



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Arqueología informática: diseño e implementación de la calculadora mecánica Millionaire con Scratch

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Pablo Baeza Boronat

Tutor: Xavier Molero Prieto

Curso 2020-2021

Resum

El objectiu principal d'aquest projecte es realitzar una virtualització de la màquina de càlcul Millionaire.

En l'inici d'aquesta memòria, s'estudia el llenguatge de programació Scratch, explicant el seus orígens i el seu funcionament, així com les seues característiques, els seus avantatges y les seues limitacions. Per a este capítol s'ha fet ús de les guíes de referència de Scratch.

A continuació, es realitza un repàs històric de l'evolució de les distintes eines de càlcul, tot analitzant també el context socioeconòmic que va propiciar la invenció de la calculadora Millionaire.

La memòria continua fent un exhaustiu anàlisi del funcionament de la màquina i els seus orígens, explicant cadascuna de les operacions que pot realitzar l'aparell i el mecanisme intern que ho permet.

Finalment, amb els coneixements adquirits en Scratch i en la màquina, s'ha programat i documentat el codi de la calculadora.

Paraules clau: Calculadora mecànica, Scratch, Millionaire, Acarreo, Màquina, Industrialització, Història de la informàtica

Resumen

El objetivo principal de este proyecto es realizar una virtualización de la máquina de cálculo Millionaire.

En el inicio de esta memoria, se estudia el lenguaje de programación Scratch, explicando sus orígenes y su funcionamiento, así como sus características, sus ventajas y sus limitaciones. Para este capítulo se ha hecho uso de las guías de referencia de Scratch.

A continuación, se realiza un repaso histórico de la evolución de las distintas herramientas de cálculo, analizando también el contexto socioeconómico que propició la invención de la calculadora Millionaire.

La memoria continúa haciendo un exhaustivo análisis del funcionamiento de la máquina y sus orígenes, explicando cada una de las operaciones que puede realizar el aparato y el mecanismo interno que lo permite.

Por último, con los conocimientos adquiridos en Scratch y en la máquina, se ha programado y documentado el código de la calculadora.

Palabras clave: Calculadora mecánica, Scratch, Millionaire, Acarreo, Máquina, Industrialización, Historia de la informática

Abstract

The main objective of this project is to realize virtualization of the Millionaire calculation machine.

At the beginning of this report, the Scratch programming language is studied, explaining its origins and operation, as well as its features, advantages and limitations. This chapter was written with the help of the Scratch reference guides.

The next part is a historical review of the evolution of the different calculation tools and the economic context that led to the invention of the Millionaire calculator.

The report continues with an exhaustive analysis of the operation of the machine and its origins, explaining each of the operations that the device can make and the internal mechanism that allows it.

Finally, with the knowledge acquired in Scratch and the machine, the calculator code has been programmed and documented.

Key words: Mechanical calculator, Scratch, Millionaire, Adden, Machine, Industrialization, History of computing

Índice general

| | |
|-------------------|-----|
| Índice general | v |
| Índice de figuras | vii |

| | | |
|----------|---|-----------|
| 1 | Introducción | 1 |
| 1.1 | Motivación | 1 |
| 1.2 | Objetivos | 2 |
| 1.3 | Estructura de la memoria | 2 |
| 1.4 | Objetivos de Desarrollo Sostenible | 3 |
| 1.4.1 | Educación de calidad | 3 |
| 1.4.2 | Industria, innovación e infraestructura | 3 |
| 1.5 | Uso de la bibliografía | 4 |
| 1.6 | Relación con asignaturas del grado | 4 |
| 1.7 | Comentarios del autor | 4 |
| 2 | El entorno de programación Scratch | 5 |
| 2.1 | ¿Por qué Scratch? | 5 |
| 2.2 | Origen y evolución de Scratch | 5 |
| 2.3 | Programación con Scratch | 6 |
| 2.3.1 | Paneles | 10 |
| 2.3.2 | Bloques | 11 |
| 2.3.3 | Objetos | 12 |
| 2.3.4 | Concurrencia | 13 |
| 2.4 | Beneficios de usar Scratch | 13 |
| 3 | Contexto histórico | 15 |
| 3.1 | Evolución de las herramientas de cálculo | 15 |
| 3.2 | La sociedad del siglo XIX | 21 |
| 4 | La calculadora Millionaire | 25 |
| 4.1 | El origen de la calculadora Millionaire | 25 |
| 4.2 | Introducción a la calculadora Millionaire | 27 |
| 4.2.1 | El panel de control | 27 |
| 4.3 | Características de la calculadora Millionaire | 27 |
| 4.4 | Funcionamiento de la calculadora Millionaire | 29 |
| 4.4.1 | Suma | 31 |
| 4.4.2 | Resta | 31 |
| 4.4.3 | Multiplicación | 32 |
| 4.4.4 | División | 34 |
| 5 | Implementación de la Millionaire en Scratch | 37 |
| 5.1 | Creación de los objetos | 37 |
| 5.2 | Movimiento de los objetos | 38 |
| 5.2.1 | Deslizadores | 38 |

| | | |
|----------|---|-----------|
| 5.2.2 | Manivela | 40 |
| 5.2.3 | Palanca de multiplicación | 40 |
| 5.2.4 | Palanca de operación | 40 |
| 5.2.5 | Operador | 40 |
| 5.2.6 | Barra de resultado | 41 |
| 5.2.7 | Barra de cociente | 41 |
| 5.2.8 | Palanca de reinicio | 41 |
| 5.3 | Programación de las operaciones | 41 |
| 5.3.1 | Suma | 46 |
| 5.3.2 | Resta | 47 |
| 5.3.3 | Multiplicación | 48 |
| 5.3.4 | División | 54 |
| 6 | Conclusiones | 59 |
| | Bibliografía | 61 |

Índice de figuras

| | | |
|------|--|----|
| 1.1 | Objetivos de Desarrollo Sostenible | 3 |
| 2.1 | Interfaz de Scratch | 7 |
| 2.2 | Panel izquierdo de la interfaz | 7 |
| 2.3 | Interfaz de disfraces | 8 |
| 2.4 | Interfaz de sonidos | 8 |
| 2.5 | Panel central de la interfaz de Scratch | 9 |
| 2.6 | Sincronización por bloqueo | 14 |
| 3.1 | Hueso de Ishango | 16 |
| 3.2 | Ábaco de Salamina | 17 |
| 3.3 | Suanpan | 19 |
| 3.4 | Ábaco de Salamina | 19 |
| 3.5 | Reloj calculante | 20 |
| 3.6 | Pascalina | 20 |
| 3.7 | Calculadora de Leibniz | 21 |
| 3.8 | Aritmómetro | 22 |
| 3.9 | Aritmómetro de Odhner | 23 |
| 3.10 | Comptómetro | 24 |
| 4.1 | Otto Steiger (1858-1923) | 26 |
| 4.2 | Ejemplar de Millionaire | 26 |
| 4.3 | Panel de control | 28 |
| 4.4 | Palanca de operaciones | 29 |
| 4.5 | Palancas de cifras | 30 |
| 4.6 | Manivela | 30 |
| 4.7 | Mecanismo de cremallera | 33 |
| 4.8 | Palanca de multiplicar | 34 |
| 4.9 | Representación de las tablas de multiplicar | 34 |
| 4.10 | Mecanismo de multiplicar | 35 |
| 4.11 | Tablas de dividir | 36 |
| 5.1 | Calculadora usada en Scratch | 38 |
| 5.2 | Manivela usada en Scratch | 38 |
| 5.3 | Calculadora sin manivela usada en Scratch | 39 |
| 5.4 | Calculadora sin partes móviles usada en Scratch | 39 |
| 5.5 | Código de movimiento de la manivela | 42 |
| 5.6 | Código de movimiento de la palanca de multiplicación | 42 |
| 5.7 | Código de movimiento de los deslizadores | 43 |
| 5.8 | Código de cambio de disfraz del operador | 43 |
| 5.9 | Código de movimiento de la palanca de operación | 44 |

| | |
|--|----|
| 5.10 Código de movimiento de resultado cuando hay acarreo | 45 |
| 5.11 Código de movimiento de la barra de cociente | 45 |
| 5.12 Código de movimiento de la palanca de reinicio | 46 |
| 5.13 Código de cambio de disfraz al realizar la suma | 47 |
| 5.14 Código de cálculo de la decenas en la suma | 48 |
| 5.15 Código de cálculo de la centena de millón para la suma | 49 |
| 5.16 Paso 1 del ejemplo de la suma | 50 |
| 5.17 Paso 2 del ejemplo de la suma | 50 |
| 5.18 Código de cálculo de la centena de millón para la resta | 51 |
| 5.19 Código de cálculo de la centena de millón para la resta | 51 |
| 5.20 Paso 1 del ejemplo de la resta | 52 |
| 5.21 Paso 2 del ejemplo de la resta | 52 |
| 5.22 Código de desplazamiento del resultado en la multiplicación | 53 |
| 5.23 Código de cálculo de la multiplicación | 53 |
| 5.24 Paso 1 del ejemplo de la multiplicación | 54 |
| 5.25 Paso 2 del ejemplo de la multiplicación | 55 |
| 5.26 Paso 3 del ejemplo de la multiplicación | 55 |
| 5.27 Paso 1 del ejemplo de la división | 56 |
| 5.28 Paso 2 del ejemplo de la división | 57 |
| 5.29 Paso 3 del ejemplo de la división | 57 |

CAPÍTULO 1

Introducción

La introducción de este proyecto comienza con la motivación, dónde se explica con qué fin se ha llevado a cabo este trabajo. Posteriormente, se presenta los objetivos, dónde se explica lo que se pretende lograr con la realización de este proyecto. Para acabar el capítulo, se explica brevemente la estructura de la memoria y los Objetivos de Desarrollo Sostenible que se impulsan con este proyecto.

1.1 Motivación

Según pasa el tiempo también lo hace la humanidad. Los avances en tecnología son más frecuentes y potentes. Estos cambios constantes generan una brecha tecnológica cada vez mayor. Por ello, las nuevas generaciones solo disponen de las herramientas actuales. Es importante conocer el origen y la historia de todas las herramientas de las que disponemos hoy en día para comprender hacia dónde nos desplazamos.

Como institución pública, la Universidad Politécnica de Valencia tiene la obligación de contribuir al estudio de estos orígenes y a la divulgación del patrimonio de la cultura informática, dando a su historia la importancia que se merece.

Con el presente trabajo, se busca apoyar la labor difusora del Museo de Informática de la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València enfocada en informar a todas las personas interesadas en conocer el recorrido evolutivo de la tecnología que nos permite a día de hoy disfrutar de sus ventajas, así como de favorecer el crecimiento y el desarrollo formativo de los estudiantes, herederos de la ciencia y la ingeniería actual. También se quiere motivar a los visitantes del museo, que para muchos será la puerta de bienvenida a los estudios universitarios [8].

Sin embargo, este proyecto también tiene un aliciente personal. En los estudios de Bachillerato, cursé la asignatura de informática. En ella, aprendimos a programar con Scratch y fue fundamental en mi decisión de realizar esta carrera. Por un lado, siempre me ha parecido infravalorado el lenguaje de programación Scratch, por lo que quería demostrar lo que se podía llegar a hacer con él. Por otro lado, me parecía poético poner fin a la etapa universitaria de la misma forma que la empecé: de la mano de Scratch.

1.2 Objetivos

Con el presente trabajo se pretende alcanzar los siguientes objetivos:

1. Ahondar en las características del lenguaje Scratch con el fin de mostrar su potencial en el ámbito educacional.
2. Analizar el entorno en el que se desarrolló la calculadora Millionaire.
3. Estudiar el impacto que supuso esta invención para el mundo científico y para la sociedad en general de aquella época.
4. Examinar la estructura de la calculadora Millionaire.
5. Reproducir virtualmente el funcionamiento de la calculadora Millionaire en el lenguaje de programación Scratch.
6. Contribuir con el Museo de Informática para seguir divulgando la historia de la informática.

1.3 Estructura de la memoria

La memoria del presente Trabajo de Fin de Grado se estructura de la siguiente forma:

En el Capítulo 1, se introduce la memoria, exponiendo la motivación y los objetivos del proyecto. También se pone en manifiesto el uso que se hace la bibliografía y el compromiso del trabajo con los Objetivos de Desarrollo Sostenible.

En el Capítulo 2, se estudia el funcionamiento del lenguaje de programación Scratch, su origen y su aplicación como herramienta primaria en la introducción a la programación.

En el Capítulo 3, se explica la evolución de las herramientas de cálculo hasta llegar al nacimiento de la calculadora Millionaire.

En el Capítulo 4, se introduce el funcionamiento de la máquina, explicando su mecanismo interno, así como su manejo para cada una de las operaciones que tiene que llevar a término el usuario o usuaria de la misma.

En el Capítulo 5, se explica cómo se ha desarrollado la programación de la máquina. Este capítulo se divide en tres secciones. La primera explica cómo se han creado los objetos y sus disfraces. La segunda parte atiende a la movilidad de cada objeto. La tercera y última parte explica cómo se ha programado el código para realizar las operaciones.

Por último, en el Capítulo 6, se finaliza el proyecto con las conclusiones.

Al final del documento se encuentra la bibliografía utilizada para poder realizar este trabajo.



Figura 1.1: Listado de los 17 Objetivos de Desarrollo Sostenible (ODS), aprobados en la Conferencia de las Naciones Unidas en 2021.

1.4 Objetivos de Desarrollo Sostenible

En 2012, se celebró en Río de Janeiro, Brasil, la Conferencia de las Naciones Unidas sobre el desarrollo sostenible. En ella se listaron 17 retos a solucionar antes de 2030, con el fin de crear un mundo más próspero y reducir las brechas económicas y culturales entre los distintos países. Estos objetivos se pueden apreciar en la Figura 1.1.

En este proyecto se ha querido aportar un pequeño grano de arena a esta meta, apoyando el objetivo número 4 y el número 9: «Educación de calidad» e «Industria, innovación e infraestructura», respectivamente, que abordamos en los siguientes apartados.

1.4.1. Educación de calidad

La educación es uno de los pilares fundamentales para la evolución y el progreso de la sociedad, favoreciendo el crecimiento socioeconómico y permitiendo alejarse de la pobreza. La simulación de la calculadora Millionaire aquí creada, busca acercar una parte de la historia de los ordenadores a cualquier parte del mundo en un formato atractivo y cómodo para el aprendizaje.

1.4.2. Industria, innovación e infraestructura

La innovación e infraestructura que permiten y potencian la industrialización sostenible e inclusiva requiere de una importante inversión. Este proyecto aporta innovación, creado una virtualización de una de las máquinas de cálculo más importantes de la historia y programándola en el lenguaje Scratch.

1.5 Uso de la bibliografía

En este proyecto, se ha utilizado el artículo *El Museo de Informática de la UPV como puerta de entrada a las enseñanzas técnicas universitarias: compromiso con la sociedad en sus actividades didácticas* [8] para entender la función difusora del Museo de Informática y ha servido como inspiración para realizar este trabajo.

Los libros de *Guía de Referencia Scratch 1.4* [3] y *Guía de Referencia Scratch 2.0* [5] los he utilizado como base para entender y explicar el funcionamiento del lenguaje Scratch en el segundo capítulo de esta memoria.

Los libros *Historia universal de las cifras* [4], *Las matemáticas a lo largo de la historia: de la prehistoria a la antigua Grecia* [6] y *Historia del cálculo* [1] han servido para tener una visión global de la evolución del cálculo a lo largo de los siglos. Y gracias al libro *The Abacus: A Brief History* [2], he podido adentrarme en el funcionamiento de las herramientas de cálculo predecesoras de la Millionaire. También se ha utilizado el libro *Historia de Europa en el Siglo XIX* [7] para entender el contexto socioeconómico del siglo XIX que favoreció la invención de la calculadora.

Por último, a partir del libro *Descripción técnica de la calculadora Millionaire* [9], se comprendió el funcionamiento interno de la calculadora, que permitió desarrollar el código que la imita.

1.6 Relación con asignaturas del grado

Este proyecto ha sido posible gracias a los conocimientos adquiridos en el grado de informática. Las asignaturas a las que más he recurrido durante la elaboración de este trabajo han sido *Fundamentos de computadores* y *Arquitectura e ingeniería de computadores* a la hora de analizar los mecanismos con la misma metodología empleada en estas materias, separando los elementos como si fuesen partes aisladas y comprendiendo lo que hace cada unidad para acabar entendiendo el conjunto.

También me he apoyado en las asignaturas *Introducción a la Informática y a la Programación* y *Programación* a la hora de analizar cómo iba a programar la máquina y a implementar dicho análisis.

Por último, he recurrido a los conocimientos obtenidos en *Interfaces Persona-Computador* para realizar la interfaz y los movimientos de los objetos, así como la interacción entre ellos.

1.7 Comentarios del autor

La palabra *Millionaire* es un extranjerismo. Como su aparición en el texto es repetitiva, se ha decidido no utilizar la cursiva a lo largo de esta memoria para referenciarla.

CAPÍTULO 2

El entorno de programación Scratch

En este capítulo se describe el entorno de programación sobre el que diseñaremos la calculadora Millionaire. En concreto, se explicará por qué se ha escogido Scratch y se hará una explicación sobre su origen y sus características.

2.1 ¿Por qué Scratch?

Para responder a «¿Por qué vamos a utilizar Scratch?» hay que entender previamente qué es Scratch.

Scratch es un software de libre acceso con versiones en línea y local, esta última previa descarga. Está disponible en casi 50 idiomas y tiene más de 2 millones de descargas. El objetivo clave de Scratch es introducir a la programación a aquellas personas interesadas que no tienen experiencia previa. Scratch no enseña a programar en lenguajes como Java o C+, sino que enseña a cómo tienes que pensar cuando programas. Está diseñado para aprender por cuenta propia gracias al *feedback* inmediato que se obtiene al ejecutar un programa.

Se ha escogido Scratch para realizar este proyecto porque es necesario hacer ver que se trata de un lenguaje más potente de lo que comúnmente se cree. La facilidad de su sintaxis y la simpleza de sus bloques hace que se menosprecie por el sector informático. Sin embargo, su capacidad para permitir la concurrencia y evitar las condiciones de carrera hacen de él un lenguaje realmente interesante. Además, se busca incluir los proyecto

2.2 Origen y evolución de Scratch

El nombre de Scratch proviene de la técnica de *Turntablism* que consiste en producir sonidos rítmicos a partir de la variación de la lectura y rotación de los discos de vinilo. Scratch absorbe su nombre para enfatizar que su objetivo es mezclar bloques de código, sonido y animaciones para realizar un programa personal, único y creativo.

Además, en informática se reconoce el término *scratching* como la reutilización de código para crear nuevos proyectos. Esta es una de las características principales de Scratch. Así pues, todo usuario puede descargarse proyectos públicos, observar su código e inspirarse en él o reciclar las partes que le interesen.

Scratch empezó como proyecto en 2003 por el grupo Lifelong Kindergarten pero no fue lanzado hasta 2007 públicamente en su web. Surgió por la necesidad y el interés que mostraban los niños y jóvenes de entre 8 y 16 años por introducirse a la programación. En ese momento, la herramienta Scratch estaba disponible únicamente en la versión de escritorio. En la página web, los usuarios podían subir sus proyectos, elaborar galerías con sus propios proyectos y consultar los de otros usuarios [3].

Sin embargo, con la implementación de Scratch 2.0 en mayo de 2013 [5], se permitió crear, editar y ejecutar los proyectos en el navegador directamente a través de la página web, sin necesidad de instalar el programa en modo local. Aunque este fue el cambio más significativo de la actualización hubo otras mejoras interesantes:

- La posibilidad de crear nuevos bloques con sus propias funciones y procedimientos.
- Aparición de dos nuevas categorías: «Eventos» y «Más bloques».
- La implementación de *backpack* que permite mover y copiar diferentes elementos (disfraces, objetos, programas, bloques de código, etc.) de un proyecto a otro.

En sus inicios, Scratch se utilizó en aprendizajes informales como bibliotecas u hogares pero fue incrementando su aparición en escuelas. Se incorporó un sistema de creación, importación y manipulación de archivos media para facilitar, motivar e instar a los jóvenes a crear sus propios proyectos y aprender a partir de la exploración y la compartición.

2.3 Programación con Scratch

Como se aprecia en la Figura 2.1, la programación en Scratch se basa en bloques coloridos que representan partes de código (condiciones, variables, expresiones, etc.) evitando así los típicos problemas sintácticos que aparecen en los inicios del aprendizaje.

La interfaz del programa es sencilla, intuitiva y fácilmente entendible por todos las personas. Se basa en una única ventana con un diseño multipanel que permite que los distintos bloques de código estén siempre visibles, facilitando así la navegación e invitando a programar. Al crear un proyecto, este se inicia con un objeto gato, que es el icono tipo del lenguaje.

«Una de las claves de Scratch es que está siempre vivo»¹. Con esta declaración, Maloney y Smith ponen de manifiesto que no hay distinción entre el paso de

¹Maloney, Resnick, Rusk, Eastwood y Silverman. *The Scratch Programming Language and Environment*. ACM Transactions on Computing Education, noviembre, 2010.

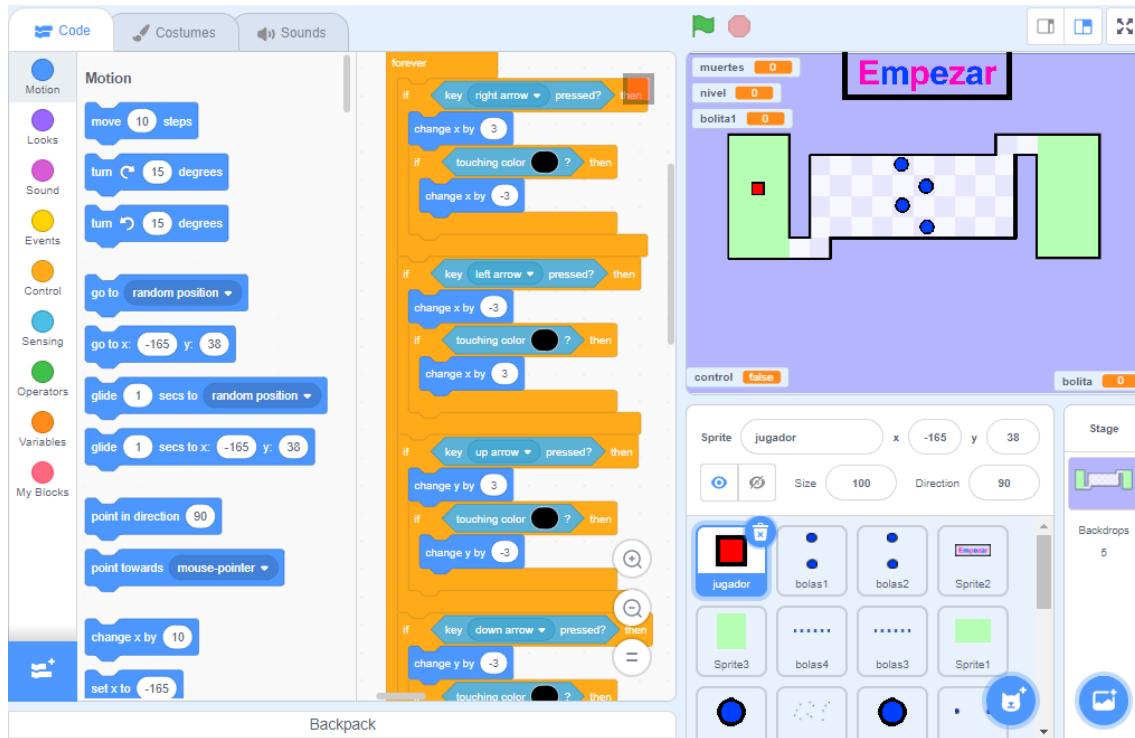


Figura 2.1: Interfaz general de Scratch. Se muestran los diferentes paneles del programa.

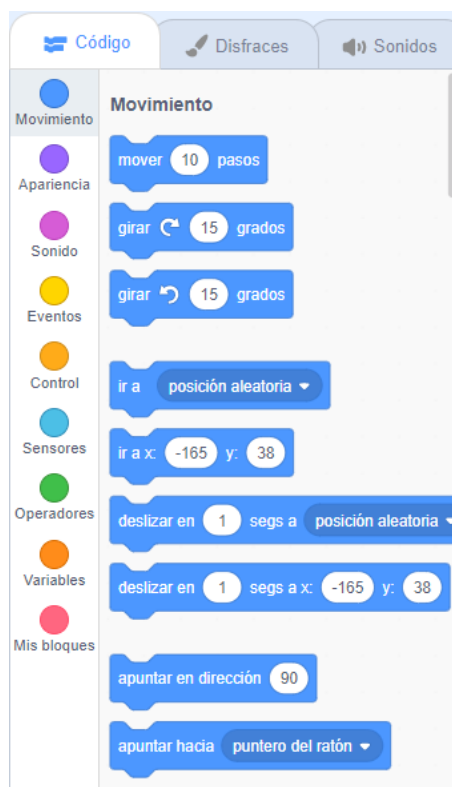


Figura 2.2: Panel izquierdo de la interfaz de Scratch donde se muestran las distintas categorías de bloques adentrándose en la sección de «Movimiento».

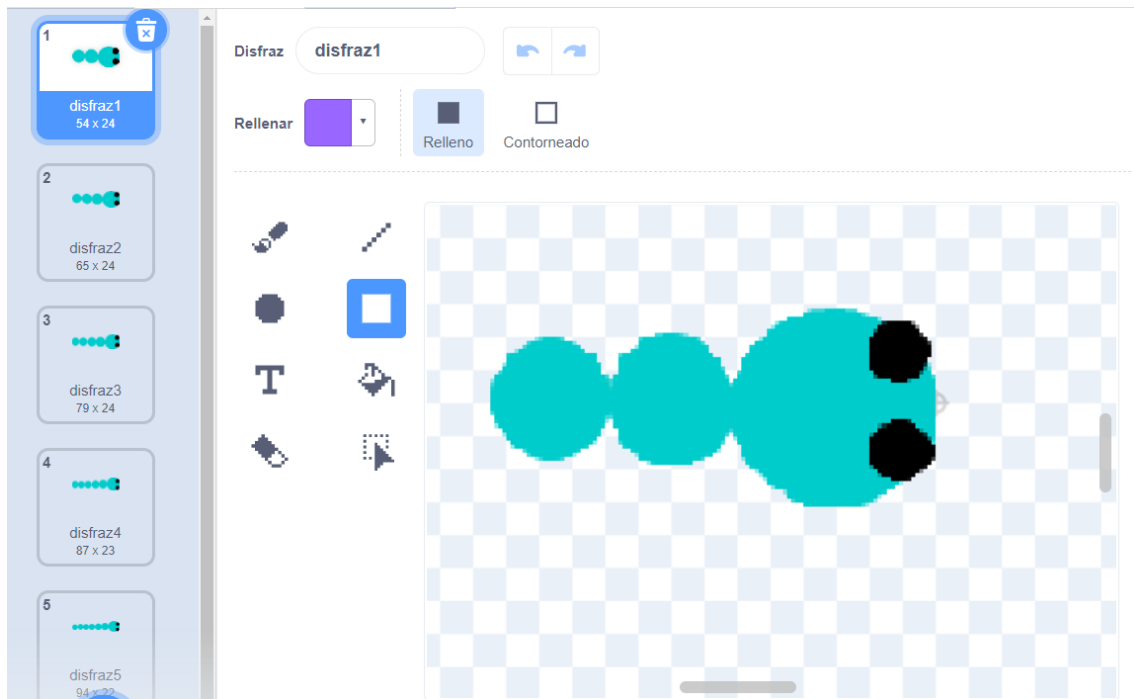


Figura 2.3: Interfaz para la personalización de los disfraces del objeto. Se permite crear o importar los diseños.



Figura 2.4: Interfaz para la carga y modificación de sonidos.

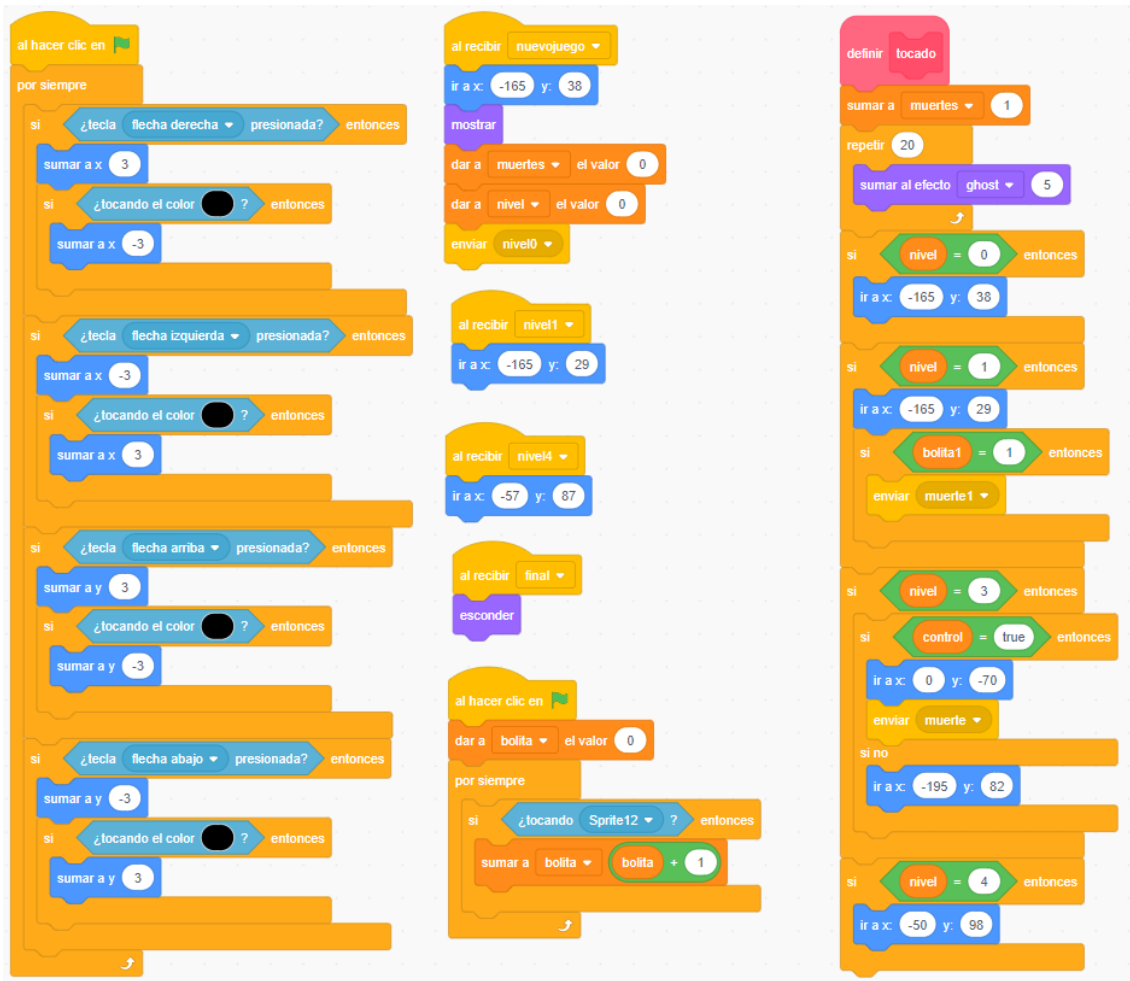


Figura 2.5: Panel central de Scratch en el que se muestra un código creado para mostrar el diseño de la interfaz principal.

compilación y el de ejecución/edición. El usuario puede hacer clic en un comando o fragmento de código para analizar qué hace en cualquier momento, gracias a que Scratch permite ejecutar los proyectos antes de tener los *scripts* completos creados. Asimismo, también se permite cambiar parámetros o añadir bloques en tiempo de ejecución.

Para facilitar a los usuarios la exploración, cada uno de los bloques tiene parámetros predefinidos que sirven como demostración para ayudar a saber lo que hacen.

Scratch no tiene mensajes de error. Esto es posible a que los bloques de código se fijan entre ellos solo cuando tiene sentido que lo hagan. Además, Scratch también elimina los errores en tiempo de ejecución haciendo todos los bloques *failsoft*, algo así como suaves al fallo. En vez de fallar con un mensaje de error, todos los bloques intentan hacer algo con sentido incluso cuando presentan fallos del tipo fuera de los límites.

2.3.1. Paneles

Como ya se ha dicho, Scratch se basa en un sistema de paneles múltiples. Cada uno de los paneles tiene una función distinta. El de la izquierda presenta las opciones disponibles para implementar los códigos separados en 9 categorías. El panel central muestra el código creado. En los paneles superior e inferior derechos se muestra la escena actual y los distintos objetos presentes en la escena, respectivamente.

El primero de ellos, el panel de la izquierda (véase Figura 2.2), presenta tres pestañas distintas:

- La primera pestaña es la de código, en ella se muestran los distintos bloques de código. Los bloques se clasifican por colores en las siguientes categorías: movimiento, apariencia, sonido, eventos, control, sensores, operadores, variables, mis bloques.
- La segunda pestaña es la de disfraces, que permite modificar e implementar diferentes diseños para un mismo objeto. De este modo, se consigue alternar la apariencia en tiempo de ejecución. Los disfraces se pueden crear de tres formas distintas: dibujando el disfraz a partir del editor mediante el botón «pintar», importando un archivo JPG, PNG, BMP o GIF a través del botón «importar» o a partir de la cámara web, por medio del botón «camera». Como se puede apreciar en la Figura 2.3, la interfaz de disfraces cuenta también con una lista de las apariencias disponibles para el objeto sobre el que se tiene el foco. En el marco central aparece el disfraz actual que se está modificando. Scratch cuenta con un sencillo editor de imágenes que permite dibujo libre, añadir texto, rellenar de un determinado color, borrar e insertar figuras geométricas.
- La tercera y última pestaña es la de sonidos. Está dirigida al diseño e implementación de distintos efectos de audio sobre distintas acciones del código. Scratch reconoce archivos en formato MP3 o los formatos no comprimidos WAV, AIF y AU codificados en 8 bits o 16 bits. Además, Scratch cuenta con

una amplia librería de sonidos disponibles y listos para incorporar al código. Observando la Figura 2.4, vemos que la interfaz de sonido tiene una columna a la izquierda con una lista de los audios ya creados. En la parte central se muestra el sonido sobre el que se está trabajando. Este se puede modificar a partir de los botones inferiores de una manera sencilla.

En la Figura 2.5 se muestra el panel central. Está formado por una pantalla móvil con todas las construcciones de código realizadas por el usuario.

En el panel inferior derecho se destacan todos los objetos en forma de miniatura. Permite centrar el foco sobre cada objeto y cambiar la visibilidad de este (visible o no), así como el tamaño, la posición y la dirección que tiene.

El último panel, el superior derecho, se muestra en tiempo real la apariencia que tiene el escenario y los objetos que están en escena. Está representado por un plano cartesiano con tiene 480 unidades de ancho y 360 de alto. Las coordenadas (0,0) marcan el centro.

Hay un quinto panel oculto en la parte inferior, la *backpack*. Este panel se encarga de almacenar los bloques de código, disfraces y objetos arrastrados por el usuario con el fin de copiarlos a otros proyectos.

2.3.2. Bloques

Aunque los bloques en Scratch se dividen en nueve categorías separadas por colores de acuerdo a su funcionalidad, se definen tres tipos principales de bloques:

- Bloques *Stack*: tienen una muesca en la parte superior y un saliente en la parte inferior que les permite encajarse entre ellos y formar pilas de código.
- Bloques *Hats*: tienen la parte superior redondeada y una muesca en la parte inferior. Son los bloques encargados de la inicialización y de capturar eventos que determinaran el comienzo de la ejecución de esa pila.
- Bloques *Reporters*: se incorporan dentro de otros bloques. Visualmente, la forma del bloque representa el tipo de dato que contiene. Cada bloque de este tipo tiene forma redondeada (*number* o *string*) o con los bordes acabados en punta (*boolean*). Estos bloques solo encajarán en los huecos de las funciones que devuelvan ese tipo de dato.

De acuerdo a su funcionalidad, Scratch realiza una división de los bloques por categorías a las que asigna un color:

- El primero de todos engloba a los bloques de movimiento. Están representados por el color azul. La funcionalidad de estos bloques es la de desplazar o rotar los diferentes objetos por el escenario.
- En segundo lugar, representados por el color morado, están los bloques de apariencia. Se encargan de variar el aspecto de los objetos. Permiten cambiar el tamaño, dar efectos o variar los disfraces.

- La tercera categoría es la de sonidos. Esta categoría permite ejecutar los sonidos importados o creados, cambiar el volumen o asignar tiempos de silencio. El color fucsia diferencia este grupo.
- La gestión de eventos se realiza a través de los bloques amarillos. En su mayoría son bloques de inicio del programa (bloques *Hats*) que se ejecutan cuando reciben una señal. Pero también hay bloques que permiten enviar señales.
- El color naranja en un tono suave representa a los bloques de control. Permiten incorporar bucles o condiciones.
- Los sensores se gestionan a través de los bloques cían. Comprueban condiciones sobre los objetos.
- Para manejar los operadores sobre las variables, ya se trata de sumas, restas, igualdades o demás operadores, se utilizan los bloques de color verde.
- El tono oscuro del naranja representa las variables creadas.
- Por último, en rosa claro se muestran los bloques creados por el propio usuario y que determinan características u operadores que el propio lenguaje no ha definido.

El color naranja se utiliza tanto para las variables (tono oscuro) como para los bloques de control (tonalidad claro). El azul cielo guarda los sensores y el verde los operadores. Por último, el color magenta almacena los bloques propios, creados por el usuario.

2.3.3. Objetos

Los objetos de Scratch, también conocidos como *sprites* encapsulan estados y comportamientos. Sin embargo, como Scratch no tiene clases ni herencia, es un lenguaje basado en objetos pero no un lenguaje orientado a ellos.

Los comandos solo se ejecutan en los objetos específicos en los que están asignados. Un objeto no puede ejecutar un comando que mueva directamente a otro objeto. Si dos objetos realizan las mismas funciones se tendrá que replicar el código del primero en el segundo.

Esto es así porque cada objeto tiene su propio bloque de código. Por una parte, facilita el entendimiento de lo que realiza cada objeto, al no tener que buscar entre las jerarquías o seguir la cadena de ejecución del programa. Por otra parte, es más tedioso de implementar varios objetos con las mismas características (como por ejemplo los ladrillos o *bricks* del famoso juego *Breakout*). Si el usuario ha de modificar dichos objetos una vez han sido creados, deberá hacerlo manualmente uno a uno.

Para la comunicación entre objetos se utiliza una transmisión de mensajes. Cuando un objeto transmite un mensaje de tipo String, como por ejemplo «Mensaje», desencadenará la ejecución de código en todos aquellos objetos que tengan un bloque de código del tipo: “al recibir «Mensaje»”.

Esta comunicación es de uno a muchos. Un solo mensaje puede desencadenar la ejecución de muchos programas de uno o varios objetos. También es asíncrona, el transmisor no espera a que los otros objetos acaben de ejecutarse; aunque tiene una variante síncrona que sí espera.

2.3.4. Concurrency

La concurrencia es considerada comúnmente como una técnica de los lenguajes avanzados. Aunque Scratch la desarrolla, evita los mecanismos de control como semáforos o monitores. En su lugar, Scratch utiliza restricciones para la alternancia de hilos. Solo permite el cambio de hilo de ejecución en dos situaciones: en un comando que espere explícitamente o al final de un bucle. El cambio no puede ocurrir a mitad de una secuencia continua o entre el *if* y su cuerpo, lo que garantiza de forma implícita, el soporte de secciones críticas. En el ejemplo de la Figura 2.6, nunca habrá un cambio de hilo entre la verificación de la condición de ratón pulsado y la instrucción de mover el objeto 10 pasos.

Aunque el modelo de concurrencia de Scratch elimina la mayoría de condiciones de carrera, no suprime todos los problemas. El más común es el que aparece cuando múltiples *scripts* son lanzados por un evento o un mensaje de otro objeto. El orden en el que se ejecutan no tiende a ser el esperado por los usuarios. Sin embargo, una vez que los usuarios entienden que un disparador lanza varios programas en orden arbitrario, entienden la solución: no tener un único disparador para varios programas si se desea un orden específico de ejecución.

2.4 Beneficios de usar Scratch

Scratch es un lenguaje enfocado a la introducción en la programación de niños.² La respuesta más común a por qué se enseña Scratch en las escuelas es que permite acabar con los estereotipos fijados de la informática: la informática es solo para chicos o la informática es para raros. Además, participar en actividades de programación en edades tempranas desarrolla el ingenio de los jóvenes y los alienta a seguir estudiando informática en secundaria o incluso en niveles más avanzados.

Malan y Leitner, de la universidad de Harvard, enseñaron Scratch en la primera semana como introducción a la programación antes de enseñar Java. Tras unas semanas enseñando Java, hicieron una encuesta sobre la utilidad de haber estudiado Scratch previamente. De los 25 de encuestados, 19 confirmaron que Scratch había sido una buena influencia y tan solo 2 declararon que había sido un malgasto de tiempo.

«A pesar de que no aprendemos la sintaxis de Java estudiando Scratch, aprendimos el tipo de pensamiento necesario para implementar programas simples. [...] Yo fui capaz de afrontar los primeros programas de Java con una idea de cómo

²Michal Armoni, Orni Meerbaum-Salant, Mordechai Ben-Ari. *From Scratch to "Real" Programming*. ACM Transactions on Computing Education, febrero, 2015.

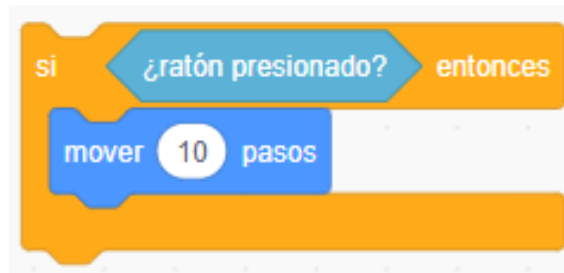


Figura 2.6: Bloque de código creado para explicar el fenómeno de la sincronización por bloqueo en Scratch.

mo encarar los problemas. Aunque no supiese cómo crear un bucle “para”, sabía cuándo era necesario porque ya los había usado en Scratch.»³

³David Malan, Henry Leitner. *Scratch for Budding Computer Scientists*. SIGCSE Technical Symposium on Computer Science Education, marzo, 2007. Traducción personal.

CAPÍTULO 3

Contexto histórico

En este capítulo se describirá el origen y evolución de herramientas de cálculo. De esta forma, se buscará entender cómo surgió la calculadora Millionaire.

3.1 Evolución de las herramientas de cálculo

Es bien sabido que el ser humano ha utilizado sus manos como principal instrumento a lo largo de la historia [2]. Por eso, no es de extrañar que la primera herramienta conocida para el cálculo fueran los dedos de las manos; sin embargo, algunas tribus del trópico también utilizaban los dedos de los pies. Aunque parezca que la capacidad de cálculo estaba muy limitada, era común el uso de nudillos o falanges para representar las docenas, las centenas o, incluso, los millares [4].

Maderas, huesos o piedras alargadas talladas fueron el primer instrumento de conteo en ser inventado. Sobre estos objetos se realizaban incisiones que representaban una unidad. De este modo era posible realizar conteos de números relativamente grandes de forma permanente. En la Figura 3.1, se observa el hueso de Ishango que data del 20.000 a. C., expuesto en el Real Instituto Belga de Ciencias Naturales. Este hueso del peroné de un babuino es una muestra de la utilización de objetos como palos de conteo.

El uso de grupos de pequeñas piedras también tiene una incuestionable antigüedad, aunque por su fugacidad resulta una tarea de alta dificultad datar esta herramienta. Sin embargo, lo que sí es conocido es que en la antigua Grecia y en la antigua Roma utilizaban las piedras como método para enseñar aritmética a los niños.

El uso de piedras para el cálculo derivó en primera instancia en los tableros de cuentas. Estas herramientas eran tablas generalmente de piedra o madera sobre la que se realizaban unos surcos para dividirla en secciones. Sobre cada sección se colocaban pequeños guijarros que tomaban un determinado valor dependiendo de la sección en la que se encontrasen. Estas tablas de contar pudieron tener su origen en Sumeria y son consideradas por algunos expertos como ábacos primitivos [2].

A través de intercambios comerciales, este ábaco rudimentario llegó a las civilizaciones griega y romana, que llamaron a este instrumento *abak* y *abacus*, respec-



Figura 3.1: Hueso de Ishango datado en el Paleolítico. Expuesto en el Real Instituto Belga de Ciencias Naturales. Hallado en el Congo Belga en 1960 por Jean de Heinzelin.

tivamente. Ambas se pueden traducir por la palabra «tabla». El primer ábaco romano o griego del que se tiene constancia, el ábaco de Salamina, fue hallado cerca de la isla de Salamina, en Grecia, y data del siglo V a.C. Este ábaco es un rectángulo de mármol blando con dos grupos de once y cinco líneas verticales cortadas de forma perpendicular por una línea horizontal, de modo que se forman dos grupos de veinte y ocho rectángulos abiertos por la cara externa. El ábaco también cuenta con anotaciones en griego antiguo que representan cantidades en el sistema de numeración de la época. Ifrah declara que los rectángulos superiores representarían múltiplos de 10 de las monedas griegas más valiosas, *dracmas* o talentos, y la parte inferior sería usada por las monedas con menor valor como *óbolos*. En la Figura 3.2 podemos apreciar este ábaco.

La Edad Media estuvo dominada por el uso del ábaco moderno debido a que con el sistema numérico romano, vigente en Europa, era muy complejo realizar operaciones aritméticas. Por ello, su uso estuvo muy popularizado entre recaudadores de impuestos, comerciantes y demás profesiones que requerían de cálculos.

Este ábaco moderno fue un utensilio portátil de origen romano. Estaba fabricado a partir de una plancha de metal rectangular sobre la que se realizaban incisiones donde irían las distintas piezas para hacer las cuentas. Las columnas de la derecha se usaban para realizar los cálculos sobre la moneda romana más pequeña de la época, el *unciae*. El resto de columnas servían para representar potencias de diez. En cada columna, las piezas de la parte inferior tenían un valor de 1 cada una de ellas; mientras que las piezas de la columna superior valdrían 5. De este modo, en cada columna se podía representar desde el 0 hasta el 9. La distribución de las columnas se puede observar en la Figura 3.4.

El único cambio que sufrió el ábaco en este periodo vino de la mano del monje Gerberto de Aurillac, que posteriormente se convertiría en el Papa Silvestre II. Gerberto estudió en sistema numérico árabe a finales del siglo X en su estancia en *Al-Andalus*. Aunque trató de introducirlo en la Europa cristiana, no fue hasta finales del siglo XII cuando se comenzó a popularizar su uso.

Lo que sí consiguió este religioso fue introducir unas pequeñas modificaciones en el ábaco que fueron popularizándose gradualmente: sustituir las actuales piedras usadas sobre el tablero (cada una de las cuales tenía un valor base de uno)

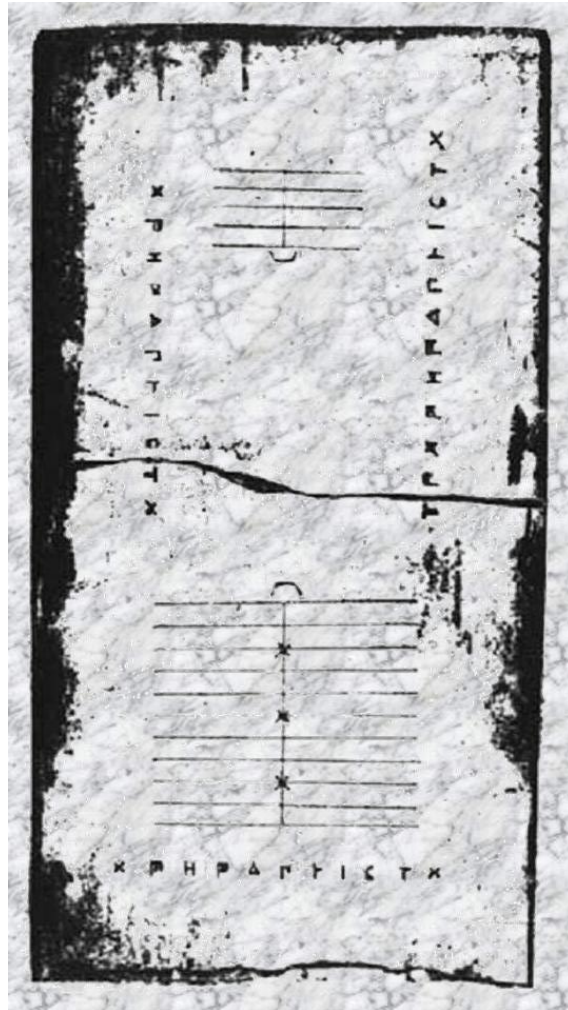


Figura 3.2: Ábaco de Salamina hallado en el siglo XIX cerca de la isla de Salamina. Data del siglo V a. C. y es el ábaco griego más antiguo descubierto hasta la fecha.

por unas piezas numeradas con las representaciones numéricas de los árabes del uno al nueve. De esta forma, se reducía considerablemente las cuentas realizadas sobre la mesa y se simplificaba el uso del instrumento.

Con la introducción y generalización del sistema numérico árabe se generó un enfrentamiento entre «algoristas» y «abacistas». Los primeros defendían el uso de los números árabe y la realización de los cálculos a papel. Por otro lado, los «abacistas» defendían el uso de la numeración tradicional romana y la utilización de los ábacos para realizar los cálculos aritméticos. Como se puede ver en el índice o el número de las páginas de este documento, Europa se decantó por el uso del sistema decimal árabe a finales de la Edad Media, eliminando así de forma gradual el ábaco de mesa.

Durante la Edad Media, en China, se desarrolló un ábaco muy similar al romano, que recibía el nombre de *suanpan* (tabla de cálculo en chino), del que no se tiene constancia hasta el siglo XII. Algunos indicios apuntan a que este ábaco surgió debido a los contactos comerciales entre chinos y romanos, aunque no hay pruebas concluyentes que lo demuestren. Este ábaco resulta de especial interés debido a que es el único que su uso sigue vigente a día de hoy.

Este instrumento se compone de un marco rectangular de madera dividido en dos regiones. Cada una de los espacios es atravesado por trece varillas, aunque la cantidad puede variar, variando la potencia de cálculo. Sobre cada una de estas varillas, que también están separadas en dos regiones, se depositan un total de siete piezas, cinco en la parte inferior, que tienen un valor base de uno, y dos sobre la parte superior, cuyo valor es de cinco. Esto quiere decir que en cada varilla se puede llegar a representar hasta un total de quince unidades. El valor de cada varilla aumenta en potencia diez de derecha a izquierda. Tradicionalmente, la varilla que representa las unidades ha sido la tercera empezando por la derecha, reservándose así dos varillas para representar los decimales, de modo que, de derecha a izquierda y en base 10, los exponentes quedarían: $-2, -1, 0, 1, 2$, etc. Este ábaco de origen chino puede observarse en la Figura 3.3.

La Edad Moderna estuvo marcada por los avances en tecnología y en conceptos matemáticos, que permitieron la aparición de nuevos sistemas de cálculo más complejos. En 1623, el alemán Wilhelm Schickard desarrolló la primera máquina capaz de calcular, a la que bautizó como «reloj calculante». Este aparato era capaz de realizar sumas y restas de forma autónoma, necesitando únicamente que un usuario introdujese los valores. También se podían realizar multiplicaciones y divisiones pero estas operaciones sí precisaban de la participación directa del usuario. El funcionamiento del reloj se basaba en un mecanismo interno formado por engranajes y cilindros que implementaban la lógica aritmética de las operaciones matemáticas. Schickard solo construyó una de estas máquinas, que fue quemada el año posterior, por lo que el reloj calculante original no ha llegado a nuestros días. Se tiene constancia de su funcionamiento gracias a la correspondencia mantenida entre el inventor y su amigo Kepler y a los bocetos y notas hallados. Gracias a estos hallazgos, podemos observar en la Figura 3.5 como debió ser aquel instrumento.

El matemático francés Blaise Pascal desarrolló dos décadas más tarde, en 1642, una máquina similar al reloj calculante de forma totalmente original. Este aparato, bautizado por el propio Pascal como la «Pascalina», o *Pascaline*, era capaz de

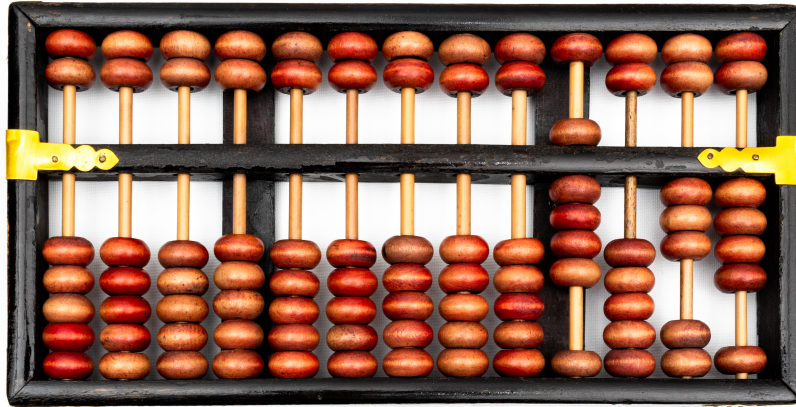


Figura 3.3: Ábaco de mano moderno de origen chino o *suanpan*. Su aparición data del siglo XII. Las dos columnas de la derecha se usan para representar las decimales. De este modo, la tercera columna está reservada para las unidades.

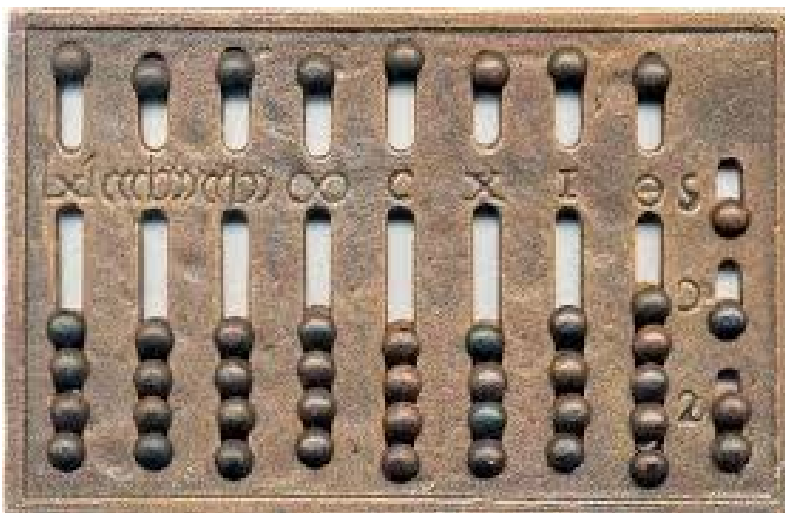


Figura 3.4: Ábaco de mano moderno de origen romano. Las columnas de la derecha se usaban para realizar cálculos sobre la moneda romana más pequeña. El resto de columnas se basaban en potencias de 10.



Figura 3.5: Réplica del reloj calculante inventado en 1623 por el astrónomo Wilhelm Schickard. Se trata de la primera máquina capaz de realizar operaciones aritméticas simples de forma autónoma.

realizar sumas mecánicamente, mostrando el resultado por las ranuras, gracias al sistema de ruedas y engranajes. La resta se podía realizar mediante la suma del complementario. Aún pareciendo muy limitada, la máquina tuvo una gran popularidad e inspiró a muchos matemáticos a hacer modificaciones y mejoras. En la Figura 3.6 se puede observar una réplica de la máquina original.

Unos años más tarde, en 1694, el matemático alemán Gottfried Leibniz fabricó la primera máquina capaz de realizar las cuatro operaciones aritméticas básicas semiautomáticamente. Solo era necesaria la acción del usuario para realizar algunos procesos mecánicos. Sin embargo, dichos procesos eran de suma complejidad y provocaban fallos humanos y dificultades en el funcionamiento. Todo esto, sumado a unos pocos fallos en la implementación suponía que fallasen al-



Figura 3.6: Réplica de la máquina de calcular inventada por el francés Blaise Pascal en 1642. La Pascalina, nombre que recibió el invento, tuvo una gran popularidad pese a ser capaz únicamente de realizar sumas de forma automática.

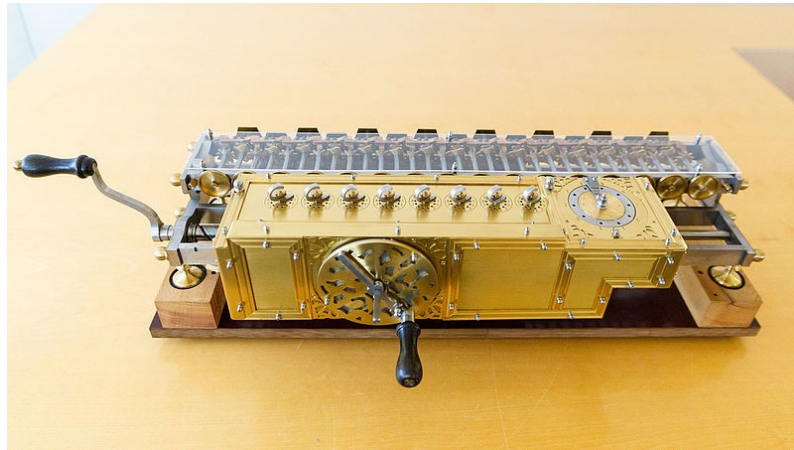


Figura 3.7: Réplica de la máquina de calcular inventada por el alemán Gottfried Leibniz en 1694. Este invento era capaz de realizar las cuatro operaciones aritméticas sencillas de forma semiautomática. Su uso no se popularizó debido a la complejidad de los procesos mecánicos necesarios para hacerla funcionar.

gunos acarreos. Esta máquina tenía como núcleo un cilindro o rueda conocido como «cilindro de Leibniz», que, como su nombre indica, era un cilindro dentado que rotando activaba la rueda de conteo del instrumento. Aunque el invento de Leibniz nunca llegó a popularizarse debido a las imperfecciones, sirvió de inspiración para las máquinas posteriores, que en su mayoría adoptaron el cilindro de Leibniz como motor de cálculo. Podemos observar una réplica de esta máquina en la Figura 3.7

3.2 La sociedad del siglo XIX

El siglo no es más que una medida temporal artificial. Hasta 1790, el término «siglo» no tenía ninguna importancia para la sociedad. Es entonces cuando los autores del momento empezaron a escribir sobre el siglo XIX. Este apelativo empezó a difundirse entre los títulos de innumerables libros por América y Europa, haciendo del siglo XIX el primer siglo en ser nombrado por un número.

Pero esta designación de «siglo XIX» vino acompañado de modernidad y cambios. Comenzó una nueva era, el inicio de la globalización, impulsada por la industrialización, el asentamiento de nuevas ideas políticas, el avance en los medios de comunicación y de transporte y el incremento de las migraciones internacionales y del comercio, propiciadas por las nuevas formas de colonización.¹

El aumento del comercio y de la migración incrementó notablemente el tráfico marítimo y ferroviario. Los barcos mercantes y militares buscaban pasar el menor tiempo posible en tierra para evitar tiempos ociosos y aparece una voluntad por sincronizar los trenes. Esto llevó a la popularización del reloj entre marinos, mercaderes y empresarios. El incremento de demanda de relojes impulsó que algunos joyeros y fábricas intentaran reducir el coste y el tiempo de fabricación de estos para aumentar sus ventas. Este trabajo permitió reducir el tamaño de los mecanismos de los relojes.

¹Sylvain Venayre *A Tour of the 19th Century. A History of the World through Objects*. Abril, 2019.



Figura 3.8: Máquina de cálculo diseñada por el francés Thomas de Colmar en 1820. Su facilidad de uso y estética la popularizó para el gran público.

La industrialización supuso un aumento en el tamaño de las fábricas y de sus empleados, a la vez que las nuevas ideas políticas favorecían las condiciones laborales de los trabajadores. El crecimiento económico y el aumento en el volumen de comercialización, así como las nuevas obligaciones legislativas respecto a la contabilidad financiera, rogaban nuevas herramientas de cálculo que agilizaran las operaciones.

En este siglo apareció la primera máquina de cálculo que realmente alcanzó popularidad más allá del ámbito científico, gozando de éxito comercial: el «Aritmómetro» o *Arithmomètre*, desarrollado por el ingeniero francés Charles Xavier Thomas de Colmar en 1822. Se trataba de una máquina basada en los mecanismos de Leibniz, aunque con sustanciales mejoras que solucionaban los fallos de su predecesora y algunas pequeñas funcionalidades nuevas, como la de poner los registros a cero. Esta máquina no supuso ningún avance a nivel de potencia de cálculo pero resultó muy importante debido a su funcionalidad y su facilidad de uso. En la Figura 3.8 puede observarse un ejemplar del Aritmómetro, donde se aprecia la sencillez de uso y la estética, pensadas para el gran público.

En 1872, Frank S. Baldwin inventó la calculadora de rueda dentada, que también fue desarrollada, de forma totalmente independiente, por Willgodt T. Odhner un año después. El modelo de Odhner, apreciable en la Figura 3.9, no empezaría a producirse hasta 1890 y es considerada por los expertos como una de las mejores calculadoras mecánicas de la historia. Surgieron numerosos clones a partir de ella, entre los que se encuentra el famoso modelo Brunsviga.

A finales de siglo, Dorr Eugene Fer patentó el comptómetro, la primera calculadora mecánica de tecla pulsada en Estados Unidos, visible en la Figura 3.10. Esta máquina tuvo un gran éxito empresarial, como avala sus más 85 años de fabricación y comercialización, y pese a que se concibió como una máquina para sumar, también era capaz de hacer restas, multiplicaciones y divisiones. Esta calculadora era extremadamente rápida, debido a que el acumulador se actuali-



Figura 3.9: Máquina de cálculo diseñada por Willgodt T. Odhner en 1873. Esta calculadora mecánica esta considerada una de las mejores de la historia y estuvo en producción desde 1890 hasta 1917, momento en el que la Revolución rusa obligó a cerrar el taller.

za sumando o restando el valor de cada tecla tan pronto como esta se presiona. Un usuario experto era capaz de añadir todos los dígitos de un número a la vez, usando tantos dedos como el número requería. Por ello, es que esta calculadora puede llegar a ser más veloz que las calculadoras electrónicas actuales.

La situación económica y comercial de Europa y América y el avance en la tecnología de las herramientas de cálculo permitió la aparición de la calculadora Millionaire.



Figura 3.10: Máquina de cálculo diseñada por el Dorr E. Fer en 1884. Tuvo un gran éxito comercial debido a su facilidad de uso y su rapidez a la hora de realizar sumas y restas. Se fabricó sin interrupción desde 1886 hasta 1974. Se mantuvo en uso hasta la década de 1990, cuando se reemplazaron los últimos ejemplares por calculadoras electrónicas.

CAPÍTULO 4

La calculadora Millionaire

En este capítulo vamos a estudiar las características de la calculadora Millionaire y sus mecanismos con el fin de comprenderla y poder implementarla en nuestro código. En primer lugar abordaremos el origen e invención de esta calculadora, a continuación detallaremos los distintos aspectos que conforman su aspecto físico y, finalmente, daremos los detalles necesarios para explicar su manejo.

4.1 El origen de la calculadora Millionaire

Hasta 1887, las máquinas de cálculo realizaban las multiplicaciones y las divisiones por sumas y restas sucesivas. Pero en ese mismo año, el francés Léon Bollée inventó una máquina capaz de utilizar tablas de multiplicar mecánicas para realizar la multiplicación. Esta máquina, aunque excelente en funcionamiento, era muy pesada, cara y de gran tamaño. Esto, sumado a que Léon perdió el interés en la fabricación de calculadoras debido al auge del automóvil, supuso que se fabricasen muy pocos ejemplares y no tuviese éxito [1].

Esta máquina fue la precursora directa de la calculadora Millionaire. La calculadora Millionaire fue desarrollada y patentada por Otto Steiger (1858-1923), un científico de origen suizo con residencia en Munich, en 1893¹ (véase la Figura 4.1). La empresa Egli, dirigida por Hans Egli y localizada en Zurich, se encargó de la manufacturación y distribución de la calculadora durante 40 años. De este modo, se llegaron a fabricar 5.000 unidades de esta máquina hasta 1935. En la Figura 4.2 se puede observar un ejemplar de este invento.

La calculadora Millionaire era insuperable por su rapidez y fiabilidad a la hora de multiplicar. Se utilizaron ampliamente en los sectores que estaban en auge debido a la segunda revolución industrial: la banca, los seguros, el transporte y la minería. No fue hasta 1930, con la aparición de las nuevas calculadoras rotativas totalmente automatizadas, que decayó su popularidad y se vio superada en términos de sencillez y potencia de cálculo.

¹Otto Steiger. “The Millionaire” – *Calculating machine for direct multiplication*

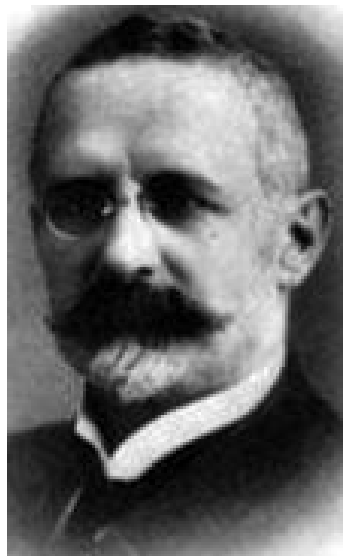


Figura 4.1: Otto Steiger, ingeniero suizo inventor de la calculadora Millionaire.

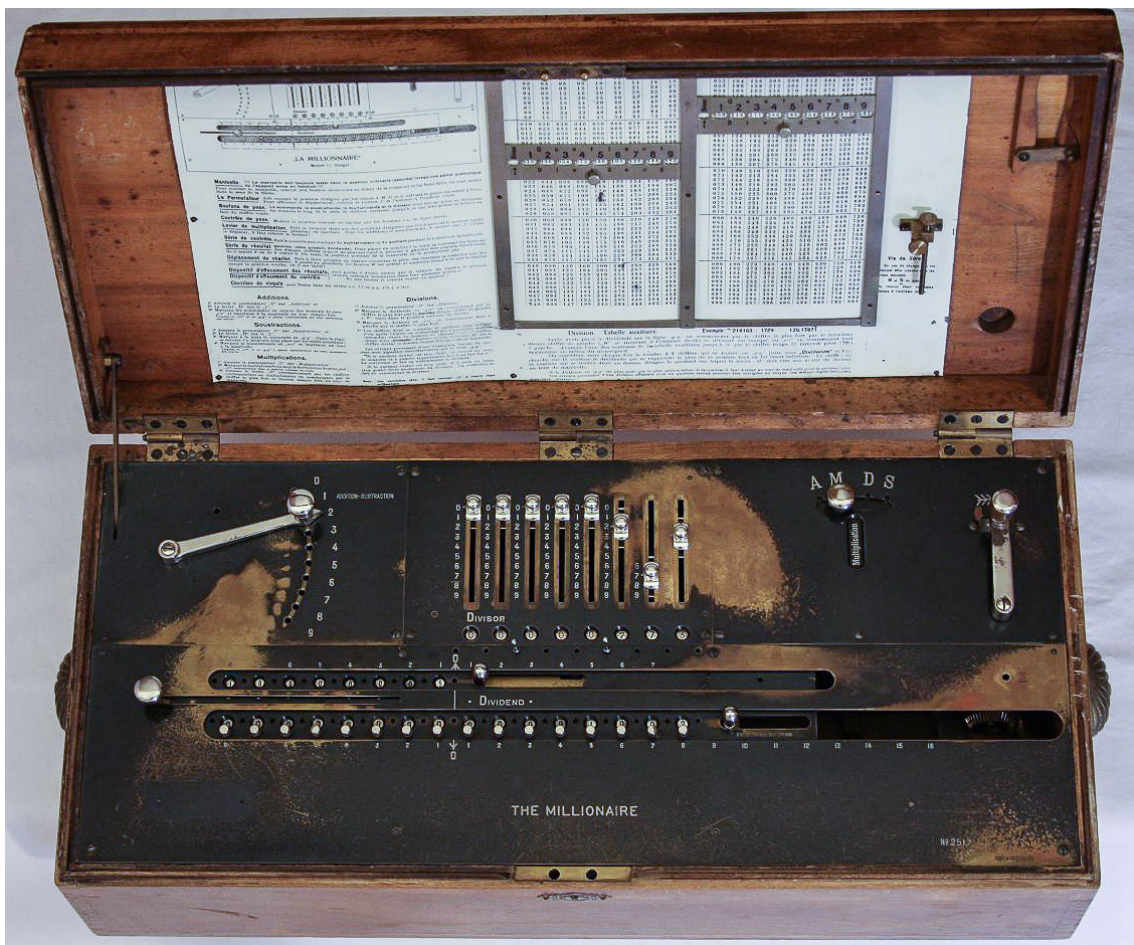


Figura 4.2: Ejemplar de la calculadora Millionaire diseñada por Otto Steiger en 1893.

4.2 Introducción a la calculadora Millionaire

La calculadora Millionaire fue diseñada como una máquina multiplicadora, no como una máquina sumadora. Verla desde esta perspectiva nos ayudará a describirla y a entenderla mejor.

4.2.1. El panel de control

El panel de control, apreciable en la Figura 4.3, tiene unas medidas de 64×26 centímetros.

Para facilitar la explicación se han enumerado las 4 partes principales del panel:

1. La palanca de control del multiplicador situada en la parte superior izquierda ajusta el multiplicador, un dígito cada vez, empezando por el más significativo. Esto se verá con más detalle en la sección dedicada a la multiplicación
2. El centro de la parte superior contiene el mecanismo de ajuste del primer operando. El mecanismo de deslizamiento, que permite seleccionar los dígitos para operar, incluye un conjunto de diales de control para ofrecer una visualización del número introducido.
3. El panel superior derecho contiene el regulador de operación, que ajusta el mecanismo para sumar, multiplicar, dividir o restar (AMDS), y la manivela de funcionamiento. La manivela da una vuelta completa en el sentido de las agujas del reloj para cada ciclo de la máquina, hasta que se apoya en un tope fijo en su posición inicial. La manivela nunca debe girarse hacia atrás.
4. La sección inferior del panel está cubierta por el carro móvil, que ocupa la mayor parte de la mitad delantera de la máquina. El panel tiene dos aberturas largas para revelar el registro del acumulador de 16 dígitos y el contador de 8 dígitos. Los mandos situados a la derecha de cada registro se tiran hacia la derecha para despejar la pantalla. El carro se desplaza automáticamente hacia la izquierda durante la multiplicación y la división, y se devuelve manualmente pulsando el pomo de su extremo izquierdo.

4.3 Características de la calculadora Millionaire

La calculadora Millionaire está formada por un mecanismo de «piñón y cremallera» (en inglés *Rack and pinion*) y permite realizar multiplicaciones directamente [9].

El primer modelo pesaba 32,66 kg, tenía unas medidas de $66 \times 30,48 \times 17,78$ centímetros y contaba con unos agarradores en los lados que facilitaban su traslado.

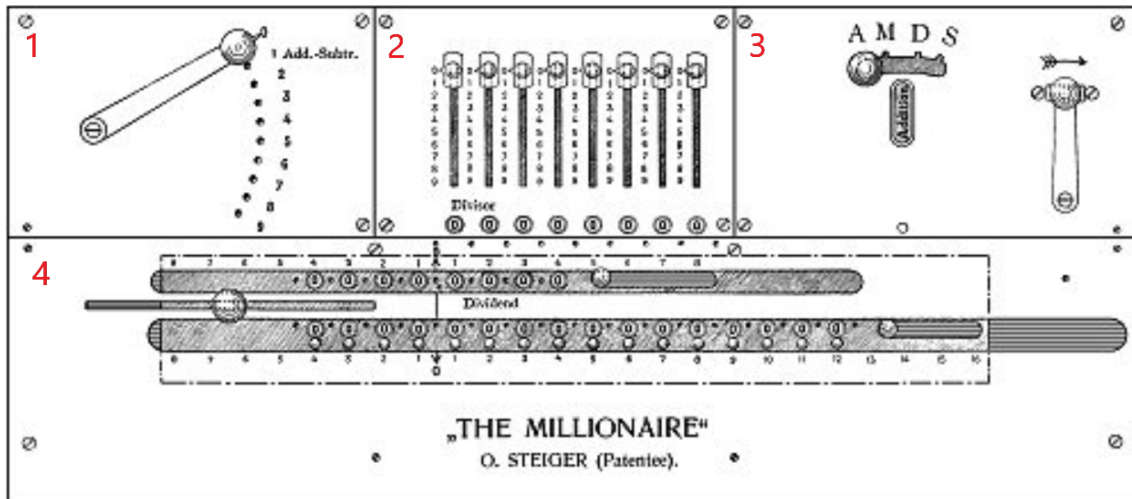


Figura 4.3: Panel de control de la calculadora Millionaire obtenido del manual de instrucciones de la máquina.

Como ya hemos comentado, la calculadora Millionaire es capaz de realizar cuatro operaciones distintas: suma, resta, multiplicación y división. La máquina cuenta con una palanca, parecida a la palanca de un coche automático, que según en qué posición se encuentre hace qué operación: en el punto más alto, la operación que realiza es la suma, representada con la «A», del inglés *add*. Si por el contrario, se coloca en segunda posición, se realiza una multiplicación, reflejada con la letra «M». La división se encuentra en el tercer peldaño de la escala, y está representada por la letra «D». Y en último lugar, la letra «S», del inglés *subtract*, fija la máquina para que opere con una resta. En la Figura 4.4 se puede apreciar esta palanca y sus cuatro posiciones.

Para seleccionar el número sobre el que operar, esta máquina dispone de ocho deslizadores. Como se observa en la Figura 4.5, cada una de estas ocho palancas cuenta con diez posiciones distintas, la más alta representa el 0 y la más baja el 9. La primera palanca empezando por la derecha representa las unidades, la segunda las decenas, la tercera las centenas; y así sucesivamente. Para facilitar la visualización del número con el que se está trabajando, la calculadora cuenta con ocho pestañas, una debajo de cada deslizador, que muestra el número seleccionado.

Cada uno de los deslizadores interactúa con un engranaje de diez muescas. Cada una de las muescas representa un dígito. Estos engranajes permiten el uso de los acarrees en las operaciones. Cuando se realiza una vuelta completa en uno de los engranajes, es decir, se pasa de la muesca que representa el dígito 9, se desplaza el engranaje inmediatamente de la izquierda en una muesca, aumentando así en una unidad.

Además de estos ocho deslizadores que seleccionan el número con el que se opera, la calculadora dispone de un deslizador a parte que permite devolver la calculadora al estado inicial, borrando el dato almacenado en resultado.

Una de las partes más importantes y llamativas de la máquina es la manivela situada en la parte derecha, que se puede observar en la Figura 4.6. Dando una



Figura 4.4: Palanca de la calculadora Millionaire utilizada para seleccionar el tipo de operación realizar. En la imagen se aprecia la ubicación de la palanca en la posición para efectuar la operación de división, y encima de la misma se pueden leer las letras A, M, D (casi tapada) y S.

vuelta a esta palanca en el sentido de las agujas del reloj, se realiza el cálculo de la operación introducida.

La calculadora cuenta con un sistema de engranajes y cremalleras que permiten realizar los cálculos en la multiplicación. Este mecanismo se explicará más adelante, en el apartado dedicado al funcionamiento de la multiplicación.

4.4 Funcionamiento de la calculadora Millionaire

Lo primero que hay que saber para utilizar la calculadora Millionaire es que, para realizar el cálculo, la máquina tiene en cuenta el tipo de operación introducido, el número marcado con los deslizadores y el número almacenado en la máquina.

Cabe destacar también, que si queremos realizar una operación completamente nueva, es decir, sin utilizar valores calculados previamente, es necesario activar el deslizador de reinicio de valores, que devolverá la máquina a la situación inicial y permitirá realizar cálculos independientes de los anteriores.

Es interesante incidir en que la manivela actúa como la tecla «= \Rightarrow » de las calculadoras actuales. De este modo, si se le da una vuelta a la manivela, hará una vez la operación. En cambio, si se rota tres veces la manivela se calculará esa operación tres veces.

Ahora, vamos a explicar cómo realizar cada una de las operaciones.²

²Nota: para explicar las operaciones se parte siempre de una situación inicial con todos los valores a 0.

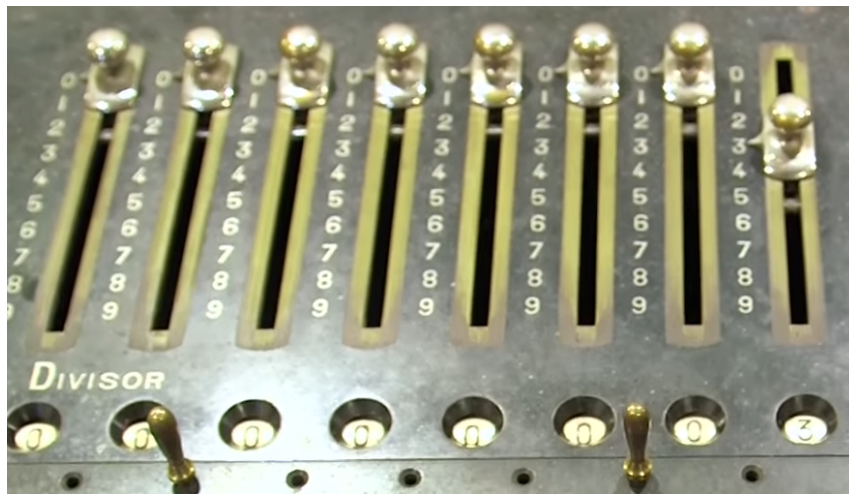


Figura 4.5: Palancas de la calculadora Millionaire utilizada para seleccionar el número con el que se quiere operar. Estas palancas se deslizan de abajo arriba y viceversa, hasta situarse en la cifra pertinente. En la imagen se aprecia que la palanca de la derecha ha seleccionado el número 3, el cual aparece en la parte inferior.



Figura 4.6: Manivela de la calculadora Millionaire utilizada para efectuar la operación aritmética entre el número que se tenía almacenado en la máquina con el valor introducido manualmente con los deslizadores.

4.4.1. Suma

La suma es la operación más sencilla en la calculadora Millionaire. Para realizarla tan solo es necesario poner la palanca del operando en la «A» de *add*, introducir uno de los sumandos en los deslizadores y añadir el otro sumando en la barra de resultado, haciendo uso de los pequeños mecanismos rotatorios que dispone la barra de resultados. Por último, es necesario rotar la manivela de resultado.

También es posible realizar la suma en dos pasos, haciendo uso únicamente de los deslizadores. Para ello, se introduce el primer sumando en los deslizadores y se rota la manivela de resultado. De este modo, se suma este sumando al valor 0 que está almacenado en la barra inferior. Posteriormente, se añade el segundo sumando haciendo uso una vez más de los deslizadores, y se obtiene el resultado final rotando por última vez la manivela. Este método, aunque *a priori* parece peor, es muy útil cuando se requiere hacer adiciones con más de dos sumandos.

Esto se ve mejor con un ejemplo. Imaginemos que partimos de una situación inicial con todos los valores a 0. Mantenemos la calculadora como está, modificando únicamente el deslizador de las unidades poniendo un 3 y con la opción de suma. Si realizamos una vuelta completa con la manivela, el resultado que se obtiene es de 3, resultado de sumar 0, del valor almacenado previamente, y 3, del valor introducido por el usuario. Sin embargo, si giramos de nuevo la manivela 360° en el sentido de las agujas del reloj, el resultado es 6. Esto es así porque cada vuelta de la manivela realiza el cálculo a partir del valor introducido con los deslizadores y el resultado de la operación anterior.

Por ello, los pasos a seguir para realizar la operación $85 + 137$ con este segundo método serían:

1. Introducir el número 85 en los deslizadores.
2. Rotar la manivela del resultado.
3. Introducir el número 137 en los deslizadores.
4. Girar la manivela para obtener el resultado final.

Cabe destacar que para realizar las sumas se necesita rotar la manivela del resultado tantas veces como sumandos tenga la operación.

4.4.2. Resta

A grandes rasgos, la resta funciona del mismo modo que la suma.

Para calcular la resta es necesario colocar la palanca de las operaciones en la posición «S» de *subtract*. Veamos un ejemplo de como realizar una resta. Pongamos que queremos calcular la diferencia entre $137 - 85$. Los pasos a seguir son:

1. Fijar la operación en «S»
2. Introducir el número 137 en la barra de resultados haciendo uso de las ruedas.

3. Introducir el número 85 en los deslizadores.
4. Girar la manivela para obtener el resultado final.

La única particularidad de la calculadora Millionaire a la hora de restar es que no tiene números negativos. Esto nos plantea la siguiente pregunta: ¿cómo se comporta la máquina frente a una operación en la que el sustraendo es mayor que el minuendo? Otto Steiger incorporó una ingeniosa pero sencilla solución a este problema.

Cuando el resultado de la resta es negativo, se realiza inmediatamente una suma y suena una campana indicando que ha habido un error. Es decir, si se realiza la resta $10 - 20$, el resultado esperado es -10 . Sin embargo, como la máquina carece de números negativos, el resultado obtenido sería 30 (resultado de sumar $10 + 20$), acompañado del tintineo de una campana, denotando un fallo.

4.4.3. Multiplicación

La característica más llamativa de esta máquina en su lanzamiento fue su capacidad para hacer multiplicaciones sin el uso de sumas consecutivas.

Esto fue posible gracias a la incorporación de las tablas de multiplicar en la memoria de la máquina. Estas tablas están constituidas por una pieza metálica con nueve pares de placas escalonadas, representando cada uno de los pares las tablas de multiplicar del 1 al 9 (una placa del par marcaba las unidades y la otra las decenas). La palanca del multiplicador, que ya hemos visto en la parte superior derecha de la Figura 4.3, eleva la placa que representa la tabla de multiplicar del número seleccionado y la alinea con el mecanismo de cremallera de la máquina. De esta forma, al realizar una vuelta a la manivela de resultado para realizar el cálculo, la placa empujará el número introducido en los *racks* obteniendo así el resultado final.

Los *racks*, apreciables en la Figura 4.7, están montados en ranuras en tres soportes de hierro fundido en la parte central trasera de la máquina, y están retenidos por tiras de cubierta atornilladas. Los bastidores están numerados del 0 en la parte trasera al 9 en la delantera, lo que corresponde a los valores fijados en los deslizadores o columnas del teclado.

A medida que la máquina va realizando el cálculo, el multiplicador empuja estas cremalleras hasta obtener el resultado. Una vez se ha terminado de empujar, los distintos *racks* vuelven a su posición inicial gracias a unos muelles que actúan como topes. Cabe destacar que la cremallera que representa el valor 0 está fija en una posición y sirve para bloquear los ejes cuando se producen ceros en los ajustes de corredera.³

La imagen 4.10 muestra los nueve pares de placas. La placa de las unidades se ve claramente a lo largo de la parte superior del bloque. Sin embargo, el par de las decenas no es visible fácilmente debido a que su valor es cero para la primera fila

³En las últimas versiones de la máquina fue necesario remachar un banda de acero para aumentar el rozamiento en los *racks*, debido a que el mecanismo los lanzaba con demasiada fuerza y se producían desajustes. Este problema no existió en la versión manual.

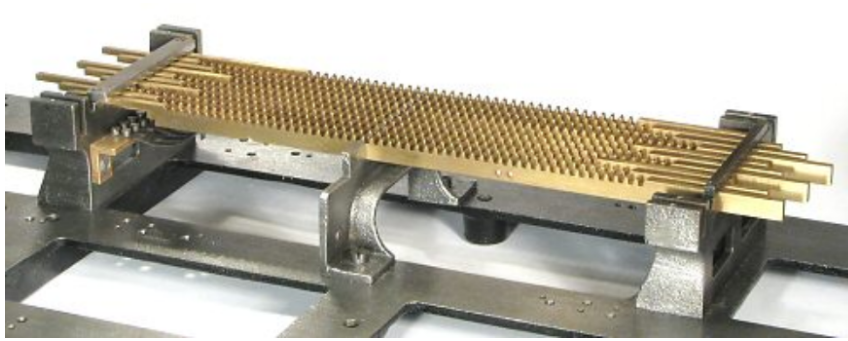


Figura 4.7: Mecanismo de cremallera o *racks* utilizado para realizar el cálculo en las multiplicaciones. Este mecanismo permite efectuar la multiplicación del multiplicando por un dígito del multiplicador mediante un único giro de manivela, independientemente del valor de este dígito.

y la primera columna. La placa más cercana a la cámara que toma la fotografía representa la tabla del 1 y la más alejada la del 9. La representación de las tablas de multiplicar almacenadas en la calculadora se puede apreciar en la Figura 4.9.

Cuando multiplicamos 7×6 mentalmente, inmediatamente sabemos el resultado: 42. Esto es posible gracias a que tenemos memorizadas las tablas de multiplicar del 0 al 9.

Sin embargo, cuando la Millionaire multiplica 7×6 , su mecanismo de control acude a la tabla de multiplicación interna y devuelve el producto parcial 4 (decenas). Éste se añade al registro, que se desplaza una posición a la izquierda. La tabla de multiplicación devuelve entonces 2 (unidades), que se añade al registro (ahora a la derecha de las decenas) para mostrar la respuesta 42. La máquina sólo requiere una vuelta de manivela para cada dígito del multiplicador, pero realiza dos ciclos de producto parcial por cada vuelta: uno para las decenas y otro para las unidades. Es decir, que si quisiésemos multiplicar 21×12 , sería necesario multiplicar 21×1 , y multiplicar posteriormente el valor calculado por 2. Al momento de multiplicar por 2, se desplaza el registro del valor almacenado, una posición a la izquierda, por lo que realmente el valor 21 que teníamos almacenado de operar 21×1 , ahora es 210. A 210 entonces se le suma el resultado de 21×2 (42) y se obtiene el resultado final: 252

La calculadora únicamente puede multiplicar por un número de un solo dígito (1-9) de golpe. Para multiplicar por números mayores, cuenta con un ingenioso mecanismo que desplaza todos los números a la izquierda y añade un 0 al final.

De esta forma, si se desea multiplicar un 1250×74 habría que:

1. Poner la máquina en modo «Multiplicación» y añadir el número 1250 haciendo uso de los deslizadores.
2. Utilizar la palanca de la multiplicación, visible en la Figura 4.8, y poner el número 7.
3. Rotar la manivela para realizar la operación
4. Volver a utilizar la palanca de la multiplicación para poner el número 4.
5. Rotar de nuevo la manivela para operar.



Figura 4.8: Palanca de la calculadora Millionaire utilizada en la multiplicación para definir por qué número se quiere multiplicar.

| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| U | T | U | T | U | T | U | T | U | T | U | T | U | T | U | T | U | T | U | T |
| 9 | 0 | 8 | 0 | 7 | 0 | 6 | 0 | 5 | 0 | 4 | 0 | 3 | 0 | 2 | 0 | 1 | 0 | | |
| 8 | 1 | 6 | 1 | 4 | 1 | 2 | 1 | 0 | 1 | 8 | 0 | 6 | 0 | 4 | 0 | 2 | 0 | | |
| 7 | 2 | 4 | 2 | 1 | 2 | 8 | 1 | 5 | 1 | 2 | 1 | 9 | 0 | 6 | 0 | 3 | 0 | | |
| 6 | 3 | 2 | 3 | 8 | 2 | 4 | 2 | 0 | 2 | 6 | 1 | 2 | 1 | 8 | 0 | 4 | 0 | | |
| 5 | 4 | 0 | 4 | 5 | 3 | 0 | 3 | 5 | 2 | 0 | 2 | 5 | 1 | 0 | 1 | 5 | 0 | | |
| 4 | 5 | 8 | 4 | 2 | 4 | 6 | 3 | 0 | 3 | 4 | 2 | 8 | 1 | 2 | 1 | 6 | 0 | | |
| 3 | 6 | 6 | 5 | 9 | 4 | 2 | 4 | 5 | 3 | 8 | 2 | 1 | 2 | 4 | 1 | 7 | 0 | | |
| 2 | 7 | 4 | 6 | 6 | 5 | 8 | 4 | 0 | 4 | 2 | 3 | 4 | 2 | 6 | 1 | 8 | 0 | | |
| 1 | 8 | 2 | 7 | 3 | 6 | 4 | 5 | 5 | 4 | 6 | 3 | 7 | 2 | 8 | 1 | 9 | 0 | | |

Figura 4.9: Representación de las tablas de multiplicar internas que tiene almacenadas la calculadora Millionaire. Nótese que la columna de las unidades y las decenas se encuentra invertida.

Por tanto, el cálculo que realmente realiza la máquina es: $1250 \times 70 + 1250 \times 4$.

4.4.4. División

La división es la operación más complicada de realizar en esta calculadora. Al contrario que en las operaciones anteriores, para utilizar esta función es necesario que el usuario sepa dividir. Del mismo modo que en la multiplicación hay que introducir paso a paso cada uno de los dígitos del multiplicador, en la división se requiere que se introduzca el numerador y el divisor. Sin embargo, también es necesario introducir manualmente el cociente dígito a dígito. Para facilitar el uso de la división, la calculadora cuenta con unas tablas que indican cuál es el cociente de cada subdivisión.

En primer lugar, se introduce manualmente en los deslizadores el valor del divisor. A continuación, se buscan los dos primeros dígitos, es decir, los más

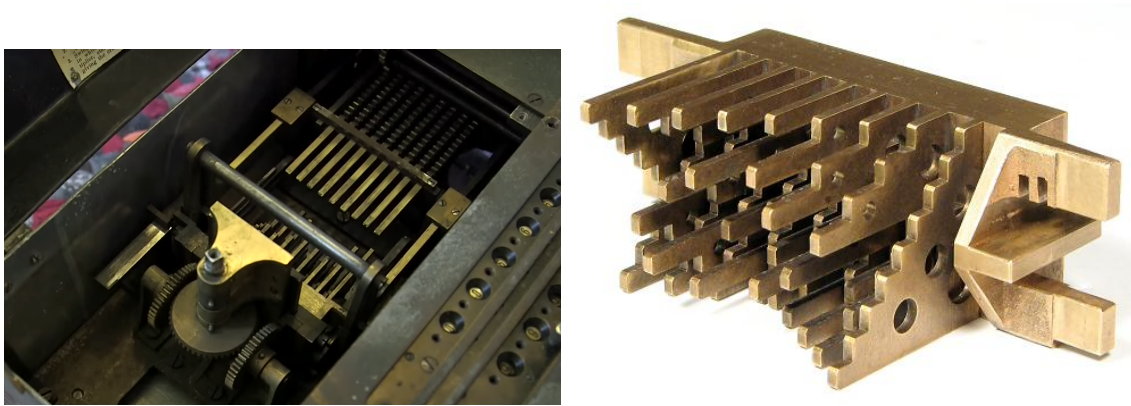


Figura 4.10: Mecanismo interno de la calculadora. Es la pieza clave que permite realizar los cálculos de las multiplicaciones.

grandes, en la tabla de soporte de la división, observable en la Figura 4.11. Se introduce el dividendo en la barra de resultados y se añade el valor localizado en la tabla de soporte a la palanca de multiplicación. Se rota la manivela y se obtiene el primer dígito del cociente. Se repite el mismo proceso hasta que el resto final sea inferior al divisor.

Para realizar la división $3550/113$, se añade el valor el número 113 en los deslizadores. Posteriormente, se consulta la tabla de soporte, utilizando la guía hasta encontrar el valor 11 en la columna de la izquierda. Buscamos el número 11 porque son los dos dígitos más elevados del divisor. A continuación, es necesario incorporar el dividendo a lo que hasta ahora ha sido la barra de resultados. Se utiliza la tabla de soporte para la división para encontrar el primer dígito del cociente. Para ello, se busca el rango más pequeño posible en la placa que contiene los primeros dígitos de 335. En este caso, el rango es $[033 - 044]$. En la parte superior de cada límite del rango se encuentra un dígito. Se selecciona el dígito del límite inferior del rango y se introduce en la palanca de multiplicación. Ahora, se gira la manivela de resultado y se obtiene el resto en la barra de resultado y el cociente en la barra intermedia. Después de esto, solo es necesario repetir el mismo proceso hasta que todo el número haya sido dividido y el resto final sea inferior al divisor.

| Divisor | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 11 | 011 | 022 | 033 | 044 | 055 | 066 | 077 | 088 | 099 |
| 12 | 015 | 030 | 045 | 060 | 075 | 090 | 105 | 120 | 135 |
| 16 | 016 | 032 | 048 | 064 | 080 | 096 | 112 | 128 | 144 |
| 17 | 017 | 034 | 051 | 068 | 085 | 102 | 119 | 136 | 153 |
| 18 | 018 | 036 | 054 | 072 | 090 | 108 | 126 | 144 | 162 |
| 19 | 019 | 038 | 057 | 076 | 095 | 114 | 133 | 152 | 171 |
| 20 | 020 | 040 | 060 | 080 | 100 | 120 | 140 | 160 | 180 |
| 21 | 021 | 042 | 063 | 084 | 105 | 126 | 147 | 168 | 189 |
| 22 | 022 | 044 | 066 | 088 | 110 | 132 | 154 | 176 | 198 |
| 23 | 023 | 046 | 069 | 092 | 115 | 138 | 161 | 184 | 207 |
| 24 | 024 | 048 | 072 | 096 | 120 | 144 | 168 | 192 | 216 |
| 25 | 025 | 050 | 075 | 100 | 125 | 150 | 175 | 200 | 225 |
| 26 | 026 | 052 | 078 | 104 | 130 | 156 | 182 | 208 | 234 |
| 27 | 027 | 054 | 081 | 108 | 135 | 162 | 189 | 216 | 243 |
| 28 | 028 | 056 | 084 | 112 | 140 | 168 | 196 | 224 | 252 |
| 29 | 029 | 058 | 087 | 116 | 145 | 174 | 203 | 232 | 261 |

Figura 4.11: Tablas incorporadas en la cara externa de la tapa de la máquina utilizadas como soporte a la hora de dividir. Contiene dos guías sobre las que se desliza una placa metálica. Esta placa hay que deslizarla hasta que los dos dígitos de la izquierda cuadren con los dos dígitos mayores del divisor. Después, se busca el rango más reducido que contenga el dividendo. Una vez localizado se introduce el número de la placa que se encuentra sobre el límite inferior de dicho rango y se introduce en la palanca de multiplicación, indicando así que este número formará parte del cociente.

CAPÍTULO 5

Implementación de la Millionaire en Scratch

Este capítulo está dedicado a exponer cómo se ha implementado el funcionamiento de la calculadora en Scratch. En primer lugar, se declarará cómo se han creado los objetos. Posteriormente, se explicará cómo se les ha dotado de movimiento. Y, por último, cómo se ha implementado cada unas de las distintas operaciones.

5.1 Creación de los objetos

Al momento de enfrentarme a una página en blanco de código, no sabía muy bien por dónde empezar. Como conocía un poco la metodología de trabajo con Scratch y había investigado para realizar este proyecto, era conocedor de que Scratch permite ver resultados de lo que estás realizando desde el primer momento, por lo que decidí crear en primer lugar los objetos para poder apreciar desde el inicio lo que estaba programando.

Estuve pensando durante un tiempo con qué herramienta de diseño gráfico realizar la parte gráfica. Probé Gimp, Paint, etc. Pero me di cuenta que parte del proyecto es demostrar la potencia y la versatilidad que tiene el lenguaje de programación Scratch, por lo que utilicé la herramienta de edición de imágenes interna a la aplicación.

De esta forma, partí como objeto principal de una imagen completa de la calculadora desde una perspectiva de planta, observable en la Figura 5.1. A partir de este objeto, hice duplicidades. Tantas como objetos tenía que crear: al principio fueron alrededor de 20 objetos, pero este número se fue incrementando conforme necesitaba nuevas funcionalidades.

De este modo, partiendo de dicha imagen, utilizaba la herramienta de borrado para aislar algunas partes. Por ejemplo, para la manivela de la esquina superior derecha, dupliqué la imagen y borré toda la máquina a excepción de esa pieza. El resultado de esto se puede apreciar en la Figura 5.2.

Esto seguía suponiendo un problema, pues cuando dotase de movimiento a este nuevo objeto, seguiría apareciendo en el fondo la manivela de la imagen primaria. Para solucionar este contratiempo, decidí coger partes de la máquina

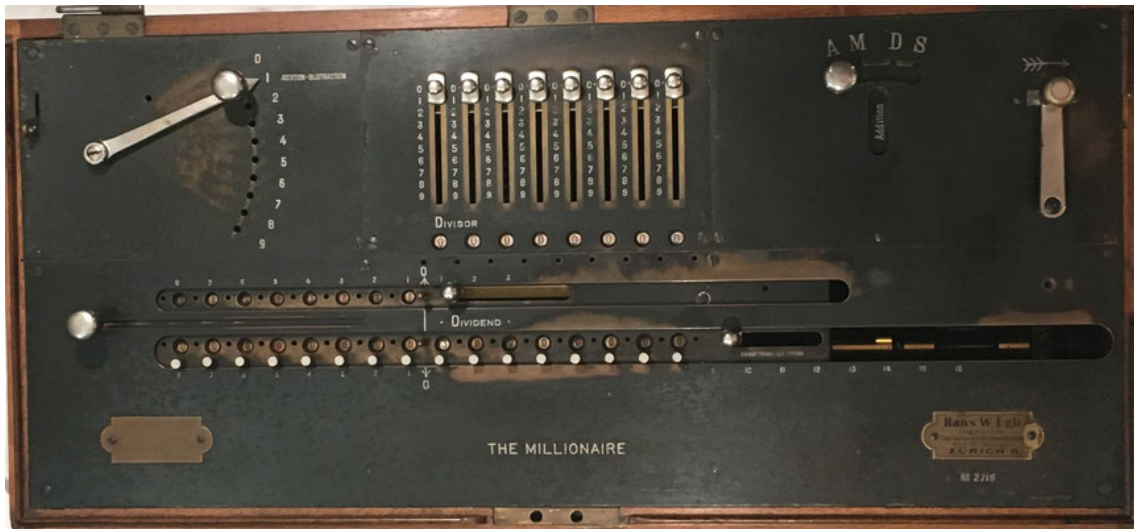


Figura 5.1: Imagen de la calculadora Millionaire vista desde un plano planta. Utilizada como base para implementar la interfaz de la máquina en Scratch.



Figura 5.2: Imagen de la manivela cuya función es realizar el cálculo. Utilizada para aislar los movimientos de la manivela en la ejecución del código.

que no tuviesen botones, ni deslizadores ni ningún tipo de mecanismo (es decir, el tablero liso) duplicarlas y pegarlas encima de la manivela. El resultado fue el que se puede apreciar en la Figura 5.3. En la Figura 5.4, se observa el resultado final del tablero, una vez suprimidos todos los mecanismos móviles en el programa.

5.2 Movimiento de los objetos

Una vez creados los principales objetos, había que dotarlos de movimiento.

5.2.1. Deslizadores

El primer objeto que animé fueron los deslizadores. En primera instancia quise hacer el movimiento arrastrando con el ratón. Después de varios intentos fallidos

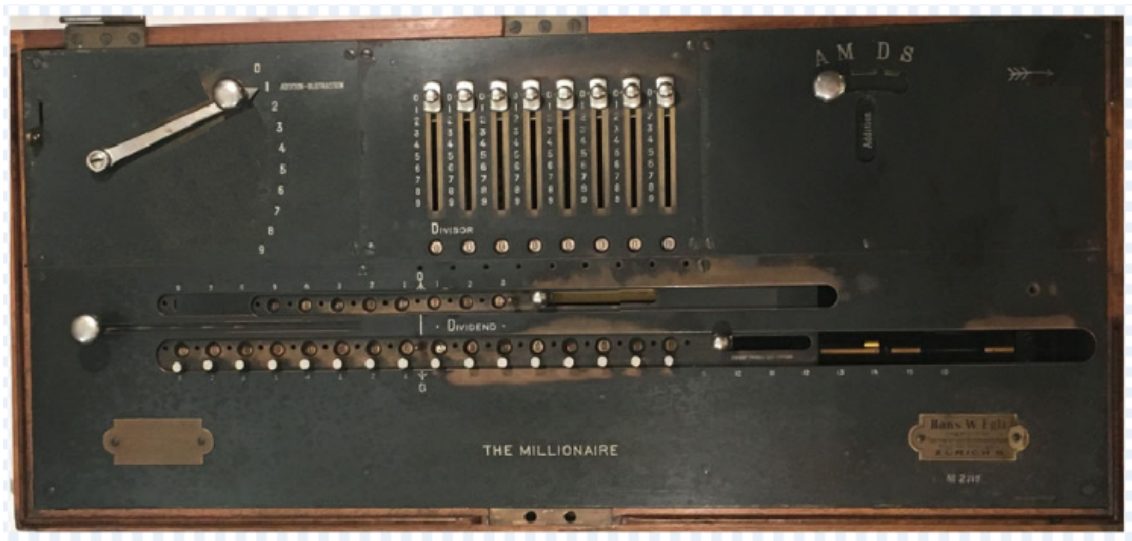


Figura 5.3: Imagen de la calculadora Millionaire una vez se ha eliminado la manivela de resultado para evitar duplicidades de objetos.

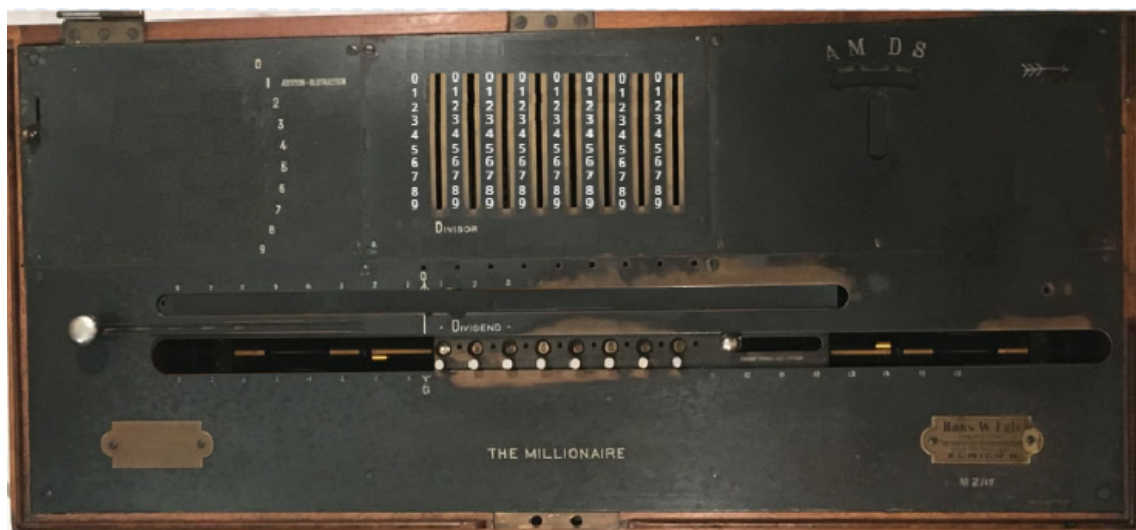


Figura 5.4: Imagen de la calculadora una vez se han eliminado todas las partes móviles. Utilizada para evitar duplicidades con los objetos aislados.

debido a la dificultad de limitar el movimiento a la hora de ser arrastrados (para evitar que se moviesen libremente por toda la pantalla) y de detectar en qué posición se quedaban, decidí simplificar el proceso. La forma en la que lo programé finalmente fue por clics: un evento que al pinchar sobre el objeto, redujese su altura 5 unidades. También incorporé un contador para que una vez hubiese alcanzado el número 9, si se volvía clicar sobre él, volviese a la posición inicial. Una vez programado este objeto, lo dupliqué 7 veces creando una variable para cada uno y cambiando el valor en el eje de ordenadas para tener todos los deslizadores operativos. Todo esto se puede apreciar en la Figura 5.7.

5.2.2. Manivela

El segundo objeto que animé fue la manivela de resultado. Este fue sencillo desde un inicio. Simplemente programé que, al pinchar sobre el objeto, realizase una vuelta de 360°. Lo único que merece la pena destacar de este objeto es que tuve que modificar el diseño levemente porque el punto central del objeto sobre el que rotaba al principio era la mitad de la manivela. En la calculadora original, esta parte rota a partir de un extremo fijo. Por ello, simplemente seleccioné el objeto completo y lo desplacé hasta que uno de los extremos ocupó la parte central del cuadro de diseño. En la Figura 5.5 se puede estudiar el código de movimiento.

5.2.3. Palanca de multiplicación

La siguiente parte de la calculadora en recibir movimiento fue la palanca de la multiplicación, situada en la parte superior izquierda. Al igual que en la manivela, este objeto tenía que rotar a partir de un extremo. Así que, una vez configurado de esta manera, realicé varias pruebas para ver cuántos grados tenía que rotar cada vez que se clicará sobre el objeto, de forma que apuntase cada vez al número pintado sobre la propia calculadora, que indica con qué cifra se está multiplicando. El resultado final del movimiento de esta palanca se puede apreciar en la Figura 5.6.

5.2.4. Palanca de operación

El último mecanismo de la parte superior de la calculadora al que doté de movimiento fue la palanca de operaciones. El movimiento de esta palanca era algo más sofisticado, debido a la forma de la guías que tenía el diseño original. La lógica que implanté fue la de reducir la altura, desplazarse sobre el eje de abscisas y volver a incrementar la altura. De esta forma, el movimiento que resultó, apreciable en la Figura 5.9, fue una copia idéntica a la de máquina original.

5.2.5. Operador

Al repasar la movilidad de todos los mecanismos me percaté de que había algo que no cuadraba todavía. En el diseño original, al cambiar la operación seleccionada, también cambiaba el cartel, indicando con qué operador se iba a rea-

lizar el cálculo. Así que creé un objeto nuevo con cuatro disfraces distintos. Cada disfraz era únicamente un letrero blanco que indicaba una de las operaciones disponibles, a saber: «Addition», «Multiplication», «Division», «Substract». Para que cambiase a la vez que la palanca del operador, programé sobre esta distintos lanzamientos de eventos que indicasen la operación con la que se estaba trabajando. De esta forma, según qué evento se captaba, se cambiaba a un disfraz o a otro. Este código tan sencillo puede verse en la Figura 5.8.

5.2.6. Barra de resultado

En esta sección se engloban todos los objetos del resultado. Es decir, las distintas esferas que muestran los dígitos, la palanca de reinicio y la barra que sostiene todos estos objetos. El movimiento depende de un evento cuyo lanzamiento aún no estaba preparado pero creé de todos modos. El movimiento es sencillo: deslizarse hacia la izquierda 10 unidades cada vez que se recibe el evento «Acarreo» y deslizarse a la posición inicial cuando se reciba el evento «Reseteo1». Este último evento se lanzará cuando se presione la palanca de reinicio para devolver la máquina al estado inicial. El lanzamiento del otro evento se explicará en el siguiente apartado, cuando veamos cómo se ejecuta el cálculo de las operaciones. En la Figura 5.10 se puede apreciar este código.

5.2.7. Barra de cociente

El movimiento de este objeto es prácticamente idéntico al anterior. Simplemente tiene una condición más que evita que se realice el movimiento cuando la variable «Coma» es igual a 1. Esta variable y su utilidad las veremos mejor en el apartado dedicado a explicar cómo se ha programado la división. Se puede estudiar este código en la Figura 5.11.

5.2.8. Palanca de reinicio

Aunque la palanca de reinicio tiene el movimiento explicado en el apartado anterior, también requiere un movimiento personalizado para cuando se realiza un clic sobre el objeto. Este movimiento simplemente desplaza la palanca hacia la derecha y luego vuelve a su posición inicial, lanzando el evento que comentábamos en el apartado anterior «Reseteo1». El código correspondiente a este movimiento se observa en la Figura 5.12.

5.3 Programación de las operaciones

En este apartado vamos a explicar cómo se han programado las distintas operaciones en Scratch. Para los cuatro tipos de operaciones, vamos a ver dos códigos distintos: uno para las cifras sobre las que se realizan los cálculos (8) y otra para las cifras que contienen los acarros (8). Para realizar los cálculos cada uno de los 16 objetos de resultado cuenta con 3 variables: una para mostrar el resultado,

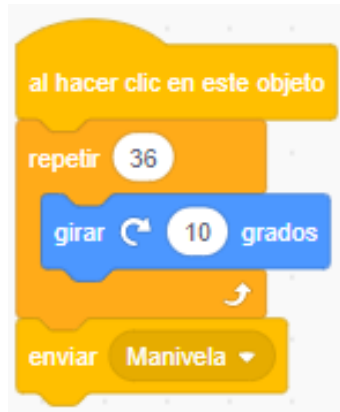


Figura 5.5: Código de movimiento de la manivela. Como se observa, al clicar sobre la manivela, el objeto rota 10 grados 36 veces, lo que supone una vuelta completa. Al terminar el bucle, envía un evento que se utilizará para determinar el momento de hacer los cálculos.



Figura 5.6: Código de movimiento de la palanca de multiplicación. Al clicar sobre ella, hace un movimiento de rotación de 6,8 grados siempre y cuando no se encuentre en la posición 9, donde deshacerá todo lo rotado. Utiliza una variable para determinar cuándo ha llegado a la última posición.



Figura 5.7: Código para mover los deslizadores, cada uno de los deslizadores cuenta con su propia variable que permite saber cuando tiene que volver al inicio. De esta forma, siempre que no se haya pinchado 9 veces sobre él, reduce el eje de ordenadas en 6 unidades. Sin embargo, cuando se pincha por novena vez vuelve a la posición inicial.



Figura 5.8: Código para cambiar el disfraz que indica con qué operación se está trabajando. Utiliza los eventos que genera la palanca de operación para saber el operador actual.

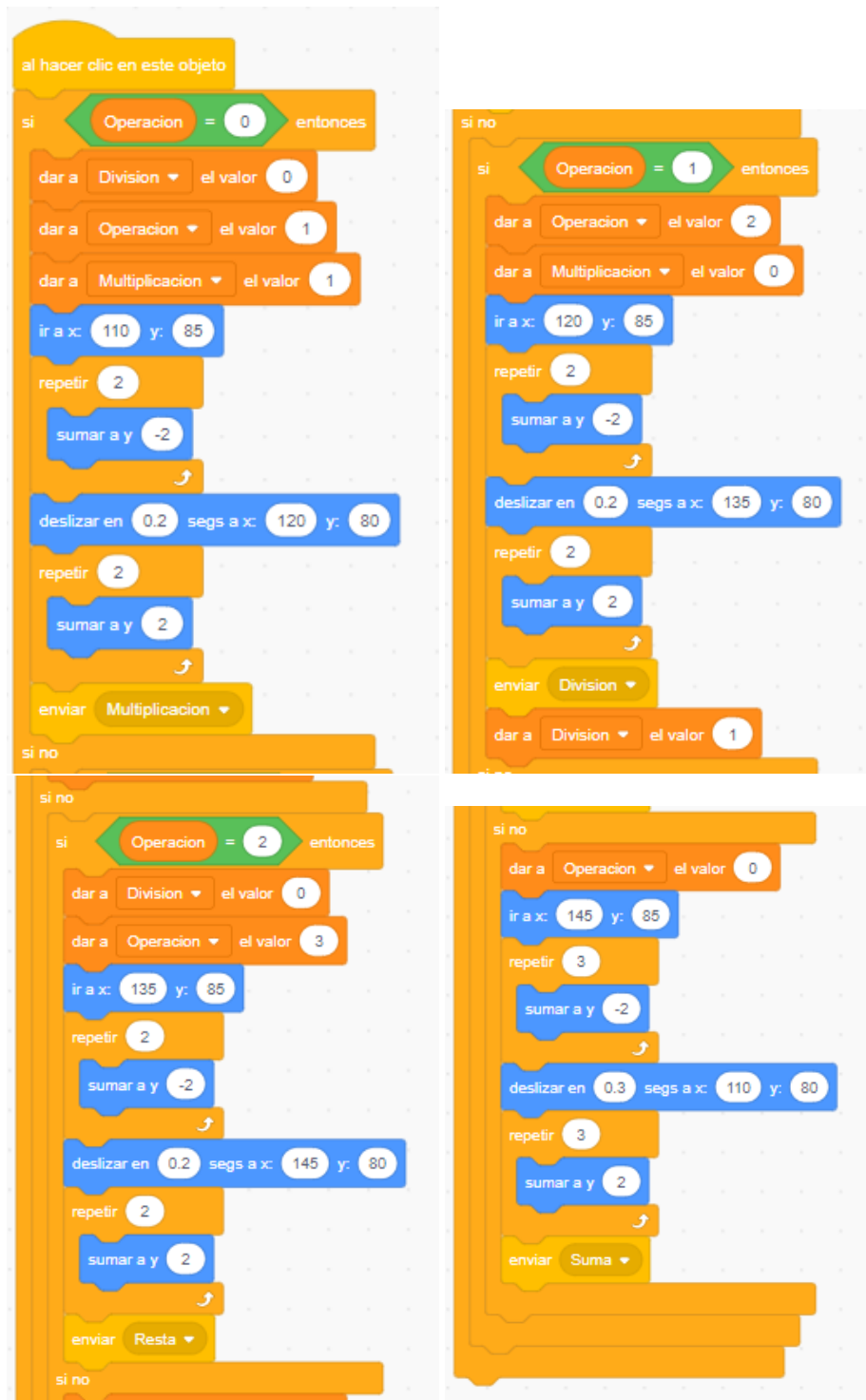


Figura 5.9: Código de movimiento de la palanca de operación. A partir de una variable se determina el movimiento que tiene que hacer para pasar a la siguiente posición. Debido a la forma que tiene la guía de la palanca, todos los movimientos tienen que reducir el valor del eje de ordenadas, desplazarse por el eje de abscisas e incrementar el valor del eje de ordenadas.

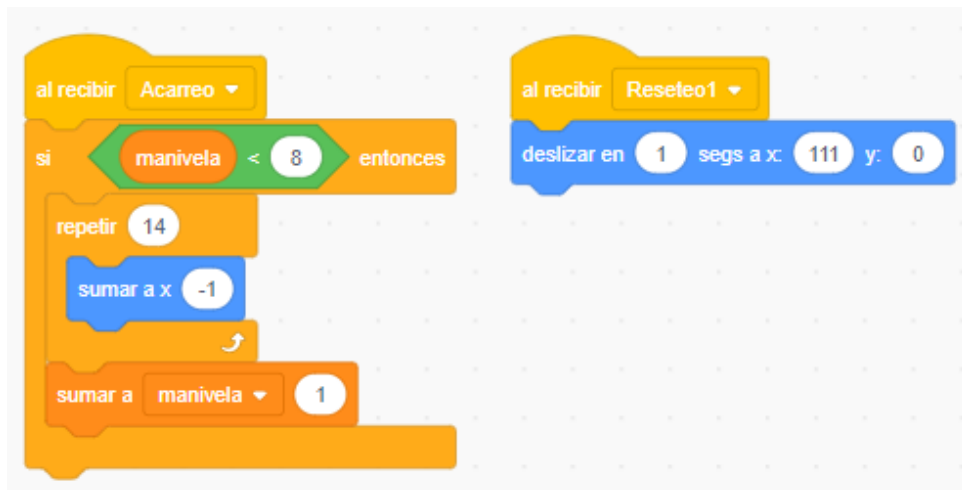


Figura 5.10: Código de la barra de resultado, las cifras y la palanca de reinicio cuando hay acarreo. Al recibir acarreo, el objeto se desplaza negativamente sobre el eje de las abscisas 14 unidades. Esto sucede siempre y cuando no haya habido 8 acarreos previos. Cuando se recibe reinicio, el objeto se desliza hasta su posición inicial.



Figura 5.11: Código de la barra de cociente, las cifras y la palanca de reinicio cuando se realiza una división. Al girar la manivela se desplaza la barra de cociente para ir almacenando el valor en la posición adecuada. Este movimiento se realiza únicamente cuando se opera con una división, el número de desplazamientos acumulados no es superior a 7 y la variable coma es igual a 0.

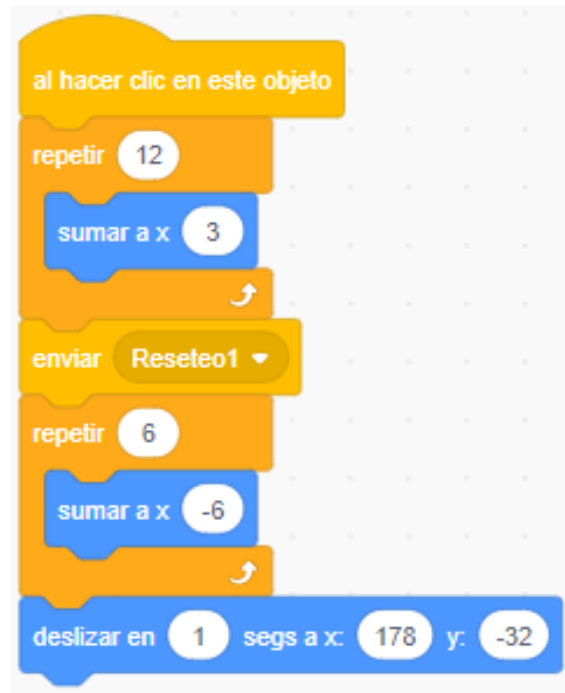


Figura 5.12: Código para llevar a cabo el movimiento de la palanca de reinicio cuando se pincha sobre ella.

otra para llevar el acarreo al siguiente objeto y otro para ayudar en el cálculo del acarreo.

Se realiza el lanzamiento secuencial de eventos para calcular las distintas cifras, necesario para evitar condiciones de carrera. Si se hiciesen todos los cálculos a partir del evento principal, no se garantizaría que el cálculo del acarreo de la cifra anterior ya se ha realizado, por lo que el resultado final podría ser incorrecto.

Como se observa en el código de la Figura 5.13, después de realizar cada cálculo, se modifica el disfraz de la cifra en consecuencia con el resultado obtenido, para así mostrar el dígito correcto. Este código está implementado en los 16 objetos resultado.

Al igual que en apartados anteriores, la explicación del código aparecerá en cada una de las figuras.

5.3.1. Suma

La primera operación implementada fue la suma. El cálculo se realiza de forma secuencial. Al pinchar sobre la manivela, se lanza un evento que recoge el objeto de resultado que representa las unidades. Este objeto realiza su cálculo y lanza un evento para que se realice el cálculo de las decenas. Nótese que se utiliza la variable «Redondeo7» como variable auxiliar. Para calcular el acarreo, se divide entre 10 el resultado, por lo que el valor obtenido puede tener decimales. Para solucionar esto, se utiliza la función «Redondeo». Por tanto, cabe la posibilidad se redondee al alza. Para evitar esto, se utiliza «Redondeo7», que tiene el valor del acarreo antes de redondear. De este modo, si al redondear «Acarreo7», el valor obtenido es mayor que el valor almacenado en la variable auxiliar, se le



Figura 5.13: Código para cambiar el disfraz del objeto a partir del resultado obtenido. Los disfraces varían entre los 10 dígitos existentes.

resta una unidad. En la Figura 5.14, se puede observar este código y estudiar su funcionamiento.

Cuando existe un acarreo en las decenas de millón, se lanza un evento que ejecuta un código distinto, realizado únicamente la suma del acarreo al valor ya almacenado. En la Figura 5.15, se puede examinar el código.

Ahora veamos un ejemplo de uso de la suma en el programa. Calculemos $51923 + 18273$. En primer lugar, como se puede observar en la Figura 5.16, se añaden los sumandos. Uno de ellos a partir de los deslizadores y el otro a partir de las ruedas. Después, nos aseguramos de la que máquina se encuentra en la opción de suma y se clicca sobre la manivela de la parte superior derecha para lanzar el cálculo. El resultado aparece en la barra inferior, como se observa en la Figura 5.17.

5.3.2. Resta

La segunda operación en ser programada fue la resta. Esta operación fue realmente sencilla de realizar, una vez las bases de la suma estaban asentadas. Al igual que con la adición, pinchar sobre la manivela lanza un evento que activa el cálculo de las unidades, una vez este está calculado, y por tanto, también su acarreo, se activa la siguiente cifra. La diferencia más grande respecto a la suma es la posibilidad de acabar en error la operación, debido a que la máquina no contempla números negativos. Por tanto, si el sustraendo es mayor que el minuendo, el código lanza un mensaje de reinicio que deja la calculadora a 0, en señal de



Figura 5.14: Código de cálculo de las decenas para la suma. El código se ejecuta cuando se recibe el evento «Res8», que indica que el cálculo de las unidades ya se ha realizado. En primer lugar, se da el valor 0 a la variable Acarreo7, para evitar colisiones con el cálculo anterior. Posteriormente, si operación es igual a 0 (lo que indica que se está realizando una suma), se calcula el acarreo a partir de la suma del número introducido, el valor almacenado del cálculo anterior y el acarreo arrastrado de la cifra inmediatamente menor. A esta operación se le aplica la función «Suelo» que redondea el número a la baja. Por último, se le da valor a la variable resultado, obteniendo el resto de la suma calculada para el acarreo, dividido entre 10.

que el cálculo no se ha realizado con éxito. Este código se puede estudiar en la Figura 5.19.

Ahora veamos un ejemplo de uso de la resta en el programa. Calculemos $51923 - 18273$. En primer lugar, como se puede observar en la Figura 5.20, se añade el sustraendo utilizando las ruedas de la barra inferior y el minuendo a partir de los deslizadores. Después, nos aseguramos de la que máquina se encuentra en la opción de resta y se clicca sobre la manivela de la parte superior derecha para lanzar el cálculo. El resultado aparece en la barra inferior, como se observa en la Figura 5.21.

5.3.3. Multiplicación

Como ya hemos visto en el capítulo anterior, la multiplicación realiza un desplazamiento con cada vuelta a la manivela, arrastrando así todos los objetos de resultado una posición a la izquierda. Este sistema suponía problemas para realizar los cálculos: si un objeto resultado está en la posición de las unidades y se desplaza a las decenas, ¿cómo saber cuándo tiene que utilizar el dígito del multiplicando de las unidades y cuándo el de las decenas? Este problema se solucionó con un sencillo código visible en la Figura 5.22, dónde se puede apreciar cómo cada objeto de resultado vuelve a su sitio una vez se ha desplazado y ocupa la posición más elevada los números que no estaban siendo utilizados. También toman el valor del dígito inmediatamente menor.

Una vez todos los objetos Resultado tienen el valor que les corresponde, se inicia el cálculo de la iteración actual. Utilizando el valor introducido en la palanca de multiplicación, se calcula el valor del dígito más pequeño y su acarreo,



Figura 5.15: Código de cálculo de las centenas de millón para la suma. El código se ejecuta cuando se recibe el evento «Res1», que indica que el cálculo de la cifra anterior ya se ha realizado. En primer lugar, se da el valor 0 a la variable Acarreo9, para evitar colisiones con el cálculo anterior. Posteriormente, si operación es igual a 0 (lo que indica que se está realizando una suma), y el valor almacenado es menor que 9, se le suma al resultado el acarreo recibido, si no se envía el desplazamiento y se cambia el valor a 0.

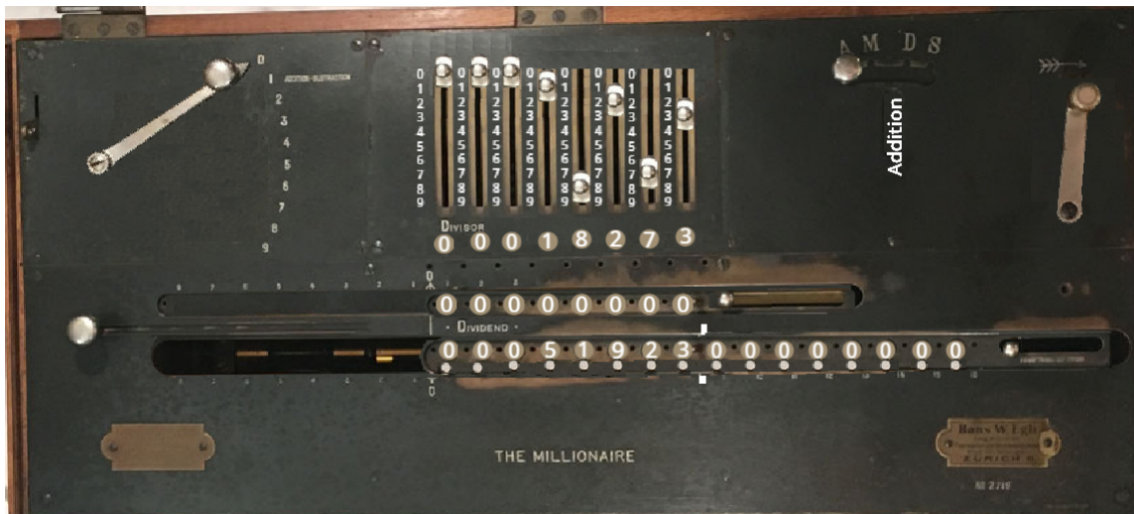


Figura 5.16: Paso 1 del ejemplo de la suma. En esta imagen se aprecia cómo se introducen los sumandos en la máquina. Nótese que la máquina se encuentra en posición de «Addition».

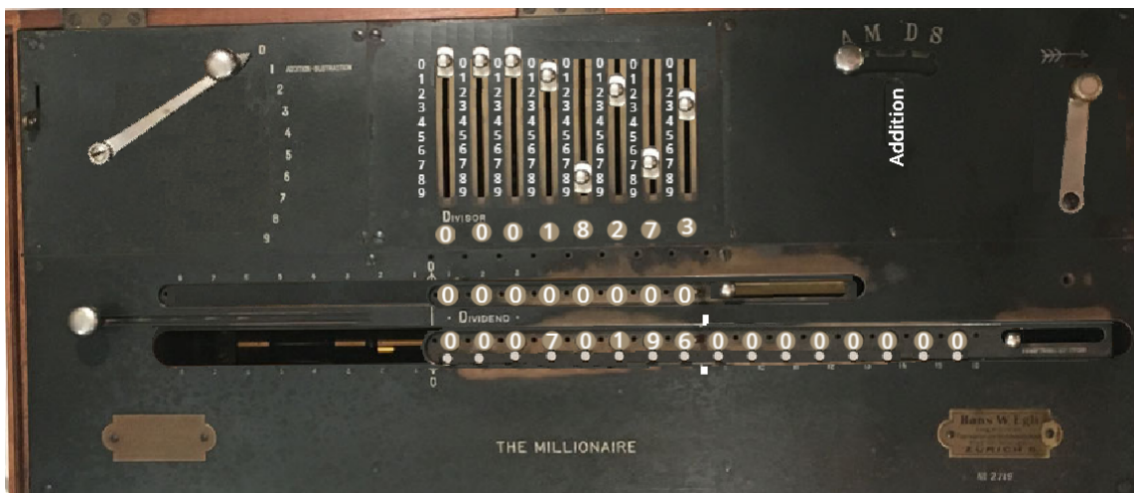


Figura 5.17: Paso 2 del ejemplo de la suma. En esta imagen se aprecia el resultado final de la operación en la barra inferior.



Figura 5.18: Código de cálculo de las centenas de millón para la resta. El código se ejecuta cuando se recibe el evento «Res8», que indica que el cálculo de la cifra anterior ya se ha realizado. Posteriormente, si operación es igual a 3 (lo que indica que se está realizando una resta), y el valor almacenado es mayor que la suma del acarreo más el valor introducido en los deslizadores, se le resta al resultado esta suma, en caso contrario, se suma 10 al resultado para que la resta no sea negativa. Se realiza la resta y se suma 1 al acarreo.



Figura 5.19: Código de cálculo de las centenas de millón para la resta. El código se ejecuta cuando se recibe el evento «Res1», que indica que el cálculo de la cifra anterior ya se ha realizado. Posteriormente, si operación es igual a 3 (lo que indica que se está realizando una resta), y el valor almacenado no es menor que el acarreo, se le resta al resultado este valor, en caso contrario. Se realiza la resta y se suma 1 al acarreo.

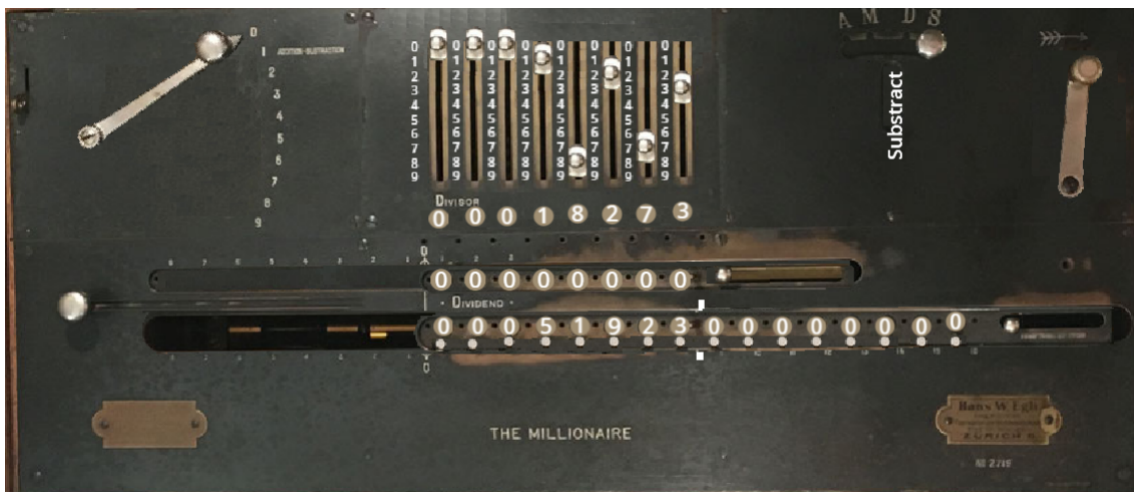


Figura 5.20: Paso 1 del ejemplo de la resta. En esta imagen se puede observar cómo se introduce el sustraendo a partir de las ruedas de la barra de resultado y el minuendo a partir de los deslizadores.

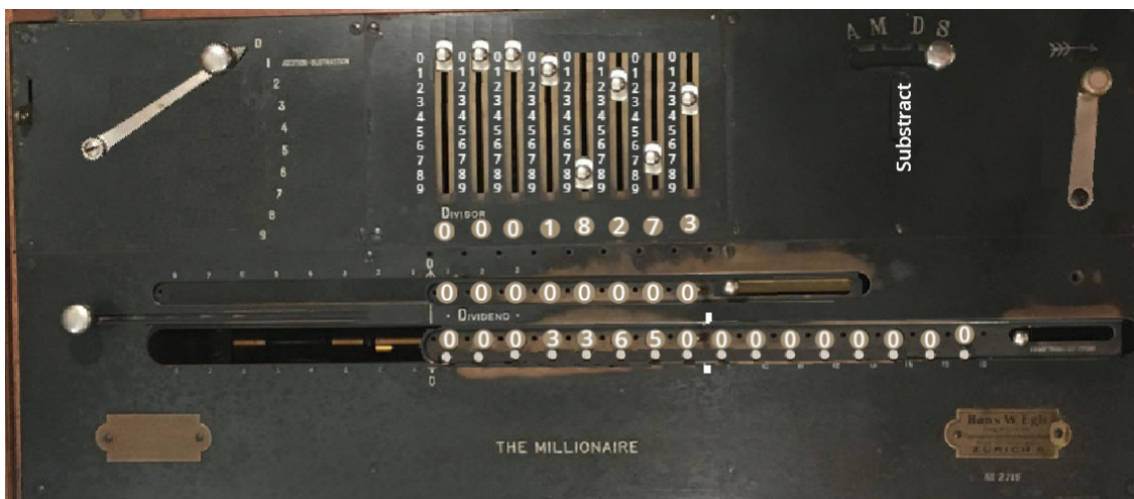


Figura 5.21: Paso 2 del ejemplo de la resta. En esta imagen se puede observar el resultado de la resta $51923 - 18273$ en la barra inferior. Menciona a la palanca de operaciones de la calculadora, que se encuentra en la posición «Subtract».

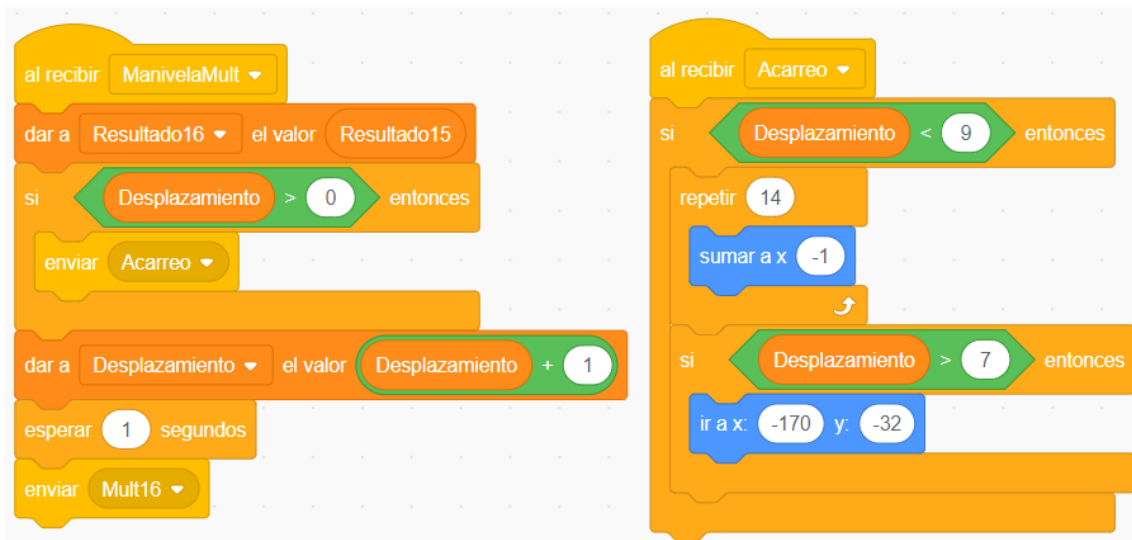


Figura 5.22: Código de movimiento de uno de los objetos de resultado. Como se observa, el objeto toma el valor del dígito inmediatamente anterior y si es la primera vez que se realiza la multiplicación, o lo que es lo mismo, «Desplazamiento» es igual a 0, se envía «Acarreo», que desplaza toda la barra de resultado. Además, se incrementa en 1 el valor de la variable desplazamiento y se envía un evento para que el siguiente dígito tome el valor de la variable inmediatamente menor.

cambiando también el disfraz del objeto, y se lanza el evento que permite a la siguiente cifra realizar sus cálculos.

Veamos esto con un ejemplo. Multipliquemos 8792×186 . En primer lugar, se introduce en los deslizadores el multiplicando, se fija la calculadora en Multiplicación, se introduce el dígito de mayor valor del multiplicador, que en este caso es el 1, y se gira la manivela. Este primer paso se puede apreciar en la Figura 5.24. El siguiente paso es más simple, introducimos el siguiente dígito de mayor valor del multiplicador con la palanca superior izquierda, en este caso, el dígito es el 8. Una vez giramos la manivela para dar fin a este paso, podemos ver en la barra inferior el resultado de la submultiplicación 8792×18 . Este segundo paso puede verse en la Figura 5.25. Por último, se añade el último dígito del multiplicador en



Figura 5.23: Código de cálculo del primer dígito del resultado en la multiplicación. Se calcula el acarreo a partir de la multiplicación del multiplicador con el dígito de las unidades del multiplicando. Posteriormente, se ajusta el acarreo del mismo modo que en la suma, con una aproximación a la baja, y se calcula el valor final de ese dígito.

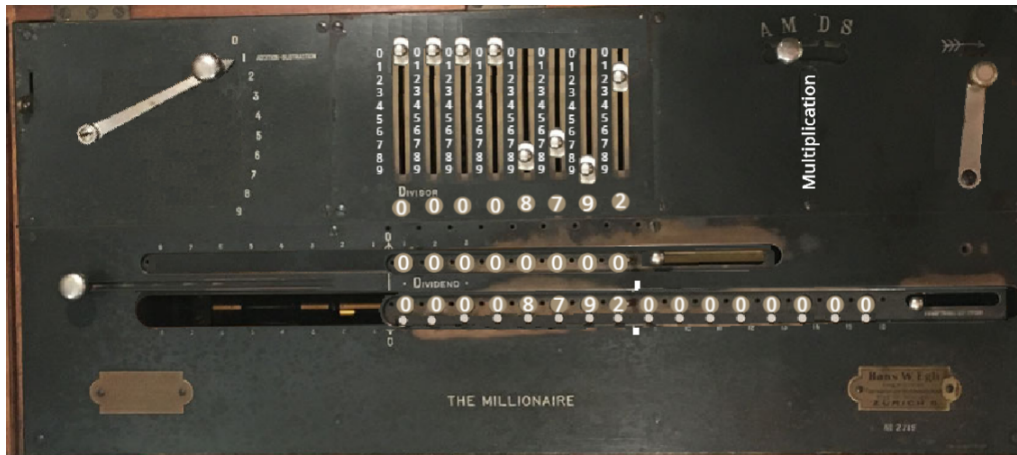


Figura 5.24: Paso 1 del código de la multiplicación. Se introduce el multiplicando con los deslizadores, se fija la calculadora en modo multiplicación y se mueve la palanca superior izquierda hasta la posición del dígito de mayor valor del multiplicador, que en este caso es 1. Nótese que en la barra inferior está almacenado el resultado de la submultiplicación realizada hasta ahora, 8792×1 .

la palanca destinada a ello y se gira la manivela, obteniendo así el resultado final en la barra inferior.

5.3.4. División

En la división, como hemos visto anteriormente, la barra de resultados deja de tener esa utilidad y pasa a ocupar la función del dividendo y los deslizadores se utilizan para marcar el divisor. La barra que queda entre estos dos elementos se utiliza para almacenar el cociente, es decir, el resultado de la división.

A diferencia de las anteriores operaciones, el cálculo de la división no se ha realizado por separado para cada uno de los objetos «Resultado», sino que se ha almacenado el valor del divisor y el valor del dividendo en una variable cada uno y a partir de ahí se ha realizado el cálculo.

Veamos un sencillo ejemplo ahora de la división en el programa. Dividamos $128/2$. El primer paso a realizar es introducir el divisor con los deslizadores, poner la máquina en modo División e introducir el dividendo en la barra inferior a partir de las ruedas destinadas a este propósito. Recaltar que es necesario desplazar la barra inferior 3 posiciones para que el número introducido no se considere 0,128. Para finalizar este primer paso, se desplaza la palanca superior derecha hasta el dígito de mayor valor al que «cabe» la subdivisión $12/2$. En la Figura 5.27 se puede observar este paso.

A continuación, con el dígito de mayor valor del cociente almacenado en la barra intermedia, se introduce con la palanca el siguiente dígito mayor al que «cabe» la subdivisión $8/2$, en este caso es 4. Remarcar que el dividendo de la subdivisión es 8 porque es el resto de la subdivisión anterior añadiendo la parte del dividendo que aún no ha sido dividida. En la Figura 5.28 se puede apreciar la situación en la que se queda la máquina después de este paso.

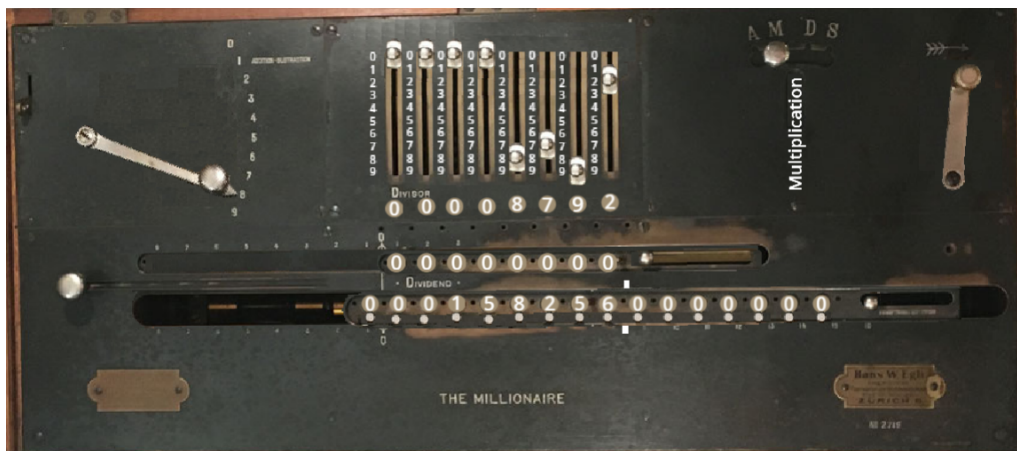


Figura 5.25: Paso 2 del código de la multiplicación. Se mueve la palanca superior izquierda hasta la posición del segundo dígito de mayor valor del multiplicador, que en este caso es 8. Nótese que en la barra inferior está almacenado el resultado de la submultiplicación realizada hasta ahora, 8792×18 .

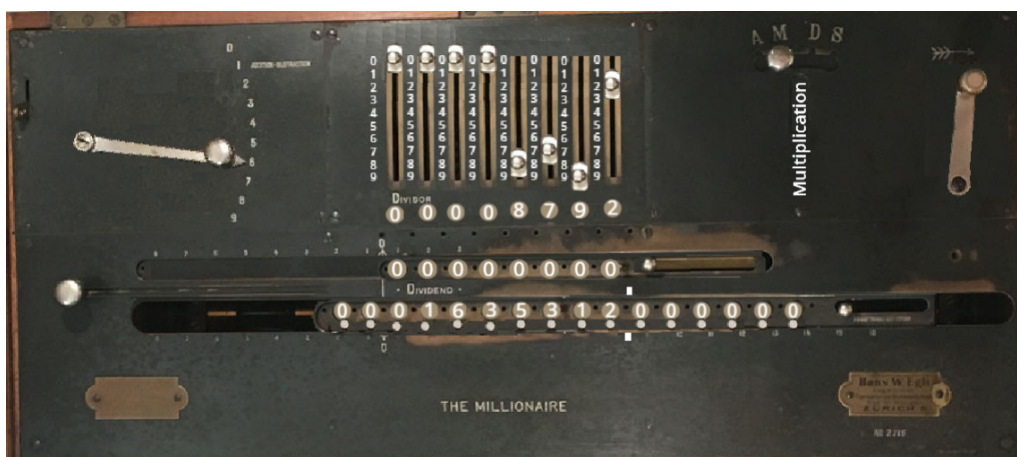


Figura 5.26: Paso 3 del código de la multiplicación. Se mueve la palanca superior izquierda hasta la posición del dígito de menor valor del multiplicador, que en este caso es 6. En la barra inferior puede observarse el resultado final de la multiplicación 8792×186 una vez se ha rotado la manivela.

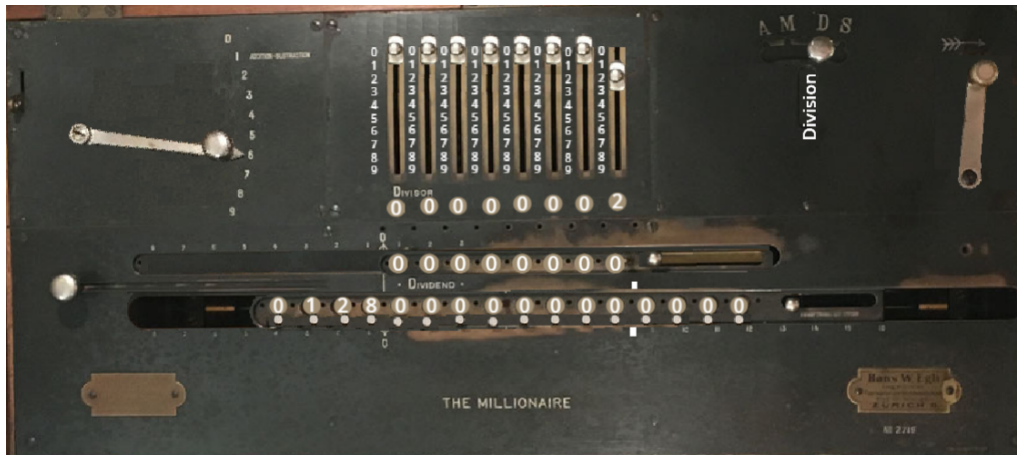


Figura 5.27: Paso 1 del código de la división. Se introduce el divisor con los deslizadores, se fija la máquina en modo División y se introduce en la barra inferior el dividendo, habiendo desplazado previamente esta barra tantas veces como dígitos tenga el dividendo. Por último, se introduce en la palanca superior izquierda el dígito mayor al que cabe la subdivisión $12/2$.

Finalmente, se rota la manivela y se almacena en la barra de cociente el último dígito introducido en con la palanca. En la barra del dividendo queda el almacenado el resto de la división, que en este caso es 0. El resultado final puede verse en la Figura [5.29](#).

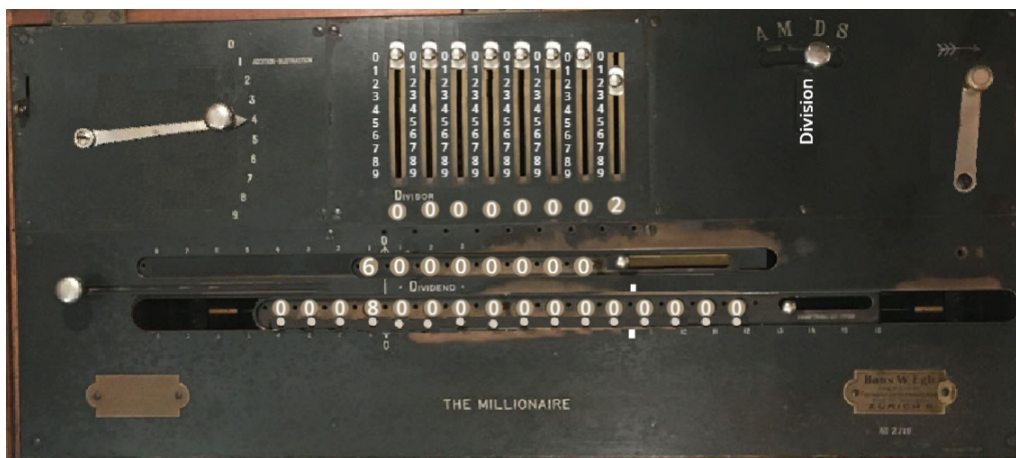


Figura 5.28: Paso 2 del código de la división. Se introduce en la palanca superior izquierda el dígito mayor al que cabe la subdivisión $8/2$. 8 porque es el resto de la subdivisión anterior añadiendo la parte del dividendo que aún no ha sido dividida. En la barra intermedia ya está almacenado el dígito de mayor valor del cociente, calculado en el paso anterior.

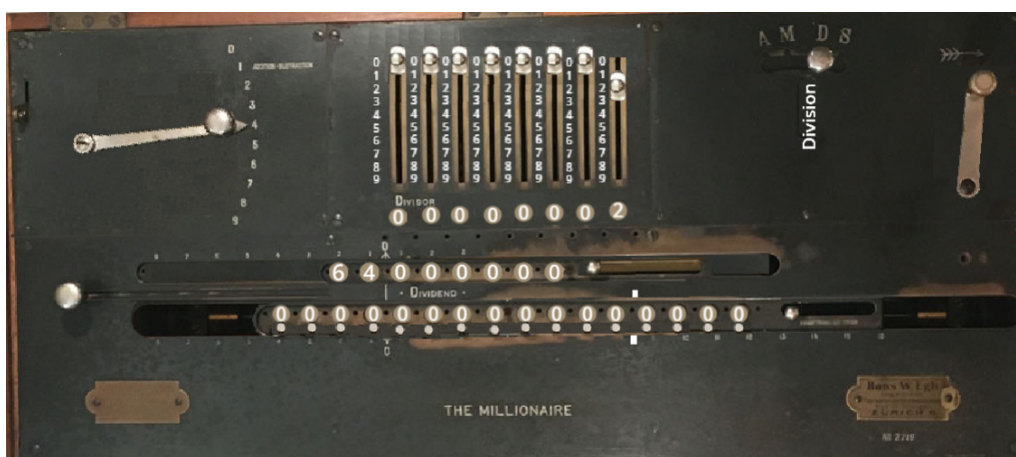


Figura 5.29: Paso 3 del código de la división. Se obtiene el resultado final en la barra intermedia, donde está almacenado el cociente. El resto de la división se queda en la barra inferior, que en este caso es 0.

CAPÍTULO 6

Conclusiones

Tras analizar Scratch, se han explicado las características y las limitaciones de este lenguaje de programación. Ello ha permitido tener un dominio absoluto de esta herramienta. Posteriormente, haciendo un repaso de la evolución histórica de las herramientas de cálculo y del contexto socioeconómico de la sociedad del siglo XIX, se ha comprendido el entorno que favoreció la invención de la máquina. Y gracias a la exhaustiva fase de preparación y análisis, se ha realizado una réplica de la calculadora mecánica Millionaire en Scratch.

Por último, creo que he alcanzado los Objetivos de Desarrollo Sostenible propuestos al inicio de la memoria y que la virtualización de la máquina es un gran activo que incorporar al Museo de Informática de la Escola Tècnica Superior d'Enginyeria Informàtica de la Universitat Politècnica de València.

Bibliografía

- [1] Blanco Vázquez, Carlos. *Historia del cálculo*. Guadalmezán, Madrid, primera edición, 2020.
- [2] Fernandes, Luis. *The Abacus: A Brief History*. Enero, 2015. Consultado en <https://www.ee.ryerson.ca/~elf/abacus/history.html>
- [3] *Guía de Referencia Scratch 1.4*. MIT Media Lab, abril, 2009. <http://eduteka.icesi.edu.co/articulos/ScratchGuiaReferencia>
- [4] Ifrah, Georges. *Historia universal de las cifras*. Espasa Calpe S.A., Madrid, sexta edición, 2008.
- [5] López García, Juan Carlos. *Guía de Referencia Scratch 2.0*. MIT Media Lab, mayo, 2013. <http://eduteka.icesi.edu.co/articulos/Scratch20>
- [6] Páez, Tomás David. *Las matemáticas a lo largo de la historia: de la prehistoria a la antigua Grecia*. Vsiion Libros, Madrid, primera edición, 2009.
- [7] Ribo, Luis. *Historia de Europa en el Siglo XIX*. Octubre, 2011. Ariel, Barcelona, primera edición.
- [8] Veloso, Abel y Molero, Xavier. *El Museo de Informática de la UPV como puerta de entrada a las enseñanzas técnicas universitarias: compromiso con la sociedad en sus actividades didácticas*. Octubre, 2019. Consultado en https://www.27cuieet.es/wp-content/uploads/2019/10/20191011_27CUIEET_BoP.pdf
- [9] Wolff, John. *Descripción técnica de la calculadora Millionaire*. Georgetown university, mayo, 2007. <http://www.johnwolff.id.au/calculators/Tech/Millionaire>

