



Universidad de Valladolid

Escuela de Ingeniería Informática

Trabajo de Fin de Grado

Grado en Ingeniería Informática
Mención Tecnologías de la Información

**DESARROLLO DE RECURSOS DE
APOYO PARA EL APRENDIZAJE DE
CONCEPTOS BÁSICOS RELATIVOS
A FUNCIONES**

Alumno:
Tomás Meroño Madriz

Tutora:
María Rosario Abril Raymundo

A mi familia y a mi pareja Noelia

Agradecimientos

Quiero agradecer en primer lugar a mi tutora por su apoyo durante el desarrollo del proyecto, sin su ayuda este proyecto no habría podido ser posible.

A mi familia por creer siempre en mí.

A mi pareja Noelia, por ser un apoyo tan grande e indispensable.

A mis amigos por animarme siempre a terminarlo.

A mis compañeros del trabajo, la oficina de Valladolid, por su preocupación constante sobre mi progreso con el proyecto.

Al profesorado de la universidad por la formación durante la carrera.

Gracias de verdad.

Resumen

En el presente documento se describe en detalle el desarrollo de un aplicación informática de apoyo al aprendizaje de conceptos básicos relativos a funciones. El objetivo de esta aplicación es desarrollar un producto que sirva como complemento de refuerzo a los alumnos de la asignatura de Fundamentos de Matemáticas del Grado de Ingeniería Informática.

Se ha diseñado e implementado una aplicación de escritorio que consta de una serie de ejercicios desarrollados mediante la librería PyQt5 utilizando un sistema gestor de base de datos MySQL.

Abstract

This document describes in detail the development of a computer application to support the learning of basic concepts related to functions. The objective of this application is to develop a product that serves as a complementary reinforcement for the students of the Fundamentals of Mathematics course of the Computer Engineering Degree.

A desktop application has been designed and implemented consisting of a series of exercises developed using the PyQt5 library and a MySQL database management system.

Índice

Índice de tablas	11
Índice de figuras	13
1. Introducción	1
1.1. Contexto	1
1.2. Motivación	2
1.3. Alcance del proyecto	2
1.4. Estructura de la memoria	2
2. Plan de Proyecto	4
2.1. Metodología implementada	4
2.1.1. Características de la metodología Kanban	4
2.1.2. Implementación de la metodología incremental en el proyecto	5
2.2. Planificación	6
2.3. Estudio de viabilidad	12
2.3.1. Estimación de los costes del proyecto	12
2.3.2. Simulación de los costes del proyecto	12
2.4. Riesgos	13
3. Ejercicios matemáticos a desarrollar	17
3.1. Dominio y recorrido	17
3.2. Funciones elementales	17
3.3. Funciones inversas	18
3.4. Máximos y mínimos	18
4. Análisis	19
4.1. Descripción del proyecto	19
4.2. Objetivos	19
4.2.1. Conceptos	20
4.3. Participantes	20
4.4. Requisitos de la aplicación	20
4.4.1. Requisitos funcionales	21
4.4.2. Requisitos no funcionales	22
4.4.3. Requisitos de información	24
4.5. Casos de uso	24
4.5.1. Diagrama	24
4.5.2. Especificación de los casos de uso	25
4.6. Modelo de dominio	32
5. Diseño	34
5.1. Arquitectura propuesta del sistema	34
5.1.1. Arquitectura basada en capas	34
5.1.2. Desarrollo basado en componentes	34
5.1.3. Diagrama de despliegue	35
5.2. Framework Qt	35
5.3. Patrones de diseño	36

5.3.1.	Patrón Model View	36
5.3.2.	Patrón Singleton	36
5.3.3.	Patrón Data Mapper	37
5.3.4.	Patrón Factoría	38
5.4.	Diseño de la base de datos	38
5.5.	Diseño de la interfaz de usuario	41
5.5.1.	Usuarios objetivo	42
5.5.2.	Objetivos de la interfaz de usuario	42
5.6.	Seguridad	43
6.	Implementación	44
6.1.	Control de versiones	44
6.2.	Implementación de la base de datos	44
6.3.	Gestión de dependencias	45
6.4.	Tipado de variables en Python	46
6.5.	Licencia	47
6.6.	Estructura del proyecto	47
6.7.	Pruebas	49
6.7.1.	Pruebas unitarias	50
6.7.2.	Prueba E2E	53
6.8.	Tecnologías utilizadas	66
6.8.1.	Python	66
6.8.2.	PyQt5	67
6.8.3.	PyQTGraph	67
6.8.4.	Pip	68
6.8.5.	Unittest	68
6.8.6.	CSS	69
6.8.7.	Git	69
6.8.8.	Github	69
6.8.9.	Pycharm	70
6.8.10.	Mypy	70
6.8.11.	Draw.io	70
6.8.12.	Plantuml	70
6.8.13.	Virtual environment	70
6.8.14.	MySQL	70
6.8.15.	Overleaf	70
6.8.16.	Coolers	70
6.8.17.	Flaticon	71
7.	Conclusiones y trabajo a futuro	72
7.1.	Conclusiones	72
7.2.	Trabajo a futuro	72
	Referencias	75
A.	Manual de instalación	78
B.	Repositorio del proyecto	79

Índice de tablas

1.	Iteración-01 Definición del proyecto	6
2.	Iteración-02 Creación del proyecto	7
3.	Iteración-03 Funciones elementales	7
4.	Iteración-04 Funciones inversas	8
5.	Iteración-05 Funcionalidades de ayuda	9
6.	Iteración-06 Dominio y recorrido	10
7.	Iteración-07 Máximos y mínimos	10
8.	Iteración-08 Estadísticas, configuración de ejercicio e información de ejercicio .	11
9.	Iteración-09 Etapa final	11
10.	Iteraciones	12
11.	Coste recursos humanos	13
12.	Coste recursos materiales	13
13.	Coste total	13
14.	Riesgos generales	15
15.	PC-01 Cambio en los requisitos	15
16.	PC-02 Desarrollo de funcionalidades extras	16
17.	ACT-01 Alumno	20
18.	RNF-01 Tiempos de respuesta	23
19.	RNF-02 Errores del sistema	23
20.	RNF-03 Coherencia de la interfaz de usuario de la aplicación	23
21.	RNF-04 Compatibilidad con Windows 10	23
22.	RNF-05 Requerimiento de seguridad	24
23.	IR-01 Información sobre resultado de un ejercicio	24
24.	CU-001 Configuración de una sesión de ejercicios	26
25.	CU-003 Realizar ejercicio: indicar si existe inversa	27
26.	CU-004 Realizar ejercicio: Seleccionar función inversa	27
27.	CU-005 Realizar ejercicio: Restringir dominio para encontrar inversa	28
28.	CU-006 Realizar ejercicio: Seleccionar función elemental	28
29.	CU-007 Realizar ejercicio: Seleccionar desplazamiento	29
30.	CU-008 Realizar ejercicio: Seleccionar máximos y mínimos	30
31.	CU-009 Realizar ejercicio: Definir dominio	31
32.	CU-010 Realizar ejercicio: Definir recorrido	32
33.	TB-001 Temas	39
34.	TB-002 Ejercicios	39
35.	TB-003 Funciones	39
36.	TB-004 Funciones_Ejercicios	40
37.	TB-005 Puntos_Ejercicios	40
38.	TB-006 Resumen_Ejercicios	40
39.	TB-007 Configuracion_Ejercicios	41
40.	TB-008 Configuracion_Pasos	41
41.	TB-009 Visualización de estadísticas	41
42.	Usuarios U-01 Alumnos	42
43.	Pruebas Unitarias-01 Modelo: ejercicios	52
44.	Pruebas Unitarias-02 Modelo: funciones	53
45.	Pruebas E2E-01 Vista de temas	53
46.	Pruebas E2E-02 Acceso a las configuraciones de ejercicio	54
47.	Pruebas E2E-03 Guardar configuración de ejercicio	54

48.	Pruebas E2E-04 Gestión de estados en la configuración de ejercicios	55
49.	Pruebas E2E-05 Acceso a estadísticas del usuario	55
50.	Pruebas E2E-06 Visualización del ejercicio indicar la función elemental	55
51.	Pruebas E2E-07 Realización correcta del ejercicio indicar la función elemental	56
52.	Pruebas E2E-08 Realización incorrecta del ejercicio indicar la función elemental	56
53.	Pruebas E2E-09 Visualización del ejercicio indicar desplazamiento	56
54.	Pruebas E2E-10 Realización correcta del ejercicio indicar desplazamiento	57
55.	Pruebas E2E-11 Realización incorrecta del ejercicio indicar desplazamiento	57
56.	Pruebas E2E-12 Visualización del ejercicio indicar si existe inversa	57
57.	Pruebas E2E-13 Realización correcta del ejercicio indicar si existe inversa	58
58.	Pruebas E2E-14 Realización incorrecta del ejercicio indicar si existe inversa	58
59.	Pruebas E2E-15 Visualización del ejercicio seleccionar la función inversa	58
60.	Pruebas E2E-16 Realización correcta del ejercicio seleccionar la función inversa	59
61.	Pruebas E2E-17 Realización incorrecta del ejercicio seleccionar la función inversa	59
62.	Pruebas E2E-18 Visualización del ejercicio restringir dominio de la función	59
63.	Pruebas E2E-19 Realización correcta del ejercicio restringir dominio de la función	60
64.	Pruebas E2E-20 Realización incorrecta del ejercicio restringir dominio de la función	60
65.	Pruebas E2E-21 Moverse entre ejercicios	61
66.	Pruebas E2E-22 Acceder a la ayuda de cada ejercicio	61
67.	Pruebas E2E-23 Acceder al vídeo de ayuda de cada ejercicio	61
68.	Pruebas E2E-24 Visualización del ejercicio indicar el dominio	62
69.	Pruebas E2E-25 Realización correcta del ejercicio indicar el dominio	62
70.	Pruebas E2E-26 Realización incorrecta del ejercicio indicar el dominio	63
71.	Pruebas E2E-27 Visualización del ejercicio indicar el recorrido	63
72.	Pruebas E2E-28 Realización correcta del ejercicio indicar el recorrido	64
73.	Pruebas E2E-29 Realización incorrecta del ejercicio indicar el recorrido	64
74.	Pruebas E2E-30 Visualización del ejercicio máximos y mínimos	65
75.	Pruebas E2E-31 Realización correcta del ejercicio máximos y mínimos	65
76.	Pruebas E2E-32 Realización incorrecta del ejercicio máximos y mínimos	66

Índice de figuras

1.	Metodología incremental. Figura obtenida de https://www.researchgate.net	6
2.	Boehm's Top 10 List	14
3.	Figura obtenida de Software Engineer, Sommerville	23
4.	Diagrama de casos de uso	25
5.	Modelo de dominio	33
6.	Diagrama de capas	34
7.	Diagrama de despliegue	35
8.	Patrón Singleton	37
9.	Patrón Data Mapper	37
10.	Patrón Factoría	38
11.	Figura obtenida de https://www.atlassian.com/es/git/tutorials/comparing-workflows/ gitflow-workflow	44
12.	Tipos de pruebas	50

1. Introducción

El proyecto va a consistir en la realización de una aplicación de matemáticas que permita ayudar a los alumnos a entender mejor determinados conceptos que se dan en la asignatura *Fundamentos de matemáticas*, conceptos relacionados con *Funciones reales de variable real*.

A lo largo de esta sección se va a exponer lo siguiente: el contexto en el que se ha desarrollado la aplicación, los motivos por los que se ha realizado el proyecto, los objetivos a alcanzar, y cómo se ha estructurado la presente memoria.

1.1. Contexto

La aplicación surge como complemento de refuerzo para la asignatura *Fundamentos de matemáticas* del Grado en Ingeniería de Informática de la Universidad de Valladolid. Esta asignatura está dentro del *Módulo 1: Fundamentos Básicos* del grado y se imparte durante el primer cuatrimestre del primer año de carrera.

La asignatura consta de una carga de trabajo valorada en 6 créditos ECTS (150 horas), que se reparte en los siguientes cuatro bloques impartidos de forma consecutiva:

- Conceptos básicos (1,6 créditos ECTS).
- Álgebra lineal (1,9 créditos ECTS).
- Cálculo diferencial en una variable (1,3 créditos ECTS).
- Cálculos integral en una variable (1,2 créditos ECTS).

En la asignatura ya se utiliza una aplicación matemática para reforzar los conocimientos impartidos en ella, *Derive*. Es una aplicación muy útil y potente que mediante las prácticas de laboratorio impartidas permite al alumno entender los conceptos tratados.

Aunque ya se desarrollan prácticas con *Derive* la aplicación surge para respaldar los conocimientos de la asignatura utilizando un enfoque distinto: crear un entorno de aprendizaje con ejercicios y ejemplos de ayuda muy específicos que incluyan casos considerados problemáticos para el alumno.

Los motivos que han llevado a realizar la aplicación sobre funciones son los siguientes:

- la importancia del bloque: el primer bloque es esencial dentro de la asignatura de *Fundamentos de matemáticas*, no se puede aprobar la asignatura sin obtener una nota mínima de 7 sobre 10 en alguna de las tres pruebas escritas que evalúan este bloque.
- la dificultad del bloque: aunque es el bloque básico de la asignatura, hay conceptos que los alumnos no terminan de dominar, sobre todo al inicio del curso que es cuando se realiza la primera evaluación de este bloque.
- el aspecto visual: los conceptos relacionados con funciones se entienden muy bien mediante representaciones gráficas. La visualización de estos conceptos ayuda mucho a los alumnos a entender correctamente los detalles de cada concepto.

1.2. Motivación

La motivación para la realización de este trabajo de fin de grado surge de una combinación de intereses personales que hicieron para mí muy atractiva la elección de este trabajo de fin de grado.

Por un lado me ha motivado mucho poder realizar un trabajo que pueda ayudar a los alumnos en su desarrollo durante la carrera. Yo he cursado esta asignatura y sé que tiene cierto grado de dificultad para los alumnos. Me ha hecho cuidar mucho los aspectos de la aplicación, pensar en qué podría ser útil para los alumnos de la asignatura. He pensado bastante en mí cuando cursé la asignatura y en cómo me podría haber ayudado a entenderla mejor.

Por otro lado quería hacer un trabajo que me aportara en el aspecto técnico, no quería realizar algo que supiera hacer, quería probar nuevas tecnologías que enriquecieran mi formación. No había creado antes una aplicación de escritorio por lo que el reto técnico me atraía mucho. Probar a hacer cosas distintas al desarrollo web habitual con frameworks como Django y Vue es algo que me llamaba mucho la atención. En proyectos de cierto tamaño no había utilizado antes PyQt ni PyQtGraph por lo que este proyecto cumplía con la condición mencionada. Además el aspecto visual de los conceptos que se iban a tratar permitía poder probar bien las bibliotecas de representación de gráficas 2D.

1.3. Alcance del proyecto

El objetivo principal de este proyecto es crear una aplicación que respalde una serie de conocimientos matemáticos para el usuario con las siguientes características:

- aplicación ejecutable desde windows.
- tratar sobre los conceptos de dominio y recorrido.
- tratar sobre el concepto de inversa.
- tratar sobre las funciones elementales y desplazamientos.
- tratar sobre los conceptos máximos y mínimos.

1.4. Estructura de la memoria

La estructura de la memoria que se ha seguido es la siguiente:

- En el primer capítulo se expone una breve introducción del proyecto, un contexto sobre los temas del proyecto, el objetivo del proyecto y la motivación por la que realizar este proyecto.
- En el segundo capítulo se expone el plan de proyecto que se va a seguir durante el desarrollo del trabajo.
- En el tercer capítulo se va a desarrollar que ejercicios matemáticos se quieren realizar.
- En el cuarto capítulo se expone la etapa de análisis del proyecto: los requisitos funcionales y no funcionales, casos de uso, el modelo de dominio, diagramas de secuencia.
- En el quinto capítulo se expone las decisiones de diseño tomadas para la aplicación.
- En el sexto capítulo se expone la implementación del proyecto.

- En el séptimo capítulo se expone unas breves conclusiones y trabajo futuro a realizar.
- Al final de la memoria se añade la bibliografía y unos anexos de utilidad: el manual de despliegue y el repositorio del código.

2. Plan de Proyecto

En esta sección se va a describir el plan de proyecto seguido durante el desarrollo de este trabajo.

La gestión del proyecto es una actividad crítica en el buen desarrollo del proyecto. Una buena gestión permite cumplir los plazos marcados, ya que en ella se tiene que delimitar correctamente el alcance y los recursos disponibles para la consecución del proyecto.

La gestión del proyecto consiste en la siguiente serie de actividades:

- especificar los objetivos y establecer el alcance del proyecto.
- escoger la metodología de trabajo a seguir, la cual establecerá los procedimientos a seguir para el buen desarrollo del trabajo.
- decidir las tareas necesarias para alcanzar los objetivos del proyecto.
- realizar una planificación para el proyecto. Los diferentes plazos para alcanzar los objetivos.
- decidir como monitorizar el progreso del proyecto, y como arreglar los retrasos.
- investigar cuales son los riesgos reales del proyecto y, a partir de ellos, desarrollar un plan de contingencia.

2.1. Metodología implementada

Se va a desarrollar el proyecto siguiendo una metodología iterativa e incremental similar a Kanban.

2.1.1. Características de la metodología Kanban

La metodología Kanban es un metodología de trabajo ágil basada en el enfoque visual del trabajo en curso, se utiliza para optimizar el flujo de trabajo en un proyecto. [25]

Los conceptos utilizados en la metodología Kanban son los siguientes:

- tareas: son el trabajo a realizar, cada una de ellas tiene una descripción y unos plazos de entrega.
- tablero Kanban: tablero dividido en varias columnas que determinan el progreso de las diferentes tareas, según cambian de estado las tareas se van moviendo por el tablero. Normalmente se usan las columnas “A realizar“, “En progreso“ y “Completado“. El tablero proporciona una visualización inmediata, clara y en tiempo real del estado del proyecto.
- límites de trabajo: existe un límite para el trabajo en curso incluido dentro del tablero, así se evitan sobrecargas de trabajo y el equipo puede centrarse en las tareas existentes en el tablero.

El uso de una metodología basada en Kanban favorece los siguientes puntos:

- mejora del flujo de trabajo: a medida que se realizan las tareas se puede identificar correctamente los problemas durante el desarrollo y optimizarlos para siguientes tareas.

- mejora continua: según el resultado de las distintas tareas hay que ir ajustando las nuevas tareas. Se tiene que identificar las áreas de mejora para poder ajustar los límites de trabajo o incluso añadir o quitar columnas al tablero.
- enfocado en la entrega continua: mediante el tablero se puede visualizar qué tareas se han realizado y cuáles no, permitiendo al gestor de proyecto decidir cuándo se han cumplido las suficientes tareas como para entregar al cliente una nueva funcionalidad. Además, se puede ajustar el tablero para cambiar las tareas y adaptarse continuamente a las exigencias del cliente.

2.1.2. Implementación de la metodología incremental en el proyecto

Se ha escogido una metodología Kanban ya que se ajusta a las necesidades del autor del proyecto, facilitando el desarrollo del proyecto.

En el proyecto se han usado tres columnas:

- a realizar: tareas pendientes sin iniciar.
- en curso: tareas realizándose. En el proyecto rara vez ha habido varias tareas en curso porque solo había un desarrollador.
- completadas: tareas terminadas para poder tener un registro del trabajo realizado y poder mejorar el flujo de trabajo.

Al ser un solo desarrollador se ha optado por limitar las tareas mostradas en el tablero Kanban mediante las iteraciones, en vez de mediante un número límite. En el tablero, en la columna “a realizar“, solo puede haber tareas relacionadas entre sí contenidas dentro de la iteración, evitando así un exceso de tareas y permitiendo saber cuándo el producto ha superado una iteración.

Al acabar las tareas de una iteración se decide la nueva iteración y sus tareas. Finalmente se añaden al tablero todas las tareas de esa iteración.

Para estimar cada tarea se ha seguido la estimación de tiempo propuesta por el PERT (Program Evaluation and Review Technique) porque al usar tecnologías nuevas para el desarrollador hay mucha incertidumbre sobre el tiempo real que va a conllevar cada tarea. La estimación propuesta tiene tres valores:

- duración más probable (ER): la estimación que el desarrollador cree que va a durar realmente la tarea.
- duración más optimista (EO): la estimación que el desarrollador cree que va a durar si hace la tarea a la primera, sin inconvenientes ni interrupciones.
- duración más pesimista (EP): la estimación límite que el desarrollador cree que va a durar si la tarea se complica.

La fórmula es la siguiente:

$$PERT = \frac{EO + 4 \cdot ER + EP}{6} \quad (1)$$

Cada iteración ha seguido las siguientes fases:

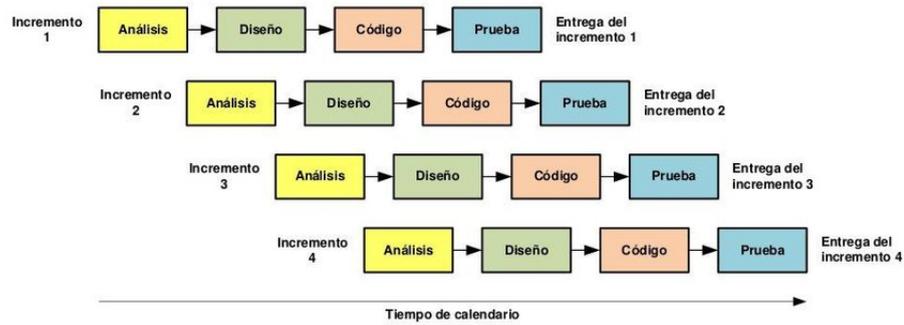


Figura 1: Metodología incremental. Figura obtenida de <https://www.researchgate.net>

2.2. Planificación

A continuación se van a describir las iteraciones realizadas en el proyecto:

Nombre	E. optimista (horas)	E. realista (horas)	E. pesimista (horas)
Repaso del temario de fundamentos.	10	10	10
Decidir las tecnologías.	8	8	8
Estudio de PyQt5.	10	10	10
Estudio de PyQt-Graph.	10	10	10
Requisitos.	5	8	10
Modelo de dominio.	2	5	8
Definición del proyecto	45	51	56

Tabla 1: Iteración-01 Definición del proyecto

Nombre	E. optimista (horas)	E. realista (horas)	E. pesimista (horas)
Creación de la aplicación.	1	1	2
Creación de la primera pantalla (Welcome page).	3	5	8
Creación de la base de datos.	5	8	13
Creación de la pantalla de temario.	5	8	13
Creación de los primeros widgets.	5	8	13
Creación de las primeras rutas de la aplicación.	3	5	13
Creación del proyecto	22	35	62

Tabla 2: Iteración-02 Creación del proyecto

Nombre	E. optimista (horas)	E. realista (horas)	E. pesimista (horas)
Actualización de la base de datos.	2	4	6
Creación de los modelos para funciones.	2	3	5
Creación del mapper para funciones.	5	8	13
Creación de la vista para funciones.	2	3	5
Creación de las rutas para funciones.	2	3	5
Creación de los componentes para funciones elementales.	8	15	21
Testing.	5	8	10
Funciones elementales	26	44	65

Tabla 3: Iteración-03 Funciones elementales

Nombre	E. optimista (horas)	E. realista (horas)	E. pesimista (horas)
Actualización de la base de datos.	2	4	6
Actualización del mapper para funciones.	5	8	13
Actualización de los modelos de funciones para resolver inversas.	2	5	8
Creación de los componentes para inversas.	10	15	26
Testing.	5	8	10
Funciones inversas	24	40	63

Tabla 4: Iteración-04 Funciones inversas

Nombre	E. optimista (horas)	E. realista (horas)	E. pesimista (horas)
Actualización de la base de datos para incluir ordenes.	1	1	2
Inclusión de botones atrás y adelante.	1	2	3
Lógica de navegación.	3	6	10
Inclusión de navegación directa por los pasos de la sesión mediante el widget combobox.	3	5	8
Lógica del combobox.	3	5	7
Inclusión de botón de ayuda.	1	1	1
Actualización de los modelos y del mapper de funciones.	5	8	10
Creación del diálogo de ayuda y lógica de navegación dentro de él.	5	10	13
Creación del proyecto	22	38	54

Tabla 5: Iteración-05 Funcionalidades de ayuda

Nombre	E. optimista (horas)	E. realista (horas)	E. pesimista (horas)
Actualización de la base de datos.	2	4	6
Actualización del mapper para funciones.	3	5	8
Actualización de los modelos de funciones para resolver dominio y recorrido.	5	10	15
Creación de los componentes para dominio y recorrido.	8	12	20
Testing.	5	8	10
Dominio y recorrido	23	39	59

Tabla 6: Iteración-06 Dominio y recorrido

Nombre	E. optimista (horas)	E. realista (horas)	E. pesimista (horas)
Actualización de la base de datos.	2	4	6
Actualización del mapper para funciones.	3	5	8
Actualización de los modelos de funciones para resolver máximos y mínimo.	10	15	20
Creación de los componentes para máximos y mínimos.	12	20	28
Testing.	5	8	10
Máximos y mínimos	32	52	72

Tabla 7: Iteración-07 Máximos y mínimos

Nombre	E. optimista (horas)	E. realista (horas)	E. pesimista (horas)
Actualización de la base de datos.	6	7	8
Actualización y creación de mappers.	3	5	8
Añadir la funcionalidad de estadísticas.	5	8	10
Añadir la funcionalidad de configuración de ejercicio.	10	14	18
Añadir la funcionalidad de información de ejercicio.	5	8	12
Testing.	5	8	10
Estadísticas, configuración de ejercicio e información de ejercicio	34	50	66

Tabla 8: Iteración-08 Estadísticas, configuración de ejercicio e información de ejercicio

Nombre	E. optimista (horas)	E. realista (horas)	E. pesimista (horas)
Repaso de ejercicios de funciones elementales.	2	4	5
Repaso de ejercicios de inversa.	3	6	8
Repaso de ejercicios de dominio y recorrido.	5	8	10
Repaso de ejercicios de máximos y mínimos.	6	10	12
Repaso de otras funcionalidades.	3	4	5
Etapas final	19	32	40

Tabla 9: Iteración-09 Etapa final

A continuación se agrupan las estimaciones de todas las iteraciones para tener una perspectiva final del proyecto:

Nombre	E. optimista (horas)	E. realista (horas)	E. pesimista (horas)
Iteración 1 Definición del proyecto.	45	51	56
Iteración 2 Creación del proyecto.	22	35	62
Iteración 3 Funciones elementales.	26	44	65
Iteración 4 Funciones inversas.	24	40	63
Iteración 5 Funcionalidades de ayuda.	22	38	54
Iteración 6 Dominio y recorrido.	23	39	59
Iteración 7 Máximos y mínimos.	32	52	72
Iteración 8 Estadísticas, configuración de ejercicio e información de ejercicio.	34	50	66
Iteración 9 Etapa final.	19	32	40
Total	247	351	537

Tabla 10: Iteraciones

2.3. Estudio de viabilidad

La última etapa del plan de proyecto es el estudio de viabilidad, el cuál determina si se puede realiza el proyecto. En esta sección se mostrará el coste que supondría llevar a cabo el proyecto en condiciones ideales, sin retrasos ni inconvenientes. Se detallará solo el gasto en recursos humanos y materiales, se podría tener en cuenta otros gastos como servicios (electricidad, internet, oficina, etc.) pero se ha optado por obviarlo.

2.3.1. Estimación de los costes del proyecto

Al ser un proyecto educativo el coste en personal es nulo. Además, el ordenador utilizado para el desarrollo es propiedad del autor, se podría considerar el desgaste pero se ha considerado insignificante. Las herramientas utilizadas gratuitas por lo que el coste en materiales también va a ser nulo.

2.3.2. Simulación de los costes del proyecto

A continuación se va a simular los posibles costes que conllevaría hacer este proyecto en un entorno real.

Para los costes humanos se tendrá un único desarrollador, como se ha realizado el proyecto, durante las 300 horas de duración del proyecto, sin tener en cuenta la seguridad social. El sueldo estimado es de 21,224€ al año [21], teniendo en cuenta que en el año 2023 hay 222 días [22] de trabajo y suponiendo una jornada de ocho horas, al año hay 1.776 horas, por lo que el precio por hora del trabajador es 11,95€/hora.

Puesto de trabajo	Coste por hora	Horas trabajadas	Cantidad de trabajadores	Coste
Desarrollador	11,95€	300	1	3585,15 €
Total				3585,15 €

Tabla 11: Coste recursos humanos

El ordenador utilizado está valorado en 800€, al ser utilizado solo para este proyecto el coste es integro asociado al proyecto, pero si se utilizará para diversos proyectos se podría repartir.

El valor de la licencia de Pycharm asciende 30,13€/mes, se estima un proyecto de una duración de seis meses, así que el coste final es 180,78€. El resto de herramientas siguen siendo gratuitas.

Herramienta software	Coste de la licencia
ordenador	800€
Pycharm	180,78€
Overleaf	0 €
Flaticon	0 €
Git	0 €
Github	0 €
Plantuml online	0 €
Drawio	0 €
Total	980,78 €

Tabla 12: Coste recursos materiales

Recursos	Coste
Humanos	3585,15 €
Materiales	980,78€ €
Total	4565,93 €

Tabla 13: Coste total

2.4. Riesgos

Los riesgos son los posibles problemas que pueden surgir durante el desarrollo del producto, y llegar a tener un gran impacto en él.

Para poder detectarlos correctamente se va a usar *Boehm's Top 10 List*, la cual se centra en los riesgos relacionados con el personal y el desarrollo del software.

1	Personnel Shortfalls: Staffing with top talent; job matching; team-building; morale-building; cross-training; prescheduling key people.
2	Unrealistic Schedules and Budgets: Detailed, multisource cost and schedule estimation; design to cost; incremental development; software reuse; requirements scrubbing.
3	Developing the wrong software functions: Organizational analysis; mission analysis; operational concept formulation; user surveys; prototyping; early users' manuals.
4	Developing the wrong user interface: Prototyping; scenarios; task analysis.
5	Gold-plating. Requirements scrubbing: prototyping; cost-benefit analysis; design to cost.
6	Continuing stream of requirements changes: High change threshold; information-hiding; incremental development (defer changes to later increments).
7	Shortfalls in externally-performed tasks: Reference-checking; pre-award audits; award-fee contracts; competitive design or prototyping; team-building.
8	Shortfalls in externally-furnished components: Benchmarking; inspections; reference checking; compatibility analysis.
9	Real-time performance shortfalls: Simulation; benchmarking; modelling; prototyping; instrumentation; tuning.
10	Straining computer science capabilities: Technical analysis; cost-benefit analysis; prototyping; reference checking.

Figura 2: Boehm's Top 10 List

Además, para calificar correctamente estos riesgos, para cada uno de ellos se va a especificar su probabilidad, el impacto que supondrían y se especificará un plan de mitigación para cada uno de ellos. Por último, en aquellos que superen un nivel de riesgo alto (4) se desarrollará un plan de contingencia.

La probabilidad y el impacto se han calculado según las estimaciones del autor del proyecto.

Ref	Descripción	Probabilidad	Impacto	Nivel de riesgo	Plan de mitigación
R01	Cambios en los requisitos en una fase tardía del proyecto	0,8	10	8	Se detallará adecuadamente los requisitos junto con la tutora para poder definir desde la primera etapa el producto a realizar.
R02	Falta de conocimiento sobre la herramienta PyQT en fases tardías del proyecto	0,7	5	3,5	Se estudiará en una fase inicial PyQT.
R03	La interfaz de usuario se desarrolla mal	0,6	5	3	Se seguirán guías de diseño para realizar una interfaz adecuada junto con las recomendaciones de la tutora.
R04	Desarrollo de funcionalidades extras	0,7	6	4,2	Se centrará el desarrollo en los objetivos marcados.
R05	Problemas de rendimiento de la aplicación	0,5	5	2,5	Buen diseño de la aplicación para evitar posibles fallos
R06	Tecnologías incorrectas para desarrollar el proyecto	0,4	8	3,2	Se buscará tecnologías con suficiente documentación y comunidad.
R07	Enfermad del desarrollador del proyecto	0,3	10	3	-
R08	Problemas con el ordenador	0,2	10	2	Usar repositorios remotos para evitar pérdidas de código.

Tabla 14: Riesgos generales

PC-01	Cambios tardíos en los requisitos
Descripción	Se hará el sistema lo más modular posible que permita introducir cambios de la forma más sencilla posible.

Tabla 15: PC-01 Cambio en los requisitos

PC-02	Desarrollo de funcionalidades extras
Descripción	Se tendrá que dejar de desarrollar estas nuevas funcionalidades y centrarse en los objetivos marcados al inicio del desarrollo.

Tabla 16: PC-02 Desarrollo de funcionalidades extras

3. Ejercicios matemáticos a desarrollar

En esta sección se van a explicar los conceptos matemáticos que se quieren desarrollar en la aplicación. Junto a cada concepto se detallarán las dificultades de aprendizaje que se intentarán subsanar.

Los conceptos matemáticos que se van a desarrollar a lo largo del proyecto están todos relacionados con las funciones, por lo que el enfoque visual en los ejercicios va a ser un componente esencial para transmitir correctamente cada concepto.

3.1. Dominio y recorrido

El objetivo de este bloque es que el alumno con el apoyo de la gráfica de la función, y en ocasiones con conocimiento añadido de la misma, detecte el dominio y recorrido de la función.

El alumno ha aprendido en cursos anteriores a obtener el dominio de funciones que tiene limitación analítica (raíces, fracciones, logaritmos), pero tiene desvinculado esto de la gráfica de la función.

El recorrido es una situación distinta. No es tan frecuente que sepan hacer ejercicios de hallar analíticamente el recorrido, por lo que no hay desvinculación en esta ocasión con el aspecto gráfico, pero tampoco saben identificar el recorrido si se les presenta la gráfica de la función.

El objetivo es solucionar estas carencias presentadas de los alumnos.

Algunos casos en los que detectar dominio y/o recorrido puede ser problemático para el alumno son:

- la función está definida a trozos.
- la presencia de funciones constantes.
- presencia de asíntotas

En este bloque hay dos tipos de ejercicios que corresponden a detectar dominio y detectar recorrido de la función.

El apoyo gráfico en la realización de estos ejercicios con el trazado de bandas horizontales (recorrido), verticales (dominio), se espera que suponga una ayuda significativa en la comprensión de los conceptos tratados.

3.2. Funciones elementales

El conocimiento de las gráficas y funciones elementales es de vital importancia para todo el cálculo. Aunque el alumno conoce algunas de cursos anteriores, en Fundamentos de Matemáticas se agrupan por familias y se amplía el muestrario. Para que el alumno supere la asignatura es muy importante que sepa reconocerlas y trazarlas. En estas gráficas el alumno tiene información relevante sobre las funciones como dominio, comportamiento en límites, acotaciones, asíntotas...

En el tipo de ejercicio diseñado en este bloque se quiere que el alumno reconozca las funciones elementales y pequeñas variaciones de ellas. Para ello se tendrá un conjunto de gráficas elementales que se le presentará al usuario de forma aleatoria en grupos de cuatro funciones.

Cabe destacar de este bloque la ayuda en la que pueden verse desplazamientos de funciones, lo que facilita al alumno la identificación de las gráficas y el manejo de desplazamientos.

3.3. Funciones inversas

El concepto de inversa resulta complicado para el alumno desde la misma definición. En la base de estos problemas está el concepto de inyectividad y el de restricción. Se pretende con los ejercicios diseñados ayudar con esto.

En este bloque se presentan tres tipos de ejercicios:

- en el primer tipo hay que determinar si la función tiene inversa. Esto es lo mismo que determinar si la función es inyectiva.
- en el segundo tipo hay que seleccionar la inversa entre un muestrario ofrecido.
- en el tercer tipo hay que restringir el dominio para que la función tenga inversa.

En la ayuda de estos ejercicios el trazado de rectas paralelas a los ejes resulta de apoyo para la buena comprensión de los conceptos.

3.4. Máximos y mínimos

Los máximos y mínimos de funciones presentan dos tipos de problemas principalmente. En primer lugar, el alumno tiene asociados los extremos relativos a dos situaciones particulares que corresponden a funciones con derivada cero en los puntos (montañitas y valles). Esto hace que no consideren extremos a puntos que lo son pero no entran en el caso particular. El segundo problema es la creencia de que los extremos absolutos son únicos. Los dos problemas son consecuencia de un estudio no cuidadoso de las definiciones. Los ejercicios propuestos pretenden ayudar con estos conceptos.

El tipo de ejercicio consiste en detectar, dada una gráfica, los máximos y mínimos, insistiendo en las situaciones “raras” para el alumno como los puntos de discontinuidad.

4. Análisis

4.1. Descripción del proyecto

El proyecto a desarrollar consiste en una aplicación informática de escritorio cuyo objetivo es ayudar a los alumnos a entender conceptos relativos sobre funciones del temario de la asignatura de Fundamento de Matemáticas. En concreto se quieren tratar los siguientes aspectos del temario:

1. dominio y recorrido de una función.
2. funciones elementales.
3. función inversa.
4. máximos y mínimos.

Estos aspectos determinan los cuatro bloques de ejercicios con los que se trabajará en la aplicación.

Para alcanzar este objetivo se requiere una aplicación que cumpla con los siguientes aspectos:

- Se necesita una muestra amplia y variada de ejercicios que permitan al usuario plantearse diferentes situaciones al tratar los diferentes conceptos matemáticos existentes dentro de la aplicación. Para permitir al usuario concentrarse en los ejercicios que él crea que son relevantes para su formación habrá que facilitar que el usuario pueda activar o desactivar los tipos de ejercicios.
- Dar al usuario una permanente retroalimentación sobre sus fallos al realizar los diferentes ejercicios, cada vez que una pregunta sea contestada erróneamente habrá que buscar la forma de mostrar gráficamente el error, mostrándolo de forma sencilla y muy visual. Es muy importante mostrarlo de forma visual porque los conceptos tratados en la aplicación están pensados para ser entendidos así.
- Permitir al usuario navegar sin restricciones por los distintos ejercicios realizados durante la sesión de ejercicios, para que durante la misma sesión pueda utilizar sus fallos para acertar las siguientes preguntas.
- Guardar registro de las respuestas del usuario para poder mostrar estadísticas que le ayuden a entender que conceptos necesita trabajar más. Este aspecto está relacionado con el primero, al permitir al usuario elegir qué tipo de ejercicios realizar puede centrarse en aquellos con peores estadísticas.
- Mostrar pequeños ejemplos sobre cada tipo de ejercicio mientras el usuario está realizando estos para que pueda tener una pequeña ayuda que pueda aplicar durante su sesión de ejercicios. El objetivo de la aplicación es ayudar al usuario a entender los distintos conceptos, por eso no importa que por esta ayuda acierte más de los que debería.

4.2. Objetivos

La aplicación debe de cumplir:

- estar diseñada para windows 10.

- reunir una muestra suficientemente amplia de ejercicios para los objetivos de aprendizaje deseados.
- guardar respuestas del usuario.
- utilizar las respuestas del usuario para generar estadísticas.
- permitir la navegación sin restricciones entre ejercicios.
- mostrar ayudas al alumno en la realización del ejercicio.

4.2.1. Conceptos

En esta sección se van a definir dos conceptos que se utilizarán a lo largo de la memoria:

- ejercicios: es una actividad concreta a resolver. Consta de una pregunta, de una ayuda para el usuario y de una solución a comprobar. Son la unidad mínima dentro de la aplicación.
- sesión de ejercicios: con este concepto nos referiremos al número y tipo de ejercicios que el alumno quiere realizar siempre dentro de un mismo bloque.

4.3. Participantes

A continuación se va a detallar el único rol posible dentro de la aplicación. El objetivo del proyecto es transmitir determinado conocimiento matemático a los alumnos, por ello, el desarrollo se va a enfocar la aplicación al uso de usuarios, facilitando conseguir este objetivo:

ACT-01	Usuario
Descripción	Es la persona que utiliza la aplicación para realizar los distintos ejercicios posibles. No necesita estar registrada.
Comentarios	La aplicación va a estar enfocada en transmitir de la mejor forma posible los conceptos matemáticos a los usuarios.

Tabla 17: ACT-01 Alumno

4.4. Requisitos de la aplicación

En este apartado se va a detallar los requisitos a cumplir del proyecto. Los requisitos especifican las propiedades y capacidades necesarias que debe tener el software para cumplir correctamente con el funcionamiento esperado. Sirven de contrato entre el cliente y el equipo de desarrollo porque detallan las funciones que el sistema podrá realizar y las restricciones que tendrá.

Los requisitos pueden ser:

- requisitos funcionales, describen el funcionamiento del sistema, los servicios que se van a proporcionar y cómo va a responder ante determinadas situaciones.
- requisitos no funcionales, especificaciones enfocadas en la descripción de las propiedades que debe tener el sistema, propiedades como el tiempo de respuesta y la fiabilidad.

A continuación se detallan los requisitos funcionales y no funcionales del proyecto refinados a lo largo del desarrollo del trabajo.

4.4.1. Requisitos funcionales

1. El sistema deberá permitir al usuario crear una sesión de ejercicios.
 - 1.1. El sistema deberá poder mostrar al usuario los diferentes temas existentes en la aplicación.
 - 1.2. El sistema deberá permitir al usuario elegir el tema de la sesión de ejercicios.
 - 1.3. El sistema deberá permitir al usuario elegir el número de ejercicios de la sesión.
 - 1.4. El sistema deberá permitir al usuario elegir el tipo de ejercicios activos en la sesión, cada tema tiene una serie de ejercicios asociados que son los cuáles podrán estar o no activos.
 - 1.5. El sistema deberá poder escoger de forma aleatoria los ejercicios de la sesión, teniendo en cuenta los parámetros definidos para la sesión por el usuario.
2. El sistema deberá poder mostrar gráficas e interactuar con ellas.
 - 2.1. El sistema deberá poder permitir definir los valores máximos para los ejes de abscisa y ordenada para una gráfica de un ejercicio concreto.
 - 2.2. El sistema deberá poder mostrar gráficas con los siguientes elementos: funciones con dominios definidos y puntos.
 - 2.3. El sistema deberá de poder diferenciar las distintas funciones contenidas dentro de la gráfica para poder resolver correctamente los ejercicios.
 - 2.4. El sistema deberá permitir al usuario seleccionar una función de la gráfica mostrada haciendo click sobre ella.
 - 2.5. El sistema deberá de poder mostrar textos en las gráficas de los ejercicios.
 - 2.6. El sistema deberá permitir al usuario conocer las coordenadas de cada cuadrícula de la gráfica para facilitar la resolución de los ejercicios.
 - 2.7. El sistema deberá permitir introducir un rango vertical u horizontal de valores dentro de la gráfica para resaltar determinadas zonas.
 - 2.8. El sistema deberá permitir al usuario introducir puntos en la gráfica.
3. El sistema deberá permitir a los usuarios realizar los ejercicios definidos en la aplicación.
 - 3.1. El sistema deberá poder mostrar la pregunta asociada al ejercicio.
 - 3.2. El sistema deberá poder mostrar al usuario distintas opciones de respuesta contenidas en botones seleccionables.
 - 3.3. El sistema deberá poder validar la respuesta del usuario sobre un determinado ejercicio.
 - 3.4. El sistema deberá poder mostrar un vídeo explicativo sobre el funcionamiento de cada ejercicio para facilitar al usuario la realización del mismo.
 - 3.5. El sistema deberá poder mostrar una ayuda gráfica y un texto explicativo sobre los conceptos matemáticos tratados en cada ejercicio.
 - 3.6. El sistema deberá poder mostrar una explicación gráfica al usuario cuando falle un ejercicio.
4. El sistema deberá permitir al usuario navegar entre los distintos ejercicios de la sesión.
 - 4.1. El sistema deberá permitir al usuario moverse hacia adelante y hacia atrás entre los distintos ejercicios de la sesión.

- 4.2. El sistema deberá permitir al usuario moverse directamente a un ejercicio concreto de la sesión.
- 4.3. El sistema deberá permitir al usuario saltarse ejercicios.
- 4.4. El sistema deberá permitir al usuario salir de la sesión de ejercicios en cualquier momento.
5. El sistema deberá poder resolver ejercicios de funciones inversas correctamente.
 - 5.1. El sistema deberá poder detectar correctamente si una función tiene inversa en un determinado dominio.
 - 5.2. El sistema deberá poder saber la función inversa de una función en un determinado dominio.
6. El sistema deberá gestionar ejercicios de dominio y recorrido correctamente.
 - 6.1. El sistema deberá poder conocer el dominio y rango de cada función en un ejercicio determinado.
 - 6.2. El sistema deberá permitir al usuario introducir un intervalo de valores numérico.
 - 6.3. El sistema deberá permitir al usuario borrar el último intervalo de valores introducido.
 - 6.4. El sistema deberá poder saber los valores equivocados del dominio y rango introducido por el usuario.
 - 6.5. El sistema deberá poder saber los valores faltantes del dominio y rango introducido por el usuario.
7. El sistema deberá gestionar ejercicios de funciones elementales correctamente.
 - 7.1. El sistema deberá poder dibujar correctamente desplazamientos sobre funciones.
 - 7.2. El sistema deberá poder detectar el desplazamiento de una determinada función.
8. El sistema deberá gestionar ejercicios de máximos y mínimos relativos y absolutos correctamente.
 - 8.1. El sistema deberá poder conocer los puntos máximos y mínimos de una función en un dominio determinado.
 - 8.2. El sistema deberá permitir al usuario seleccionar el tipo de punto introducido en la gráfica, máximo o mínimo relativo o absoluto.
 - 8.3. El sistema deberá poder conocer los puntos equivocados de la respuesta del usuario.
9. El sistema deberá gestionar el resultado de los ejercicios.
 - 9.1. El sistema deberá poder almacenar la respuesta del usuario a un ejercicio concreto.
 - 9.2. El sistema deberá permitir al usuario visualizar sus estadísticas en la resolución de los ejercicios.

4.4.2. Requisitos no funcionales

Para obtener y delimitar correctamente los requisitos no funcionales, se va a seguir el modelo [F]URPS, centrado en la usabilidad, fiabilidad, rendimiento y soporte. También se va a tener en cuenta el modelo propuesto en (*Sommerville, 2000*) expuesto en la siguiente imagen:



Figura 3: Figura obtenida de Software Engineer, Sommerville

ID	Nombre	Descripción
RNF-01	Tiempo de respuesta	La aplicación no deberá de tardar en responder a las interacciones del alumno más de un segundo.

Tabla 18: RNF-01 Tiempos de respuesta

ID	Nombre	Descripción
RNF-02	Errores del sistema	La probabilidad de fallo de la aplicación al realizar los ejercicios deberá de ser menor del 1 %, al estar los ejercicios probados correctamente.

Tabla 19: RNF-02 Errores del sistema

ID	Nombre	Descripción
RNF-03	Coherencia de la interfaz de usuario de la aplicación	La aplicación se tendrá que desarrollar de forma uniforme permitiendo mantener a lo largo de la aplicación el mismo estilo y facilitando al usuario el uso de ella.

Tabla 20: RNF-03 Coherencia de la interfaz de usuario de la aplicación

ID	Nombre	Descripción
RNF-07	Compatibilidad con Windows 10	La aplicación tiene que poder ejecutarse en Windows 10 para permitir un uso sencillo para los alumnos.

Tabla 21: RNF-04 Compatibilidad con Windows 10

ID	Nombre	Descripción
RNF-05	Requerimiento de seguridad	La aplicación no deberá permitir la interacción directa entre el alumno y la base de datos.

Tabla 22: RNF-05 Requerimiento de seguridad

4.4.3. Requisitos de información

A continuación se va a mostrar aquellas entidades que se van a recopilar a lo largo de la aplicación:

IR-01	Información sobre el resultado de un ejercicio
	Tipo de ejercicio
	Respuesta del alumno
	Es correcta la respuesta
	Identificador del ejercicio

Tabla 23: IR-01 Información sobre resultado de un ejercicio

4.5. Casos de uso

Los casos de uso son un componente esencial para poder obtener todos los requisitos de la aplicación y definir correctamente los flujos. Describen como usarán los usuarios la aplicación, por lo que permiten completar los requisitos ya obtenidos y encontrar otros que puedan faltar en el proyecto.

4.5.1. Diagrama

A continuación se expone el diagrama de casos de uso de la aplicación:

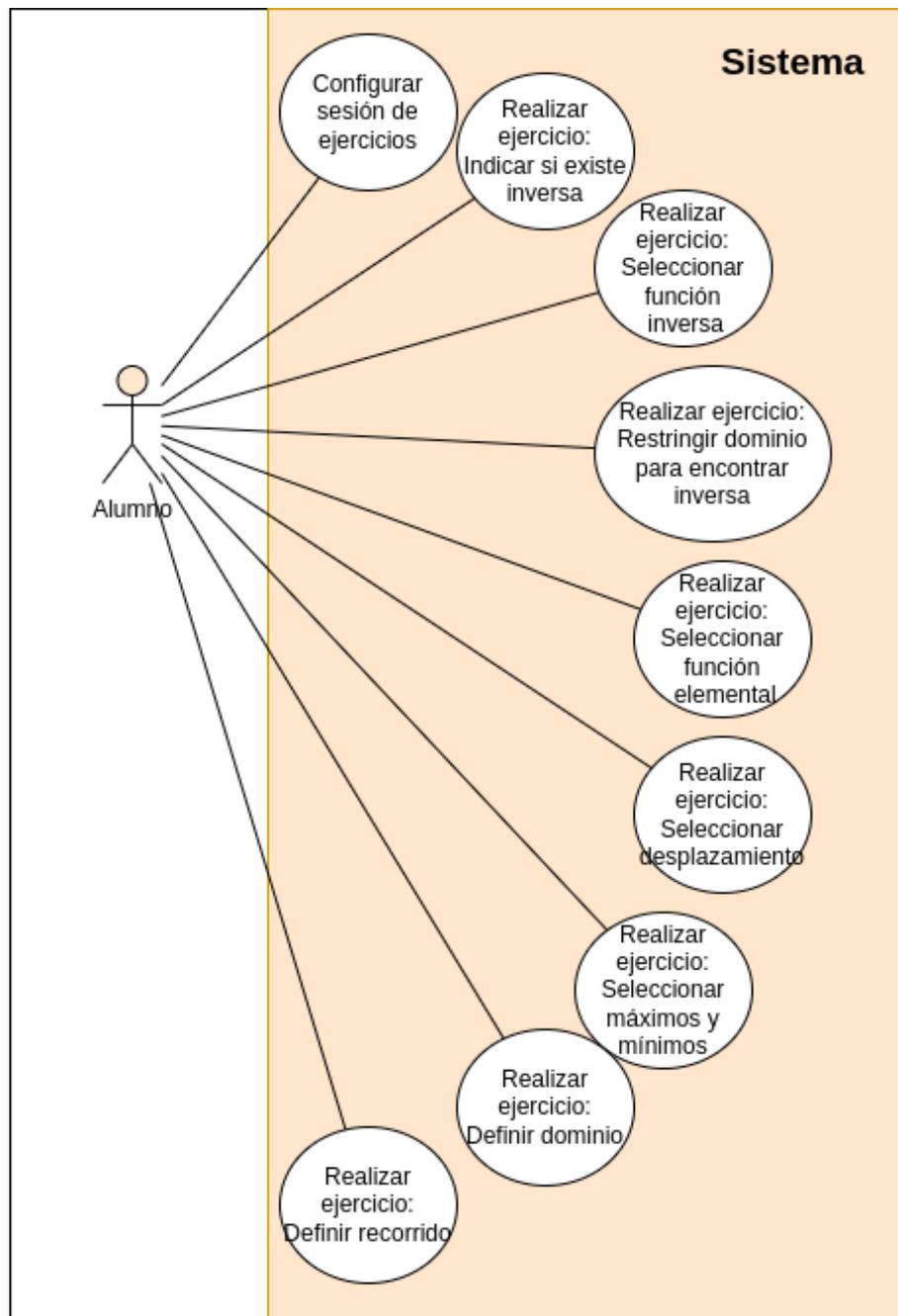


Figura 4: Diagrama de casos de uso

4.5.2. Especificación de los casos de uso

A continuación se va a mostrar la especificación de los casos de uso más importantes dentro de la aplicación:

CU-001	Configuración de una sesión de ejercicios	
Descripción	El caso de uso describe el proceso a través del cual un usuario puede configurar una sesión de ejercicios.	
Actor	ACT-01	
Precondición	-	
Secuencia normal	Paso	Acción
	1	El usuario solicita configurar una sesión de ejercicios de un tema concreto.
	2	El sistema muestra al usuario todas las opciones disponibles configurables: muestra el número de ejercicios de la sesión y el tipo de ejercicios activos.
	3	El usuario solicita cambiar el número de ejercicios de la sesión.
	4	El sistema comprueba que el número de ejercicios seleccionados no es menor que uno ni mayor que el número de ejercicios máximos disponibles.
	5	El usuario introduce que tipos de ejercicios quiere desactivar y activar en el temario.
	6	El sistema comprueba que siempre haya al menos un tipo de ejercicio activo.
	9	El usuario solicita guardar la configuración.
	10	El sistema guarda la configuración pedida por el usuario.
	Excepciones	Paso
4'		Si el número de ejercicios introducidos no cumple la validación, el valor del número de ejercicios de la sesión vuelve a estar en el mismo estado anterior.
7'		Si no existe al menos un tipo ejercicio activo el sistema activa el último ejercicio desactivado.
Postcondición	El usuario tendrá un tema configurado según sus preferencias.	

Tabla 24: CU-001 Configuración de una sesión de ejercicios

CU-001	Realizar ejercicio: indicar si existe inversa	
Descripción	El caso de uso describe el proceso a través del cual un usuario resuelve el tipo de ejercicio en él que indica si una función en un dominio determinado tiene inversa.	
Actor	ACT-01	
Precondición	El usuario está realizando el ejercicio <i>Indicar si existe inversa</i>	
Secuencia normal	Paso	Acción
	1	El sistema muestra al usuario el ejercicio a resolver: muestra la pregunta, representa la gráfica e incluye las opciones Sí o No como posibles respuestas.
	2	El usuario indica su respuesta al ejercicio.
	3	El sistema valida la respuesta.
	4'	Si la respuesta es correcta el sistema muestra un texto indicando al usuario que la respuesta ha sido correcta.
4''	Si la respuesta es incorrecta el sistema muestra dos puntos que con imágenes repetidas.	
Excepciones	Paso	Acción
	-	-
Postcondición	El usuario habrá resuelto un ejercicio de indicar si existe inversa y se habrá guardado en el sistema su respuesta.	

Tabla 25: CU-003 Realizar ejercicio: indicar si existe inversa

CU-001	Realizar ejercicio: Seleccionar función inversa	
Descripción	El caso de uso describe el proceso a través del cual un usuario resuelve el tipo de ejercicio de selección de la función inversa de la función principal.	
Actor	ACT-01	
Precondición	El usuario está realizando el ejercicio <i>Selección de inversa</i>	
Secuencia normal	Paso	Acción
	1	El sistema muestra al usuario el ejercicio a resolver: muestra la pregunta, representa la gráfica e incluye cuatro funciones como posibilidades de ser la inversa de la función principal.
	2	El usuario realiza un click sobre la función que cree que es la inversa.
	3	El sistema valida la respuesta.
	4'	Si la respuesta es correcta el sistema muestra un texto indicando al usuario que la respuesta ha sido correcta.
4''	Si la respuesta es incorrecta el sistema muestra en rojo la función seleccionada por el usuario, muestra la bisectriz del primer y tercer cuadrante y muestra en verde la respuesta correcta.	
Excepciones	Paso	Acción
	-	-
Postcondición	El usuario habrá resuelto un ejercicio de selección de inversa y se habrá guardado en el sistema su respuesta.	

Tabla 26: CU-004 Realizar ejercicio: Seleccionar función inversa

CU-001	Realizar ejercicio: Restringir dominio para encontrar inversa	
Descripción	El caso de uso describe el proceso a través del cual un usuario resuelve el tipo de ejercicio de restricción de dominio sobre una función para la existencia de inversa en el dominio restringido.	
Actor	ACT-01	
Precondición	El usuario está realizando el ejercicio <i>Restricción de dominio para funciones</i>	
Secuencia normal	Paso	Acción
	1	El sistema muestra al usuario el ejercicio a resolver: muestra la pregunta, representa la gráfica e incluye un rango de valores que marcan cierta parte de la gráfica, inicialmente -1 a 1.
	2	El usuario mueve el rango de valores.
	3	El usuario solicita validar su respuesta.
	4	El sistema valida la respuesta.
	5'	Si la respuesta es correcta el sistema muestra un texto indicando al usuario que la respuesta ha sido correcta.
5''	Si la respuesta es incorrecta el sistema muestra en rojo dos puntos del rango de valores seleccionado por el usuario que tengan imágenes repetidas.	
Excepciones	Paso	Acción
	-	-
Postcondición	El usuario habrá resuelto un ejercicio de restricción de dominio para funciones y se habrá guardado en el sistema su respuesta.	

Tabla 27: CU-005 Realizar ejercicio: Restringir dominio para encontrar inversa

CU-001	Realizar ejercicio: Seleccionar función elemental	
Descripción	El caso de uso describe el proceso a través del cual un usuario resuelve el tipo de ejercicio de selección de una función elemental.	
Actor	ACT-01	
Precondición	El usuario está realizando el ejercicio <i>Selección de la función elemental</i>	
Secuencia normal	Paso	Acción
	1	El sistema muestra al usuario el ejercicio a resolver: muestra la pregunta, representa la gráfica e incluye una serie de botones con diferente expresiones de funciones elementales como posibles respuestas
	2	El usuario selecciona una respuesta.
	3	El sistema valida la respuesta.
	4'	Si la respuesta es correcta el sistema muestra un texto indicando al usuario que la respuesta ha sido correcta.
	4''	Si la respuesta es incorrecta el sistema muestra en rojo la función seleccionada junto con su expresión, y muestra en la gráfica la expresión esperada en blanco encima de la función principal.
Excepciones	Paso	Acción
	-	-
Postcondición	El usuario habrá resuelto un ejercicio de selección de función elemental y se habrá guardado en el sistema su respuesta.	

Tabla 28: CU-006 Realizar ejercicio: Seleccionar función elemental

CU-001	Realizar ejercicio: Seleccionar desplazamiento	
Descripción	El caso de uso describe el proceso a través del cual un usuario resuelve el tipo de ejercicio de selección del desplazamiento sobre una función.	
Actor	ACT-01	
Precondición	El usuario está realizando el ejercicio <i>Selección del desplazamiento de una función</i>	
Secuencia normal	Paso	Acción
	1	El sistema muestra al usuario el ejercicio a resolver: muestra la pregunta, representa la gráfica e incluye una serie de botones con diferentes expresiones de funciones desplazadas como posibles respuestas
	2	El usuario selecciona una respuesta.
	3	El sistema valida la respuesta.
	4'	Si la respuesta es correcta el sistema muestra un texto indicando al usuario que la respuesta ha sido correcta.
4''	Si la respuesta es incorrecta el sistema muestra en rojo la función seleccionada junto con su expresión, y muestra en la gráfica la expresión esperada en blanco encima de la función principal.	
Excepciones	Paso	Acción
	-	-
Postcondición	El usuario habrá resuelto un ejercicio de selección de desplazamiento y se habrá guardado en el sistema su respuesta.	

Tabla 29: CU-007 Realizar ejercicio: Seleccionar desplazamiento

CU-001	Realizar ejercicio: Seleccionar máximos y mínimos	
Descripción	El caso de uso describe el proceso a través del cual un usuario resuelve el tipo de ejercicio de detección de máximos y mínimos relativos absolutos.	
Actor	ACT-01	
Precondición	El usuario está realizando el ejercicio <i>Detección de mínimos y máximos de una función</i>	
Secuencia normal	Paso	Acción
	1	El sistema muestra al usuario el ejercicio a resolver: muestra la pregunta y representa la gráfica.
	2	El usuario selecciona un punto en la gráfica haciendo click.
	3	El sistema muestra al usuario el punto seleccionado, muestra el tipo de punto posible (máximo o mínimo relativo o absoluto). También muestra al usuario un posible rango de valores.
	4	El usuario introduce su respuesta sobre el punto seleccionado.
	5	El sistema dibuja en la gráfica el punto o la función constante seleccionada por el usuario.
	6	El usuario solicita validar su respuesta.
	7	El sistema validar su respuesta.
	8'	Si la respuesta es correcta el sistema muestra un texto indicando al usuario que la respuesta ha sido correcta.
8''	Si la respuesta es incorrecta el sistema muestra en naranja los puntos que faltan en la respuesta del usuario, y muestra en rojo los puntos incorrectos introducidos. Junto a cada punto muestra una pequeña descripción que indica el tipo de punto o constante que es.	
Excepciones	Paso	Acción
	-	-
Postcondición	El usuario habrá resuelto un ejercicio de máximos y mínimos y se habrá guardado en el sistema su respuesta.	

Tabla 30: CU-008 Realizar ejercicio: Seleccionar máximos y mínimos

CU-001	Realizar ejercicio: Definir dominio	
Descripción	El caso de uso describe el proceso a través del cual un usuario resuelve el tipo de ejercicio de definición del dominio de una función.	
Actor	ACT-01	
Precondición	El usuario está realizando el ejercicio <i>Definición del dominio</i>	
Secuencia normal	Paso	Acción
	1	El sistema muestra al usuario el ejercicio a resolver: muestra la pregunta, dos botones para añadir y borrar conjunto de valores y representa la gráfica.
	2	El usuario selecciona el botón de añadir un conjunto de valores.
	3	El sistema muestra al usuario una caja de texto donde introducir el conjunto de valores para el dominio.
	4	El usuario introduce un conjunto de valores.
	5	El sistema comprueba que el conjunto de valores es válido.
	6	El usuario solicita borrar el último conjunto de valores.
	7	El sistema borra el último conjunto de valores introducido por el usuario.
	8	El usuario solicita validar su respuesta.
	9	El sistema valida su respuesta.
	10'	Si la respuesta es correcta el sistema muestra un texto indicando al usuario que la respuesta ha sido correcta.
10''	Si la respuesta es incorrecta el sistema muestra en que valores se ha confundido el usuario.	
Excepciones	Paso	Acción
	5'	Si el conjunto de valores introducido no es válido, el caso de uso comienza de nuevo en el ejercicio 4
Postcondición	El usuario habrá resuelto un ejercicio de definición de dominio y se habrá guardado en el sistema su respuesta.	

Tabla 31: CU-009 Realizar ejercicio: Definir dominio

CU-001	Realizar ejercicio: Definir recorrido	
Descripción	El caso de uso describe el proceso a través del cual un usuario resuelve el tipo de ejercicio de definición del recorrido de una función.	
Actor	ACT-01	
Precondición	El usuario está realizando el ejercicio <i>Definición del recorrido</i>	
Secuencia normal	Paso	Acción
	1	El sistema muestra al usuario el ejercicio a resolver: muestra la pregunta, dos botones para añadir y borrar conjunto de valores y representa la gráfica.
	2	El usuario selecciona el botón de añadir un conjunto de valores.
	3	El sistema muestra al usuario una caja de texto donde introducir el conjunto de valores para el recorrido.
	4	El usuario introduce un conjunto de valores.
	5	El sistema comprueba que el conjunto de valores es válido.
	6	El usuario solicita borrar el último conjunto de valores.
	7	El sistema borra el último conjunto de valores introducido por el usuario.
	8	El usuario solicita validar su respuesta.
	9	El sistema valida su respuesta.
	10'	Si la respuesta es correcta el sistema muestra un texto indicando al usuario que la respuesta ha sido correcta.
10''	Si la respuesta es incorrecta el sistema muestra en que valores se ha confundido el usuario.	
Excepciones	Paso	Acción
	5'	Si el conjunto de valores introducido no es válido, el caso de uso comienza de nuevo en el ejercicio 4
Postcondición	El usuario habrá resuelto un ejercicio de definición de recorrido y se habrá guardado en el sistema su respuesta.	

Tabla 32: CU-010 Realizar ejercicio: Definir recorrido

4.6. Modelo de dominio

A continuación se mostrará el modelo de dominio de la aplicación:

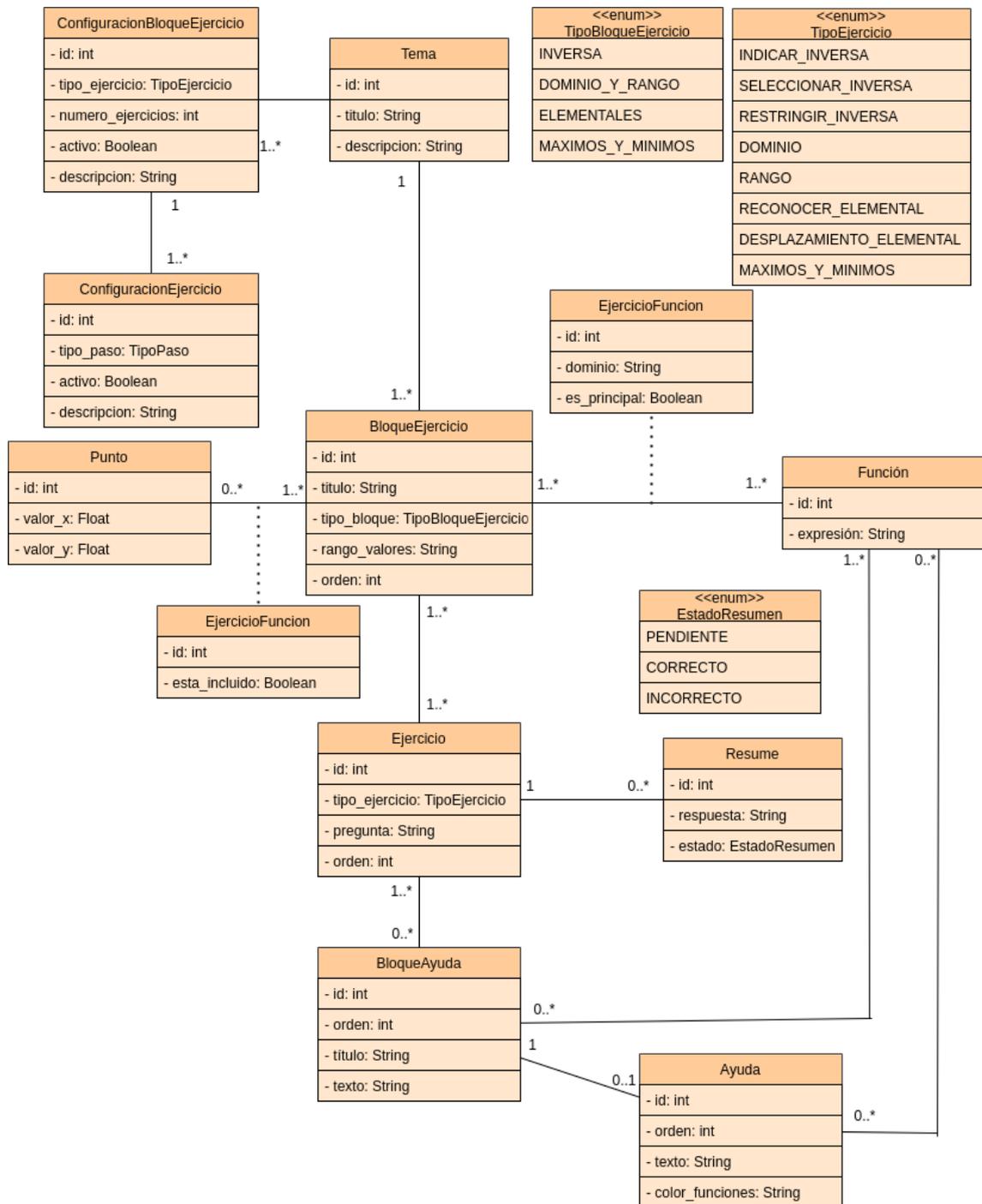


Figura 5: Modelo de dominio

5. Diseño

El diseño es una etapa fundamental en el proceso de desarrollo de software, en él se va a establecer la estructura y los detalles de cada aspecto de la aplicación. El diseño de la aplicación define dos características fundamentales de un proyecto informático:

- **mantenibilidad:** característica que indica la facilidad de una aplicación a ser modificada para corregir errores, para cambios de funcionalidad o para inclusión de nuevas funcionalidades.
- **rendimiento:** característica que indica la eficiencia de una aplicación para realizar una tarea en un determinado tiempo.

A continuación se va a detallar todo el proceso de diseño realizado durante el proyecto: la arquitectura software de la aplicación, las decisiones relativas a la interfaz de usuario, el estado final de la base de datos, los patrones de diseño aplicados y la seguridad, entre otros aspectos.

5.1. Arquitectura propuesta del sistema

5.1.1. Arquitectura basada en capas

Arquitectura a partir de la cual se organiza el sistema en capas, donde cada capa cumple un propósito y se comunica con las capas adyacentes de manera controlada.

Se proponen las siguientes capas:

- **capa de presentación:** corresponde a la interfaz de usuario, se encarga de toda la interacción con el usuario. En la aplicación esta capa está compuesta por componentes, vistas y widgets.
- **capa de negocio:** corresponde a las clases que contienen toda la lógica de negocio. Se encarga de coordinar la interacción entre la capa de presentación y la capa de acceso a datos. En la aplicación esta capa está compuesta por los modelos.
- **capa de persistencia:** corresponde a las clases que se encargan del acceso a los datos almacenados. Proporciona la funcionalidad para realizar operaciones CRUD sobre datos almacenados. En la aplicación esta capa corresponde a la base de datos.

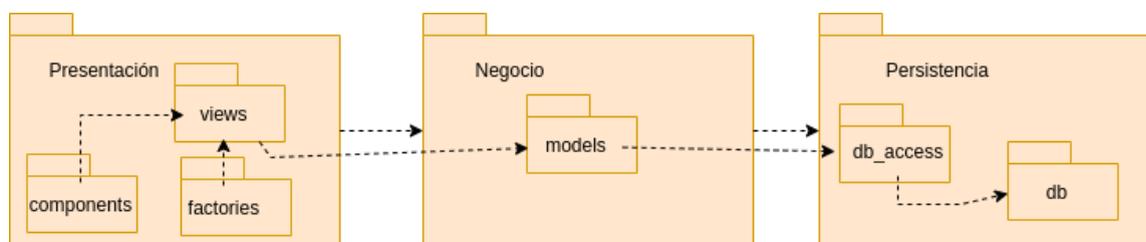


Figura 6: Diagrama de capas

5.1.2. Desarrollo basado en componentes

El desarrollo basado en componentes consiste en el uso de componentes reutilizables para formar sistemas más grandes. Un componente es una clase que encapsula una funcionalidad concreta y puede ser reutilizable a lo largo de la aplicación para crear componentes más grandes.

En la aplicación se usan componentes de PyQt5 y componentes creados en la aplicación, estos últimos han sido creados como clases envoltorio para algunos componentes de PyQt5, obteniendo así una interfaz común a lo largo del proyecto, ayudando a la cohesión de la interfaz de usuario de la aplicación. El uso de componentes junto con las factorías ha servido para acelerar el desarrollo de la aplicación.

5.1.3. Diagrama de despliegue

Un diagrama de despliegue es diagrama utilizado para visualizar la configuración física del sistema. Su objetivo principal es mostrar cómo se distribuyen los diferentes elementos del sistema en el entorno de despliegue.

Como es una aplicación de escritorio y tampoco hay ninguna petición externa que se ejecute, el diagrama de despliegue se reduce a la ejecución que se realiza en el entorno del cliente.

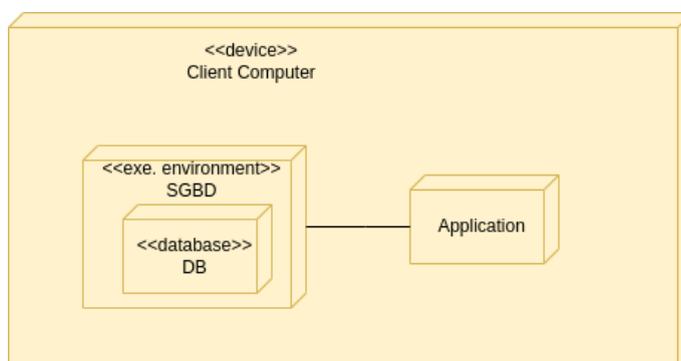


Figura 7: Diagrama de despliegue

5.2. Framework Qt

Qt es un framework de desarrollo de aplicaciones multiplataforma de código abierto.

Qt usa un sistema de señales y ranuras, que permite la comunicación entre objetos de manera eficiente y desacoplada. Las señales son los eventos generados por los objetos Qt cuando ocurre sobre ellos determinados eventos (click, hover, etc.), mientras que las ranuras son métodos que se conectan a estas señales para manejarlas. Al usar este sistema, Qt aplica el patrón de diseño Model View en vez de Model View Controller porque las señales y ranuras sustituyen el papel del controlador.

Qt proporciona una amplia gama de widgets para su uso en la interfaz de usuario. Estos widgets facilitan la creación de elementos de interfaz de usuario. Para facilitar la creación de widgets Qt ofrece Qt Designer, una herramienta gráfica para el diseño de interfaz de usuario.

A continuación se enumeran los patrones de diseño que usa Qt y han sido utilizados a lo largo de la aplicación de forma directa:

- Patrón Observador: para notificar y actualizar correctamente la interfaz de usuario cuando ha ocurrido algún evento en ella. [11]

- Patrón Comando: encapsula las solicitudes de cambios en objetos parametrizables. [11]
- Patrón Singleton: para gestionar ciertos componentes como QApplication.

Se ha utilizado este framework en el proyecto por su facilidad para el desarrollo de aplicaciones multiplataforma y sus librerías gráficas.

5.3. Patrones de diseño

Los patrones de diseño son soluciones probadas para problemas comunes que surgen durante el desarrollo de una aplicación informática.

A continuación se van a describir los patrones utilizados a lo largo del desarrollo de la aplicación.

5.3.1. Patrón Model View

En Qt se promueve el uso del patrón MV (Model View) sobre el patrón tradicional Modelo Vista Controlador (MVC) porque se adapta mejor a PyQt, ya que usa señales y ranuras que sustituyen al controlador del MVC. [19]

Al igual que el patrón MVC, al separar la lógica de datos y su representación visual, permite un mayor grado de desacoplamiento, facilitando el desarrollo y los cambios a lo largo del proyecto. También promueve la reutilización de componentes al estar abstraídos de la lógica de datos.

En el proyecto se va a implementar este patrón pero con ligeros cambios, sin integrar directamente los modelos con los componentes, sino permitiendo a la vista una mayor flexibilidad reaccionando a los eventos producidos en la interfaz de usuario.

Se ha seguido este patrón porque PyQt está pensado para ello, por lo que para facilitar el desarrollo se ha implementado este patrón con ciertas modificaciones.

5.3.2. Patrón Singleton

El patrón Singleton define una solución que garantiza una única instancia de una clase y un acceso global a ella. Se usa sobre todo para clases que necesiten un control estricto de acceso y se quiera acceso global, por ejemplo para el acceso a base de datos, la gestión de las conexiones. [16]

Para aplicar el patrón Singleton se ha definido un método estático de clase que comprueba si existe la instancia y la devuelve, o crea una nueva si no existiera y la almacena para futuras llamadas.

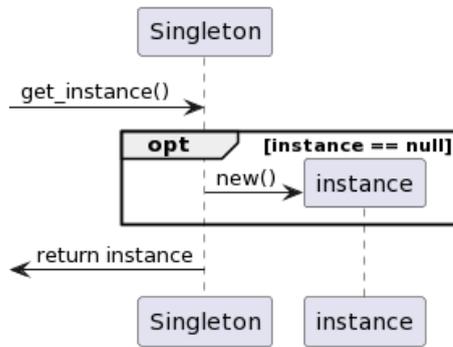


Figura 8: Patrón Singleton

El patrón se ha utilizado en el proyecto para el acceso a base de datos, permitiendo tener una única instancia global con la conexión a base de datos bien definida (abriendo y cerrando correctamente la conexión).

5.3.3. Patrón Data Mapper

El patrón Data Mapper es un patrón de diseño cuyo objetivo principal es separar la capa de persistencia de datos de la capa de dominio, permitiendo una mayor abstracción y desacoplamiento entre ambas capas. [17]

El patrón se ha aplicado mediante clases DataMapper para cada clase de la capa de dominio, estas clases DataMapper reciben el resultado de la query realizada mediante el Singleton y crean objetos de dominio a partir de los datos recibidos. Mediante esta separación se puede añadir o modificar datos de la capa de persistencia de forma sencilla, por ejemplo para añadir el número máximo de ejercicios seleccionables en la configuración de un tipo de ejercicio.

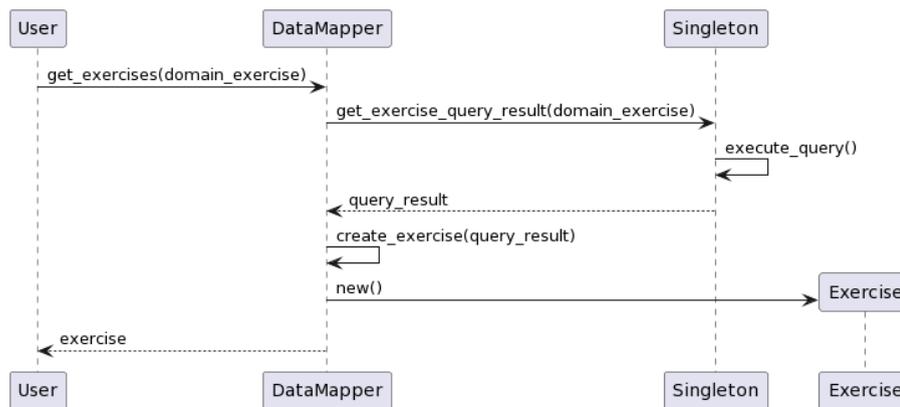


Figura 9: Patrón Data Mapper

Se ha utilizado este patrón de diseño en la aplicación por los modelos Paso y Ayuda, son objetos que siempre son rellenados de la misma manera dependiendo del tipo de ejercicio, por ello en vez de almacenarlo en base de datos se ha optado por crearlo en ejecución en los DataMappers dependiendo del tipo de ejercicio. [18]

5.3.4. Patrón Factoría

El patrón Factoría se encarga de proporcionar clases fábrica dedicadas a la creación y configuración de objetos. Proveen una interfaz común dentro de la aplicación para la creación de determinados objetos, facilitando la gestión de estos objetos.

Se ha creado clases factoría sobre los widgets de PyQt5 para facilitar su gestión, en las clases de componentes y vistas se han creado los widgets mediante estas factorías.

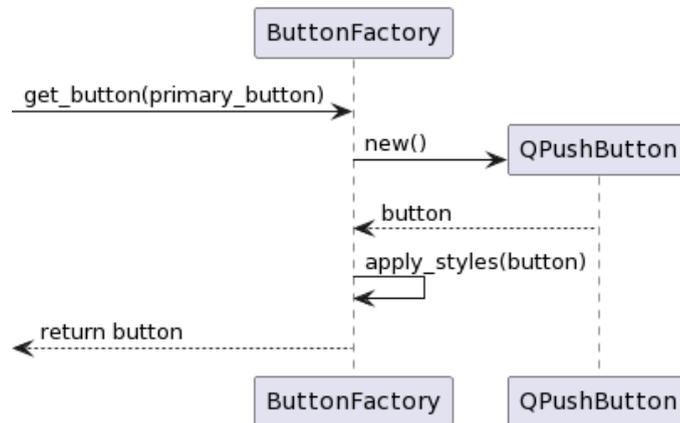


Figura 10: Patrón Factoría

El patrón Factoría se ha utilizado en la aplicación para la creación de componentes, proporcionando una interfaz común y favoreciendo la cohesión de la aplicación en términos de interfaz de usuario. Además ha permitido la creación de forma eficiente de los componentes, evitando todo el código *boiler plate* asociado a la creación de widgets, por ejemplo ha evitado la creación constante de objetos *QPushButton* de PyQt5.

5.4. Diseño de la base de datos

Para el diseño de la base de datos se va a optar por una base de datos relacional para tener una base de datos estructurada y con integridad referencial.

Como se ha explicado en el apartado *Patrones de diseño aplicados* 5.3.3, se ha abstraído la capa de Modelo de Dominio de la capa de Persistencia de Datos, siguiendo el patrón *Data Mapper*. Al utilizar este patrón no ha sido necesario guardar en base de datos los siguientes datos:

- información relativa a los pasos: los pasos de los ejercicios son datos constantes, siempre tienen los mismos atributos (orden, pregunta, ayuda relacionada) para un mismo tipo de ejercicio, por lo que se ha decidido en no persistirlo en base de datos.
- información relativa a la ayuda de cada ejercicio: al igual que los pasos, la ayuda de cada ejercicio es siempre la misma (mismas funciones, mismos puntos, mismos intervalos de ayuda, etc.) por lo que se ha preferido no persistirlo en base de datos.
- vídeo relativo a la ayuda de cada ejercicio: al igual que la ayuda, el vídeo es el mismo para cada tipo de ejercicio, por lo que se ha preferido no persistirlo.

- el máximo número seleccionable para un tipo de ejercicio: no ha hecho falta persistirlo, al hacer la query a base de datos se ha calculado el máximo de ejercicios existentes de ese tipo en base de datos.

A partir del modelo de dominio 4.6 y las consideraciones anteriores se ha obtenido las siguientes tablas:

Tabla: Temas					
Atributo	Tipo	Valor	Clave	Restricciones	Descripción
id	Integer	not null	PK	-	Identificador a nivel interno
title	String	not null	-	-	Título del tema
description	String	-	-	-	Descripción del tema
Comentario	Tabla que almacena los posibles temas dentro de la aplicación				

Tabla 33: TB-001 Temas

Tabla: Ejercicios					
Atributo	Tipo	Valor	Clave	Restricciones	Descripción
id	Integer	not null	PK	-	Identificador a nivel interno
exercise_type	String	not null	-	-	Tipo de ejercicio
plot_range	String	-	-	-	Rango de valores a mostrar de las abscisas y ordenadas en la gráfica del ejercicio. Por defecto '5, 5'.
topic_id	Integer	not null	FK	on cascade	Clave foránea que indica a que tema pertenece el ejercicio.
Comentario	Tabla que los ejercicios existentes dentro de la aplicación				

Tabla 34: TB-002 Ejercicios

Tabla: Funciones					
Atributo	Tipo	Valor	Clave	Restricciones	Descripción
id	Integer	not null	PK	-	Identificador a nivel interno
expression	String	not null	-	-	Expresión de la función
Comentario	Tabla que almacena las funciones existentes dentro de la aplicación				

Tabla 35: TB-003 Funciones

Tabla: Funciones_Ejercicios					
Atributo	Tipo	Valor	Clave	Restricciones	Descripción
id	Integer	not null	PK	-	Identificador a nivel interno
exercise_id	Integer	not null	FK	on cascade	Clave foránea que indica el ejercicio al que pertenece esta entrada
graph_id	Integer	not null	FK	on cascade	Clave foránea que indica la función a la que pertenece esta entrada
domain	String	-	-	-	Indica el dominio de la función, por defecto '(-inf, +inf)
is_main_graphic	Integer	-	-	-	Indica si es una función principal dentro del ejercicio
Comentario	Tabla intermedia que almacena la relación entre ejercicios y funciones				

Tabla 36: TB-004 Funciones_Ejercicios

Tabla: Puntos_Ejercicios					
Atributo	Tipo	Valor	Clave	Restricciones	Descripción
id	Integer	not null	PK	-	Identificador a nivel interno
exercise_id	Integer	not null	FK	on cascade	Clave foránea que indica el ejercicio al que pertenece esta entrada
x_value	Integer	not null	-	-	Valor de la x en el punto
y_value	Integer	not null	-	-	Valor de la y en el punto
is_included	Integer	not null	-	-	Indica si el punto está incluido en el ejercicio
Comentario	Tabla intermedia que almacena la relación entre puntos y ejercicios				

Tabla 37: TB-005 Puntos_Ejercicios

Tabla: Resumen_Ejercicios					
Atributo	Tipo	Valor	Clave	Restricciones	Descripción
id	Integer	not null	PK	-	Identificador a nivel interno
is_correct	Integer	not null	-	-	Indica si el usuario ha respondido correctamente
step_type	String	not null	-	-	Indica el ejercicio del ejercicio al que pertenece el resumen
response	String	not null	-	-	Almacena la respuesta enviada por el usuario
exercise_id	Integer	not null	FK	on cascade	Clave foránea al ejercicio del resumen.
graph_id	Integer	not null	FK	on cascade	Clave foránea a la función principal correcta.
Comentario	Tabla intermedia que almacena las respuestas del usuario				

Tabla 38: TB-006 Resumen_Ejercicios

Tabla: Configuracion Ejercicios					
Atributo	Tipo	Valor	Clave	Restricciones	Descripción
id	Integer	not null	PK	-	Identificador a nivel interno
exercise_type	String	not null	-	-	Tipo de ejercicio configurado
description	String	-	-	-	Breve descripción de ayuda para el usuario
exercise_num	Integer	-	-	-	Número de ejercicios seleccionado
is_active	Integer	-	-	-	Indica si el tipo de ejercicio está activo
topic_id	Integer	not null	FK	on cascade	Clave foránea que indica el tema
Comentario	Tabla que almacena la configuración de un tipo de ejercicio para un tema concreto				

Tabla 39: TB-007 Configuracion Ejercicios

Tabla: Configuracion Pasos					
Atributo	Tipo	Valor	Clave	Restricciones	Descripción
id	Integer	not null	PK	-	Identificador a nivel interno
step_type	String	not null	-	-	Tipo de ejercicio configurado
description	String	-	-	-	Breve descripción de ayuda para el usuario
is_active	Integer	-	-	-	Indica si el tipo de ejercicio está activo
exercise_setting_id	Integer	not null	FK	on cascade	Clave foránea que indica la configuración de ejercicio
Comentario	Tabla que almacena la configuración de los pasos dentro de un tipo de ejercicio				

Tabla 40: TB-008 Configuracion Pasos

Tabla: Configuracion Pasos					
Atributo	Tipo	Valor	Clave	Restricciones	Descripción
id	Integer	not null	PK	-	Identificador a nivel interno
step_type	String	not null	-	-	Tipo de ejercicio configurado
description	String	-	-	-	Breve descripción de ayuda para el usuario
is_active	Integer	-	-	-	Indica si el tipo de ejercicio está activo
exercise_setting_id	Integer	not null	FK	on cascade	Clave foránea que indica la configuración de ejercicio
Comentario	Tabla que almacena la configuración de los pasos dentro de un tipo de ejercicio				

Tabla 41: TB-009 Visualización de estadísticas

5.5. Diseño de la interfaz de usuario

A continuación se va a detallar el diseño de la interfaz de usuario.

5.5.1. Usuarios objetivo

El diseño está centrado en los usuarios que van a usar la aplicación. La aplicación está pensada para alumnos de Ingeniería Informática.

Usuarios: Alumnos	
Características	<ul style="list-style-type: none">■ facilidad de aprendizaje: el alumno comprende fácilmente como utilizar la interfaz de usuario.■ curiosidad: el alumno tiene interés por comprobar el funcionamiento al completo de la aplicación.■ familiaridad con patrones de diseño de interfaces.
Características	Las características mencionadas muestran un tipo de persona que no necesita una interfaz de usuario para novatos, sino una persona muy capaz de aprender rápidamente el funcionamiento. También habrá que tener cuidado con la aplicación para evitar que los alumnos la puedan romper.

Tabla 42: Usuarios U-01 Alumnos

5.5.2. Objetivos de la interfaz de usuario

A continuación se van a explicar las reglas que se han seguido para el diseño de la interfaz. [23] [24]

Consistencia. Se buscará que la aplicación tenga consistencia interna, haciendo que para el usuario la aplicación sea fácil de usar al ser similar a lo largo de todas sus vistas. Para ello se van a realizar componentes de terceros y creados en específico para el desarrollo de esta aplicación.

Retroalimentación constante al usuario. Todas las acciones del usuario tienen que tener una retroalimentación asociada que sirva de guía al usuario, para permitirle saber que efectos están teniendo sus acciones. Ejemplos de estas retroalimentaciones serán el mensaje de guardado de configuración de tipo de ejercicio y las validaciones de respuesta de usuario a los pasos de los ejercicios.

Facilitar la acción de revertir. El usuario tiene que poder deshacer sus acciones en determinados casos para evitar cierta frustración de tener que hacer las cosas a la primera. Por ejemplo en los ejercicios de máximo y mínimos o los ejercicios de dominio y recorrido, durante estos ejercicios, para facilitar su resolución, el usuario puede deshacer.

Prevenir mejor que dar errores, manejar los errores de manera simple. Comprobar los campos que se puedan antes de que el usuario se introduzca errores y haya que recuperar el estado. Por ejemplo para introducir el dominio no se le permite añadir caracteres erróneos, evitando así fallos en el sistema.

Facilidad de aprendizaje promoviendo que el usuario reconozca la aplicación más que la tenga que recordar. Hacer una aplicación usable para un alumno.

Diseño centrado en el usuario. Todo tiene que estar pensado según un alumno de la universidad y sus necesidades.

Para realizar el diseño de la interfaz de usuario de una forma barata (tiempo y dinero) se han usados bocetos realizados por el autor del proyecto a mano.

5.6. Seguridad

Al ser una aplicación de educativa no se guarda ningún tipo de datos sobre usuarios, solamente se almacena todo lo relacionado con los ejercicios.

Se ha cuidado la aplicación para que el usuario no pueda dejar en un estado inconsistente a la BD. Para ello se ha protegido el sistema de inyecciones SQL, código malicioso para ejecutar en la base de datos. Todas las interacciones del usuario con la base de datos se verifican y se restringe su ejecución libre para garantizar la integridad y seguridad del sistema.

6. Implementación

6.1. Control de versiones

A continuación se va a explicar la estrategia de ramas que se ha seguido durante el desarrollo de la aplicación.

Mediante Git la creación de ramas es trivial, facilitando beneficiarse del uso de ramas: permite desarrollo paralelo entre diversos desarrolladores, mantener el código principal estable en una rama y poder hacer pruebas en otras ramas, tener un registro de las diferentes versiones del proyecto, evitar la inclusión de código conflictivo dejándolo en una rama aparte.

Al ser solo un desarrollador en la aplicación se ha simplificado la estrategia para hacer una estrategia eficiente y acorde a la situación real del proyecto:

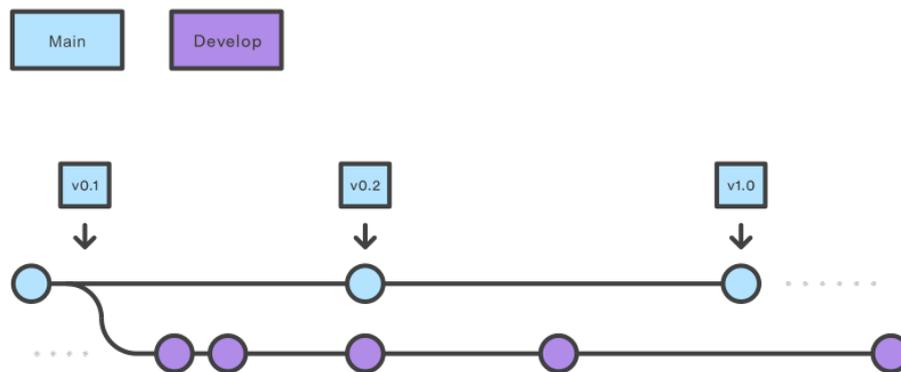


Figura 11: Figura obtenida de <https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow>

Existen dos ramas de trabajo: “dev“, donde se hacen los commits de las tareas de cada iteración, y “main“, la rama principal a partir de la cual se sacan las versiones que usarán los usuarios.

El flujo es el siguiente:

1. en la rama “dev“ se han ido realizando los commits de las distintas tareas realizadas.
2. cuando se alcanza determinada funcionalidad se realiza un merge a “master“ de la rama de “dev“ para tener en “master“ toda la funcionalidad.
3. se saca un tag a partir de la rama “master“.

Durante el desarrollo de este proyecto solo se ha considerado realizar un tag al final del desarrollo al ser el momento de la primera versión que cumplía con los objetivos planteados.

6.2. Implementación de la base de datos

En este apartado, se presentará una introducción a la implementación de la base de datos utilizando el sistema de gestión de bases de datos relacional (SGBD) MySQL.

Se ha elegido un SGBD relacional por los siguientes aspectos:

- el modelo de datos es estructurado, encajando en un modelo de columnas y filas para la organización de los datos de la aplicación.
- proveer integridad referencial en los datos de la aplicación, permitiendo así una consistencia y coherencia a lo largo de la aplicación.
- flexibilidad y escalabilidad, al estar la aplicación dividida en módulos está pensada para escalar, con este tipo de SGBD se sigue la misma línea.
- consultas de alto nivel, un SGBD permite hacer consultas complejas entre los datos de la aplicación.

La elección MySQL viene determinada por lo siguiente:

- su compatibilidad con distintos sistemas operativos, es multiplataforma. Para poder alcanzar el objetivo de desarrollar la aplicación para windows 10.
- la documentación existente, todas las dudas relacionadas van a poder ser resueltas al tener una documentación muy extensa y una gran comunidad.

Para que el alumno pueda utilizar la aplicación solamente ejecutándola se ha generado un pequeño código introducido en la clase “*database_creator.py*” para crear y poblar la base de datos. El script comprueba si la base de datos ya está creada y si no lo está se ejecuta para crearla y poblarla. Con esta funcionalidad se permite que la aplicación sea usable de verdad por los alumnos, sin tener estos que ejecutar por ellos mismos un script que pueble la base de datos.

6.3. Gestión de dependencias

Un proceso clave para poder migrar la aplicación a windows 10 es manejar correctamente las dependencias del proyecto.

La gestión de dependencias es el proceso de administrar y controlar las bibliotecas utilizadas en un proyecto. En una buena gestión de dependencias las bibliotecas utilizadas están fijadas a versiones concretas, para reducir los posibles conflictos.

Los conflictos de dependencias son los problemas generados cuando se requiere una librería en diferentes versiones, pueden ser generados por:

- se requiere diferentes versiones para una misma librería. Este requerimiento puede ser un conflicto entre la versión requerida por el proyecto y la versión requerida de forma global.
- se requiere de forma indirecta diferentes versiones de la misma librería. Librerías requeridas directamente importan versiones diferentes de la misma librería.

Para Python el gestor de paquetes estándar es Pip. Permite instalar y desinstalar paquetes. Una buena práctica es mantener los requisitos de la aplicación en un archivo *requirements.txt* con el número de versión concreto para cada librería. Si no se pone un número de versión la librería podría ejecutarse en una versión distinta de la esperada, versión que podría ser incompatible con la aplicación.

Para abstraer toda esta lógica surgen los entornos virtuales (virtualenv en Python). Herramienta que permiten crear entornos virtuales aislados para proyectos de Python, evitando conflictos

distintos entre proyectos.

Para evitar conflictos indirectos en el archivo de *requirements.txt* se ha especificado cada versión necesitada de la siguiente forma:

```
PyQt5==5.15.4
```

Para crear un entorno virtual, activarlo e instalar las dependencias se han seguido los siguientes pasos:

```
python3 -m venv virtualenv # Creación del entorno
source virtualenv/bin/activate # Activación del entorno
pip install -r requirements.txt # Instalación de bibliotecas
```

6.4. Tipado de variables en Python

Una característica muy utilizada en el proyecto es el tipado de variables introducido en Python en la versión 3.5. Es una característica opcional en el Python que utilizan herramientas externas como los IDEs, linters y herramientas de análisis de código estático.

Este tipado variable se ha utilizado por toda la aplicación en los parámetros de los métodos y en los atributos de cada objeto.

```
class Point:
def __init__(self, x: float, y: float, is_included: bool = True):
self.x: float = x
self.y: float = y
self.is_included: bool = is_included
```

Ha sido especialmente útil para el desarrollo del proyecto por las siguientes razones que han permitido un desarrollo mucho mejor y más eficiente:

- detección de errores en el paso de argumentos: permite saber si el tipo de cada variable pasada al método es el esperado, permitiendo que el comportamiento del método sea el esperable.
- mantenibilidad: el tipado sirve de documentación permitiendo añadir código y corregir bugs de forma más fácil.
- ayuda del IDE: permitir al IDE saber el tipo de variable con la que está tratando facilita que te ayude mostrando la documentación de esa variable.

Para comprobar el estado del tipado dentro de la aplicación se puede usar la herramienta Mypy, herramienta de análisis estático para tipos en Python. En el análisis de datos la mayoría de errores son por asignaciones iniciales a valor nulo.

Algunos ejemplos de la salida de ejecutar la herramienta Mypy:

- *.../domain_definition_component.py:159: error: Argument "alignment" to "addWidget" of "QBoxLayout" has incompatible type "int"; expected "Union[Alignment, AlignmentFlag]" [arg-type]*
- *.../domain_definition_component.py:237: error: Incompatible types in assignment (expression has type "float", variable has type "int") [assignment]*
- *.../function_exercise_page.py:34: error: Incompatible types in assignment (expression has type "None", variable has type "QComboBox") [assignment]*

6.5. Licencia

Este proyecto usa una licencia Creative Commons. En concreto de Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) que permite copiar y distribuir el material del proyecto [27] siempre que se atribuya el crédito suficiente a esta obra y no se use para motivos comerciales. Si la obra es cambiada no se permite su distribución.

6.6. Estructura del proyecto

A continuación se va a presentar una muestra de la estructura del proyecto, para permitir comprender los aspectos más relevantes de la organización del proyecto:

```

- function/
  - components/
    - domain_components/
      - domain_definition_component.py
      - range_definition_component.py
      - ...
    - elementary_graph_components/
  - ...
  - component.py
  - function_exercise_component.py
  - ...
- data_mappers/
  - function_exercise_data_mapper.py
  - step_data_mapper.py
  - ...
- factories/
  - plot_factory.py
- models/
  - function_exercise.py
  - function.py
  - ...
- tests/
  - models

```

```

        - test_function.py
        - ...
    - views/
        - function_exercise_page.py
- media/
    - images/
        - save.png
        - ...
    - videos/
        - inverse_concept_info_data.mp4
        - ...
- project_conf/
    - app/
        - controller.py
        - py_math_app.py
    - components/
        - window.py
        - dialog.py
    - data_mappers/
        - topic_data_mapper.py
    - database/
        - database_access_singleton.py
        - database_creator.py
    - factories/
        - button_factory.py
        - label_factory.py
        - ...
    - models/
        - topic.py
        - exercise_setting.py
        - ...
    - views/
        - welcome_page.py
        - topics_page.py

- main.py

```

La aplicación está pensada para estar dividida en módulos. Ahora mismo solo existe un módulo para funciones “*functions*“, dónde está almacenada toda la lógica relativa a los ejercicios para funciones.

En cada módulo se sigue la siguiente estructura:

- la carpeta “*components*” almacena los componentes que se van a usar en ese módulo.
- la carpeta “*data_mappers*” almacena las clases que transforman los resultados SQL a clases de Python.
- la carpeta “*factories*” almacena los widgets de PyQt que se van a utilizar en ese módulo.
- la carpeta “*models*” almacena las clases de Python que representan las clases del modelo de dominio.
- la carpeta “*tests*” almacena los tests de ese modulo, esta carpeta sigue la misma estructura de carpetas que el módulo en el que se encuentran.
- la carpeta “*views*” almacena las vistas de ese módulo. Accesibles para el usuario a través de las rutas de la aplicación.

Además del módulo de funciones, existe una carpeta principal “*project_conf*” dónde se almacena funcionalidad general relativa a la aplicación, preparada para ser usada por todos los módulos.

En la carpeta principal se almacenan clases relativas al funcionamiento de la aplicación, cada módulo puede hacer uso de ellas junto a las existentes en él.

- en la carpeta “*app*” se encuentra la inicialización de la aplicación “*py_math_app.py* y la clase controladora para las vistas y las rutas de la aplicación *controller.py*.”
- en la carpeta “*database*” se encuentra la clase que inicializa y puebla todas las tablas de la base de datos “*database_creator.py*” y la clase que accede a base de datos “*database_access_singleton.py*”.
- en la carpeta “*models*” se encuentran los modelos del modelo de dominio comunes a todos los módulos.
- en la carpeta “*data_mappers*” se encuentran las clases que siguen este patrón comunes a todos los módulos.
- en la carpeta “*factorie*” se encuentran los widgets comunes a todos los módulos, por ejemplo el “*button_factory.py*”.
- en la carpeta “*views.py*” se encuentran las vistas iniciales accesibles para el alumno, la vista inicial y la de elección de tema.

Por último, ahora mismo hay una carpeta “*media*” donde se almacena todo el multimedia de la aplicación.

6.7. Pruebas

Las pruebas de software son un proceso por el cual evaluamos la funcionalidad del código, definiendo el comportamiento esperado y detectando errores para casos específicos. Hacer pruebas exhaustivas permite comprobar que no hay bugs en el código, y que al introducir nueva funcionalidad se detecta como afecta en otras funcionalidades (fallos o cambios). [15]

Las pruebas se pueden clasificar de muchas formas distintas:

- Según la forma de ejecutar el código:

- Manuales. Requieren de interacción externa
 - Automáticas. No requieren de interacción externa, a través de una herramienta externa se pueden lanzar.
- Según los requisitos que verifican:
 - Funcionales. Comprueban la funcionalidad del software.
 - No funcionales. Comprueban las características relacionadas con requisitos no funcionales (rendimiento, seguridad).
 - Según la funcionalidad que abarquen:
 - Unitarias. Comprueban pequeñas partes del código de forma independiente, se debería hacer mocks de toda funcionalidad externa que no forme parte de lo que se está probando.
 - Integración. Comprueban varias partes del código a la vez para ver como funcionan en conjunto.
 - E2E. Comprueban todo el flujo completo de la aplicación.

Como se puede ver en la siguiente figura, los tests unitarios son más rápidos y baratos que los tests E2E pero abarcan mucho menos, por eso suele haber muchos tests unitarios para poder cubrir con garantías la funcionalidad del proyecto. En el proyecto se han hecho tests unitarios automáticos mediante unittest y test manuales E2E, detallados en los apartados siguientes.



Figura 12: Tipos de pruebas

6.7.1. Pruebas unitarias

Para realizar las pruebas unitarias se ha utilizado el framework de testing de Python *Unittest*.

En la aplicación no se han realizado pruebas unitarias para componentes y vistas de la aplicación por los siguientes motivos:

- considerarse excesivamente complejas y costosas. Poder probar y tener en cuenta todos los casos posibles iba a ser muy difícil.
- muy susceptibles a cambios, el trabajo realizado para testear la aplicación podría ser poco útil si hay que cambiar los tests en cada iteración.
- se ha considerado que los tests finales E2E servirán para probar la interfaz de usuario correctamente

Teniendo lo anterior en cuenta se ha testeado solamente los modelos de la capa de dominio por lo siguiente:

- detectar errores en la lógica de la aplicación.
- tener lógica esencial y compleja para la aplicación.
- reconocer cambios durante las iteraciones que puedan afectar (queriendo o no) y poder actuar en consecuencia.
- evitar tener que probar en cada iteración lógica que no es susceptible a cambios.

Los tests desarrollados van a ser *tests de caja blanca* para poder hacer un testing profundo y exhaustivo únicamente del código de los modelos. Permitiendo hacer mock de módulos independientes que no tengan interés en el test.

Modelo: ejercicios	
Comprobar el valor del dominio introducido por el usuario	<ul style="list-style-type: none"> ■ validar el dominio cuando la expresión es idéntica a la esperada. ■ validar el dominio cuando la expresión no es idéntica a la esperada pero la expresión es válida. ■ no validar el dominio cuando ni la expresión ni los valores son los esperados.
Comprobar el valor del dominio para un ejercicio	<ul style="list-style-type: none"> ■ obtener el valor correcto de la expresión esperada para el dominio en un ejercicio determinado.
Comprobar los valores asociados a una expresión de dominio	<ul style="list-style-type: none"> ■ obtener los números asociados a una expresión de dominio.
Comprobar los máximos y mínimos de un ejercicio	<ul style="list-style-type: none"> ■ obtener todos los máximos y mínimos para un ejercicio con una única función. ■ obtener todos los máximos y mínimos para un ejercicio con múltiples funciones sin solapar. ■ obtener todos los máximos y mínimos para un ejercicio con múltiples funciones con solapamiento. ■ obtener todos los máximos y mínimos para un ejercicio que incluya puntos. ■ obtener todos los máximos y mínimos para un ejercicio que incluya funciones constantes.

Tabla 43: Pruebas Unitarias-01 Modelo: ejercicios

Modelo: funciones	
Comprobar el dominio	<ul style="list-style-type: none"> ■ obtener dominio de una función sin asíntotas. ■ obtener dominio de una función con asíntotas.
Comprobar el recorrido	<ul style="list-style-type: none"> ■ obtener recorrido de una función sin asíntotas. ■ obtener recorrido de una función con asíntotas.
Comprobar las asíntotas	<ul style="list-style-type: none"> ■ obtener asíntotas horizontales. ■ obtener asíntotas horizontales cuando la función no tiene (comprobar su comportamiento).
Comprobar máximos y mínimos	<ul style="list-style-type: none"> ■ obtener puntos máximos y mínimos.

Tabla 44: Pruebas Unitarias-02 Modelo: funciones

En futuras iteraciones se podría aumentar la cobertura de los tests a otras partes de la aplicación, no solamente a los modelos, pero por falta de tiempo durante las iteraciones solo se ha probado lo más importante de la aplicación, la lógica de negocio contenida dentro de los modelos de la capa de dominio.

6.7.2. Prueba E2E

Para comprobar el flujo completamente se ha realizado las siguientes pruebas E2E:

Acceso a la vista de temas	
Entrada	El usuario inicia la aplicación y hace click en entrar.
Resultado esperado	<ul style="list-style-type: none"> ■ al usuario se le muestran todos los temas disponibles en la aplicación. ■ se muestra junto a cada tema un botón de configuración. ■ se muestra el botón de estadísticas para el usuario.
Resultado	Ok.
Comentarios	-

Tabla 45: Pruebas E2E-01 Vista de temas

Acceso a las configuraciones de ejercicio	
Entrada	El usuario hace click sobre el botón de configuración de un ejercicio.
Resultado esperado	<ul style="list-style-type: none"> ■ Al usuario se le muestra un modal de configuración donde aparece el número de ejercicios a realizar y el estado de cada ejercicio (activo o inactivo). ■ el resto de botones de configuración quedan desactivados hasta que el usuario cierre esta configuración. ■ el usuario no puede acceder a los ejercicios ni a sus estadísticas (desactivados) hasta que cierre esta configuración.
Resultado	Ok.
Comentarios	Prueba realizada para todas las configuraciones.

Tabla 46: Pruebas E2E-02 Acceso a las configuraciones de ejercicio

Guardar configuración de ejercicio	
Entrada	El usuario cambia la configuración del ejercicio de forma válida.
Resultado esperado	<ul style="list-style-type: none"> ■ al usuario se le muestra un mensaje informativo. ■ la configuración se ha guardado correctamente en base de datos. ■ la configuración aplicada a ese tipo de ejercicio cuando el usuario acceda a él es la aplicada por última vez.
Resultado	Ok.
Comentarios	Prueba realizada para todas las configuraciones usando las distintas opciones de configuración mostradas al usuario.

Tabla 47: Pruebas E2E-03 Guardar configuración de ejercicio

Gestión de estados en la configuración de ejercicios	
Entrada	El usuario entra y cambia la configuración del ejercicio.
Resultado esperado	<ul style="list-style-type: none"> ■ los botones de estado de ejercicios se muestran correctamente. Si solo hay un tipo de ejercicio activado, al usuario se le va a mostrar el tipo de ejercicio desactivado para que siempre haya al menos un tipo de ejercicio. Si hay más de un tipo se muestran todos activados. ■ para el conjunto de pasos de un tipo de ejercicio se comprueba lo anterior, mostrando el botón desactivado si es el único de ese tipo de ejercicio activado.
Resultado	Ok.
Comentarios	Prueba realizada para todas las configuraciones para cuando se muestra inicialmente el modal y para cuando el usuario realiza cambios.

Tabla 48: Pruebas E2E-04 Gestión de estados en la configuración de ejercicios

Acceso a estadísticas de usuario	
Entrada	El usuario solicita visualizar sus estadísticas.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra sus estadísticas en una ventana aparte. Los datos mostrados son correctos. ■ se desactivan el resto de botones de la interfaz de la pantalla principal, los botones de selección de temas y los botones de configuración de cada tema.
Resultado	Ok.
Comentarios	-.

Tabla 49: Pruebas E2E-05 Acceso a estadísticas del usuario

Visualización del ejercicio indicar función elemental	
Entrada	El usuario accede al ejercicio función elemental.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestran los botones de navegación por los ejercicios. ■ se muestra el botón de información del ejercicio mediante vídeo. ■ se muestran cuatro botones con posibles soluciones de funciones elementales, entre las que está la respuesta.
Resultado	Ok.
Comentarios	-.

Tabla 50: Pruebas E2E-06 Visualización del ejercicio indicar la función elemental

Realización correcta del ejercicio indicar función elemental	
Entrada	El usuario accede al ejercicio función elemental y lo resuelve correctamente.
Resultado esperado	<ul style="list-style-type: none"> ■ se pinta la gráfica en verde. ■ se deshabilitan los botones de respuesta y el usuario ya no puede interactuar con ellos.
Resultado	Ok.
Comentarios	-.

Tabla 51: Pruebas E2E-07 Realización correcta del ejercicio indicar la función elemental

Realización incorrecta del ejercicio indicar función elemental	
Entrada	El usuario accede al ejercicio función elemental y lo resuelve incorrectamente.
Resultado esperado	<ul style="list-style-type: none"> ■ se pinta en rojo la gráfica seleccionada por el usuario y en verde la correcta. ■ se deshabilitan los botones de respuesta y el usuario ya no puede interactuar con ellos.
Resultado	Ok.
Comentarios	-.

Tabla 52: Pruebas E2E-08 Realización incorrecta del ejercicio indicar la función elemental

Visualización del ejercicio indicar desplazamiento	
Entrada	El usuario accede al ejercicio indicar desplazamiento.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestran los botones de navegación por los ejercicios. ■ se muestra el botón de información del ejercicio mediante vídeo. ■ se muestran cuatro botones con posibles soluciones de desplazamientos, entre las que está la respuesta.
Resultado	Ok.
Comentarios	-.

Tabla 53: Pruebas E2E-09 Visualización del ejercicio indicar desplazamiento

Realización correcta del ejercicio indicar desplazamiento	
Entrada	El usuario accede al ejercicio indicar desplazamiento y lo resuelve correctamente.
Resultado esperado	<ul style="list-style-type: none"> ■ se pinta la gráfica en verde. ■ se deshabilitan los botones de respuesta y el usuario ya no puede interactuar con ellos.
Resultado	Ok.
Comentarios	-.

Tabla 54: Pruebas E2E-10 Realización correcta del ejercicio indicar desplazamiento

Realización incorrecta del ejercicio indicar desplazamiento	
Entrada	El usuario accede al ejercicio indicar desplazamiento y lo resuelve incorrectamente.
Resultado esperado	<ul style="list-style-type: none"> ■ se pinta en rojo la gráfica seleccionada por el usuario y en verde la correcta. ■ se deshabilitan los botones de respuesta y el usuario ya no puede interactuar con ellos.
Resultado	Ok.
Comentarios	-.

Tabla 55: Pruebas E2E-11 Realización incorrecta del ejercicio indicar desplazamiento

Visualización del ejercicio indicar si existe inversa	
Entrada	El usuario accede al ejercicio indicar si existe inversa.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestran dos botones con las dos posibles soluciones Sí o No. ■ el usuario ya no puede interactuar con los botones.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 56: Pruebas E2E-12 Visualización del ejercicio indicar si existe inversa

Realización correcta del ejercicio indicar si existe inversa	
Entrada	El usuario accede al ejercicio indicar si existe inversa y lo resuelve.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra un mensaje de acierto. ■ el usuario ya no puede interactuar con los botones.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 57: Pruebas E2E-13 Realización correcta del ejercicio indicar si existe inversa

Realización incorrecta del ejercicio indicar si existe inversa	
Entrada	El usuario accede al ejercicio indicar si existe inversa y lo resuelve.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra un mensaje de error. ■ se muestra dos puntos iguales que impidan tener inversa, estos dos puntos están unidos por una línea horizontal que indica que están al mismo nivel, tienen el mismo valor para su imagen. ■ el usuario ya no puede interactuar con los botones.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 58: Pruebas E2E-14 Realización incorrecta del ejercicio indicar si existe inversa

Visualización del ejercicio seleccionar la función inversa	
Entrada	El usuario accede al ejercicio seleccionar la función inversa.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestran tres posibles funciones inversas junto con la función principal en blanco. Si la inversa coincide con la función se muestran solo dos posibles funciones y la función se muestra en blanco.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 59: Pruebas E2E-15 Visualización del ejercicio seleccionar la función inversa

Realización correcta del ejercicio seleccionar la función inversa	
Entrada	El usuario accede al ejercicio seleccionar la función inversa y lo resuelve correctamente.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestran solo las funciones siguientes: la función seleccionada en verde y la función principal en blanco. ■ el usuario ya no puede interactuar con la gráfica.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 60: Pruebas E2E-16 Realización correcta del ejercicio seleccionar la función inversa

Realización incorrecta del ejercicio seleccionar la función inversa	
Entrada	El usuario accede al ejercicio seleccionar la función inversa y lo resuelve incorrectamente.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestran solo las funciones siguientes: la función seleccionada en rojo, la función esperada en verde y la función principal en blanco. ■ el usuario ya no puede interactuar con la gráfica.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 61: Pruebas E2E-17 Realización incorrecta del ejercicio seleccionar la función inversa

Visualización del ejercicio restringir dominio de la función	
Entrada	El usuario accede al ejercicio restringir dominio de la función.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra un intervalo manejable por el usuario. El intervalo mostrado no puede salirse de la función pero si se pueden cruzar los extremos entre sí.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 62: Pruebas E2E-18 Visualización del ejercicio restringir dominio de la función

Realización correcta del ejercicio restringir dominio de la función	
Entrada	El usuario accede al ejercicio restringir dominio de la función y lo resuelve correctamente.
Resultado esperado	<ul style="list-style-type: none"> ▪ se muestra la región seleccionada en verde y ya no es seleccionable. ▪ se deshabilita el botón de validar ejercicio.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 63: Pruebas E2E-19 Realización correcta del ejercicio restringir dominio de la función

Realización incorrecta del ejercicio restringir dominio de la función	
Entrada	El usuario accede al ejercicio restringir dominio de la función y lo resuelve incorrectamente.
Resultado esperado	<ul style="list-style-type: none"> ▪ se muestra en rojo dos puntos de la gráfica que tengan la misma imagen junto a una línea que los una indicando que están al mismo nivel. ▪ la región ya no es seleccionable. ▪ el botón de validar ya no es seleccionable.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 64: Pruebas E2E-20 Realización incorrecta del ejercicio restringir dominio de la función

Moverse entre ejercicios	
Entrada	El usuario utiliza las flechas de continuar y volver atrás y el desplegable para seleccionar un ejercicio en concreto.
Resultado esperado	<ul style="list-style-type: none"> ■ el botón de continuar solo se activa cuando está resuelto el ejercicio. ■ el botón de continuar es substituido por un botón que indica finalización de la sesión cuando se llega al final. ■ el botón de atrás solo se activa cuando hay ejercicio anterior. ■ el desplegable solo se activa cuando el usuario ha resuelto al menos un ejercicio. ■ el estado se maneja correctamente y siempre se muestra el ejercicio que el usuario ha pedido.
Resultado	Ok.
Comentarios	Se ha probado con todas las posibles combinaciones de pasos activados.

Tabla 65: Pruebas E2E-21 Moverse entre ejercicios

Acceder a la ayuda de cada ejercicio	
Entrada	El usuario utiliza el botón de ayuda.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra una serie de gráficas de ayuda junto con un texto. ■ se puede mover perfectamente entre las gráficas de ayuda mediante las flechas siguiendo la misma lógica que la navegación entre ejercicios.
Resultado	Ok.
Comentarios	Se ha probado con todos los tipos de ejercicio.

Tabla 66: Pruebas E2E-22 Acceder a la ayuda de cada ejercicio

Acceder al vídeo de ayuda de cada ejercicio	
Entrada	El usuario utiliza el botón de ayuda visual.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra un vídeo con textos explicativos. ■ se puede interactuar correctamente con el vídeo.
Resultado	Ok.
Comentarios	Se ha probado con todos los tipos de ejercicio.

Tabla 67: Pruebas E2E-23 Acceder al vídeo de ayuda de cada ejercicio

Visualización del ejercicio indicar dominio de la función	
Entrada	El usuario accede al ejercicio indicar dominio de la función.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra dos botones para añadir intervalos numéricos. ■ se muestra un caja de texto con los intervalos introducidos hasta el momento. ■ todos los intervalos introducidos se pueden editar haciendo click en ellos.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 68: Pruebas E2E-24 Visualización del ejercicio indicar el dominio

Realización correcta del ejercicio indicar dominio de la función	
Entrada	El usuario accede al ejercicio indicar dominio de la función y lo resuelve correctamente.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra un mensaje de validación correcta en verde. ■ el alumno ya no puede interactuar con la gráfica. ■ el alumno ya no puede interactuar con los botones de añadir y eliminar dominio. ■ el alumno ya no puede solicitar de nuevo la evaluación del ejercicio.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 69: Pruebas E2E-25 Realización correcta del ejercicio indicar el dominio

Realización correcta del ejercicio indicar dominio de la función	
Entrada	El usuario accede al ejercicio indicar dominio de la función y lo resuelve incorrectamente.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra un mensaje de validación incorrecta en rojo. ■ se muestra al alumno dos botones a partir de los cuales puede cambiar entre su respuesta y la respuesta correcta. ■ el alumno ya no puede interactuar con la gráfica. ■ el alumno ya no puede interactuar con los botones de añadir y eliminar dominio. ■ el alumno ya no puede solicitar de nuevo la evaluación del ejercicio.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 70: Pruebas E2E-26 Realización incorrecta del ejercicio indicar el dominio

Visualización del ejercicio indicar recorrido de la función	
Entrada	El usuario accede al ejercicio indicar recorrido de la función.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra dos botones para añadir intervalos numéricos. ■ se muestra un caja de texto con los intervalos introducidos hasta el momento. ■ todos los intervalos introducidos se pueden editar haciendo click en ellos.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 71: Pruebas E2E-27 Visualización del ejercicio indicar el recorrido

Realización correcta del ejercicio indicar recorrido de la función	
Entrada	El usuario accede al ejercicio indicar recorrido de la función y lo resuelve correctamente.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra un mensaje de validación correcta en verde. ■ el alumno ya no puede interactuar con la gráfica. ■ el alumno ya no puede interactuar con los botones de añadir y eliminar recorrido. ■ el alumno ya no puede solicitar de nuevo la evaluación del ejercicio.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 72: Pruebas E2E-28 Realización correcta del ejercicio indicar el recorrido

Realización correcta del ejercicio indicar recorrido de la función	
Entrada	El usuario accede al ejercicio indicar recorrido de la función y lo resuelve incorrectamente.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra un mensaje de validación incorrecta en rojo. ■ se muestra al alumno dos botones a partir de los cuales puede cambiar entre su respuesta y la respuesta correcta. ■ el alumno ya no puede interactuar con la gráfica. ■ el alumno ya no puede interactuar con los botones de añadir y eliminar recorrido. ■ el alumno ya no puede solicitar de nuevo la evaluación del ejercicio.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 73: Pruebas E2E-29 Realización incorrecta del ejercicio indicar el recorrido

Visualización del ejercicio máximos y mínimos	
Entrada	El usuario accede al ejercicio máximos y mínimos.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra un botón de eliminación de puntos. ■ se muestra un botón de validación debajo de la gráfica. ■ los puntos o constantes introducidos por el usuario se pintan sobre la gráfica.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 74: Pruebas E2E-30 Visualización del ejercicio máximos y mínimos

Realización correcta del ejercicio máximos y mínimos	
Entrada	El usuario accede al ejercicio máximos y mínimos y lo resuelve correctamente.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra un mensaje de validación correcta en verde. ■ el usuario ya no puede interactuar con la gráfica. ■ se deshabilita el botón de solicitud de validación.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 75: Pruebas E2E-31 Realización correcta del ejercicio máximos y mínimos

Realización correcta del ejercicio máximos y mínimos	
Entrada	El usuario accede al ejercicio máximos y mínimos y lo resuelve incorrectamente.
Resultado esperado	<ul style="list-style-type: none"> ■ se muestra un mensaje de validación incorrecta en rojo. ■ los puntos que el usuario haya introducido erróneamente están marcados en amarillo. ■ los puntos que le falten al usuario por introducir están marcados en naranja. ■ el usuario ya no puede interactuar con la gráfica. ■ se deshabilita el botón de solicitud de validación.
Resultado	Ok.
Comentarios	Se ha probado con todos los ejercicios.

Tabla 76: Pruebas E2E-32 Realización incorrecta del ejercicio máximos y mínimos

6.8. Tecnologías utilizadas

En esta sección se va a detallar las tecnologías y herramientas utilizadas durante el desarrollo de este proyecto. La elección de las tecnologías adecuadas es fundamental para realizar un desarrollo eficiente y exitoso de la aplicación.

A continuación se va a dar una descripción general de cada una y su papel en el desarrollo de la aplicación.

6.8.1. Python

Python es un lenguaje multiplataforma de alto nivel, interpretado, con tipado dinámico y orientado a objetos. Es un lenguaje muy popular por su facilidad de aprendizaje, la gran comunidad que tiene y su amplia aplicación en diversos áreas. La popularidad de python se puede verificar en su posición entre los lenguajes de programación en índices como TIOBE o RedMonk o su uso en proyectos reales.

El índice TIOBE (The Importance of Being Earnest) clasifica los lenguajes según la demanda laboral; resultados de motores de búsqueda como Google, Yahoo, etc.; los cursos de formación disponible, entre otros factores. En el índice TIOBE Python ocupa la primera posición. [6]

El índice RedMonk se basa en el principio de que la popularidad de un lenguaje depende de la comunidad de desarrolladores detrás de él y su relevancia en un lenguaje particular. En el índice RedMonk ocupa el segundo lugar. [7]

Python se usa en proyectos tan importantes como Django, framework python para el desarrollo de aplicaciones [10], Spotify [8] o Dropbox [9].

Fue creado por Guido van Rossum entre 1989 y 1991, momento en el que sacó la primera

versión del lenguaje, la versión 0.9.0. El objetivo era crear un lenguaje que siguiera la filosofía del lenguaje de programación ABC, un lenguaje simple, legible y amigable con los principiantes, características resumidas en el documento *PEP 20 – The Zen of Python*. [5]

En el proyecto se ha usado Python como lenguaje de programación para el desarrollo de la aplicación. Se ha elegido por los siguientes motivos:

- ser un lenguaje multiplataforma, para poder portar la aplicación a Windows 10.
- las bibliotecas existentes para el lenguaje, poder tener varias opciones Matplotlib, PyQt-Graph.
- la gran comunidad que tiene, para tener mayor facilidad para resolver los problemas que surgen en el desarrollo software.
- por la experiencia previa y comodidad del desarrollador con el lenguaje.

6.8.2. PyQt5

PyQt5 es una biblioteca de Python para el desarrollo de aplicaciones de escritorio mediante el framework Qt. Destaca por la integración con otras bibliotecas de python, la amplia gama de widgets disponible, el soporte multiplataforma y el diseño visual mediante Qt Designer.

Una característica muy importante de Qt es su sistema de señales, los objetos Qt emiten señales cuando ocurren eventos sobre ellos, señales que se pueden manejar y conectar a métodos u otros objetos Qt. Este sistema permite un gran desacoplamiento, una separación clara entre los objetos emisores y receptores, facilitando el desarrollo de la aplicación. [11]

PyQt surge en 1998 cuando Riverbank Computing Ltd. adquiere una licencia de uso de Qt para desarrollar PyQt. El objetivo era crear aplicaciones de escritorio con interfaz de usuario mediante Python. Con el paso del tiempo Riverbank Computing Ltd. ha ido lanzando nuevas versiones de PyQt. Entre ellas destaca PyQt5, una versión potente del framework y con un soporte muy grande por parte de la empresa y la comunidad de desarrolladores. [12]

En este proyecto se ha usado PyQt5, una versión muy popular del framework PyQt, como la librería para desarrollar la aplicación de escritorio. Se ha usado porque es altamente portable, una librería para crear aplicaciones de escritorio muy versátil, dispone de una cantidad enorme de widgets para modelar la interfaz de usuario, y tiene librerías asociadas como PyQtGraph.

6.8.3. PyQtGraph

PyQtGraph es una biblioteca de Python para la visualización de datos mediante gráficos. Está construida mediante PyQt y numpy, por lo que está diseñada para ser utilizada con PyQt y soportar el uso grandes conjuntos de datos. [13]

PyQtGraph surge en 2009 para crear una biblioteca de visualización de datos en tiempo real, su gran ventaja frente a Matplotlib.

Se ha utilizado PyQtGraph para visualizar correctamente todas las gráficas usadas en la aplicación. La elección de PyQtGraph ha sido realizada por su facilidad de uso con PyQt5 y la magnitud de los cálculos a ejecutar en la aplicación, nada excesivamente complejo como para necesitar una aplicación como Matplotlib.

6.8.4. Pip

Pip (Pip Installs Packages) es un gestor de paquetes para Python. Fue creado para facilitar la instalación y eliminación de las bibliotecas y paquetes de Python, así como sus dependencias, utilizadas en un proyecto. [26]

Pip surgió a inicios de la década de los 2000 cuando Ian Bicking desarrollo un módulo para la gestión de paquetes que mejorará *easy_install*, gestor anteriormente utilizado, porque se quería un sistema más predecible y centralizado. Pip terminó saliendo en 2008 y se incluyó en la versión 2.7 de Python.

Se ha utilizado Pip en el proyecto porque es el gestor de paquetes más utilizado para Python.

6.8.5. Unittest

Es un framework de pruebas unitarias para Python. En este framework cada prueba tiene tres posibles resultados:

- OK: indica que la prueba ha sido un éxito.
- FAIL: indica que la prueba ha fallado en alguna sentencia, dando lugar a una excepción `AssertionError`.
- ERROR: indica que durante la prueba ha ocurrido excepción distinta a `AssertionError`. Al ejecutar el test ocurre un error.

Unittest agrupa en clases (*suites* de tests) conjuntos de tests, permitiendo manejar los tests pertenecientes a la clase de forma conjunta.

Cada clase de test tiene que heredar de la clase `TestCase`, la clase base con los métodos y propiedades necesarios para poder lanzar correctamente los tests. Esta clase contiene los siguientes métodos de preparación y limpieza para los tests:

- `setUp`: método de preparación que se ejecuta antes de cada test.
- `tearDown`: método de limpieza que se ejecuta al finalizar cada test.
- `setUpClass`: método de preparación que solo se ejecuta una vez para toda la clase, antes de iniciar los tests contenidos en ella. Debe ser un método de clase con el decorador `@classmethod`.
- `tearDownClass`: método de limpieza que se ejecuta una vez para toda la clase, al finalizar el último test contenido en ella. Al igual que el método anterior, es un método de clase con el decorador `@classmethod`.

A continuación se van a mostrar la mayoría de los métodos de comprobación del framework, mediante estos métodos se puede testear el funcionamiento de cada prueba:

Método	Descripción
assertEqual(a, b)	a == b
assertNotEqual(a, b)	a != b
assertTrue(a)	bool(a) is True
assertFalse(a)	bool(a) is False
assertIs(a, b)	a is b
assertIsNone(a)	a is None
assertIn(a)	a in b
assertIsInstance(a, b)	isinstance(a, b)
assertGreater(a, b)	a > b
assertLessEqual(a, b)	a ≤ b
assertRegex(s, r)	r.search(s)

Se ha usado unittest para los tests unitarios de la aplicación por ser un módulo de Python para pruebas automatizadas muy completo, ha permitido realizar todos los tests unitarios necesarios durante el desarrollo de la aplicación, y también se ha utilizado por la experiencia del desarrollador con este framework de pruebas.

6.8.6. CSS

Es un lenguaje de estilos (*Cascading Style Sheets* utilizado para mejorar la presentación de la interfaz de usuario, mediante reglas agrupadas permite definir como deben de ser mostrados visualmente los elementos afectados por cada regla.

Fue propuesto en 1994 por desarrolladores del CERN, entre ellos Håkon Wium Lie, CTO de Opera Software [20].

En este proyecto se ha utilizado para definir el aspecto visual de los componentes dentro de la aplicación, ha sido esencial para definir una interfaz consistente a lo largo de toda la aplicación.

6.8.7. Git

Git es un sistema de control de versiones distribuido. Un sistema de control de versiones permite gestionar correctamente todas las actualizaciones de código: tener almacenado todos los cambios en el código de la aplicación, gestionar conflictos entre los cambios de los distintos desarrolladores, facilitar la búsqueda de errores en el código e incluso revertir cambios de forma sencilla.

Git fue creado en 2005 por Linus Torvalds para el desarrollo del kernel de linux. [14]

Se ha utilizado Git en el proyecto como sistema de control de versiones. No ha habido una razón específica para usar Git en vez de Subversión, ambos son sistemas que cumplían las necesidades para el desarrollo correcto de la aplicación.

6.8.8. Github

Se ha usado Github para alojar el código de la aplicación en repositorios en línea usando Git. Tener almacenado el código en repositorios remotos ha evitado pérdidas de código por problemas en la máquina local de desarrollo de la aplicación.

6.8.9. Pycharm

PyCharm es un IDE (entorno de desarrollo integrado) creado por JetBrains específicamente para el desarrollo de aplicaciones Python. Se ha utilizado como IDE en la aplicación sobre todo por la posibilidad de depurar el código de manera sencilla y cómoda.

6.8.10. Mypy

Mypy es una herramienta de análisis de anotaciones estáticas para Python. Ayuda a detectar errores de tipado de variable de Python. Las anotaciones de tipo no tienen en efecto en tiempo de ejecución pero permiten mejorar la calidad y mantenibilidad del código.

6.8.11. Draw.io

Aplicación online y gratuita para crear y editar diagramas de forma fácil, permite exportar los diagramas realizados a multitud de formatos. Se ha utilizado en el proyecto por su facilidad permitiendo hacer los diagramas de clase y despliegue.

6.8.12. Plantuml

PlantUML es una herramienta de código abierto que permite crear diagramas UML utilizando una sintaxis específica, para crear diagramas hay que escribir un pequeño script en vez de manejar elementos gráficos como se suele hacer en otras aplicaciones. Se ha utilizado en el proyecto para los diagramas de secuencia.

6.8.13. Virtual environment

Un entorno virtual de Python es una herramienta que permite crear y gestionar entornos aislados para proyectos Python, entornos donde se puedan instalar dependencias específicas sin interferir con el sistema global u otros proyectos. Se ha utilizado en la aplicación para facilitar la instalación de la aplicación usando un entorno aislado de los sistemas del usuario.

6.8.14. MySQL

MySQL es un sistema de gestión de bases de datos relacional. Permite almacenar, organizar y manipular grandes cantidades de datos. MySQL es un gestor de base de datos es multiplataforma y compatible con diversos lenguajes de programación, lo que facilita la integración de la base de datos en aplicaciones y sitios web.

En el proyecto se ha utilizado por ser un gestor de bases de datos relacional y su facilidad de integración con Python.

6.8.15. Overleaf

Editor online de documentos en formato Latex. Se ha utilizado para desarrollar la memoria del proyecto.

6.8.16. Coolors

Aplicación web para generar de forma sencilla una adecuada paleta de colores. Se ha utilizado en el proyecto para crear de forma sencilla la gama de colores utilizada a lo largo de toda la aplicación.

6.8.17. Flaticon

Aplicación web para obtener una amplia colección de iconos en distintos formatos. Para el proyecto se ha creado una cuenta en la aplicación permitiendo así poder editar los iconos y agregar a los iconos la paleta de colores de la aplicación.

7. Conclusiones y trabajo a futuro

7.1. Conclusiones

Se han cumplido todos los objetivos funcionales propuestos al comienzo del TFG desarrollando una aplicación de apoyo al aprendizaje de las matemáticas para la asignatura de Fundamentos de Matemáticas del Grado de Ingeniería Informática. La aplicación ha incorporado las ayudas y apoyos de retroalimentación que se ha considerado que era valioso introducir a lo largo del desarrollo del proyecto.

El único objetivo no funcional que no se ha podido cumplir es la conversión de la aplicación a un ejecutable de windows 10, se ha intentado utilizar tecnologías como *py-installer* pero han resultado en intentos negativos.

El desarrollo ha sido complejo sobre todo para poder conseguir plasmar en la aplicación ese refuerzo de conceptos que se quería conseguir. Se ha cambiado el enfoque varias veces sobre cómo transmitir al alumno sus errores en cada tipo de ejercicio, pero finalmente se ha conseguido para cada ejercicio uno con el que estoy contento y considero que cumple con el objetivo.

A lo largo del desarrollo de este trabajo he aprendido mucho. El desarrollo de la aplicación ha sido muy enriquecedor al tener que crear una aplicación compleja desde cero, forzando a utilizar todo lo aprendido a lo largo de los cuatro años de carrera. Ha sido un proceso extenso pero que me ha permitido entender mejor conceptos dados en cursos pasados y relacionarlos entre sí.

A nivel de gestión de proyecto he experimentado como algunos cambios en la planificación pueden provocar retrasos considerables en el desarrollo del proyecto. También he ido mejorando mucho en la estimación de las tareas, al principio he estimado con mucha incertidumbre, que según iba realizando tareas iba disminuyendo.

A nivel técnico he aprendido muchos conceptos informáticos relacionados con las aplicaciones de escritorio, la gestión de eventos, la intercepción y propagación de señales, el uso de widgets, entre otras cosas... Además, el objetivo de desarrollar una aplicación para windows, aunque no se haya conseguido, me ha permitido realizar tareas que no había hecho antes, aprendiendo mucho de como poder hacer una aplicación multiplataforma.

Considero que el desarrollo en PyQt ha sido más complejo de lo esperado y menos productivo que si hubiera sido realizado mediante frameworks web. Sin embargo, si tuviera que elegir de nuevo las tecnologías a utilizar para este proyecto seguirían siendo las mismas, he aprendido mucho saliendo de mi zona de confort y desarrollando una aplicación como la entregada.

7.2. Trabajo a futuro

La línea principal de trabajo a futuro es la portabilidad de la aplicación a otras plataformas, en concreto a windows, para permitir a la tutora y a los alumnos un fácil acceso a ella.

Además de la portabilidad, hay otro aspecto a futuro que considero que es el trabajo más inmediato a realizar. La inclusión de dos nuevas sesiones de ejercicios de libre creación:

- módulo de ejercicios de repaso: utilizando las estadísticas del usuario se podría configurar una sesión con los ejercicios que no superaran cierto porcentaje de acierto, editable por el alumno.
- módulo de ejercicios a petición: permitir al usuario crear su propia sesión de ejercicios.

Estos módulos son posibles ahora mismo gracias a las configuraciones de ejercicios, mediante la asociación de todos los tipos de ejercicios a los módulos el alumno podría llevar a cabo sesiones muy completas y a petición.

Ya se ha explicado anteriormente la arquitectura modular seguida en la construcción de la aplicación, la aplicación está diseñada para soportar nuevos módulos de la asignatura. Es una línea de trabajo a futuro que habría que tratar con los profesores de matemáticas.

Sobre el módulo de funciones se podrían añadir los siguientes ejercicios:

- añadir ejercicios de detección de asíntotas, otro concepto que puede conllevar cierta dificultad para el alumnado.
- determinar si las funciones están o no acotadas.

Otras pequeñas funcionalidades que se han pensado son las siguientes:

- gradiente css en las sesiones de ejercicios según el porcentaje de acierto. A mayor porcentaje el botón de sesión estaría más iluminado. Para hacer más visual la aplicación.
- poder filtrar las estadísticas por fecha.

Referencias

- [1] Wikipedia. (2021). *Derive (computer algebra system) - Wikipedia*. Recuperado el 01/07/2021 de [https://en.wikipedia.org/wiki/Derive_\(computer_algebra_system\)](https://en.wikipedia.org/wiki/Derive_(computer_algebra_system))
- [2] Wikipedia. (2022). *Statgraphics - Wikipedia*. Recuperado el 01/07/2021 de <https://en.wikipedia.org/wiki/Statgraphics>
- [3] Scrum Guide. (2020). *Scrum Guide — Scrum Guides*. Recuperado el 01/07/2021 de <https://scrumguides.org/scrum-guide.html>
- [4] Python org (2021) *Welcome to Python.org*. Recuperado el 05/03/2022 de <https://www.python.org/>.
- [5] Peters, T. (2020) *PEP 20 – The Zen of Python*. Recuperado el 05/03/2022 de <https://www.python.org/dev/peps/pep-0020/>
- [6] TIOBE (2023) *TIOBE Index - TIOBE*. Recuperado el 01/07/2023 de <https://www.tiobe.com/tiobe-index/>
- [7] REDMONK (2023) *The RedMonk Programming Language Rankings: January 2023 – tecosystems*. Recuperado el 01/07/2023 de <https://redmonk.com/sogradey/2023/05/16/language-rankings-1-23/>
- [8] van der Meer, G. (2013) *How we use Python at Spotify - Spotify Engineering : Spotify Engineering*. Recuperado el 01/07/2023 de <https://engineering.atspotify.com/2013/03/how-we-use-python-at-spotify/>
- [9] Bélanger, M.. DeVille D. (2018) *How we rolled out one of the largest Python 3 migrations ever - Dropbox*. Recuperado el 01/07/2023 de <https://dropbox.tech/application/how-we-rolled-out-one-of-the-largest-python-3-migrations-ever>
- [10] Django (2023) *The web framework for perfectionists with deadlines — Django*. Recuperado el 01/07/2023 de <https://www.djangoproject.com/>
- [11] PyQt (2023) *Signals & Slots — Qt Core 5.15.14*. Recuperado el 01/07/2023 de <https://doc.qt.io/qt-5/signalsandslots.html>
- [12] Riverbank Computing Ltd. (2023) *Riverbank Computing — Introduction*. Recuperado el 01/07/2023 de <https://www.riverbankcomputing.com/software/pyqt/>
- [13] PyqtGraph (2023) *PyQtGraph - Scientific Graphics and GUI Library for Python*. Recuperado el 01/07/2023 de <https://www.pyqtgraph.org/>
- [14] Git (2023) *Git*. Recuperado el 01/07/2023 de <https://git-scm.com/>
- [15] Software Testing Material (2021) *What Is Software Testing? Definition, Basics, Types, Methods, Approaches* Recuperado el 01/07/2023 de <https://www.softwaretestingmaterial.com/software-testing/>
- [16] Refactoring Guru (2023) *Singleton* Recuperado el 03/07/2023 de <https://refactoring.guru/es/design-patterns/singleton>
- [17] Fowler, M. (2002) *P of EAA: Data Mapper* Recuperado el 03/07/2023 de <https://martinfowler.com/eaCatalog/dataMapper.html>

- [18] Refactoring Guru (2023) *Factory Method* Recuperado el 03/07/2023 de <https://refactoring.guru/es/design-patterns/factory-method>
- [19] PyQt (2023) *Model/View Programming - Qt for Python* Recuperado el 03/07/2023 de <https://doc.qt.io/qtforpython-6/overviews/model-view-programming.html>
- [20] Wikipedia (2021) *CSS - Wikipedia* Recuperado el 01/07/2023 de <https://en.wikipedia.org/wiki/CSS>
- [21] Indeed (2023) *Sueldo de Programador/a junior en España* Recuperado el 06/07/2023 de https://es.indeed.com/career/programador-junior/salaries?from=top_sb
- [22] Asime (2023) *DISTRIBUCIÓN DE LA JORNADA LABORAL PARA 2023* Recuperado el 06/07/2023 de <https://asime.es/wp-content/uploads/2022/12/DISTRIBUCION-DE-LA-JORNADA-LABORAL-PARA-2023.pdf>
- [23] Nick Babich (2016) *Golden Rules of User Interface Design — by Nick Babich — UX Planet* Recuperado el 06/07/2023 de <https://uxplanet.org/golden-rules-of-user-interface-design-19282aeb06b>
- [24] Alejandra Martínez Monés (2017) *IPC-Usabilidad* Recuperado el 06/07/2023.
- [25] Kanbanize (2023) *What Is Kanban? Explained in 10 Minutes — Kanbanize* Recuperado el 06/07/2023 de <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban>
- [26] Wikipedia (2023) *pip (package manager) - Wikipedia* Recuperado el 06/07/2023 de [https://en.wikipedia.org/wiki/Pip_\(package_manager\)](https://en.wikipedia.org/wiki/Pip_(package_manager))
- [27] Creative Commons (2021) *Creative Commons — Attribution-NonCommercial-NoDerivatives 4.0 International — CC BY-NC-ND 4.0* Recuperado el 10/07/2023 de <https://creativecommons.org/licenses/by-nc-nd/4.0/>
- [28] Mypy (2023) *mypy 1.4.1 documentation* Recuperado el 06/07/2023 de <https://mypy.readthedocs.io/en/stable/>
- [29] O'REILLY (2023) *1. Layered Architecture - Software Architecture Patterns [Book]* Recuperado el 02/07/2023 de <https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html>
- [30] *Apuntes de Fundamentos de Matemáticas.* Recuperado el 09/07/2023.

A. Manual de instalación

Para instalar la aplicación se necesita un ordenador con sistema operativo linux, el utilizado en el desarrollo de la aplicación ha sido Ubuntu 20.04.

1. descomprimir el zip adjunto en la entrega del TFG con el código de la aplicación.
2. creación de un entorno virtual en las carpetas del proyecto.

```
python3 -m venv nombre_entorno
```

3. activar el entorno virtual.

```
source nombre_entorno/bin/activate
```

4. instalación de los requisitos redactados en el archivo *requirements.txt*.

```
pip install -r requirements.txt
```

5. ejecutar el archivo inicial *main.py*

```
python3 main.py
```

6. si la aplicación da un error inicial puede ser porque tenga que re-instalarse una librería de PyQt5 *libxcb-xinerama0*.

```
sudo apt-get install --reinstall libxcb-xinerama0
```

B. Repositorio del proyecto

El código está alojado en el siguiente repositorio: <https://github.com/tomero98/pymath/tree/master>