 €	1,50 €
			Coste total		<u>151,50 €</u>
7.2	P07.02	Diseño de la interfaz de la aplicación			
	30,00	h	Ingeniero Industrial Junior	30,00 €	900,00 €
	5,00	h	Ingeniero Industrial Senior	60,00 €	300,00 €
			1,000 % Costes indirectos	1200,00 €	12,00 €
			Coste total		<u>1212,00 €</u>
7.3	P07.03	Herramienta para la creación de horarios			
	80,00	h	Ingeniero Industrial Junior	30,00 €	2400,00 €
	10,00	h	Ingeniero Industrial Senior	60,00 €	600,00 €
			1,000 % Costes indirectos	3000,00 €	30,00 €
			Coste total		<u>3030,00 €</u>

2. PRESUPUESTOS PARCIALES

N.º	Código Ud.	Descripción			
Total					
8. Validación de la herramienta sobre un caso real					
8.1	P08.01	Metodología empleada			
	2,00	h	Ingeniero Industrial Junior	30,00 €	60,00 €
			1,000 % Costes indirectos	60,00 €	0,60 €
			Coste total		<u>60,60 €</u>
8.2	P08.02	Validación de la herramienta			
	30,00	h	Ingeniero Industrial Junior	30,00 €	900,00 €
	10,00	h	Ingeniero Industrial Senior	60,00 €	600,00 €
			1,000 % Costes indirectos	1500,00 €	15,00 €
			Coste total		<u>1515,00 €</u>
8.3	P08.03	Análisis de los resultados			
	10,00	h	Ingeniero Industrial Junior	30,00 €	300,00 €
			1,000 % Costes indirectos	300,00 €	3,00 €
			Coste total		<u>303,00 €</u>

Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia**2.2. SÍNTESIS POR CAPÍTULOS****4. Descripción del entorno**

N. º	Descripción		Medición	Precio	Importe
1.1	Introducción				
		Total:	1,000	30,30 €	30,30 €
1.2	Estructura del sistema educativo				
		Total:	1,000	30,30 €	30,30 €
1.3	Leyes educativas				
		Total:	1,000	30,30 €	30,30 €
1.4	Estructura de la institución educativa				
		Total:	1,000	30,30 €	30,30 €
1.5	Cursos escolares				
		Total:	1,000	30,30 €	30,30 €
1.6	Profesorado				
		Total:	1,000	30,30 €	30,30 €
Parcial 1: Descripción del entorno					181,80 €

2. PRESUPUESTOS PARCIALES

5. Antecedentes históricos

N. º	Descripción	Medición	Precio	Importe
2.1	Sistemas de información y bases de datos			
	Total:	1,000	60,60 €	60,60 €
2.2	Métodos y técnicas de resolución de problemas			
	Total:	1,000	90,90 €	90,90 €
2.3	Métodos para resolver la gestión de horarios escolares			
	Total:	1,000	30,30 €	30,30 €
Parcial 2: Antecedentes históricos				181,80 €

Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

6. Análisis de la situación actual

N. o	Descripción		Medición	Precio	Importe
3.1	Descripción de incidencias				
		Total:	1,000	30,30 €	30,30 €
3.2	Identificación de las causas				
		Total:	1,000	60,60 €	60,60 €
3.3	Análisis ES / NO ES y causa-raíz				
		Total:	1,000	90,90 €	90,90 €
3.4	Diseño de requerimientos				
		Total:	1,000	30,30 €	30,30 €
Parcial 3: Análisis de la situación actual					212,10 €

2. PRESUPUESTOS PARCIALES

7. Selección de alternativas

N. º	Descripción		Medición	Precio	Importe
4.1	Criterios de evaluación				
		Total:	1,000	30,30 €	30,30 €
4.2	Método AHP				
		Total:	1,000	60,60 €	60,60 €
4.3	Introducción de posibles alternativas				
		Total:	1,000	90,90 €	90,90 €
4.4	Elección final de la alternativa				
		Total:	1,000	151,50 €	151,50 €
Parcial 4: Selección de alternativas					333,30 €

Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

8. Modelado

N. º	Descripción		Medición	Precio	Importe
5.1	Índices				
		Total:	1,000	60,60 €	60,60 €
5.2	Parámetros y variables de decisión				
		Total:	1,000	424,20 €	424,20 €
5.3	Función objetivo				
		Total:	1,000	30,30 €	30,30 €
5.4	Restricciones y dominios				
		Total:	1,000	909,00 €	909,00 €
Parcial 5: Modelado					1424,10 €

2. PRESUPUESTOS PARCIALES

9. Análisis y elección de las características de la solución

N. º	Descripción	Medición	Precio	Importe
6.1	Análisis y elección del lenguaje de programación			
	Total:	1,000	90,90 €	90,90 €
6.2	Análisis y elección del Solver			
	Total:	1,000	181,80 €	181,80 €
6.3	Elección del programa de importación y exportación de datos			
	Total:	1,000	60,60 €	60,60 €
6.4	Elección final del software			
	Total:	1,000	30,30 €	30,30 €
Parcial 6: Análisis y elección de las características de la solución				363,60 €



Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

10. Diseño de la aplicación y la herramienta

N. o	Descripción	Medición	Precio	Importe
7.1	Metodología empleada			
	Total:	1,000	151,5 €	151,50 €
7.2	Diseño de la interfaz de la aplicación			
	Total:	1,000	1212,00 €	1212,00 €
7.3	Herramienta para la creación de horarios			
	Total:	1,000	3030,00 €	3030,00 €
Parcial 7: Diseño de la aplicación y la herramienta				4393,50 €

2. PRESUPUESTOS PARCIALES

11. Validación de la herramienta sobre el caso real

N. o	Descripción	Medición	Precio	Importe
8.1	Metodología empleada			
	Total:	1,000	60,60 €	60,60 €
8.2	Validación de la herramienta			
	Total:	1,000	1515,00 €	1515,00 €
8.3	Análisis de los resultados			
	Total:	1,000	303,30 €	303,30 €
Parcial 8: Validación de la herramienta sobre el caso real				1878,90 €



Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

Presupuesto de Ejecución Material

1. Descripción del entorno	181,80 €
2. Antecedentes teóricos	181,80 €
3. Análisis de la situación actual	212,10 €
4. Selección de alternativas	333,33 €
5. Modelado	1424,10 €
6. Análisis y elección de las características de la solución	363,60 €
7. Diseño de la aplicación y la herramienta	4393,50 €
8. Validación de la herramienta sobre el caso real	1878,90 €
Total:	10.848,03 €

El Presupuesto de Ejecución Material asciende a la cantidad de DIEZ MIL OCHOCIENTOS CUARENTA Y OCHO EUROS CON TRES CÉNTIMOS.

3. RESUMEN DEL PRESUPUESTO

Presupuesto de Ejecución por Contrata

1. Descripción del entorno	181,10 €
2. Antecedentes teóricos	181,10 €
3. Análisis de la situación actual	212,10 €
4. Selección de alternativas	333,33 €
5. Modelado	1424,10 €
6. Análisis y elección de las características de la solución	363,60 €
7. Diseño de la aplicación y la herramienta	4393,50 €
8. Validación de la herramienta sobre el caso real	1878,90 €
Presupuesto de Ejecución Material (PEM)	10.848,03 €
13% de gastos generales	1.410,25 €
6% de beneficio industrial	650,89 €
Presupuesto de Ejecución por Contrata (PEC = PEM + GG + BI)	11.640,17 €
21% de IVA	2.444,44 €
Presupuesto de Ejecución por Contrata con IVA	14.084,61 €

El Presupuesto de Ejecución por Contrata con IVA asciende a la cantidad de CATORCE MIL OCHENTA Y CUATRO EUROS CON SESENTA Y UN CÉNTIMOS.

4. ANÁLISIS DE VIABILIDAD ECONÓMICA

4.1. INTRODUCCIÓN

El objetivo de este apartado es analizar la viabilidad económica del uso de la herramienta de creación de horarios, en comparación con el método manual (utilizado actualmente en el caso real de estudio), y calcular el retorno de inversión (payback period).

4.2. ESTIMACIÓN DE AHORROS EN TIEMPO

Actualmente, la elaboración manual del horario involucra principalmente al jefe de estudios. Este proceso puede tomar semanas de planificación y ajustes, estimando que, en promedio, se dedican unas 50 horas al año en la construcción de estos horarios.

- **Horas dedicadas manualmente:** 50 horas.
- **Costo promedio por hora del jefe de estudios** (asumiendo un salario de 40€/hora, valor estimado):

$$\text{Coste promedio por hora} = 40\text{€}$$

$$\text{Coste total anual} = 50 \text{ horas} \times 40 \text{ €/hora} = 2000\text{€}$$

Con la implementación de la herramienta, el tiempo estimado para la creación del horario podría reducirse a tan solo 20 horas anuales, ya que gran parte del proceso se automatizaría, únicamente habría que validar la ejecución herramienta y analizar si son adecuados o no los resultados. Esto supondría un ahorro de 80 horas anuales.

- **Horas dedicadas a la herramienta:** 20 horas.
- **Coste promedio por hora del jefe de estudios:**

$$\text{Coste promedio por hora} = 40\text{€}$$

$$\text{Coste total anual con la herramienta} = 20 \text{ horas} \times 40 \text{ €/hora} = 800\text{€}$$

4.3. CÁLCULO DEL PAYBACK PERIOD

Si el coste de desarrollo e implementación estimado de la herramienta es de unos 3000€, el ahorro anual estimado generado por la herramienta será:

$$\text{Ahorro anual} = 2000 - 800 = 1200\text{€}$$

El payback period se calcularía de la siguiente manera:

$$\text{Payback period} = \frac{\text{Coste total de la herramienta}}{\text{Ahorro anual}} = \frac{3000\text{€}}{\frac{1200\text{€}}{\text{año}}} = 2,5 \text{ años}$$

Por lo tanto, el período de retorno de la inversión sería de 2,5 años.

4.4. IMPACTO EN LA SATISFACCIÓN DEL ALUMNADO

Aunque este aspecto no puede cuantificarse económicamente, la mejora en la planificación y la equidad de los horarios podría aumentar la satisfacción de los alumnos. Al contar con horarios más eficientes, los estudiantes podrán gestionar mejor su tiempo, lo que podría generar un impacto positivo en su bienestar y rendimiento académico. Esto no tiene un valor económico directo, pero es importante considerar el impacto cualitativo en el aprendizaje y la experiencia general de los estudiantes.

4.5. CONCLUSIONES

El uso de esta herramienta no solo genera ahorros significativos en tiempo y costes para el responsable de la planificación, sino que también podría mejorar la satisfacción de los alumnos, lo cual puede contribuir indirectamente a un mejor rendimiento académico.

ANEXOS

ANEXO 1. CODIFICACIÓN COMPLETA DE LA HERRAMIENTA EN PYTHON

Tal y como se ha dicho a lo largo de los capítulos referentes a la herramienta, en este se anexo se adjunta la codificación completa de los modelos de programación matemática realizados para la consecución de la herramienta diseñada.

Modelo 1

```
import pulp as pl
from pulp import LpProblem, LpVariable, lpSum, LpMaximize, LpStatus,
value, GUROBI_CMD
import pandas as pd
from openpyxl.styles import Border, Side, Font, Alignment
from openpyxl.utils import get_column_letter

solver_list = pl.listSolvers()
print(solver_list)

solver_list = pl.listSolvers(onlyAvailable=True)
print(solver_list)

# ÍNDICES

I = ["I1", "I2", "I3", "I4", "I5", "I6", "I7"]

I_O = ["I1", "I7"]

D = ["L", "M", "X", "J", "V"]

G = ["1A", "1B", "2A", "2B", "3A", "3B", "4A", "4B", "1ART", "1CI",
"1HUM", "2ART", "2CI", "2HUM"]

# Función para leer y cargar los datos de los índices E (Específicas),
O (Optativas) y K (Profesores)
def leer_listado_especificas(BaseDatos):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='E', header=None)
    # Buscar el índice de la fila donde están las asignaturas
    indice_especificas = df[df[0] == 'Código'].index[0] + 1
    # Leer la columna de asignaturas desde ese índice hacia abajo
    especificas = df.loc[indice_especificas:, 0].tolist()
    return especificas

def leer_listado_optativas(BaseDatos):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='O', header=None)
    # Buscar el índice de la fila donde están las asignaturas
    indice_optativas = df[df[0] == 'Código'].index[0] + 1
```

Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

```
# Leer la columna de asignaturas desde ese índice hacia abajo
optativas = df.loc[indice_optativas:, 0].tolist()
return optativas

def leer_listado_profesores(BaseDatos):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='Profesores', header=None)
    # Buscar el índice de la fila donde están los profesores
    indice_profesores = df[df[0] == 'ID'].index[0] + 1
    # Leer la columna de profesores desde ese índice hacia abajo
    profesores = df.loc[indice_profesores:, 0].tolist()
    return profesores

# PARÁMETROS
# Parámetro HSe. Número de horas semanales que deben impartirse para
# el grupo g del bloque de asignaturas específicas e
def leer_horas_semanales_especificas(BaseDatos, G, E):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='HSbe', index_col=0, header=0)
    # Limpiar espacios en blanco de columnas e índice
    df.columns = df.columns.str.strip()
    df.index = df.index.str.strip()
    HSe = {}
    for e in E:
        for g in G:
            if g in df.columns and e in df.index:
                valor_horas = df.at[e, g]
                HSe[(e, g)] = valor_horas
            else:
                print(f"Advertencia: Grupo '{g}' o bloque '{e}' no
encontrado en el DataFrame HSe")
    return HSe

# Parámetro HSo. Número de horas semanales que deben impartirse para
# el grupo g de optativas o
def leer_horas_semanales_optativas(BaseDatos, G, O):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='HSbo', index_col=0, header=0)

    # Limpiar espacios en blanco de columnas e índice
    df.columns = df.columns.str.strip()
    df.index = df.index.str.strip()

    HSo = {}
    for o in O:
        for g in G:
            if g in df.columns and o in df.index:
                valor_horas = df.at[o, g]
                HSo[(o, g)] = valor_horas
            else:
                print(f"Advertencia: Grupo '{g}' o optativa '{o}' no
encontrado en el DataFrame HSo")
    return HSo

# Parámetro CPe. Parámetro binario que toma valor 1 en caso de que el
# profesor k pueda impartir
```

ANEXO 1. CODIFICACIÓN COMPLETA DE LA HERRAMIENTA EN PYTHON

```
# el bloque de asignaturas específicas e, toma valor 0 si no está
capacitado
def leer_datos_CPe(BaseDatos, E, K):
    xls = pd.ExcelFile(BaseDatos)

    # Leer los datos de la hoja CPe
    df_cpe = pd.read_excel(xls, sheet_name='CPbe', index_col=0,
header=0)

    # Inicializar el diccionario CPe
    CPe = {}

    # Iterar sobre cada profesor y bloque de específicas para obtener
el valor binario correspondiente
    for profesor in K:
        for especifica in E:
            # Obtener el valor de la celda correspondiente
            valor_celda = df_cpe.at[profesor, especifica]
            # Asignar el valor binario al diccionario CPe
            CPe[(profesor, especifica)] = valor_celda

    return CPe

# Parámetro CPo. Parámetro binario que toma valor 1 en caso de que el
profesor k pueda impartir
# el bloque de asignaturas optativas o, toma valor 0 si no está
capacitado
def leer_datos_CPo(BaseDatos, O, K):
    xls = pd.ExcelFile(BaseDatos)

    # Leer los datos de la hoja CPo
    df_cpo = pd.read_excel(xls, sheet_name='CPbo', index_col=0,
header=0)

    # Inicializar el diccionario CP
    CPo = {}

    # Iterar sobre cada profesor y bloque de optativas para obtener el
valor binario correspondiente
    for profesor in K:
        for optativa in O:
            # Obtener el valor de la celda correspondiente
            valor_celda = df_cpo.at[profesor, optativa]
            # Asignar el valor binario al diccionario CPo
            CPo[(profesor, optativa)] = valor_celda

    return CPo

# Ruta del archivo Excel
BaseDatos2 = '/Users/alexgarcia/Desktop/TFM/Base de
datos/BaseDatos2.xlsx'

# Leer y guardar los datos en listas
E = leer_listado_especificas(BaseDatos2)
O = leer_listado_optativas(BaseDatos2)
K = leer_listado_profesores(BaseDatos2)
HSe = leer_horas_semanales_especificas(BaseDatos2, G, E)
```

Diseño de una herramienta mediante programación matemática para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

```
HSo = leer_horas_semanales_optativas(BaseDatos2, G, O)
CPe = leer_datos_CPe(BaseDatos2, E, K)
CPo = leer_datos_CPo(BaseDatos2, O, K)

# Impresiones para verificar
print("Específicas (E):", E)
print("Optativas (O):", O)
print("Profesores (K):", K)
print("Horas semanales específicas (HSe):", HSe)
print("Horas semanales optativas (HSo):", HSo)
print("Parámetro CPe:", CPe)
print("Parámetro CPo:", CPo)

# Parámetro CS. Coste de satisfacción del profesor k en la franja
horaria i del día de la semana d
def procesar_excel_a_diccionarios(Horario_Docentes):
    # Leer el archivo de Excel
    xl = pd.ExcelFile(Horario_Docentes, engine='openpyxl')
    # Diccionario que contendrá todos los horarios por persona
    CS = {}
    # Procesar cada hoja del archivo de Excel
    for hoja in xl.sheet_names:
        # Leer la hoja actual en un DataFrame
        df = xl.parse(hoja)
        # Eliminar filas donde 'día de la semana' es NaN
        df.dropna(subset=[df.columns[0]], inplace=True)
        coste_profesor = {} # Diccionario para el profesor actual
        # Obtener los nombres de las columnas (intervalos de horas)
        intervalos_horas = df.columns[1:] # Excluir la primera
columna de días
        # Iterar sobre cada fila del DataFrame
        for index, row in df.iterrows():
            dia_semana = row[0] # El día de la semana está en la
primera columna
            if pd.isna(dia_semana):
                continue # Saltar filas donde el día de la semana es
NaN
            # Iterar sobre cada intervalo de horas
            for hora in intervalos_horas:
                valor_entero = row[hora]
                if pd.isna(valor_entero):
                    continue # Saltar valores binarios NaN
                # Crear la clave del diccionario
                clave = (hoja, hora, dia_semana)
                # Asignar el valor correspondiente
                coste_profesor[clave] = valor_entero
        # Agregar el diccionario de la persona al diccionario general
        CS.update(coste_profesor)

    return CS

# Ejemplo de uso
Horario_Docentes =
'/Users/alexgarcia/Desktop/TFM/Horario_Docentes.xlsx'
CS = procesar_excel_a_diccionarios(Horario_Docentes)
print(CS)
```

ANEXO 1. CODIFICACIÓN COMPLETA DE LA HERRAMIENTA EN PYTHON

```
# VARIABLES DE DECISIÓN
# Variable Z. Variable binaria que toma valor 1 en caso de que para el
grupo g en la franja horaria i
# se asigna el bloque de específicas e impartida por el profesor k en
el día d, 0 en caso contrario
Z = LpVariable.dicts("Z", (G, I, E, K, D), cat='Binary')

# Variable T. Variable binaria que toma valor 1 en caso de que para el
grupo g en la franja horaria i
# se asigna el bloque de optativas o impartida por el profesor k en el
día d, 0 en caso contrario
T = LpVariable.dicts("T", (G, I, O, K, D), cat='Binary')

# Variable V. Variable binaria que toma valor 1 en caso de que para el
grupo g
# se imparte el bloque de específicas e por el profesor k, 0 en caso
contrario
V = LpVariable.dicts("V", (G, E, K), cat='Binary')

# Variable W. Variable binaria que toma valor 1 en caso de que para el
grupo g
# se imparte el bloque de optativas o por el profesor k, 0 en caso
contrario
W = LpVariable.dicts("W", (G, O, K), cat='Binary')

# Definir el problema como de maximización
prob = LpProblem("Horario", LpMaximize)

# FO: Maximizar la satisfacción de las preferencias horarias de los
profesores
prob += (
    lpSum(Z[g][i][e][k][d] * CS.get((k, i, d), 0) for g in G for i in
I for e in E for k in K for d in D) +
    lpSum(T[g][i][o][k][d] * CS.get((k, i, d), 0) for g in G for i in
I for o in O for k in K for d in D)
)

# RESTRICCIONES

# 1. Solamente un profesor único k puede impartir la específica e al
grupo g
for g in G:
    for e in E:
        prob += lpSum(V[g][e][k] for k in K) == 1

# 2. Solamente un profesor único k puede impartir la optativa o al
grupo g
for g in G:
    for o in O:
        prob += lpSum(W[g][o][k] for k in K) == 1

# 3. Asignación específica limitada a profesores capacitados
for g in G:
    for e in E:
        for k in K:
            if CPe[(k, e)] == 0:
                prob += V[g][e][k] == 0
```

Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

```
# 4. Asignación optativa limitada a profesores capacitados
for g in G:
    for o in O:
        for k in K:
            if CPo[(k, o)] == 0:
                prob += W[g][o][k] == 0

# 5. En caso de que un profesor k esté asignado a la específica e del
# grupo g en cualquier momento, es el responsable
for g in G:
    for e in E:
        for k in K:
            prob += lpSum(Z[g][i][e][k][d] for i in I for d in D) ==
HSe.get((e, g), 0) * V[g][e][k]

# 6. En caso de que un profesor k esté asignado a la optativa o del
# grupo g en cualquier momento, es el responsable
for g in G:
    for o in O:
        for k in K:
            prob += lpSum(T[g][i][o][k][d] for i in I for d in D) ==
HSo.get((o, g), 0) * W[g][o][k]

# 7. Para un grupo g la específica e en una franja horaria i
# puede impartirse como máximo una vez en un día d
for g in G:
    for e in E:
        for d in D:
            prob += lpSum(Z[g][i][e][k][d] for i in I for k in K) <= 1

# 8. Para un grupo g la optativa o en una franja horaria i puede
# impartirse como máximo una vez en un día d
for g in G:
    for o in O:
        for d in D:
            prob += lpSum(T[g][i][o][k][d] for i in I for k in K) <= 1

# 9. Todas las horas semanales específicas deben impartirse
for g in G:
    for e in E:
        prob += lpSum(Z[g][i][e][k][d] for i in I for k in K for d in
D) == HSe.get((e, g), 0)

# 10. Todas las horas semanales optativas deben impartirse
for g in G:
    for o in O:
        prob += lpSum(T[g][i][o][k][d] for i in I for k in K for d in
D) == HSo.get((o, g), 0)

# 11. Cada grupo g solo puede tener una específica e o una optativa o
# en una franja horaria i
for g in G:
    for i in I:
        for d in D:
            prob += (
                lpSum(Z[g][i][e][k][d] for e in E for k in K) +
```

ANEXO 1. CODIFICACIÓN COMPLETA DE LA HERRAMIENTA EN PYTHON

```
        lpSum(T[g][i][o][k][d] for o in O for k in K)
        ) <= 1
# 12. Un profesor k solo puede impartir una asignatura j, una
# específica e o una optativa o
# en una franja horaria i de un día d
for i in I:
    for k in K:
        for d in D:
            prob += (
                lpSum(Z[g][i][e][k][d] for g in G for e in E) +
                lpSum(T[g][i][o][k][d] for g in G for o in O)
            ) <= 1

# 13. Las optativas o solo pueden impartirse en la primera o en la
# última franja horaria i
for g in G:
    for o in O:
        for k in K:
            for d in D:
                for i in I:
                    if i not in I_o:
                        prob += T[g][i][o][k][d] == 0

# Crear una instancia del solucionador GUROBI
gurobi_solver = GUROBI_CMD()

import time

StartTim = time.time()
print("Start time")

# Optimización
status = prob.solve(gurobi_solver)

FinishTime = time.time()
print("Finish time")

# Mostrar el resultado
print("Estado:", LpStatus[prob.status])
print("Horario óptimo:")

# Imprimir los horarios de cada profesor
for k in K:
    print(f"\n{k}:")
    for d in D:
        print(f"-{d}:")
        for i in I:
            for g in G:
                for e in E:
                    if value(Z[g][i][e][k][d]) == 1:
                        print(f" {i}: {g}, {e}")
            for o in O:
                if value(T[g][i][o][k][d]) == 1:
                    print(f" {i}: {g}, {o}")

# Imprimir los horarios de cada grupo
for g in G:
```


Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

```

print(f"Grupo: {g}")
for d in D:
    for i in I:
        for e in E:
            for k in K:
                if value(Z[g][i][e][k][d]) == 1:
                    print(f" - {d}, {i}: {e}")
        for o in O:
            for k in K:
                if value(T[g][i][o][k][d]) == 1:
                    print(f" - {d}, {i}: {o}")

print("Nivel de satisfacción:", value(prob.objective))

# Inicializar el diccionario para recoger los horarios de cada
profesor adecuadamente
horarios_profesor = {prof: {i: {d: "" for d in D} for i in I} for g in
G for prof in K}

# Llenar los horarios con la información de la solución
for g in G: # Debes iniciar con el grupo para tener acceso a las
asignaturas y profesores específicos de ese grupo
    for k in K:
        for d in D:
            for i in I:
                for e in E:
                    # Asegurarse que el acceso a la variable de
decisión es correcto
                    # y que el profesor 'k' enseña 'e' en 'g'
                    if Z[g][i][e][k][d].varValue == 1:
                        # Añadir al horario del profesor 'k' que
enseña en 'g',
                        # en el día 'd', franja 'i', el bloque
específico 'e'
                        if f"{e} ({g})" not in
horarios_profesor[k][i][d]: # Verificar que no se añada duplicados
horarios_profesor[k][i][d] += f"{e}
({g})\n"
                for o in O:
                    # Asegurarse que el acceso a la variable de
decisión es correcto
                    # y que el profesor 'k' enseña 'o' en 'g'
                    if T[g][i][o][k][d].varValue == 1:
                        # Añadir al horario del profesor 'k' que
enseña en 'g',
                        # en el día 'd', franja 'i', el bloque
optativo 'o'
                        if f"{o} ({g})" not in
horarios_profesor[k][i][d]: # Verificar que no se añada duplicados
horarios_profesor[k][i][d] += f"{o}
({g})\n"

# Guardar los horarios de cada profesor en un archivo Excel
with pd.ExcelWriter(Horario_Docentes, engine='openpyxl', mode='a') as
writer:

    for k, horario in horarios_profesor.items():

```

ANEXO 1. CODIFICACIÓN COMPLETA DE LA HERRAMIENTA EN PYTHON

```
# Obtener la hoja correspondiente al profesor (K1, K2, K3,
...))
if f"{k}" in writer.book.sheetnames:
    sheet = writer.book[f"{k}"]
else:
    sheet = writer.book.create_sheet(title=f"K{k}")

# Ajustar el ancho de las columnas y el tamaño de la fuente
for col in range(1, len(D) + 2):
    col_letter = get_column_letter(col)
    sheet.column_dimensions[col_letter].width = 14 # Ancho de
columna

    for cell in sheet[col_letter]:
        cell.font = Font(size=14) # Tamaño de la fuente

# Escribir los días de la semana en las celdas B8 a H8
for col, day in enumerate(D, start=2):
    sheet.cell(row=8, column=col, value=day).font =
Font(bold=True, size=14) # Negrita y Fuente 14
    sheet.cell(row=8, column=col).alignment =
Alignment(horizontal='center') # Centrado

# Escribir las franjas horarias en las celdas A9 a A13
for row, time_slot in enumerate(I, start=9):
    sheet.cell(row=row, column=1, value=time_slot).font =
Font(bold=True, size=14) # Negrita y Fuente 14
    sheet.cell(row=row, column=1).alignment =
Alignment(horizontal='center') # Centrado

# Escribir el horario en la hoja (ajustado para las nuevas
posiciones)
for row, time_slot in enumerate(I, start=9):
    for col, day in enumerate(D, start=2):
        cell = sheet.cell(row=row, column=col,
value=horario[time_slot][day])
        cell.alignment = Alignment(horizontal='center') #
Centrado

        cell.font = Font(size=14) # Tamaño de la fuente

# Bordes de todas las celdas
border = Border(left=Side(style='thin', color='000000'),
                right=Side(style='thin', color='000000'),
                top=Side(style='thin', color='000000'),
                bottom=Side(style='thin', color='000000'))

# Aplicar bordes a todas las celdas en el rango ajustado
for row in sheet.iter_rows(min_row=8, max_row=15, min_col=1,
max_col=len(D) + 1):
    for cell in row:
        cell.border = border

# Abrir automáticamente el archivo Excel
import os

# Ruta del archivo Excel
ruta_excel = '/Users/alexgarcia/Desktop/TFM/Horario_Docentes.xlsx'
```

Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

```
# Verificar si el archivo existe
if os.path.exists(ruta_excel):
    # Abrir el archivo Excel con el programa predeterminado
    os.system(f'open "{ruta_excel}"')
else:
    print("El archivo Excel no se encontró en la ruta especificada.")

# Crear una estructura de datos para representar el horario por grupo
horarios_grupo = {g: {i: {d: "" for d in D} for i in I} for g in G}

# Llenar los horarios con la información de la solución
for g in G: # Inicia con el grupo para acceder correctamente a las
asignaturas y profesores de ese grupo
    for d in D:
        for i in I:
            for e in E:
                for k in K:
                    # Asegúrate de que el acceso a las variables de
decisión sea correcto
                    if Z[g][i][e][k][d].varValue == 1:
                        # Añadir el bloque específico 'e' al horario
del grupo 'g' en el día 'd' y franja 'i'
                        if f"{e} (Prof: {k})" not in
horarios_grupo[g][i][d]: # Evitar duplicados
                            horarios_grupo[g][i][d] += f"{e} (Prof:
{k})\n"
                        for o in O:
                            for k in K:
                                # Asegúrate de que el acceso a las variables de
decisión sea correcto
                                if T[g][i][o][k][d].varValue == 1:
                                    # Añadir el bloque optativo 'o' al horario del
grupo 'g' en el día 'd' y franja 'i'
                                    if f"{o} (Prof: {k})" not in
horarios_grupo[g][i][d]: # Evitar duplicados
                                        horarios_grupo[g][i][d] += f"{o} (Prof:
{k})\n"

# Guardar los horarios de cada grupo en un archivo Excel
Horario_Grupos = '/Users/alexgarcia/Desktop/TFM/Horario_Grupos.xlsx'
with pd.ExcelWriter(Horario_Grupos, engine='openpyxl', mode='a') as
writer:
    for g, horario in horarios_grupo.items():
        # Comprobar si la hoja del grupo ya existe
        if g in writer.book.sheetnames:
            sheet = writer.book[g] # Obtener la hoja existente
        else:
            raise ValueError(f"La hoja para el grupo {g} no existe en
el archivo")

    # Ajustar el ancho de las columnas y el tamaño de la fuente
    for col in range(1, len(D) + 2):
        col_letter = get_column_letter(col)
        sheet.column_dimensions[col_letter].width = 21 # Ancho de
columna
        for cell in sheet[col_letter]:
            cell.font = Font(size=14) # Tamaño de la fuente
```

ANEXO 1. CODIFICACIÓN COMPLETA DE LA HERRAMIENTA EN PYTHON

```
# Escribir los días de la semana en las celdas B1 a H1
(ajustar las celdas si es necesario)
for col, day in enumerate(D, start=2):
    sheet.cell(row=1, column=col, value=day).font =
Font(bold=True, size=14) # Negrita y Fuente 14
    sheet.cell(row=1, column=col).alignment =
Alignment(horizontal='center') # Centrado

# Escribir las franjas horarias en las celdas A2 a A8 (ajustar
las celdas si es necesario)
for row, time_slot in enumerate(I, start=2):
    sheet.cell(row=row, column=1, value=time_slot).font =
Font(bold=True, size=14) # Negrita y Fuente 14
    sheet.cell(row=row, column=1).alignment =
Alignment(horizontal='center') # Centrado

# Escribir el horario en la hoja
for row, time_slot in enumerate(I, start=2):
    for col, day in enumerate(D, start=2):
        cell = sheet.cell(row=row, column=col,
value=horario[time_slot][day])
        cell.alignment = Alignment(horizontal='center') #
Centrado
        cell.font = Font(size=14) # Tamaño de la fuente

# Bordes de todas las celdas
border = Border(left=Side(style='thin', color='000000'),
                right=Side(style='thin', color='000000'),
                top=Side(style='thin', color='000000'),
                bottom=Side(style='thin', color='000000'))

# Aplicar bordes a todas las celdas
for row in sheet.iter_rows(min_row=1, max_row=len(I) + 1,
min_col=1, max_col=len(D) + 1):
    for cell in row:
        cell.border = border

# Abrir automáticamente el archivo Excel
import os

# Ruta del archivo Excel
ruta_excel = '/Users/alexgarcia/Desktop/TFM/Horario_Grupos.xlsx'

# Verificar si el archivo existe
if os.path.exists(ruta_excel):
    # Abrir el archivo Excel con el programa predeterminado
    os.system(f'open "{ruta_excel}"')
else:
    print("El archivo Excel no se encontró en la ruta especificada.")
```

Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

Modelo 2

```
import pulp as pl
from pulp import LpProblem, LpVariable, lpSum, LpMaximize, LpStatus,
value, GUROBI_CMD
import pandas as pd
from openpyxl.styles import Border, Side, Font, Alignment
from openpyxl.utils import get_column_letter

solver_list = pl.listSolvers()
print(solver_list)

solver_list = pl.listSolvers(onlyAvailable=True)
print(solver_list)

# ÍNDICES

I = ["I1", "I2", "I3", "I4", "I5", "I6", "I7"]

D = ["L", "M", "X", "J", "V"]

G = ["1A", "1B", "2A", "2B", "3A", "3B", "4A", "4B", "1ART", "1CI",
"1HUM", "2ART", "2CI", "2HUM"]

# Función para leer y cargar los datos de los índices J (Asignaturas)
y K (Profesores)
def leer_listado_asignaturas(BaseDatos):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='Generales', header=None)

    # Buscar el índice de la fila donde están las asignaturas
    indice_asignaturas = df[df[0] == 'Código'].index[0] + 1

    # Leer la columna de asignaturas desde ese índice hacia abajo
    asignaturas = df.loc[indice_asignaturas:, 0].tolist()

    return asignaturas

def leer_listado_profesores(BaseDatos):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='Profesores', header=None)

    # Buscar el índice de la fila donde están los profesores
    indice_profesores = df[df[0] == 'ID'].index[0] + 1

    # Leer la columna de profesores desde ese índice hacia abajo
    profesores = df.loc[indice_profesores:, 0].tolist()

    return profesores

# PARÁMETROS
# Parámetro HS. Número de horas semanales que deben impartirse para el
grupo g de la asignatura j
def leer_horas_semanales(BaseDatos, G, J):
    xls = pd.ExcelFile(BaseDatos)
    df = pd.read_excel(xls, sheet_name='HS', index_col=0, header=0)
    # Limpiar espacios en blanco de columnas e índice
```

ANEXO 1. CODIFICACIÓN COMPLETA DE LA HERRAMIENTA EN PYTHON

```
df.columns = df.columns.str.strip()
df.index = df.index.str.strip()
HS = {}
for j in J:
    for g in G:
        if g in df.columns and j in df.index:
            valor_horas = df.at[j, g]
            HS[(j, g)] = valor_horas
        else:
            print(f"Advertencia: Grupo '{g}' o asignatura '{j}' no
encontrado en el DataFrame HS")
    return HS

# Parámetro CP. Parámetro binario que toma valor 1 en caso de que el
profesor k pueda impartir la asignatura j
# Toma valor 0 si no está capacitado
def leer_datos_CP(BaseDatos, J, K):
    xls = pd.ExcelFile(BaseDatos)

    # Leer los datos de la hoja CP
    df_cp = pd.read_excel(xls, sheet_name='CP', index_col=0, header=0)

    # Inicializar el diccionario CP
    CP = {}

    # Iterar sobre cada profesor y asignatura para obtener el valor
binario correspondiente
    for profesor in K:
        for asignatura in J:
            # Obtener el valor de la celda correspondiente
            valor_celda = df_cp.at[profesor, asignatura]
            # Asignar el valor binario al diccionario CP
            CP[(profesor, asignatura)] = valor_celda

    return CP

# Ruta del archivo Excel
BaseDatos2 = '/Users/alexgarcia/Desktop/TFM/Base de
datos/BaseDatos2.xlsx'

# Leer y guardar los datos en listas
J = leer_listado_asignaturas(BaseDatos2)
K = leer_listado_profesores(BaseDatos2)
HS = leer_horas_semanales(BaseDatos2, G, J)
CP = leer_datos_CP(BaseDatos2, J, K)

# Impresiones para verificar
print("Asignaturas (J):", J)
print("Profesores (K):", K)
print("Horas semanales generales (HS):", HS)
print("Parámetro CP:", CP)

# Parámetro CS. Coste de satisfacción del profesor k en la franja
horaria i del día de la semana d
def procesar_excel_a_diccionarios(Horario_Docentes):
    # Leer el archivo de Excel
    xl = pd.ExcelFile(Horario_Docentes, engine='openpyxl')
```

**Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia**

```

# Diccionario que contendrá todos los horarios por persona
CS = {}
# Procesar cada hoja del archivo de Excel
for hoja in xl.sheet_names:
    # Leer la hoja actual en un DataFrame
    df = xl.parse(hoja)
    # Eliminar filas donde 'día de la semana' es NaN
    df.dropna(subset=[df.columns[0]], inplace=True)
    coste_profesor = {} # Diccionario para el profesor actual
    # Obtener los nombres de las columnas (intervalos de horas)
    intervalos_horas = df.columns[1:] # Excluir la primera
columna de días
    # Iterar sobre cada fila del DataFrame
    for index, row in df.iterrows():
        dia_semana = row[0] # El día de la semana está en la
primera columna
        if pd.isna(dia_semana):
            continue # Saltar filas donde el día de la semana es
NaN

        # Iterar sobre cada intervalo de horas
        for hora in intervalos_horas:
            valor_entero = row[hora]
            if pd.isna(valor_entero):
                continue # Saltar valores binarios NaN
            # Crear la clave del diccionario
            clave = (hoja, hora, dia_semana)
            # Asignar el valor correspondiente
            coste_profesor[clave] = valor_entero
        # Agregar el diccionario de la persona al diccionario general
        CS.update(coste_profesor)

return CS

Horario_Docentes =
'/Users/alexgarcia/Desktop/TFM/Horario_Docentes.xlsx'
CS = procesar_excel_a_diccionarios(Horario_Docentes)
print(CS)

# Lectura de los horarios de los grupos para ver las franjas libres
def leer_horario_ocupado_grupos(Horario_Grupos, G):
    xl = pd.ExcelFile(Horario_Grupos, engine='openpyxl')
    ocupacion_grupos = {}

    for grupo in G:
        df = pd.read_excel(xl, sheet_name=grupo, header=None)
        ocupacion_grupo = {}

        for row in range(2, 9): # Filas 2 a 8
            for col in range(2, 7): # Columnas B a F
                dia_semana = df.iloc[0, col - 1] # Los días de la
semana están en la fila 1
                franja_horaria = df.iloc[row - 1, 0] # Las franjas
horarias están en la columna A
                if pd.notna(df.iloc[row - 1, col - 1]): # Si hay algo
en la celda
                    ocupacion_grupo[(franja_horaria, dia_semana)] =
df.iloc[row - 1, col - 1]

```

ANEXO 1. CODIFICACIÓN COMPLETA DE LA HERRAMIENTA EN PYTHON

```
    ocupacion_grupos[grupo] = ocupacion_grupo
    return ocupacion_grupos

# Lectura de los horarios de los profesores para ver las franjas
libres
def leer_horario_ocupado_profesores(Horario_Docentes, K):
    xl = pd.ExcelFile(Horario_Docentes, engine='openpyxl')
    ocupacion_profesores = {}

    for profesor in K:
        df = pd.read_excel(xl, sheet_name=profesor, header=None)
        ocupacion_profesor = {}

        for row in range(9, 16): # Filas 9 a 15
            for col in range(2, 7): # Columnas B a F
                dia_semana = df.iloc[0, col-1] # Los días de la
                semana están en la fila 1
                franja_horaria = df.iloc[row-1, 0] # Las franjas
                horarias están en la columna A
                if pd.notna(df.iloc[row-1, col-1]): # Si hay algo en
                la celda
                    ocupacion_profesor[(franja_horaria, dia_semana)] =
                    df.iloc[row-1, col-1]

        ocupacion_profesores[profesor] = ocupacion_profesor
    return ocupacion_profesores

# Leer los horarios ocupados
Horario_Grupos = '/Users/alexgarcia/Desktop/TFM/Horario_Grupos.xlsx'
Horario_Docentes =
'/Users/alexgarcia/Desktop/TFM/Horario_Docentes.xlsx'
ocupacion_grupos = leer_horario_ocupado_grupos(Horario_Grupos, G)
ocupacion_profesores =
leer_horario_ocupado_profesores(Horario_Docentes, K)

# VARIABLES DE DECISIÓN
# Variable X. Variable binaria que toma valor 1 en caso de para el
grupo g
# en la franja horaria i se asigna la asignatura j impartida por el
profesor k en el día d, 0 en caso contrario
X = LpVariable.dicts("X", (G, I, J, K, D), cat='Binary')

# Variable Y. Variable binaria que toma valor 1 en caso de que para el
grupo g
# se imparte la asignatura j por el profesor k, 0 en caso contrario
Y = LpVariable.dicts("Y", (G, J, K), cat='Binary')

# Definir el problema como de maximización
prob = LpProblem("Horario", LpMaximize)

# FO: Maximizar la satisfacción de las preferencias horarias de los
profesores
prob += (
    lpSum(X[g][i][j][k][d] * CS.get((k, i, d), 0) for g in G for i in
    I for j in J for k in K for d in D)
)
```


Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

```
# RESTRICCIONES
# 1. Solamente un profesor único k puede impartir la asignatura j al
grupo g
for g in G:
    for j in J:
        prob += lpSum(Y[g][j][k] for k in K) == 1

# 2. Asignación general limitada a profesores capacitados
for g in G:
    for j in J:
        for k in K:
            if CP[(k, j)] == 0:
                prob += Y[g][j][k] == 0

# 3. En caso de que un profesor k esté asignado a la asignatura j del
grupo g en cualquier momento, es el responsable
for g in G:
    for j in J:
        for k in K:
            prob += lpSum(X[g][i][j][k][d] for i in I for d in D) ==
HS.get((j, g), 0) * Y[g][j][k]

# 4. Para un grupo g la asignatura j en una franja horaria i puede
impartirse como máximo una vez en un día d
for g in G:
    for j in J:
        for d in D:
            prob += lpSum(X[g][i][j][k][d] for i in I for k in K) <= 1

# 5. Todas las horas semanales deben impartirse
for g in G:
    for j in J:
        prob += lpSum(X[g][i][j][k][d] for i in I for k in K for d in
D) == HS.get((j, g), 0)

# 6. Cada grupo g solo puede tener una asignatura j en una franja
horaria i de un día d
for g in G:
    for i in I:
        for d in D:
            prob += lpSum(X[g][i][j][k][d] for j in J for k in K) <= 1

# 7. Cada profesor k solo puede tener una asignatura j en una franja
horaria i de un día d
for i in I:
    for k in K:
        for d in D:
            prob += lpSum(X[g][i][j][k][d] for g in G for j in J) <= 1

# 8. No se pueden realizar asignaciones en franjas ocupadas
for g in G:
    for i in I:
        for d in D:
            for j in J:
                for k in K:
```

ANEXO 1. CODIFICACIÓN COMPLETA DE LA HERRAMIENTA EN PYTHON

```
        if (i, d) in ocupacion_grupos[g] or (i, d) in
ocupacion_profesores[k]:
            prob += X[g][i][j][k][d] == 0 # No asignar en
frangas ocupadas

# Crear una instancia del solucionador GUROBI
gurobi_solver = GUROBI_CMD()

import time

StartTim = time.time()
print("Start time")

# Optimización
status = prob.solve(gurobi_solver)

FinishTime = time.time()
print("Finish time")

# Mostrar el resultado
print("Estado:", LpStatus[prob.status])
print("Horario óptimo:")

# Imprimir los horarios de cada profesor
for k in K:
    print(f"\n{k}:")
    for d in D:
        print(f"-{d}:")
        for i in I:
            for g in G:
                for j in J:
                    if value(X[g][i][j][k][d]) == 1:
                        print(f"  {i}: {g}, {j}")

# Imprimir los horarios de cada grupo
for g in G:
    print(f"Grupo: {g}")
    for d in D:
        for i in I:
            for j in J:
                for k in K:
                    if value(X[g][i][j][k][d]) == 1:
                        print(f"  - {d}, {i}: {j}")

print("Nivel de satisfacción:", value(prob.objective))

# Inicializar el diccionario para recoger los horarios de cada
profesor adecuadamente
horarios_profesor = {prof: {i: {d: "" for d in D} for i in I} for g in
G for prof in K}

# Llenar los horarios con la información de la solución
for g in G: # Debes iniciar con el grupo para tener acceso a las
asignaturas y profesores específicos de ese grupo
    for k in K:
        for d in D:
            for i in I:
```

Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

```
for j in J:
    # Asegurarse que el acceso a la variable de
    decisión es correcto
    # y que el profesor 'k' enseña 'j' en 'g'
    if X[g][i][j][k][d].varValue == 1:
        # Añadir al horario del profesor 'k' que
        enseña en 'g',
        # en el día 'd', franja 'i', la asignatura 'j'
        if f"{j} ({g})" not in
horarios_profesor[k][i][d]: # Verificar que no se añada duplicados
            horarios_profesor[k][i][d] += f"{j}
({g})\n"

# Guardar los horarios de cada profesor en un archivo Excel
with pd.ExcelWriter(Horario_Docentes, engine='openpyxl', mode='a') as
writer:

    for k, horario in horarios_profesor.items():
        # Obtener la hoja correspondiente al profesor (K1, K2, K3,
        ...)
        if f"{k}" in writer.book.sheetnames:
            sheet = writer.book[f"{k}"]
        else:
            sheet = writer.book.create_sheet(title=f"K{k}")

        # Ajustar el ancho de las columnas y el tamaño de la fuente
        for col in range(1, len(D) + 2):
            col_letter = get_column_letter(col)
            sheet.column_dimensions[col_letter].width = 14 # Ancho de
            columna

            for cell in sheet[col_letter]:
                cell.font = Font(size=14) # Tamaño de la fuente

        # Escribir los días de la semana en las celdas B8 a H8
        for col, day in enumerate(D, start=2):
            sheet.cell(row=8, column=col, value=day).font =
            Font(bold=True, size=14) # Negrita y Fuente 14
            sheet.cell(row=8, column=col).alignment =
            Alignment(horizontal='center') # Centrado

        # Escribir las franjas horarias en las celdas A9 a A13
        for row, time_slot in enumerate(I, start=9):
            sheet.cell(row=row, column=1, value=time_slot).font =
            Font(bold=True, size=14) # Negrita y Fuente 14
            sheet.cell(row=row, column=1).alignment =
            Alignment(horizontal='center') # Centrado

        # Escribir el horario en la hoja (ajustado para las nuevas
        posiciones)
        for row, time_slot in enumerate(I, start=9):
            for col, day in enumerate(D, start=2):
                cell = sheet.cell(row=row, column=col,
                value=horario[time_slot][day])
                cell.alignment = Alignment(horizontal='center') #
                Centrado

                cell.font = Font(size=14) # Tamaño de la fuente
```

ANEXO 1. CODIFICACIÓN COMPLETA DE LA HERRAMIENTA EN PYTHON

```
# Bordes de todas las celdas
border = Border(left=Side(style='thin', color='000000'),
               right=Side(style='thin', color='000000'),
               top=Side(style='thin', color='000000'),
               bottom=Side(style='thin', color='000000'))

# Aplicar bordes a todas las celdas en el rango ajustado
for row in sheet.iter_rows(min_row=8, max_row=15, min_col=1,
max_col=len(D) + 1):
    for cell in row:
        cell.border = border

# Abrir automáticamente el archivo Excel
import os

# Ruta del archivo Excel
ruta_excel = '/Users/alexgarcia/Desktop/TFM/Horario_Docentes.xlsx'

# Verificar si el archivo existe
if os.path.exists(ruta_excel):
    # Abrir el archivo Excel con el programa predeterminado
    os.system(f'open "{ruta_excel}")')
else:
    print("El archivo Excel no se encontró en la ruta especificada.")

# Crear una estructura de datos para representar el horario por grupo
horarios_grupo = {g: {i: {d: "" for d in D} for i in I} for g in G}

# Llenar los horarios con la información de la solución
for g in G: # Inicia con el grupo para acceder correctamente a las
asignaturas y profesores de ese grupo
    for d in D:
        for i in I:
            for j in J:
                for k in K:
                    # Asegúrate de que el acceso a las variables de
decisión sea correcto
                    if X[g][i][j][k][d].varValue == 1:
                        # Añadir la asignatura 'j' al horario del
grupo 'g' en el día 'd' y franja 'i'
                        if f"{j} (Prof: {k})" not in
horarios_grupo[g][i][d]: # Evitar duplicados
                            horarios_grupo[g][i][d] += f"{j} (Prof:
{k})\n"

# Guardar los horarios de cada grupo en un archivo Excel
Horario_Grupos = '/Users/alexgarcia/Desktop/TFM/Horario_Grupos.xlsx'
with pd.ExcelWriter(Horario_Grupos, engine='openpyxl', mode='a') as
writer:
    for g, horario in horarios_grupo.items():
        # Comprobar si la hoja del grupo ya existe
        if g in writer.book.sheetnames:
            sheet = writer.book[g] # Obtener la hoja existente
        else:
            raise ValueError(f"La hoja para el grupo {g} no existe en
el archivo")
```

Diseño de una herramienta mediante programación matemática para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

```
# Ajustar el ancho de las columnas y el tamaño de la fuente
for col in range(1, len(D) + 2):
    col_letter = get_column_letter(col)
    sheet.column_dimensions[col_letter].width = 21 # Ancho de
columna

    for cell in sheet[col_letter]:
        cell.font = Font(size=14) # Tamaño de la fuente

# Escribir los días de la semana en las celdas B1 a H1
(ajustar las celdas si es necesario)
for col, day in enumerate(D, start=2):
    sheet.cell(row=1, column=col, value=day).font =
Font(bold=True, size=14) # Negrita y Fuente 14
    sheet.cell(row=1, column=col).alignment =
Alignment(horizontal='center') # Centrado

# Escribir las franjas horarias en las celdas A2 a A8 (ajustar
las celdas si es necesario)
for row, time_slot in enumerate(I, start=2):
    sheet.cell(row=row, column=1, value=time_slot).font =
Font(bold=True, size=14) # Negrita y Fuente 14
    sheet.cell(row=row, column=1).alignment =
Alignment(horizontal='center') # Centrado

# Escribir el horario en la hoja
for row, time_slot in enumerate(I, start=2):
    for col, day in enumerate(D, start=2):
        cell = sheet.cell(row=row, column=col,
value=horario[time_slot][day])
        cell.alignment = Alignment(horizontal='center') #
Centrado

        cell.font = Font(size=14) # Tamaño de la fuente

# Bordes de todas las celdas
border = Border(left=Side(style='thin', color='000000'),
                right=Side(style='thin', color='000000'),
                top=Side(style='thin', color='000000'),
                bottom=Side(style='thin', color='000000'))

# Aplicar bordes a todas las celdas
for row in sheet.iter_rows(min_row=1, max_row=len(I) + 1,
min_col=1, max_col=len(D) + 1):
    for cell in row:
        cell.border = border

# Abrir automáticamente el archivo Excel
import os

# Ruta del archivo Excel
ruta_excel = '/Users/alexgarcia/Desktop/TFM/Horario_Grupos.xlsx'

# Verificar si el archivo existe
if os.path.exists(ruta_excel):
    # Abrir el archivo Excel con el programa predeterminado
    os.system(f'open "{ruta_excel}"')
else:
    print("El archivo Excel no se encontró en la ruta especificada.")
```

ANEXO 2. CODIFICACIÓN DE LA INTERFAZ VISUAL EN VBA

```
Sub CrearMenuDesplegable ()

    Dim ws As Worksheet
    Set ws = ThisWorkbook.Sheets("MENU") ' Cambia "HojaPrincipal" por
    el nombre de tu hoja

    ' Acceder al ComboBox de formulario por nombre y llenarlo con
    opciones
    With ws.Shapes("Grupos").ControlFormat
        .RemoveAllItems
        .AddItem "1A"
        .AddItem "1B"
        .AddItem "2A"
        .AddItem "2B"
        .AddItem "3A"
        .AddItem "3B"
        .AddItem "4A"
        .AddItem "4B"
        .AddItem "1ART"
        .AddItem "1CI"
        .AddItem "1HUM"
        .AddItem "2ART"
        .AddItem "2CI"
        .AddItem "2HUM"
    End With

    ' Asignar los valores al ComboBox2 (PROFESORES)
    With ws.Shapes("Profesores").ControlFormat
        .RemoveAllItems
        Dim i As Integer
        For i = 1 To 39
            .AddItem "K" & i
        Next i
    End With

    'Asignar los valores al ComboBox3 (GENERACIÓN HORARIO)
    With ws.Shapes("Horario").ControlFormat
        .RemoveAllItems
        .AddItem "GEN"
        .AddItem "DEP"
    End With

End Sub

Sub IrAGrupo ()
    Dim ws As Worksheet
    Dim grupoSeleccionado As String
    Dim indiceSeleccionado As Integer
```

**Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia**

```
' Obtener la referencia a la hoja donde está el ComboBox de
formulario
Set ws = ThisWorkbook.Sheets("MENU")

' Obtener el índice seleccionado en el ComboBox de formulario
indiceSeleccionado = ws.Shapes("Grupos").ControlFormat.Value

' Si no se ha seleccionado nada, mostramos un mensaje
If indiceSeleccionado = 0 Then
    MsgBox "Por favor selecciona un grupo."
    Exit Sub
End If

' Obtener el valor seleccionado basado en el índice del ComboBox
grupoSeleccionado =
ws.Shapes("Grupos").ControlFormat.List(indiceSeleccionado)

' Comprobar si la hoja seleccionada existe
On Error Resume Next
Set ws = ThisWorkbook.Sheets(grupoSeleccionado)
On Error GoTo 0

If ws Is Nothing Then
    MsgBox "La hoja para el profesor " & grupoSeleccionado & " no
existe."
    Exit Sub
End If

' Mostrar la hoja del profesor seleccionado y activarla
ws.Visible = xlSheetVisible
ws.Activate

' Ocultar la hoja del menú
ThisWorkbook.Sheets("MENU").Visible = xlSheetVeryHidden

End Sub

Sub IrAProfesor()
' Llevar a la hoja correspondiente de profesores
Dim ws As Worksheet
Dim profesorSeleccionado As String
Dim comboBoxIndex As Integer

' Obtener la referencia a la hoja donde está el ComboBox de
formulario
Set ws = ThisWorkbook.Sheets("MENU")

' Obtener el índice seleccionado en el ComboBox de formulario
comboBoxIndex = ws.Shapes("Profesores").ControlFormat.Value

' Si no se ha seleccionado nada, mostramos un mensaje
If comboBoxIndex = 0 Then
    MsgBox "Por favor selecciona un profesor."
    Exit Sub
End If

' Obtener el valor seleccionado basado en el índice del ComboBox
```

```
profesorSeleccionado =  
ws.Shapes("Profesores").ControlFormat.List(comboBoxIndex)  
  
' Comprobar si la hoja seleccionada existe  
On Error Resume Next  
Set ws = ThisWorkbook.Sheets(profesorSeleccionado)  
On Error GoTo 0  
  
If ws Is Nothing Then  
    MsgBox "La hoja para el profesor " & profesorSeleccionado & "  
no existe."  
    Exit Sub  
End If  
  
' Mostrar la hoja del profesor seleccionado y activarla  
ws.Visible = xlSheetVisible  
ws.Activate  
  
' Ocultar la hoja del menú  
ThisWorkbook.Sheets("MENU").Visible = xlSheetVeryHidden  
  
End Sub  
  
Sub ControlComboBox()  
    Dim comboValor As String  
    Dim ws As Worksheet  
  
    ' El ComboBox está en la hoja "MENU"  
    Set ws = ThisWorkbook.Sheets("MENU")  
  
    ' Leer el valor del combobox (combobox de formulario)  
    comboValor =  
ws.Shapes("Horario").ControlFormat.List(ws.Shapes("Horario").ControlFo  
rmat.ListIndex)  
  
    ' Verificar qué opción se ha seleccionado  
    If comboValor = "GEN" Then  
        ' Llamar al sub CopiarHojas para la opción GENERAR  
        Call CopiarHojas  
    ElseIf comboValor = "DEP" Then  
        ' Llamar al sub VaciarHojas para la opción DEPURAR  
        Call VaciarHojas  
    End If  
End Sub  
  
Sub CopiarHojas()  
    Dim wbFuenteGrupos As Workbook  
    Dim wbFuenteDocentes As Workbook  
    Dim wsFuente As Worksheet  
    Dim wsDestino As Worksheet  
    Dim rutaGrupos As String  
    Dim rutaDocentes As String  
    Dim nombreHoja As String  
  
    ' Rutas de los archivos a copiar  
    rutaGrupos =  
"/Users/alexgarcia/Desktop/TFM/Horario_Grupos_Completo.xlsx"
```


Diseño de una herramienta mediante programación matemática
para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

```
rutaDocentes =  
"/Users/alexgarcia/Desktop/TFM/Horario_Docentes_Completo.xlsx"  
  
' Abrir los archivos fuente  
Set wbFuenteGrupos = Workbooks.Open(rutaGrupos)  
Set wbFuenteDocentes = Workbooks.Open(rutaDocentes)  
  
' Copiar las hojas del archivo de grupos  
For Each wsFuente In wbFuenteGrupos.Worksheets  
    nombreHoja = wsFuente.Name  
    On Error Resume Next  
    Set wsDestino = ThisWorkbook.Sheets(nombreHoja)  
    On Error GoTo 0  
    If Not wsDestino Is Nothing Then  
        wsFuente.Cells.Copy Destination:=wsDestino.Cells  
    End If  
Next wsFuente  
  
' Copiar las hojas del archivo de docentes  
For Each wsFuente In wbFuenteDocentes.Worksheets  
    nombreHoja = wsFuente.Name  
    On Error Resume Next  
    Set wsDestino = ThisWorkbook.Sheets(nombreHoja)  
    On Error GoTo 0  
    If Not wsDestino Is Nothing Then  
        wsFuente.Cells.Copy Destination:=wsDestino.Cells  
    End If  
Next wsFuente  
  
' Cerrar los archivos fuente sin guardar cambios  
wbFuenteGrupos.Close False  
wbFuenteDocentes.Close False  
End Sub  
  
Sub VaciarHojas()  
    Dim ws As Worksheet  
  
    ' Iterar sobre cada hoja en la interfaz de Excel  
    For Each ws In ThisWorkbook.Worksheets  
        ' Verificar si la hoja debe ser vaciada (en caso de que no sea  
        una hoja especial como "Interfaz")  
        If ws.Name <> "MENU" Then  
            ' Solo borrar el contenido de las celdas  
            ws.Cells.ClearContents  
        End If  
    Next ws  
End Sub  
  
Sub Ocultar()  
    Dim A As Integer  
    For A = 2 To 56  
        Sheets(A).Visible = xlVeryHidden  
        Sheets("MENU").Select  
        Range("a1").Select  
    Next  
End Sub
```

```
Sub Mostrar()  
Dim A As Integer  
For A = 2 To 56  
Sheets(A).Visible = True  
Sheets("MENU").Select  
Range("a1").Select  
Next  
End Sub  
  
Sub VolverAlMenu()  
    ' Ocultar la hoja activa y volver al menú  
    Dim hojaActual As Worksheet  
    Set hojaActual = ActiveSheet  
  
    ' Mostrar la hoja del menú  
    ThisWorkbook.Sheets("MENU").Visible = xlSheetVisible  
    ThisWorkbook.Sheets("MENU").Activate  
  
    ' Ocultar la hoja actual  
    hojaActual.Visible = xlSheetVeryHidden  
End Sub
```

ANEXO 3. MANUAL DE USUARIO DE LA APLICACIÓN

1. Introducción

Esta aplicación permite visualizar y gestionar horarios de grupos y profesores. Además, ofrece la funcionalidad de generar o eliminar horarios automáticamente. Incluye opciones para mostrar u ocultar hojas de datos según las necesidades del usuario.

2. Requisitos

- Microsoft Excel y entorno Python instalados.
- Ficheros de horarios correspondientes a grupos y profesores cargados en la aplicación.

3. Menú principal



1. GRUPOS

Este menú contiene una lista de todos los grupos disponibles. Selecciona un grupo para visualizar su horario.

- *Cómo utilizarlo:*
 - Haz clic en el menú desplegable "GRUPOS".
 - Selecciona el grupo cuyo horario deseas ver.

- El horario aparecerá automáticamente en la hoja correspondiente.

2. PROFESORES

Este menú contiene una lista de todos los profesores disponibles. Selecciona un profesor para visualizar su horario.

- *Cómo utilizarlo:*
 - Haz clic en el menú desplegable "PROFESORES".
 - Selecciona el profesor cuyo horario deseas ver.
 - El horario aparecerá automáticamente en la hoja correspondiente.

3. GENERACIÓN HORARIO

Este menú ofrece dos opciones para gestionar la generación de horarios.

- **GEN:** Genera un nuevo horario basado en la información proporcionada.
- **DEP:** Elimina cualquier horario generado previamente.
- *Cómo utilizarlo:*
 - Haz clic en el menú desplegable "GENERACIÓN HORARIO".
 - Selecciona "GEN" para generar un horario nuevo o "DEP" para eliminar el existente.

4. MOSTRAR

Esta opción es un botón que permite visualizar las hojas ocultas del archivo Excel (por defecto están todas las hojas ocultas exceptuando la del menú principal).

- *Cómo utilizarlo:*
 - Haz clic en "MOSTRAR" para visualizar todas las hojas disponibles.

5. OCULTAR

Esta opción es un botón que permite ocultar las hojas del archivo Excel, menos la hoja del menú principal

- *Cómo utilizarlo:*
 - Haz clic en "OCULTAR" para ocultar todas las hojas visibles.

4. Pasos para el uso

- 1) Inicia la aplicación
- 2) Usa los menús desplegables "GRUPOS" o "PROFESORES" para ver horarios específicos.
- 3) Asegúrate que el código Python de la herramienta ejecuta correctamente
- 4) Utiliza el menú "GENERACIÓN HORARIO" para generar o depurar horarios.
- 5) Usa los botones "MOSTRAR" y "OCULTAR" para gestionar la visibilidad de las hojas.

5. Resolución de problemas

- Si no se muestra un horario, asegúrate de que el grupo o profesor seleccionado tiene horarios disponibles en la base de datos.
- Si no se genera un horario, verifica que los datos de entrada sean correctos.

ANEXO 3. RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030

En este último Anexo se procede al análisis de la relación que tiene este Trabajo de Fin de Máster con los Objetivos de Desarrollo Sostenible (ODS) dispuestos en la Agenda 2030. También se hará referencia a las metas correspondientes de cada uno esos objetivos.

3.1. Grado de relación del trabajo con los objetivos de desarrollo sostenible

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.	X			
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.	X			
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.		X		
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

ANEXO 3. RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO
SOSTENIBLE DE LA AGENDA 2030

3.2. Grado de relación del trabajo con los ODS

Meta del ODS 4. Educación de calidad	Alto	Medio	Bajo	No Procede
4.3. De aquí a 2030, asegurar el acceso igualitario de todos los hombres y las mujeres a una formación técnica, profesional y superior de calidad, incluida la enseñanza universitaria		X		
4.4. De aquí a 2030, aumentar considerablemente el número de jóvenes y adultos que tienen las competencias necesarias, en particular técnicas y profesionales, para acceder al empleo, el trabajo decente y el emprendimiento	X			
4.c. De aquí a 2030, aumentar considerablemente la oferta de docentes calificados, incluso mediante la cooperación internacional para la formación de docentes en los países en desarrollo, especialmente los países menos adelantados y los pequeños Estados insulares en desarrollo			X	

Meta del ODS 8. Trabajo decente y crecimiento económico	Alto	Medio	Bajo	No Procede
8.2. Lograr niveles más elevados de productividad económica mediante la diversificación, la modernización tecnológica y la innovación, entre otras cosas centrándose en los sectores con gran valor añadido y un uso intensivo de la mano de obra		X		
8.3. Promover políticas orientadas al desarrollo que apoyen las actividades productivas, la creación de puestos de trabajo decentes, el emprendimiento, la creatividad y la innovación, y fomentar la formalización y el crecimiento de las microempresas y las pequeñas y medianas empresas, incluso mediante el acceso a servicios financieros			X	
8.6. De aquí a 2020, reducir considerablemente la proporción de jóvenes que no están empleados y no cursan estudios ni reciben capacitación.			X	

Meta del ODS 10. Reducir la desigualdad en los países y entre ellos	Alto	Medio	Bajo	No Procede
10.3. Garantizar la igualdad de oportunidades y reducir la desigualdad. De resultados, incluso eliminando leyes, políticas y prácticas discriminatorias y promoviendo legislaciones, políticas y medidas adecuadas a ese respecto			X	

Diseño de una herramienta mediante programación matemática para la gestión y organización de un horario anual en una escuela de secundaria en Valencia

3.3. Descripción de la alineación del TFM con los ODS más relacionados

La relación que guarda este trabajo con los Objetivos de Desarrollo Sostenible (ODS) es considerable, particularmente con el ODS 4, ODS 8 y ODS 10.

El **ODS 4** busca asegurar una educación inclusiva, equitativa y de calidad para todas las personas. Este proyecto tiene como finalidad crear un horario anual que cumpla con las expectativas de todas las partes involucradas, garantizando que sea justo y de alta calidad. El objetivo es que tanto profesores como estudiantes experimenten una sensación de bienestar en su rutina diaria, influenciada por el horario que deben seguir (meta 4.3). Además, esta herramienta estará disponible para todos los docentes y, si es necesario, también para los estudiantes, lo que les permitirá adquirir habilidades y competencias relacionadas con su uso (metas 4.4 y 4.c).

En relación con el **ODS 8**, enfocado en promover el trabajo decente y el crecimiento económico, la optimización en la planificación del horario podría generar beneficios para la escuela al reducir costos en recursos (meta 8.2). Además, esta mejora podría impulsar el crecimiento tecnológico y la innovación dentro de la institución (meta 8.3). Por último, es probable que este nuevo enfoque en la creación de horarios atraiga a un mayor número de estudiantes interesados en el proceso (meta 8.6).

En cuanto al **ODS 10**, centrado en reducir la desigualdad dentro de los países y entre ellos, la herramienta propuesta contribuye a garantizar la igualdad de oportunidades en la creación de horarios, asegurando una distribución equitativa para todos los involucrados. Esta optimización podría ayudar a minimizar cualquier posible sesgo o práctica discriminatoria en la asignación de recursos, promoviendo así políticas y prácticas más justas (meta 10.3).

La relevancia de este trabajo radica en su contribución directa a la consecución de varios Objetivos de Desarrollo Sostenible (ODS) y sus metas específicas. Al optimizar la planificación y creación de horarios, se facilita una distribución equitativa de los recursos docentes, lo que impacta de manera positiva en la calidad educativa. Esta herramienta garantiza que tanto profesores como estudiantes tengan una experiencia más organizada y eficiente, favoreciendo un entorno educativo más inclusivo y accesible para todos.

Además, el desarrollo de esta herramienta, sustentada en una programación matemática, promueve la innovación y mejora la eficiencia en la gestión educativa. Esto no solo reduce los tiempos y costes en la planificación, sino que también fomenta el uso de soluciones tecnológicas en el ámbito académico, pudiendo impulsar el crecimiento tecnológico y apoyar políticas que promuevan el desarrollo sostenible de la institución.

Cabe destacar que el proyecto no se limita a aspectos teóricos, sino que tiene un enfoque práctico al generar bases de datos y herramientas concretas que pueden ser aplicadas directamente en la gestión y planificación escolar.



ANEXO 3. RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030

En resumen, este Trabajo de Fin de Máster establece una conexión directa con los Objetivos de Desarrollo Sostenible, particularmente con el ODS 4, el ODS 8 y el ODS 10, así como con las metas 4.3, 4.4, 4.c, 8.2, 8.3, 8.6 y 10.3. A través de la creación de una herramienta para la gestión de horarios, este proyecto tiene el potencial de generar un impacto positivo tanto en el ámbito educativo como en el laboral, contribuyendo de manera significativa a la consecución de los ODS y avanzando hacia un futuro más equitativo y sostenible.