



UNIVERSIDAD DE GRANADA

TRABAJO FIN DE GRADO
INGENIERÍA EN INFORMÁTICA

LAB 2223

Entorno colaborativo virtual para aprendizaje visual de conceptos

Autor

Miguel Carracedo Rodríguez

Director

Juan José Ramos Muñoz



Escuela Técnica Superior de Ingenierías Informática y
de Telecomunicación

Granada, julio de 2023



**UNIVERSIDAD
DE GRANADA**

LAB 2223

Entorno colaborativo virtual para aprendizaje visual de conceptos

Autor

Miguel Carracedo Rodríguez

Directores

Juan José Ramos Muñoz

LAB 2223: Entorno colaborativo virtual para aprendizaje visual de conceptos

Miguel Carracedo Rodríguez

Palabras clave: motor gráfico, Godot, GDScript, Java, Android

Resumen

Entorno colaborativo virtual para aprendizaje visual de conceptos o LAB es un proyecto centrado en facilitar a los alumnos cursando el grado de nutrición humana y dietética en la facultad de farmacia el aprendizaje de las distintas actividades que se realizan en el laboratorio de las asignaturas correspondientes al segundo curso de la carrera (toxicología alimentaria, nutrición II).

Para ello se ha desarrollado una aplicación interactiva con ayuda del motor gráfico Godot en 2D donde cada estudiante podrá seleccionar la actividad que desee simular para comprobar que sabe realizarla correctamente.

De esta manera el estudiante puede asegurarse de que ha comprendido paso a paso la actividad que realizará en el laboratorio y a su vez su profesor podrá comprobar que sus alumnos entienden el procedimiento.

La aplicación se encuentra estructurada en distintas actividades, cada una correspondiente a una práctica de laboratorio distinta donde el estudiante tendrá que seguir los pasos que se le indican para su correcta realización. Según se vayan completando las actividades se irá indicando en el menú principal su estado (pendiente de hacer o finalizada) para que el profesor pueda llevar a cabo el seguimiento del progreso de cada alumno.

La aplicación obtenida durante el desarrollo del proyecto permite ser ampliada en el futuro en caso de que así se desee. Se indica en este documento la funcionalidad que podría ser añadida a la aplicación que quedó pendiente de ser implementada.

LAB 2223: Entorno colaborativo virtual para aprendizaje visual de conceptos

Miguel Carracedo Rodríguez

Keywords: graphical engine, Godot, GDScript, Java, Android

Abstract

Virtual collaborative environment for visual learning of concepts or LAB is a project focused on facilitating the learning of different activities that are carried out in the laboratory of the subjects corresponding to the second degree course (food toxicology, nutrition II).

An interactive application has been developed with the help of Godot 2D graphics engine where each student can select the activity they want to simulate to check that they know how to do it correctly.

In this way, the students can make sure that they have understood the activity that he will carry out in the laboratory step by step and, in turn, his teacher will be able to verify that his students understand the procedure.

The application is structured in different activities, each one corresponding to a different laboratory practice where the student will have to follow the steps indicated for its correct completion.

As the activities completed, their status (pending to be done or completed) will be indicated in the main menu so that the teacher can monitor the progress of each student.

The application obtained during the development of the project can be expanded in the future if desired. This document indicates functionality that could be added to the application that was pending to be implemented.

Yo, **Miguel Carracedo Rodríguez**, alumno de la titulación Grado en Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 45959184D, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Miguel Carracedo Rodríguez

Granada a 13 de julio de 2023

D. **Juan José Ramos Muñoz**, Profesor del Área de Ingeniería Telemática del Departamento de Teoría de la Señal, Telemática y Comunicaciones de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado ***Entorno colaborativo virtual para aprendizaje visual de conceptos (LAB2223)***, ha sido realizado bajo su supervisión por **Juan José Ramos Muñoz**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 13 de julio de 2023

Los directores:

Juan José Ramos Munoz

Agradecimientos

En primer lugar deseo darle las gracias a mis padres, por su apoyo, preocupación y por haberme aguantado todos estos años. Fue gracias a ellos que continué desarrollando el proyecto de forma constante incluso en ocasiones donde no me apetecía.

Quiero también darle las gracias a mi tutor, por haberme ayudado durante todo el desarrollo siendo de gran ayuda a la hora de resolver dudas y la comunicación con el cliente. Gracias también por ser comprensivo y ajustar el avance del proyecto en base a los problemas personales que surgieron durante su desarrollo.

Dar las gracias también a todos los amigos y familiares que me ayudaron dándome ideas de como desarrollar la aplicación y me ayudaron a probarla para encontrar errores.

Agradecer también a todos los desarrolladores de bibliotecas sin las cuales habría sido más tediosa la programación del proyecto.

Gracias a todos por haberme ayudado en esta etapa final de mi vida universitaria y haberme facilitado lo máximo posible la realización de este proyecto.

Índice general

Introducción.....	21
1.1. Contexto y motivación.....	21
1.2. Objetivos.....	22
1.3. Estructura del documento.....	22
Contexto.....	24
2.1. Dominio del problema a abordar.....	24
2.2. Estado del arte.....	24
2.2.1. Aplicaciones similares.....	25
2.2.2. Laboratorios virtuales de años pasados.....	26
2.3. Tecnologías.....	26
2.3.1. Herramientas para la gestión del proyecto.....	28
Gestión del proyecto y planificación.....	29
3.1. Metodología de desarrollo del proyecto.....	29
3.1.1. Desarrollo incremental.....	29
3.1.2. Scrum.....	31
3.1.2.1 Historias de Usuario.....	31
3.1.3. Backlog.....	32
3.1.4. Sprints.....	32
3.1.5. Burndown.....	32
3.2. Planificación temporal.....	34
3.2.1. Brainstorming inicial.....	34
3.2.2. Seguimiento.....	35
3.2.3. Retrospectiva.....	35
3.3. Trello.....	35
3.4. Estimación de coste.....	36
3.4.1. Software.....	36
3.4.2. Hardware.....	37
3.4.3. Recursos humanos.....	37
3.4.4. Coste total.....	38
3.5. Reuniones con el cliente.....	39
3.6. Decisiones tomadas.....	40
3.7. Uso de GitHub y copias de seguridad.....	40
Diseño.....	42
4.1. Diseño general.....	42
4.2. Paquetes.....	42
4.3. Clases.....	43
4.4. Caso de uso.....	45
4.5. Conclusiones y decisiones tomadas.....	45
Implementación.....	47
5.1. Godot Engine.....	47
5.2. GDScript.....	47
5.3. Estructura de las actividades.....	48

5.4. Bibliotecas.....	52
5.6. Estructura del proyecto.....	52
5.7. Exportación de la aplicación.....	54
5.8. Código.....	55
Pruebas.....	57
6.1. Tests.....	58
6.2. Betas y pruebas con el cliente.....	58
6.3. Configuración para ejecutar la aplicación.....	59
Conclusiones y trabajos futuros.....	60
7.1. Conclusiones.....	60
7.2. Trabajos futuros.....	61
Bibliografía.....	62

Índice de figuras

Figura 1: Logotipo de Godot Engine.....	28
Figura 2: Representación del desarrollo incremental.....	30
Figura 3: Ejemplo de una gráfica burndown.....	33
Figura 4: Diagrama de Gantt sobre la aplanificación real.....	33
Figura 5: Entorno de trabajo de Trello.....	36
Figura 6: Diagrama de paquetes.....	43
Figura 7: Diagrama de clases simplificado.....	44
Figura 8: Actividad 1 en la etapa inicial del proyecto.....	46
Figura 9: Selección del material en la etapa inicial del proyecto.....	46
Figura 10: Menú principal de la aplicación.....	48
Figura 11: Pantalla inicial de cada actividad.....	49
Figura 12: Sección de materiales de cada actividad.....	50
Figura 13: Sección de calentar de la actividad 1.....	51
Figura 14: Indicadores del progreso de cada actividad.....	52
Figura 15: Árbol de directorios del proyecto.....	54
Figura 16: Código de transición al menú principal.....	55
Figura 17: Código de salto a una escena mediante un botón.....	56

Índice de tablas

Tabla 1: Ventajas e inconvenientes de los laboratorios virtuales.....	25
Tabla 2: Incremento del proyecto por sprints.....	31
Tabla 3: Ideas recogidas del brainstorming con el tutor.....	34
Tabla 4: Ideas recogidas del brainstorming con los clientes.....	34
Tabla 5: Recursos software utilizados en el proyecto.....	37
Tabla 6: Recursos humanos utilizados en el proyecto.....	38
Tabla 7: Coste total del desarrollo del proyecto.....	38
Tabla 8: Reuniones semanales con el cliente.....	39
Tabla 9: Pruebas realizadas a lo largo del desarrollo del proyecto.....	57

Capítulo 1

Introducción

1.1. Contexto y motivación

Este proyecto consiste en el desarrollo de un laboratorio virtual para las asignaturas toxicología alimentaria y nutrición II del grado de nutrición humana y dietética de la universidad de Granada [14].

Este trabajo se enmarca en un proyecto de innovación docente cuyo objetivo es el de poder facilitar el aprendizaje de las actividades realizadas en el laboratorio de prácticas de una forma original y desde una perspectiva diferente a través de las distintas actividades que se realizan en dicho proyecto.

La idea de este proyecto surgió gracias a la colaboración entre la facultad de Farmacia y la Escuela Técnica Superior en Ingenierías Informática y de Telecomunicación con el objetivo de ayudar a los alumnos y profesores de las asignaturas de a poder facilitar el aprendizaje del procedimiento a seguir a la hora de realizar ciertas actividades de laboratorio (selección de material y mezclas entre los reactivos). Para el autor de este trabajo ha supuesto simular un primer encargo profesional, con contacto entre clientes y desarrolladores, de forma que al final se obtiene un producto para estos el cual sea capaz de satisfacer las necesidades que se hayan especificado.

Para ello, se decide crear una aplicación interactiva 2D. Tras el estudio de las herramientas disponibles se decide utilizar el motor gráfico Godot Engine (en la sección de estado del arte se realizará la comparativa con otros motores similares), compatible en ordenadores con el sistema operativo Windows y móviles con el sistema operativo Android. Dicha aplicación contiene distintas actividades donde cada una de estas simula una práctica de laboratorio que los estudiantes tendrán que realizar. Cada actividad necesitará ser realizada de una forma específica para que se indique como finalizada correctamente y su estado cambie de no completada a completada. Posteriormente el profesor podrá comprobar de forma individual los resultados obtenidos por cada uno de sus alumnos y ver si han sido capaces de entender los conceptos explicados en clase.

Como resultado se ha desarrollado una aplicación compatible con Android y Windows la cual plantea una serie de actividades a realizar por el/la estudiante. Dichas actividades simulan una actividad de la parte de prácticas de la asignatura permitiendo seleccionar el material y los reactivos necesarios para su correcta realización. De esta manera se le permite al estudiante repasar los pasos para realizar la práctica en su casa mediante el laboratorio virtual evitando que la única forma de poder repasarlos sea a la hora de realizar la práctica en el laboratorio físico.

1.2. Objetivos

Los objetivos de este proyecto son la creación de una aplicación tanto para ordenador como para móvil que permita a los estudiantes simular un laboratorio virtual en el que se pueden repasar que material y reactivos son necesarios para realizar las prácticas de laboratorio de las asignaturas de toxicología y nutrición II. A su vez permitirá al profesor comprobar de forma individual que cada alumno comprende los conceptos explicados en clase.

1.3. Estructura del documento

En aras de presentar el trabajo de forma estructurada, este documento se ha dividido en distintos capítulos. A continuación se indica que puntos se tratarán en cada capítulo así como un breve resumen de su contenido:

- ◆ En este Capítulo 1 (actual) se ha explicado como surge este proyecto así como los objetivos principales de este y un breve resumen de la estructura de este documento
- ◆ A continuación, en el capítulo 2 se abordará el entorno en el que se desarrolló la aplicación y de donde se obtuvieron las referencias para su desarrollo así como aplicaciones con una estructura similar a la desarrollada.
- ◆ En el capítulo 3 se explicará el tipo de planificación utilizada para el desarrollo del proyecto junto con las estimaciones del coste total del desarrollo de este.
- ◆ En el capítulo 4 se hablará del diseño de la aplicación y las distintas decisiones tomadas durante su desarrollo. Seguido por la manera en la que se resolvieron los problemas que fueron surgiendo.
- ◆ En el capítulo 5 se explicará como se fue implementando el proyecto en el motor elegido junto con la forma en la se encuentra estructurado este. También se hablará de las distintas bibliotecas utilizados como soporte de la aplicación junto con el uso de GitHub para el almacenamiento de copias de seguridad.
- ◆ En el capítulo 6 se explicarán como se llevaron a cabo las distintas pruebas en la aplicación para la detección de errores, bugs y otros elementos que pudiesen dañar la integridad del proyecto. Al final de este capítulo de explica-

rán las distintas reuniones realizadas con el cliente y los objetivos de cada una de estas así como las conclusiones obtenidas al final de cada reunión.

- ◆ Por último, en el capítulo 7 se abordan los posibles añadidos que quedaron pendientes de ser implementados en la aplicación así como posibles mejoras de cara al futuro. En el último apartado se realiza una conclusión de los resultados obtenidos durante la realización de proyecto a nivel de tiempo y rendimiento y una opinión personal del mismo.
- ◆ En capítulo 8 corresponde a la bibliografía donde se hace referencia a la documentación utilizada en todo el proyecto.

Capítulo 2

Contexto

2.1. Dominio del problema a abordar

En el segundo curso de la carrera en nutrición y dietética se encuentran dos asignaturas en las que se realizan prácticas en el laboratorio sobre distintos productos nutricionales (toxicología alimentaria y nutrición II). En estas asignaturas la mayor parte del aprendizaje que se lleva a cabo en las aulas se realiza de forma escrita ya sea mediante exámenes o una serie de pruebas de otro tipo que se puedan evaluar. Surge así la idea de crear una aplicación que permita a los estudiantes comprobar su conocimiento de la materia estudiada en la que a la vez que estos vayan realizando las actividades, el profesor pueda hacer un seguimiento individual del progreso de cada estudiante. Esta aplicación esta enfocada en la creación de un laboratorio virtual 2D donde los estudiantes puedan simular distintas prácticas de laboratorio que han sido explicadas en la parte teórica de la asignatura para así poder ser evaluados de forma distinta.

A pesar de que la gran parte de los alumnos a los que esta aplicación esta enfocada se encuentren familiarizados con la tecnología la aplicación a desarrollar no debe de ser muy compleja y poco intuitiva a la hora de navegar por ella. Se pretende crear una aplicación que no solo permita a los estudiantes poner a prueba su conocimiento si no también ser capaz de enseñarles los conceptos teóricos explicados de forma simple haciendo posible evaluarlos y enseñarles a la vez.

2.2. Estado del arte

Actualmente existen varios laboratorios virtuales muy similares a los que se ha generado en este proyecto. Entre todos estos laboratorios virtuales los que más destacan son “AULA EN RED” [15], “Biomodel” [16], “Chemical Reaction” [17], “CHEMLAB” [18] entre otros, no solo por su gran nivel de detalle y facilidad de uso sino también por tener una instalación gratuita lo cual hace su utilidad mucho mayor. Una de las principales diferencias entre estos laboratorios virtuales y el generado en el proyecto es que este último está mucho más enfocado a las actividades que los estudiantes de la facultad de farmacia realizarán en el laboratorio pudiendo tener un nivel de detalle y explicación mayor que si estuviera generalizado.

A continuación se muestra una tabla reuniendo las ventajas e inconvenientes que cada uno de los motores gráficos anteriormente mencionados presentan:

Laboratorio Virtual	Ventaja	Inconveniente
AULA EN RED	Información en español con un gran detalle a nivel visual del material utilizado.	Las actividades que se pueden realizar están muy simplificadas y carecen de suficiente detalle.
Biomodel	Las actividades que se pueden realizar poseen un gran nivel de complejidad y profundidad.	Ninguna de las actividades está relacionada con las realizadas por los estudiantes.
Chemical Reaction	Las explicaciones de los procedimientos y las medidas de seguridad son muy detalladas con enlaces externos demostrativos.	Toda la información se muestra en inglés lo cual puede suponer un problema para personas no familiarizadas con este lenguaje
CHEMLAB	Fácil manejo de la interfaz y muy intuitivo a la hora de usarse por primera vez.	Este laboratorio virtual no posee actividades relacionadas con las realizadas por los estudiantes en la facultad de nutrición y dietética.

Tabla 1: Ventajas e inconvenientes de los laboratorios virtuales

Teniendo en cuenta la tabla anterior, este proyecto intentará adoptar aquellos aspectos más relevantes de cada uno de los laboratorios virtuales y a su vez evitar los problemas o dificultados que estos presentan.

2.2.1. Aplicaciones similares

Como se mencionó anteriormente, actualmente en el mercado se encuentran una gran variedad de laboratorios virtuales similares al creado en el proyecto. De todas formas es necesario destacar, que a la hora de elegir un modelo del que partir hubo una gran influencia por parte de los laboratorio virtuales “CLOUDLABS” [20] y “Kahoot” [19] ya que una mezcla de estas dos aplicaciones era lo que se tenía en mente inicialmente. Por un lado se pretendía desarrollar una aplicación que en el momento permitiera al estudiante saber si las decisiones tomadas eran correctas y dependiendo de como de rápido fuese el desarrollo se podría crear alguna funcionalidad que permitiera a los alumnos competir entre ellos, es esta parte la que se parece más a aplicaciones como “Kahoot”. La inspiración sacada de “CLOUDLABS” pertenece principalmente a la forma y estructura en la que crear un laboratorio virtual y como hacer que el estudiante seleccione cada material y reactivo.

La inspiración obtenida influyó principalmente a la forma en la que el proyecto se encuentra estructurado así como los objetivos necesarios para poder finalizar la actividad de forma correcta. Dicho esto, se intento no hacer uso excesivo de estas aplicaciones e intentar desarrollar el proyecto de una forma original donde solo se tomaría inspiración de estas aplicaciones en caso de duda o posible mejora de la estructura actual.

2.2.2. Laboratorios virtuales de años pasados

Cabe destacar que si bien esta es la primera vez que se realiza un proyecto colaborativo de este tipo, a la hora de planificar su desarrollo y organizar las secciones más importantes de la aplicación me inspiré en proyectos colaborativos de años pasados que poseían una estructura similar a pesar de tener un objetivo completamente distinto. Esto me ayudó a escoger un método de planificación que se ajustase a los objetivos del proyecto y programar una serie de objetivos a completar de forma progresiva.

2.3. Tecnologías

A la hora de crear el laboratorio virtual, se identifican una gran variedad de motores de videojuegos en los que poder desarrollarlo. Los principales motores gráficos que tenía en mente eran Unity, Unreal Engine, GameMaker Studio 2, Stencyl, Armory3D, Defold y Godot. A continuación se explican los motivos por los que cada uno de estos motores eran opciones validas para la realización del proyecto.

- ◆ El motor gráfico Unity [22] dependiendo de la licencia puede utilizarse de forma gratuita y al ser un motor gráfico muy popular en el mercado hay mucha documentación sobre este lo cual facilita la resolución de dudas o problemas. El echo de que sea muy popular viene acompañado de tener una comunidad muy activa a la hora de resolver dudas o querer optimizar el código. La curva de aprendizaje de este motor gráfico es muy fácil ya que entre otras cosas se puede utilizar C# como lenguaje de programación, lenguaje con el que estoy muy familiarizado. Por último, comentar que Unity ofrece una gran variedad de opciones en lo que respecta a contenidos en su tienda oficial ya sean scripts, música, texturas, etc. lo que haría más fácil el desarrollo de la aplicación.
- ◆ Unreal Engine [21] era otra de las opciones que llamaba la atención a la hora de desarrollar el proyecto ya que el desarrollador se encontraba familiarizado con su entorno al haberlo utilizado bastante en el pasado. Al igual que Unity, Unreal Engine puede utilizarse de forma gratuita y es considerado el motor gráfico más potente en el mercado actualmente. Muchas de sus ventajas son similares a las de Unity como por ejemplo la gran comunidad que tiene y facilidad a la hora de buscar documentación. Esto se debe a que tanto Unity como Unreal Engine son los dos motores de videojuegos más

populares actualmente en el mercado.

- ◆ GameMaker Studio 2 [23] es otro motor gráfico que me llamó la atención debido a lo simple que era su interfaz y facilidad de uso junto con una integración con git y otras herramientas internas. Pero decidí no utilizarlo debido a problemas como que su licencia gratuita es muy limitada y a no ser que decidiese pagar una suscripción me encontraría muy limitado a la hora de desarrollar el proyecto. Además solo soporta GML lo cual puede ser un problema en el futuro si poseo un código muy extenso limitando el uso de distintos lenguajes de programación. Por último comentar que su comunidad no es tan grande como los motores gráficos anteriores haciendo más difícil la búsqueda de soluciones.
- ◆ Stencyl [24], Armory3D [25] y Defold [26] eran motores que compartían algunas características con los anteriores pero decidí no utilizarlos debido entre otras cosas a su interfaz poco intuitiva, la poca experiencia que tenía con estos motores, limitación o incompatibilidad con ciertos lenguajes de programación y poca documentación disponible en algunos casos.
- ◆ Mi última opción era el motor gráfico Godot el cual nunca había utilizado en el pasado. Godot posee una gran flexibilidad a la hora de la creación y compaginación de distintas componentes además de ser completamente gratuito. Su documentación junto con su comunidad es muy extensa a pesar de no ser tan grande como las comunidades de Unity o Unreal Engine. Al ser un motor de código abierto y permite la programación en varios lenguajes de programación. Tiene también su propio lenguaje incorporado llamado GDScript el cual esta bastante simplificado y es fácil de entender.

Al final valorando todas las opciones, los principales candidatos era por un lado Unity y Unreal Engine debido a que el desarrollador del proyecto ya estaba familiarizado con estos motores y sabía que no tendría problemas a la hora de buscar documentación durante el desarrollo del proyecto. Por otro lado se encontraba Godot el cual atraía bastante por ser completamente gratuito, además era una buena oportunidad para poder familiarizarse con este motor el cual el desarrollador del proyecto había oído hablar de él muchas veces pero nunca tuvo oportunidad de probarlo.

Tras haber tenido en cuenta todas las ventajas e inconvenientes que cada motor presentaba se decidió desarrollar una aplicación con el motor gráfico Godot no solo por ser una buena oportunidad para aprender a utilizarlo sino también por que las posibilidades que presentaba y la ayuda de terceros que se podía obtener para facilitar el desarrollo del proyecto era muy extensa.



Figura 1: Logotipo de Godot Engine
[29]

2.3.1. Herramientas para la gestión del proyecto

Durante el desarrollo del proyecto se utilizaron aplicaciones externas como Trello para facilitar su seguimiento y planificación así como GitHub para almacenar copias de seguridad y sus distintas versiones. En el siguiente capítulo se explicarán las tecnologías y la metodología usada para la planificación del proyecto.

Capítulo 3

Gestión del proyecto y planificación

A continuación se describen las distintas metodologías adoptadas en el proyecto para facilitar su desarrollo y seguimiento de este. También se hablará de aplicaciones externas usadas para facilitar la coordinación de cada fase iterativa del proyecto.

3.1. Metodología de desarrollo del proyecto

Para la realización de este proyecto se utilizó un desarrollo incremental mediante la metodología Scrum. A continuación se explica por qué se tomaron estas decisiones de desarrollo y cómo se adaptaron al proyecto.

3.1.1. Desarrollo incremental

El desarrollo incremental es un tipo de desarrollo software donde inicialmente se desarrollará el producto cumpliendo los requisitos mínimos, una vez que se haya finalizado esta parte se irá refinando e incrementando su funcionalidad a medida que el proyecto va avanzando. La principal ventaja de este tipo de desarrollo es la constante disposición de un producto funcional y presentable al cliente el cual gracias a esto puede aportar retroalimentación más precisa y de forma más rápida.

La principal ventaja de haber adoptado este tipo de desarrollo frente a otros es la tranquilidad de saber que en el caso de que el cliente decidiese cambiar drásticamente la finalidad de algunos aspectos del proyecto siempre se contará con una versión funcional de este que cumple con un serie de requisitos mínimos.

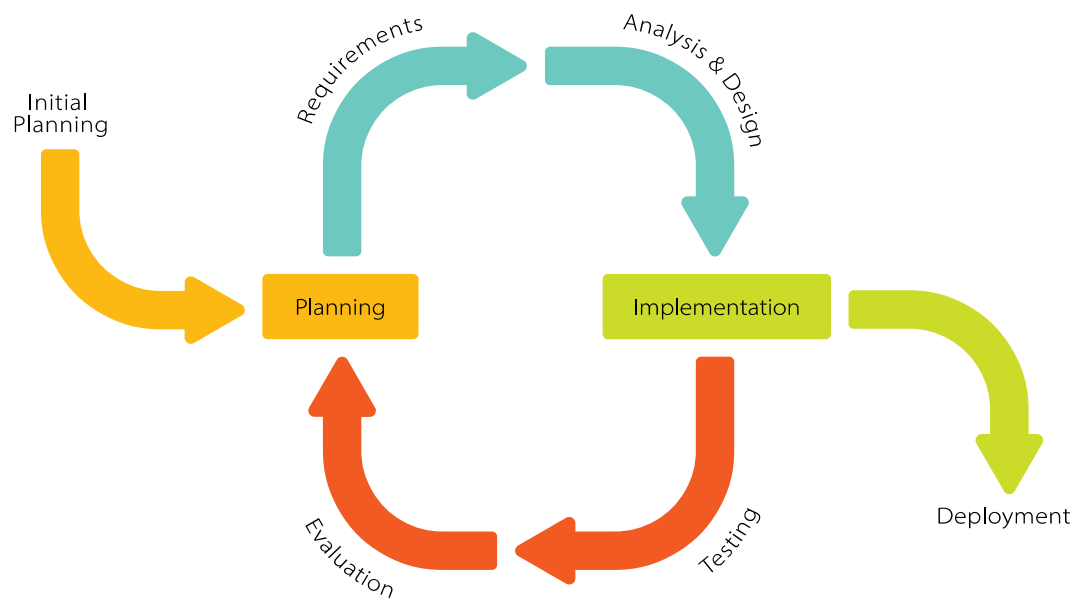


Figura 2: Representación del desarrollo incremental
[30]

Como el proyecto surgió a partir de una idea de crear una aplicación funcional para los alumnos la mejor estrategia a seguir sería crear primero un producto funcional con las historias de usuario más importantes implementadas. Una vez se tuviese esta aplicación funcional todavía en un estado muy básico y simple se iría añadiendo más funcionalidad y opciones más complejas a medida que el proyecto iba avanzando.

El principal objetivo de este tipo de desarrollo era el poder disponer de un producto funcional en todo momento. De esta forma si el cliente deseaba hacer añadidos al producto o alguna serie de cambios en su estructura, se dispondría siempre de un producto funcional con mayor o menor funcionalidad pero que cumple con los requisitos más importantes. Una vez los añadidos se hubieran finalizado, la versión anterior sería actualizada y el proceso se volvería a repetir. Es por esto que el desarrollo incremental era la opción más deseada para el desarrollo de este tipo de proyecto.

Las distintas fases que siguieron al desarrollo del proyecto fueron las siguientes:

Estado	Sprints
Aplicación con un menú principal y actividades	1, 2
Cada actividad es diferente de otra simulando una actividad específica	3, 4, 5
Seguimiento del profesor de cada actividad por el alumno	6, 7
Mayor detalle de las actividades	8
Aplicación funcional en Android	9

Tabla 2: Incremento del proyecto por sprints

3.1.2. Scrum

Scrum [6] es una metodología para el desarrollo ágil de software donde se aplican de manera regular una serie de buenas prácticas de desarrollo en equipo permitiendo así el mejor resultado posible a la hora de realizar proyectos.

El principal motivo por el que la metodología Scrum fue elegida frente a otras metodologías de desarrollo software fue la posibilidad que ofrece de realizar la planificación del proyecto a medida que la planificación va avanzando, dicha posibilidad no es posible en otras formas de desarrollo software más convencionales.

Es especialmente óptima para este tipo de proyecto ya que los clientes constantemente cambiaban las demandas y objetivos del proyecto, cambiando parte de la funcionalidad, añadiendo nueva e incluso eliminando funcionalidad que ya había sido implementada.

3.1.2.1 Historias de Usuario

La metodología Scrum anteriormente explicada hace uso de las Historias de Usuario para reflejar un requisito o aspecto funcional del proyecto. Estas Historias de Usuario están compuestas por:

- ◆ Un nombre con el formato *Como [rol] quiero [acción a realizar]*, el rol es el usuario o miembro del equipo que realiza la Historia de Usuario (ej. Usuario, Ingeniero de Software, Administrador...). Un ejemplo aplicable a este proyecto sería: Como alumno quiero poder visualizar mi progreso de forma individual en cada una de las actividades.
- ◆ Una prioridad que viene indicada por un número el cual indica que Historias de Usuario deben de realizarse antes que otras.

- ◆ El esfuerzo, representado como un número de la serie de Fibonacci el cual indica cuanto tomará implementar la historia de usuario (estos números no representan unidades de tiempo, son su propia unidad relativa a la planificación del proyecto).
- ◆ Un dueño el cual se asegura de que la Historia de Usuario a sido implementada cprrectamente.

A lo largo de todo el proyecto se hizo uso de dichas historias de usuario para facilitar la implementación y la distinción de los requisitos funcionales más importantes del proyecto.

3.1.3. Backlog

El backlog contiene una lista de todas las Historias de Usuario ordenadas por su prioridad . Esta lista será utilizada a la hora de asignar Historias de Usuario a cada uno de los Sprints del proyecto. El backlog fue actualizado a medida que los Sprints se iban completando y las historias de usuario con mayor finalidad eran implementadas en el proyecto.

3.1.4. Sprints

Un Sprint es cada uno de los ciclos del proyecto donde se realizarán las Historias de Usuario. Estos Sprints sirven para poder estructurar el proyecto de mejor forma y permitir la re-planificación a medida que el proyecto va avanzando.

Una vez se finalice un Sprint el siguiente paso a realizar será asignar las Historias de Usuario al Sprint siguiente. Además también se realiza una retrospectiva donde se aborda lo bueno, lo malo y las dudas o problemas que haya tenido el equipo durante la realización del Sprint. Por último también se indicará una reseña donde el equipo reflexiona de como se ha aplicado la metodología durante el desarrollo del Sprint y que se podría hacer para mejorar en cuanto al siguiente Sprint. En este último caso como el equipo de desarrolladores del proyecto estaba formado por una única persona este último apartado muy rellenado con una opinión personal de lo anteriormente mencionado indicando que aspectos del desarrollo se hicieron bien y cuales podrían haberse mejorado.

3.1.5. Burndown

Un Burndown es una gráfica donde el eje X representa el paso del tiempo en el Sprint mientras que el eje Y indica la cantidad de Historias de Usuario o tareas que quedan por realizar. Esta gráfica está compuesta por una recta decreciente la cual representa el flujo

de trabajo ideal (a medida que avanza el tiempo se van finalizando Historias de Usuario), y una serie de puntos conectados que representa la cantidad de trabajo restante en cada instancia de tiempo del eje X.

Esta gráfica muestra de forma visual como de bien o mal se ajusta el esfuerzo real al planificado, o en otras palabras, como de bien se esta avanzando en el proyecto respecto a lo que se ha planificado.

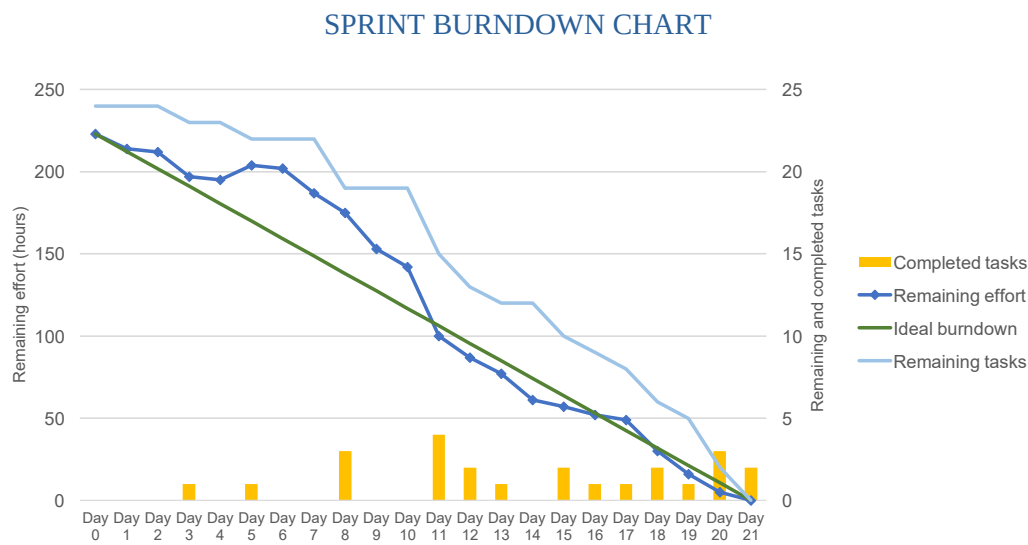


Figura 3: Ejemplo de una gráfica burndown [31]

Por lo general, a lo largo de la realización del proyecto se siguió el avance que se muestra en la imagen anterior siendo un desarrollo continuo con días en los que se avanzaba más que otros.

La siguiente figura muestra la planificación real que se adaptó en el proyecto:

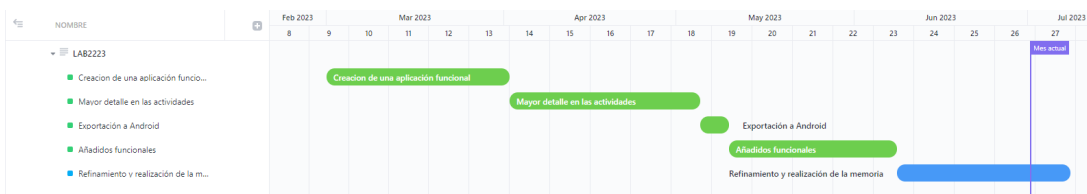


Figura 4: Diagrama de Gantt sobre la aplanificación real [32]

3.2. Planificación temporal

3.2.1. Brainstorming inicial

El proyecto comienza con una reunión con el tutor donde se proponen ideas relacionadas con la funcionalidad del proyecto, en esta reunión las ideas más interesantes son anotadas dando lugar a la siguiente tabla:

Idea	Prioridad
Desarrollo de un laboratorio virtual para la simulación de actividades	5
Entorno multijugador con diversas acciones	4
Modelo competitivo entre los jugadores	3
Compatible en ordenadores, móviles y tablets	2
Importar el proyecto a realidad virtual	1

Tabla 3: Ideas recogidas del brainstorming con el tutor

El catorce de febrero se programa una visita al centro donde se encuentran los clientes, en esta visita se hace un intercambio de ideas y posibles añadidos respecto a la funcionalidad del proyecto, las ideas más importantes son recogidas en la siguiente tabla:

Idea	Prioridad
Desarrollo de una aplicación con actividades para realizar de forma virtual	5
Permitir al estudiante seleccionar paso a paso el procedimiento a seguir	4
Permitir al profesor hacer un seguimiento del progreso del alumno	3
Compatible en ordenadores y móviles	2
Añadir una versión competitiva entre los alumnos	1

Tabla 4: Ideas recogidas del brainstorming con los clientes

3.2.2. Seguimiento

El proyecto comienza siendo desarrollado con las ideas principales en mente con el objetivo de tener cuanto antes una aplicación funcional que poder enseñarle al cliente. Una vez se ha desarrollado dicha aplicación todavía en un estado muy simple, esta le es mostrada al cliente el cual aportara su opinión junto con posibles ideas o mejoras que se le puedan hacer. A partir de este punto las reuniones con el cliente se irán haciendo de manera periódica (1 día a la semana) para una mejor coordinación y realización de los objetivos programados. Estas reuniones se realizaron de forma online con ayuda de Google Meet desde el día 17 abril hasta la finalización del proyecto el 12 de julio. Las reuniones se realizan desde las 10:00 hasta las 11:00 (11:30 en caso de alargarse por motivos externos) en la mayoría de los casos cada miércoles de la semana.

3.2.3. Retrospectiva

Observando como el proyecto se fue desarrollando a lo largo de los meses el principal punto que permitió su desarrollo rápido y exacto fueron las reuniones semanales con el cliente como guía para el refinamiento de dicho proyecto. Si bien en algunos casos parte de la funcionalidad que se había implementado tuvo que ser retirada del proyecto por lo general se consiguió un desarrollo incremental y coordinado entre los desarrolladores y el cliente.

Mas adelante en la sección de reuniones con el cliente se muestra cada reunión realizada indicando los aspectos tratados en cada una de ellas y los objetivos programados para la siguiente reunión.

3.3. Trello

La aplicación web Trello [12] fue una herramienta indispensable para la realización del proyecto. Trello permite gestionar el flujo de trabajo de un proyecto de manera visual mediante el uso de postits para permitir mayor facilidad al supervisar las tareas y coordinación. En este proyecto Trello fue utilizado a diario como un recordatorio de los principales objetivos a realizar aquel día.

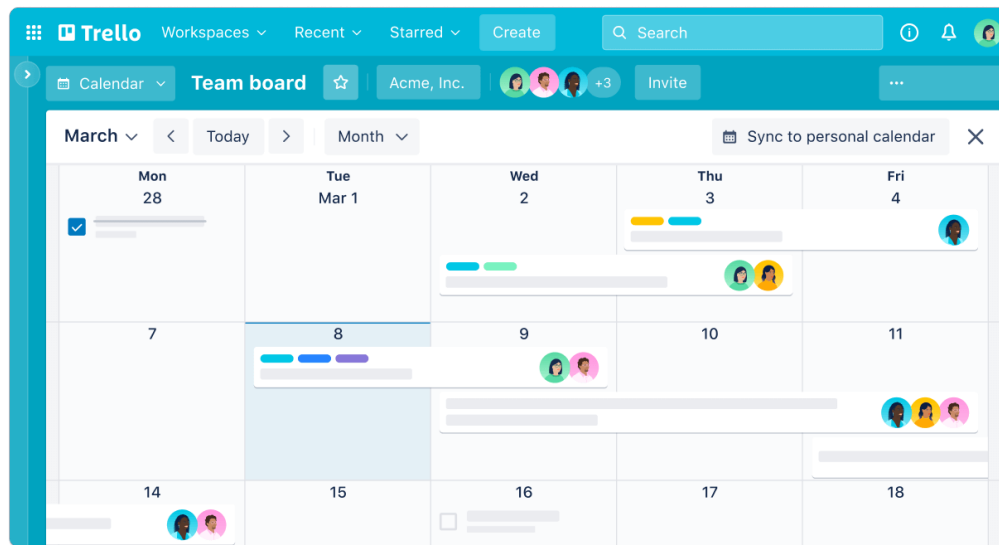


Figura 5: Entorno de trabajo de Trello
[12]

3.4. Estimación de coste

A continuación se explican los distintos recursos utilizados para el desarrollo del proyecto junto con una estimación del coste total de este.

3.4.1. Software

Se ha utilizado una gran cantidad de bibliotecas las cuales serán explicadas más adelante en la sección de código, además de esto se han utilizado los siguientes recursos:

- ◆ Godot Engine, valorado en 0€. Ya que es gratis para individuos y organizaciones.
- ◆ Windows 10 pro, valorado en 199,99€. Ya que el ordenador de sobremesa no contaba con un sistema operativo de fábrica
- ◆ Android One y otras versiones, valorado en 0€ ya que el precio viene incluido en el precio del dispositivo móvil,
- ◆ LibreOffice, valorado en 0€. Ya que es gratis para individuos y organizaciones.

- ◆ Trello, valorado en 0€. Ya que es gratis para individuos y organizaciones.
- ◆ Google Meet, valorado en 0€. Ya que es gratis para individuos y organizaciones.
- ◆ GitHub, valorado en 0€. Ya que es gratis para individuos y organizaciones.
- ◆ Visual Paradigm, valorado en 0€.

Recurso	Estimación del precio
Godot Engine	0€ (gratis)
Windows 10 pro	199,99€
Android One	99,99€
LibreOffice	0€ (gratis)
Trello	0€ (gratis)
Google Meet	0€ (gratis)
GitHub	0€ (gratis)
Visual Paradigm	0€ (gratis)
TOTAL	299,98 €

Tabla 5: Recursos software utilizados en el proyecto

3.4.2. Hardware

Para el desarrollo de este proyecto se ha utilizado unicamente un ordenador de sobremesa construido por partes valorado en 1300€, y un teléfono móvil Android sobre el que realizar las pruebas valorado en 299,99€. Obtenemos en esta sección un total de 1599,99€.

3.4.3. Recursos humanos

Los recursos humanos del proyecto son 2 grupos:

- ◆ El alumno Miguel Carracedo Rodríguez, como ingeniero de software, el cual se estima que ha invertido 47.5 sesiones a media jornada, es decir, 273.5 horas.
- ◆ El tutor como supervisor del proyecto,

Recurso	Estimación del precio
Ingeniero de software	19,31€/hora
Tutor del proyecto	1587€/mes
TOTAL	14.816,27 €

Tabla 6: Recursos humanos utilizados en el proyecto

3.4.4. Coste total

Teniendo en cuenta que el precio de un profesor en la UGR es de 1587€ al mes y el proyecto duro unos 5 meses obtenemos una estimación de 7935€.

Suponiendo que el salario medio por hora de un ingeniero de software en Andalucía es de 19,31€, se estima que el coste humano es de 5.281,28€, añadiéndole el resto de recursos utilizados el coste total de desarrollar el proyecto es de 6.881,27€ .

Recurso	Estimación del precio total
Software	299,98€
Hardware	1599,99€
Recursos humanos	14.816,27€
TOTAL	16.716,24 €

Tabla 7: Coste total del desarrollo del proyecto

3.5. Reuniones con el cliente

Las reuniones con el cliente se realizaron de forma semanal a partir de mediados de abril, fecha en la que se le pudo presentar el producto desarrollado con una funcionalidad mínima. En estas reuniones semanales se le presentaban los avances realizados en el proyecto así como las dudas que hubiesen surgido desde la semana anterior. Al final de cada reunión se indicaba en un documento los avances realizados junto con los objetivos programados para la siguiente semana. Estas reuniones comenzaban a las 10:00 de la mañana y solían durar entre 30 minutos y 1 hora.

A continuación se muestra una tabla resumiendo cada una de las reuniones con el cliente donde se indican los aspectos de la aplicación que se habían completado junto con que objetivos se intentarían completar para la semana siguiente.

Fecha	Aspectos explicados en la reunión	Aspectos programados para la siguiente reunión
21/4/2023	Explicación de las actividades realizadas y el punto de partida de la aplicación. Actividad 1 explicada	Mayor detalle en la sección de materiales así como una serie de cambios enviados por correo
8/5/2023	Explicación de los avances realizados y posibles añadidos futuros. Actividad 2 y 3 explicadas	Creación de una opción de ayuda y submenú en ciertas actividades
17/5/2023	Muestra de los añadidos solicitados en la anterior reunión y explicación de la actividad 4 y 5 explicadas	El cliente solicitó importar el proyecto a móvil, tablet y ordenador
24/5/2023	Demostración del funcionamiento en móviles Android. Actividades 6 y 7 explicadas	Mayor detalle general a la hora de explicar cada actividad
31/5/2023	Se explicaron los añadidos y mejoras realizadas desde la última sesión	Pequeños cambios de nomenclatura y cambios estructurales
7/6/2023	Exposición del estado final de la aplicación y aspectos finales que realizar	Cambio masivo de la estructura del proyecto con ideas imposibles de programar en el tiempo restante
14/6/2023	Explicación del funcionamiento de la aplicación e ideas para posibles cambios futuros	Pendiente de contactar para realizar posibles cambios en el futuro

Tabla 8: Reuniones semanales con el cliente

Estas reuniones se llevaron a cabo hasta finales de Junio fecha en la que el proyecto entro en un estado de finalización y refinamiento ya que había que realizar entre otras cosas la memoria del mismo.

3.6. Decisiones tomadas

A mediados del desarrollo del proyecto se observó como por motivos temporales algunos aspectos funcionales de la aplicación podrían no ser implementados a tiempo. Hubo entonces que tomar una decisión importante. Por una parte se podría diseñar las actividades más importantes con un gran nivel de profundidad y complejidad que permitirían al alumnos comprobar sus conocimientos con más detalle pero solo para tres de las nueve actividades programadas. La otra opción era simplificar todas las actividades haciendo que el estudiante pudiese comprobar sus conocimientos en menor medida que antes pero con una variedad de actividades mayor. Tras pensarse y discutirlo con el cliente se llegó a la conclusión de que la segunda opción era la más favorable ya que esta a pesar de no tener un nivel de complejidad muy alto seguía permitiendo a los alumnos repasar los conceptos de todas las actividades explicadas en clase.

En conclusión las decisiones tomadas permiten al estudiante simular una gran variedad de actividades a diferencia de tener un número reducido de actividades donde la complejidad puede confundir al usuario.

3.7. Uso de GitHub y copias de seguridad

El uso de GitHub durante el desarrollo del proyecto fue mínimo ya que se utilizó únicamente para actualizar el proyecto cada vez que una nueva versión había sido refinada y estaba libre de errores. La estructura estaba formada por un repositorio donde se iba subiendo el código del proyecto a medida que este avanzaba. Su estructura era una única rama master de la que partía todo y en pocas ocasiones se creaban ramas individuales temporales para el almacenamiento de ideas futuras.

En cuanto a las copias de seguridad además de tener el propio proyecto en el ordenador local guardado, también se tenía todo el código subido a GitHub y una copia en un disco duro externo con una copia exacta del proyecto en el ordenador local que era actualizada cada vez que se finalizaba una sesión de desarrollo. Esto fue de gran ayuda ya que durante el desarrollo del proyecto debido a conflictos con el sistema de almacenamiento en la nube OneDrive se borró prácticamente todo el contenido relacionado al proyecto que se encontraba en el ordenador local. Gracias a las copias de seguridad externas en cuestión de minutos se volvía a disponer del proyecto en el ordenador local sin haber perdido tiempo

volviendo a implementar las últimas versiones.

Capítulo 4

Diseño

4.1. Diseño general

El diseño del proyecto ha ido desarrollándose a medida que este se iba implementando. Esto quiere decir que su diseño no ha sido reflejado en un diagrama antes de implementarlo ya que al haber sido guiado por las historias de usuario y las reuniones con el cliente estos diagramas iban siendo modificados constantemente según avanzase su implementación.

La aplicación a desarrollar consiste en un laboratorio virtual donde el estudiante realizará distintos tipos de actividades. Inicialmente se encontrará en el menú principal donde se le mostrarán todas las actividades a realizar. A partir de ahí el estudiante podrá decidir que actividad desea hacer primero. Cada actividad se dividirá en distintas secciones, una para seleccionar el material y los reactivos, otra para mezclarlos y otra para realizar distintas reacciones con ellos. Una vez el estudiante crea haber finalizado la actividad de forma correcta podrá comprobarlo pulsando un botón. Tras esto se le revelará el resultado obtenido y en caso de haber realizado mal la actividad se le indicará aquellos aspectos en los que ha fallado.

Una vez el estudiante haya finalizado una actividad podrá volver a realizarla si así lo desea o simplemente puede pasar a realizar otra actividad. En el menú principal también habrá un botón que permita cerrar la aplicación.

4.2. Paquetes

La división de módulos del proyecto se ha hecho de tal manera que cada funcionalidad de este está desacoplada lo máximo posible del resto. De esta manera facilitamos la inyección de nueva funcionalidad al módulo sin la posibilidad de que entre en conflicto con otras.

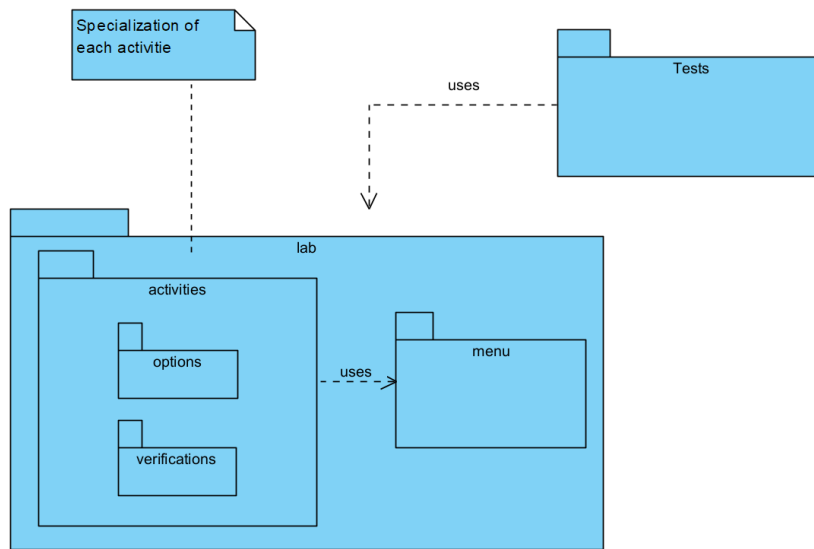


Figura 6: Diagrama de paquetes
[1]

4.3. Clases

En el diagrama de clases mostrado a continuación, se puede observar como se tiene una clase central Actividad de la que heredan el resto de clases hijas. Esta clase principal contiene las variables globales y estructura base de la que parten el resto. De esta forma solo habrá que implementar una pequeña parte del código para cada actividad ya que las funciones principales se encuentran implementadas en la clase padre.

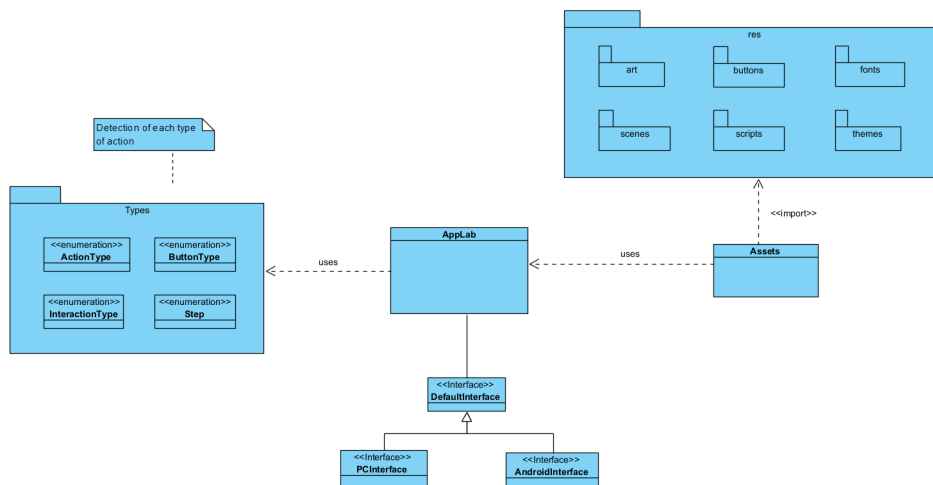


Figura 7: Diagrama de clases simplificado
[1]

A continuación se indican las clases del proyecto y las funciones principales de cada una de ellas:

- ◆ Botones: Esta clase contiene la configuración y comportamiento de cada uno de los botones del proyecto. Estos botones pueden ser los que te permiten moverte entre las actividades y el menú principal o simplemente botones para seleccionar un material.
- ◆ Fondos: Esta clase se encarga de controlar que fondo de pantalla aparecerá en cada momento de la actividad. Por ejemplo a la hora de cerrar un menú y abrir otro esta clase será ocultando un fondo de pantalla y hará visible otro.
- ◆ Comprobar: Se encarga de realizar comprobaciones a la hora de verificar si el resultado es correcto. Todas las actividades del proyecto llamará a funciones de esta clase cuando se quiera comprobar si ya actividad se ha realizado de forma correcta. También es utilizada para comprobar de fondo el nivel de progreso en el que el estudiante se encuentra en una actividad.
- ◆ Singleton: Esta clase contiene una serie de variables globales que se llaman cuando queremos realizar mezclas entre distintos reactivos.
- ◆ `_pressed()`: Esta no es una clase, si no una función llamada en todas las actividades la cual diferenciará cada tipo de botón ya que leerá su identificador asociado para saber cual es.

4.4. Caso de uso

La aplicación está implementada como una secuencia linear en la que el usuario irá seleccionando las distintas opciones que considere correctas hasta llegar a el resultado final. Una vez obtenido este resultado el usuario podrá comprobar si es correcto.

A continuación se muestra un caso de uso en desde que el estudiante entra en la aplicación hasta que finaliza una actividad :

1. Seleccionar la actividad que se desea realizar desde el menú principal.
2. Tras haber leído el enunciado ir a la siguiente sección donde se realiza la actividad.
3. Una vez en esta sección, se selecciona el material y los reactivos que se consideren necesarios.
4. Tras esto los mezclamos según se indica en el procedimiento a seguir.
5. Una vez obtenida la mezcla final podemos pulsar el botón de comprobar el cual nos indica si la actividad se realizó de forma correcta.
6. Por último el usuario tiene las opciones de volver a realizar la actividad o pasar a la siguiente.
7. En caso de querer continuar más tarde el estudiante puede cerrar la aplicación desde el menú principal.

4.5. Conclusiones y decisiones tomadas

Durante todo el desarrollo del proyecto hubo que tomar decisiones relacionadas con el diseño de la aplicación. Inicialmente se estableció una estructura para el menú principal y la forma de acceder a cada actividad de forma individual al igual que la manera de volver de vuelta al menú principal. Esta estructura inicial le gustó al cliente y quedó implementada sin ser modificada hasta que se obtuvo el producto final.

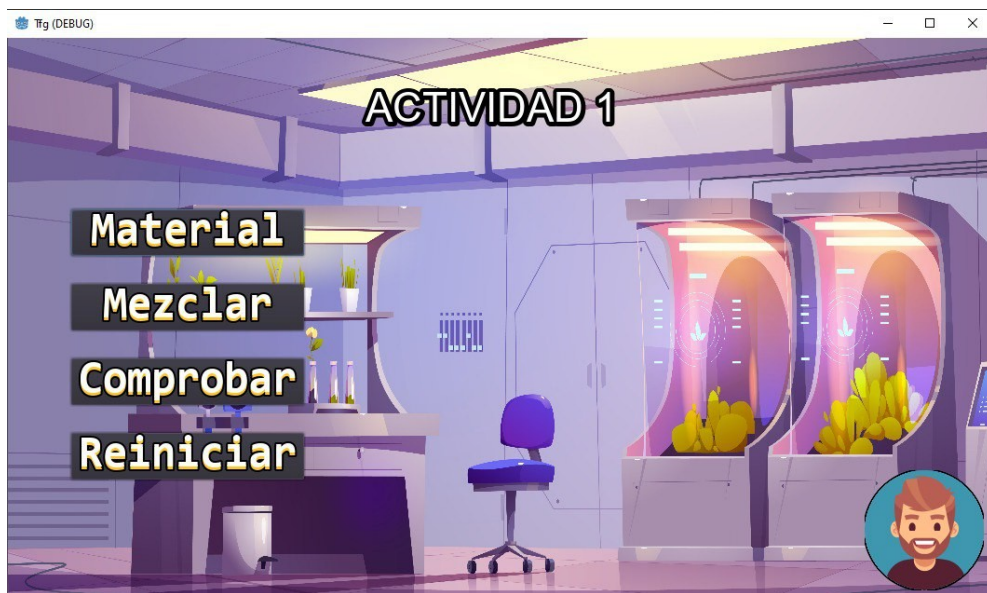


Figura 8: Actividad 1 en la etapa inicial del proyecto
[9]

La siguiente parte del proceso consistió en decidir como estructurar cada actividad. Esta parte tuvo muchas alteraciones a lo largo del desarrollo de la aplicación. Con cada iteración se fue creando un interfaz más intuitiva y fácil de navegar. A su vez el cliente fue aportando su opinión para facilitar la forma en la que se diseñaba la estructura de cada actividad.



Figura 9: Selección del material en la etapa inicial del proyecto
[9]

Capítulo 5

Implementación

5.1. Godot Engine

El desarrollo de esta aplicación se hizo mediante el uso del motor gráfico Godot como se comentó anteriormente. Hay varias versiones de Godot Engine pero la que se utilizó para la realización del proyecto fue la más reciente en el momento siendo esta Godot Engine 3.5. Como se comentó anteriormente, este motor es completamente gratuito por lo que se hizo uso completo de las características que ofrecía.

A pesar de no estar familiarizado con este motor y su interfaz, el aprendizaje de este fue relativamente sencillo ya que la interfaz era muy intuitiva y se disponía de una gran cantidad de documentación en internet, la cual se podía consultar en caso de dudas a la hora de implementar algún tipo de funcionalidad en Godot Engine.

Además Godot Engine posee una estructura formada por un árbol de escenas donde cada una de estas escenas está formada por un árbol de nodos. Dichos nodos permiten una gran variedad de funciones y se especializan según la funcionalidad deseada.

5.2. GDScript

GDScript es el lenguaje que Godot Engine utiliza para la creación de escenas tanto 3D como 2D y las posibles interacciones que se hagan en estas. También soporta lenguajes como C# y Java pero como el desarrollo en estos dos lenguajes de programación podría dar lugar a complicaciones futuras a la hora de conectarlos se decidió utilizar GDScript en la totalidad del proyecto ya que es un lenguaje de programación cercano a C# y muy fácil de aprender.

Este lenguaje no supuso ningún tipo de complicación durante el desarrollo del proyecto ya que era muy similar a otros lenguajes con los que me encontraba familiarizados y era más sencillo que la mayoría permitiendo programar cosas que en otros lenguajes de programación no me habría sido posible por problemas de dependencias.

5.3. Estructura de las actividades

Al iniciar la aplicación el usuario se encontrará con un menú principal. Este menú principal es muy simple mostrando únicamente 9 botones pertenecientes a cada una de las actividades que se pueden realizar así como un botón que cierra la aplicación. Se puede navegar por estos botones mediante el teclado, el ratón e incluso gamepad en el caso de que este estuviese conectado.

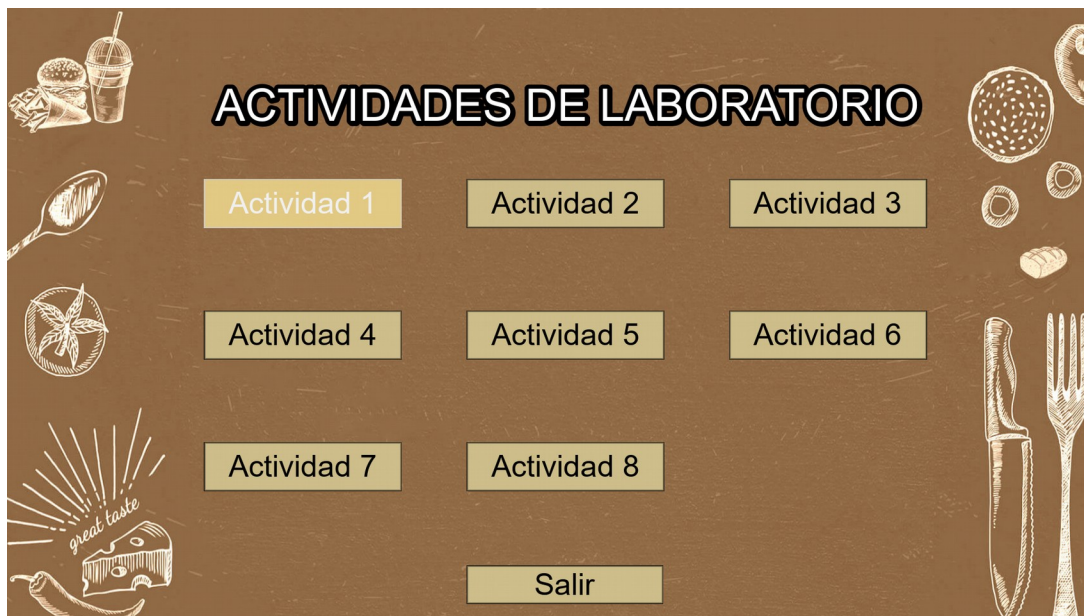


Figura 10: Menú principal de la aplicación
[9]

Una vez se haya seleccionado la actividad que se desea realizar, la pantalla cambiará y pasará a mostrarnos el título de la actividad seleccionada, una breve explicación de este y una serie de botones que se explican a continuación.

El botón de ayuda situado en la esquina inferior derecha sirva para ayudar al estudiante recordar el procedimiento que hay que realizar para completar la actividad de manera satisfactoria. Este botón se implementó con la idea de que tal vez el enunciado de la actividad podría no ser suficiente para ayudar al estudiante a realizar la actividad. De todas formas la ventana desplegable que se muestra una vez se haya pulsado el botón solo será visible en esta escena por lo que el estudiante no la tendrá presente a la hora de realizar las mezclas necesarias evitando de esta manera hacer la actividad demasiado sencilla ya que solo habrá que copiar y pegar lo que se indica en el apartado de ayuda.

Los tres botones principales que se encuentran en el centro de la pantalla son los de Material, Reiniciar y Comprobar. Esos dos últimos botones como su nombre indica son

usados para reiniciar la actividad en caso de que el estudiante se encuentre perdido y no sepa en que paso se encuentra o si desea volver a realizar la actividad para adquirir más destreza. El botón de Comprobar es usado para verificar si el procedimiento seguido es el correcto y la actividad a finalizado de manera correcta. El botón de Material llevará a una nueva ventana donde se realizará la actividad.

En la esquina inferior izquierda también se encuentra situado una fecha hacia atrás utilizada para volver al menú principal si el usuario lo desea. De todas formas si esta fecha es pulsada el progreso realizado en la actividad se perderá y el usuario tendrá que volver a realizarla de nuevo si no está finalizada.

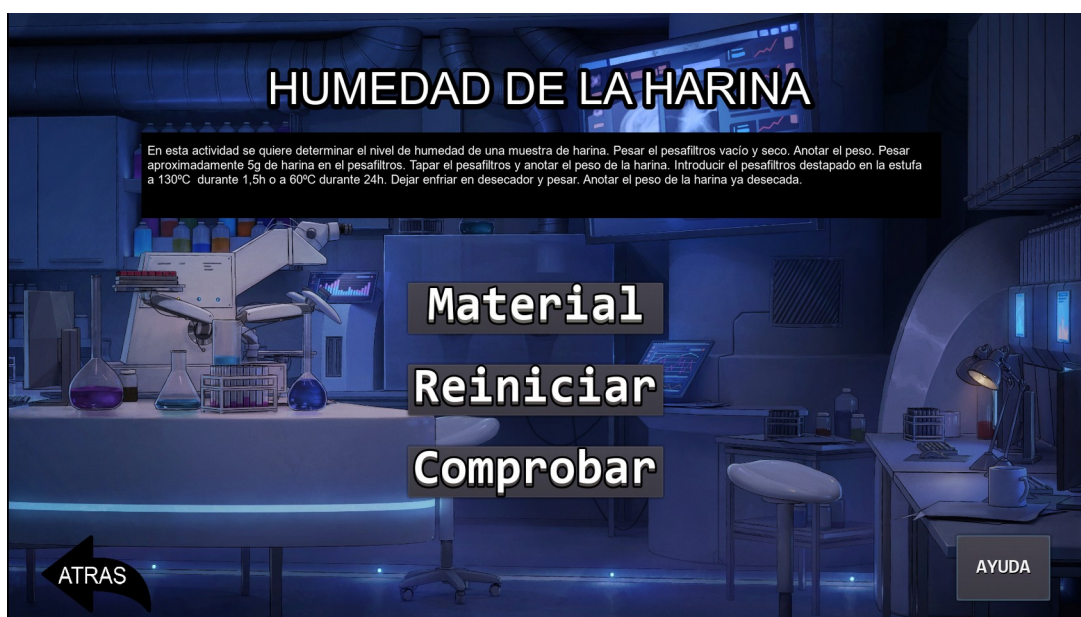


Figura 11: Pantalla inicial de cada actividad
[9]

Si se hubiese pulsado el botón de Material accederíamos a la siguiente pantalla:



Figura 12: Sección de materiales de cada actividad
[9]

Como se puede observar en la imagen anterior disponemos de tres botones en la parte izquierda de la pantalla y una serie de materiales a seleccionar. También se disponen del botón Comprobar y la fecha para ir hacia atrás, ambos con la misma función que en la escena anterior.

Esta escena cambiará dependiendo de la actividad que se esté realizando ya que el material para cada actividad será distinto y las opciones de que hacer con este material también lo serán. Para no hacer extremadamente largo la explicación del funcionamiento de cada actividad utilizaremos la imagen anterior como ejemplo.

A la derecha aparecen una serie de materiales de los que el estudiante debe seleccionar aquellos que considere relevantes para la realización de la práctica. Cada vez que se haya seleccionado un material se abrirá una ventana donde se pregunta con que sustancia se quiere rellenar, el estudiante indicará que sustancia utilizará seleccionándola en el menú desplegable. Cuando se quiera seleccionar otro material se deberá clicar en el botón Nuevo material y el proceso se repetirá. Cuando todo el material se haya seleccionado el estudiante podrá utilizar este material y experimentar en el con los otros dos botones, los cuales variarán dependiendo de la actividad.

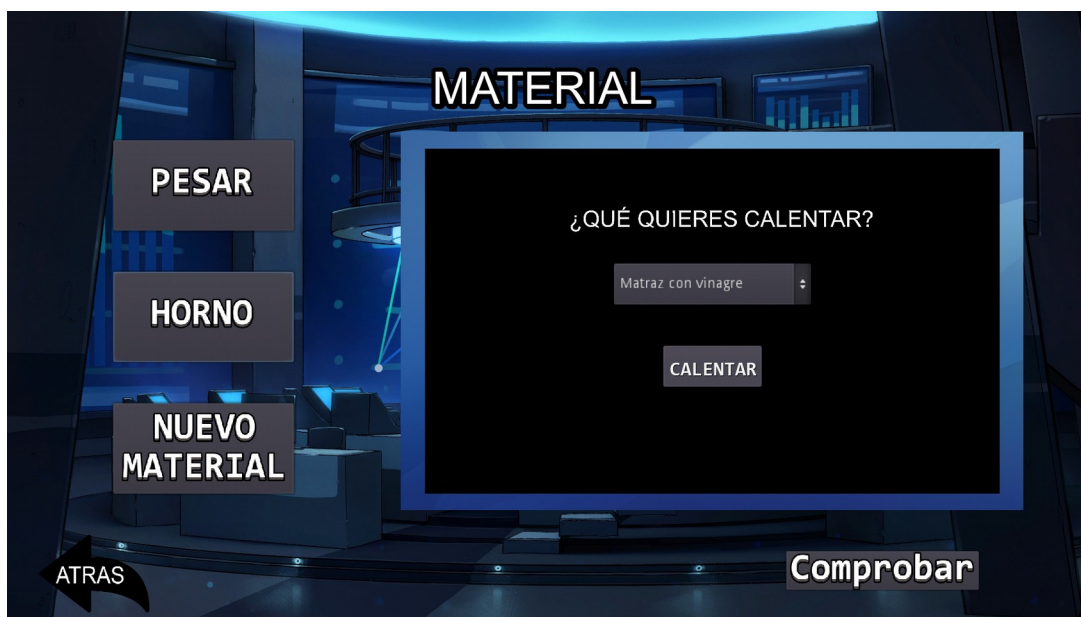


Figura 13: Sección de calentar de la actividad 1
[9]

Una vez se hayan realizado todos los pasos de la práctica el estudiante clicará en el botón Comprobar y el resultado obtenido aparecerá en pantalla. Si el resultado no es el deseado y la práctica se realizó de manera errónea, un mensaje en rojo le pedirá al estudiante que lo intente de nuevo. En caso contrario aparecerá un mensaje de felicitación en verde. Esta última acción se verá reflejada en el menú principal mostrando aquellas actividades que se han realizado de manera errónea y aquellas que se han finalizado correctamente.



Figura 14: Indicadores del progreso de cada actividad
[9]

De esta manera el profesor podrá comprobar el progreso de cada estudiante de forma individual.

5.4. Bibliotecas

Durante el desarrollo del proyecto se usaron una gran variedad de bibliotecas de código abierto. Algunas de ellas se dejaron de usar según el proyecto avanzaba ya que se encontraron mejores maneras de implementar el código que haciendo uso de estas.

También se hizo uso de la función que posee Godot conocida como AssetLib la cual permite descargarse plugins directamente desde Godot. Estos plugins fueron creados por otros desarrolladores y se encuentran en la página oficial de Godot para su libre descarga al igual que en la sección AssetLib del motor. Esta función contiene una serie de librerías muy específicas que se usaron para pequeños cambios en la funcionalidad del proyecto.

Por último comentar que en caso de que se hiciese el proyecto en un lenguaje externo como C# para su correcta integración en el motor gráfico habría que hacer uso de librerías como GDNative la cual permite la integración de C# en el programa.

5.6. Estructura del proyecto

El proyecto se encuentra dividido en una serie de carpetas y módulos donde se almacena distinto tipo de datos ya sean códigos, texturas u de otro tipo. A continuación se indican cada una de las carpetas del proyecto así como la función y el contenido de cada una de ellas:

- ◆ **Res:** Esta es la carpeta padre de la que se derivan el resto de carpetas, ella está todo el proyecto y es creada por defecto por Godot para almacenar toda la información una vez se ha creado un proyecto nuevo.
- ◆ **Art:** Se encuentra situada como hija de la carpeta res (padre). La carpeta art contiene todas las texturas, fondos de pantalla y otras imágenes utilizadas en el proyecto así como una serie de animaciones y efectos utilizados al inicio de la aplicación para simular una transición progresiva de la pantalla de inicio al menú principal.
- ◆ **Buttons:** Esta carpeta es descendiente de la carpeta padre. En ella se almacenan distintas configuraciones para los botones utilizados en la aplicación variando su forma, tamaño, efecto al ser pulsado, etc.
- ◆ **Fonts:** Como su nombre indica aquí se guardan todas las fuentes que son utilizadas en la aplicación. Esta carpeta es pequeña ya que en todo el proyecto solo se utilizan dos tipos distintos de fuentes.
- ◆ **Scenes:** Aquí se guardan cada una de las escenas de la aplicación. Cada escena es un ventana a la que el usuario podrá acceder dependiendo de las opciones que seleccione. Estas escenas poseen muchos estados distintos en los que dependiendo del progreso realizado se visualizarán una serie de elementos relacionados con el punto en el que el usuario se encuentre en la actividad.
- ◆ **Scripts:** Esta carpeta esta muy relacionada con la carpeta anterior ya que la mayoría de las escenas tienen un script asignado donde se codifica lo que ocurre con los distintos elementos de la escena (botones, animaciones, paso del tiempo). Estos scripts se encuentran programados en GDScript y su longitud variará dependiendo de la funcionalidad de la escena a la que corresponden. Como ejemplo el script asignado a la escena del menú principal es muy sencillo ya que esta escena es solo una serie de botones con las distintas actividades a realizar. Por otra parte el script asignado a la escena de la Actividad 5 es muy complejo ya que esta actividad es muy larga y requiere que el usuario realiza una serie de pasos para realizarla de forma correcta.
- ◆ **Themes:** Esta es la última carpeta de este proyecto. En ella se almacenan las variaciones que se realizaron en cada uno de los textos de la aplicación. A la hora de escribir un texto que se quiere que aparezca por pantalla primero se debe de asignar una fuente a este texto (situada en la carpeta Fonts), una vez hecho esto existe la opción de asignarle un tema a esta fuente. Este tema puede ser uno previamente creado o uno completamente nuevo. Cada tema consiste en modificar la fuente asignada al texto y darle por ejemplo más fondo o más sombra

además de poder modificar su tamaño y su interacción con la escena.

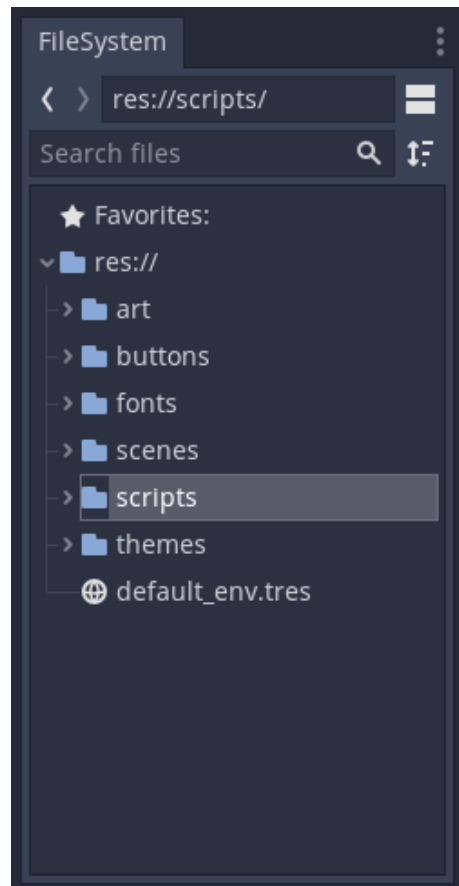


Figura 15: Árbol de directorios del proyecto

[9]

5.7. Exportación de la aplicación

Al estar haciendo uso del motor gráfico Godot, una vez se había finalizado de implementar el código y no se detectaba ningún tipo de error en este lo único que había que hacer era compilarlo y automáticamente Godot Engine se encargaría del resto abriendo una ventana que simularía la pantalla del dispositivo seleccionado. Una vez abierta esta pantalla la aplicación se lanzaría automáticamente y se podrían realizar las pruebas que fuesen necesarias para comprobar su correcto funcionamiento.

5.8. Código

Para hacer el código más extensible a la hora de programar y evitar la redundancia, el código se ha programado con la idea de crear una serie de clase hijas pertenecientes a cada una de las actividades que heredarán de una clase padre con la gran parte de la funcionalidad implementada. De esta forma las clases hijas solo tendrán que contener la información específica para esa actividad mientras que toda la funcionalidad general se encontrará implementada en la clase padre.

También se ha diseñado un script aparte para el menú principal el cual contiene funciones como la conexión a cada un de las actividades, un transición a la hora de iniciar la aplicación, etc. Por último también se ha de mencionar el script Singleton.gd el cual contiene variables globales necesarias para la comprobación de esta de cada actividad e indicar si una actividad se ha realizado correctamente.

Estos scripts se encuentran en la carpeta “scripts” dentro del proyecto en Godot y exceptuando el script padre, el cual contiene una gran cantidad de funcionalidad, el resto poseen poco código ya que utilizan la función de herencia al máximo.

A screenshot of the Godot code editor showing a script. The script is written in GDScript and is part of a class that extends Node2D. It defines a function _ready() which contains several lines of code: it plays a 'Fade in' animation, yields for 3 seconds, plays a 'Fade out' animation, yields for 2 seconds, and then changes the scene to 'res://scenes/MenuPrincipal.tscn'. The code is as follows:

```
1 extends Node2D
2
3 func _ready():
4     $AnimationPlayer.play("Fade in")
5     yield(get_tree().create_timer(3), "timeout")
6     $AnimationPlayer.play("Fade out")
7     yield(get_tree().create_timer(2), "timeout")
8     get_tree().change_scene("res://scenes/MenuPrincipal.tscn")
9
```

Figura 16: Código de transición al menú principal
[9]


```

1  | extends Button
2
3  export var reference_path = ""
4  export(bool) var start_focused = false
5
6  func _ready():
7  > | if(start_focused):
8  > | | grab_focus()
9  > | |
10 > | connect("mouse_entered",self,"_on_Button_mouse_entered")
11 > | connect("pressed",self,"_on_Button_Pressed")
12
13 > func _on_Button_mouse_entered():
14 > | grab_focus()
15
16 > func _on_Button_Pressed():
17 > | if(reference_path != ""):
18 > | | get_tree().change_scene(reference_path)
19 > | else:
20 > | | get_tree().quit()
21

```

Figura 17: Código de salto a una escena mediante un botón
[9]

El resto del código implementado en las clases hijas es muy simple, siendo únicamente una serie de condiciones booleanas y botones que ofrecen una variedad de opciones al ser pulsados por el usuario.

Capítulo 6

Pruebas

En esta sección se hablará de las distintas pruebas realizadas a la aplicación antes de ser presentada al cliente para asegurarse de su correcto funcionamiento y estructura.

A continuación se muestra una tabla con todas las pruebas realizadas en el proyecto junto con el objetivo y resultado de cada una de ellas:

Fecha	Objetivo	Resultado
01/03/2023	Prueba de la estabilidad general de la aplicación	Necesario mejorar el menú principal
14/03/2023	Comprobación de todas las posibles combinaciones de reactivos en las actividades 1 y 2	Todos las mezclas se realizan correctamente
30/03/2023	Prueba de navegación en el menú principal a cada una de las actividades	Fallo encontrado al acceder a la actividad 3. Pendiente de arreglar.
7/04/2023	Prueba de las especificaciones de material de la actividad 2	Pequeños fallos a la hora de seleccionar las imágenes, arreglados en el momento
13/04/2023	Comprobación de todas las posibles combinaciones de reactivos en las actividades 3,4,5 y 6	Todo funciona correctamente
2/05/2023	Prueba realizada sobre las distintas opciones de ayuda de cada actividad	Fallos de sincronización entre la imagen de fondo y el texto. Pendientes de arreglar.
30/05/2023	Prueba general del resultado obtenido en cada actividad reflejándose en el menú principal	Desalineación en las actividades 4, 5 y 6 a la hora de mostrar los resultados arreglados en el momento
14/06/2023	Comprobación de todas las posibles combinaciones de reactivos en las actividades 7 y 8	Todo funciona correctamente
30/06/2023	Prueba general de la aplicación en su totalidad y con todas las posibles combinaciones	Pequeños fallos de redireccionamiento arreglados en el momento

Tabla 9: Pruebas realizadas a lo largo del desarrollo del proyecto

6.1. Tests

A la hora de realizar los tests o pruebas para comprobar que la aplicación funciona correctamente inicialmente la persona encargada de realizar la aplicación comprueba que funciona como se espera al realizar alguna de las actividades. En caso de que no se haya detectado ningún problema el siguiente paso será probar la aplicación con conocidos y personas que no se encuentren familiarizadas con como está diseñada la aplicación. Si este grupo de personas tampoco encuentra ningún tipo de problema se guardará el estado actual de la aplicación con su versión correspondiente y se pasará a desarrollar la siguiente funcionalidad.

En el caso de que a la hora de ser testeada por los desarrolladores se encuentre algún tipo de bug o problema de navegación, este se intentará resolver en el momento teniendo la máxima prioridad. Si las personas que detectan el problema no se encuentran familiarizadas con la aplicación primero se analiza que tipo de problema han encontrado. En caso de ser un bug o fallo de la aplicación se realizará lo mismo que antes intentando resolverlo lo antes posible. En el caso que el problema sea de navegación o fallo para entender el camino a seguir, se anotará en un blog de notas para esta sección y posteriormente se analizará y decidirá que cambios son necesarios para hacer la aplicación más intuitiva.

6.2. Betas y pruebas con el cliente

Cuando se crea haber avanzado lo suficiente como para enseñarle al cliente el resultado de los cambios realizados a la aplicación se organizará una reunión con este. En la reunión se utilizará la última versión funcional de la aplicación (beta) la cual servirá para mostrar de forma visual al cliente las decisiones tomadas y cambios realizados desde la última reunión con este.

Una vez finalizado el ejemplo demostrativo del funcionamiento de la aplicación a los clientes, estos aportarán retroalimentación de posibles cambios o problemas que hayan detectado así como correcciones a la nomenclatura en caso de haberlas. Todos estos cambios serán añadidos a la lista de actividades pendientes de realizar con una prioridad muy alta y posteriormente serán implementadas y mostradas en la siguiente reunión con el cliente.

6.3. Configuración para ejecutar la aplicación

Para ejecutar la aplicación y poder realizar las pruebas en el ordenador es necesario tener la versión de prueba de Godot la cual nos permite ejecutar proyectos externos sin la necesidad de tener la versión completa de Godot instalada. En caso de querer realizar pruebas en el móvil Android se utilizó el móvil de la persona que lo desarrollaba activando las opciones de desarrollador en este para poder probar aplicaciones externas.

Por lo general las pruebas en el ordenador se realizaron de manera más rápida que en el móvil debido a la cantidad de tiempo que tomaba la corrección de errores y volver a compilar el proyecto para simularlo en Android.

Capítulo 7

Conclusiones y trabajos futuros

7.1. Conclusiones

Una vez finalizada la aplicación, mi idea general era que esta era muy simple y había conseguido implementar la funcionalidad mínima para que fuera un producto presentable. Pero según fui creando este informe y comprobando la cantidad de tiempo invertido junto con los objetivos iniciales y los finales llegué a la conclusión de que el desarrollo no había sido lento o mal programado y que teniendo en cuenta las problemas que tuve durante la realización de la aplicación y el tiempo invertido en esta el producto final que se había obtenido no era tan simple como pensaba. De todas formas sigo pensando que había una gran cantidad de funcionalidad que mejoraría la aplicación considerablemente y que no pude realizar por problemas de tiempo.

Este Trabajo de Fin de Grado no solo me ha permitido desarrollar una aplicación funcional simulando las condiciones de trabajo del mundo laboral si no que también me ha permitido adquirir experiencia con los clientes mediante las reuniones periódicas que tuve con estos en las que a medida que el tiempo fue avanzando fui adquiriendo más soltura a la hora de explicar los objetivos y problemas encontrados.

En cuanto a lo aprendido en este proyecto no solo adquirí experiencia para el mundo laboral, también aprendí a manejar el motor gráfico Godot y codificar en GDScript lenguaje que a pesar de no ser utilizado fuera de este motor gráfico me pareció muy fácil de entender y en las etapas finales del proyecto me resultaba muy cómodo programar con este. También aprendí a adaptar mi forma de trabajo al tener que planificar y registrar las horas dedicadas al proyecto en la hoja de cálculo de Scrum, hacer revisiones de los Sprints y utilizar Trello entre otras aplicaciones para facilitar el seguimiento del desarrollo de la aplicación.

Si bien inicialmente los objetivos programados para el proyecto eran demasiados no haciendo posible la implementación de todos estos, se ha obtenido un producto muy útil que será utilizado en el ámbito docente por profesores y alumnos en los siguientes años. Gracias tanto a la constancia de mi tutor Juan José Ramos Muñoz, como a la mía, he sido capaz de desarrollar un producto que a pesar de no contar con toda la funcionalidad que deseaba, sigue siendo un producto bastante refinado.

7.2. Trabajo futuro

A continuación se muestran una serie de apartados con los posibles añadidos que se le podrían realizar al proyecto en los meses posteriores a la finalización de este. Estos añadidos son en su mayor parte una mejora de la funcionalidad de la aplicación y añadidos a ciertas actividades:

1. Una opción que muestre de forma detallada en qué paso de la actividad se encuentra el estudiante, así como qué pasos ya ha realizado y cuáles le quedan por realizar.
2. Un cronómetro que le indique cuánto tiempo tardó en realizar la actividad y en caso de decidir volver a realizarla en un periodo menor de tiempo que este nuevo tiempo quede registrado borrando el anterior. De esta manera los alumnos podrían competir por quien resolvió la actividad de forma más rápida haciendo esta un poco más atractiva.
3. Añadir las actividades que en el último momento el cliente decidió omitir ya que nos les daría tiempo a darlas en clase a pesar de que ya se encontraban implementadas en la aplicación.
4. Tener una opción de valoración de la aplicación así como posibles mejoras a realizar (esto podrían ser correos de forma anónima a una base de datos).
5. Añadir una serie de animaciones que hagan que cada actividad sea un poco más entretenida de realizar.
6. En general hacer toda la aplicación un poco más detallada tanto a nivel visual como en las explicaciones que se ofrecen.

Bibliografía

- [1] Visual Paradigm, Agile project tools to improve productivity, <https://www.visual-paradigm.com/> , 9/07/2023.
- [2] GitHub, Let's build from here, <https://github.com/>, 11/07/2023.
- [3] Github-Precios, Get the complete developer platform, <https://github.com/pricing>, 5/04/2023.
- [4] Asus, Asus gaming laptops, <https://www.asus.com/es/laptops/for-gaming/tuf-gaming/asus-tuf-gaming-f15-2023/>, 8/04/2023.
- [5] Indeed, Salario-medio de un ingeniero de software en Andalucía, <https://es.indeed.com/?from=gnav-title-webapp> , 27/06/2023.
- [6] Scrum, Welcome to the home of Scrum, <https://www.scrum.org/>, 12/07/2023.
- [7] StackOverflow, Every developer has a tab open to Stack Overflow, <https://stackoverflow.com/>, 29/06/2023.
- [8] Windows, Windows-10-Pro, <https://www.microsoft.com/en-us/windows/get-windows-11?activetab=pivot:overviewtab>, 01/03/2023.
- [9] Godot-Engine, The game engine you've been waiting for, <https://godotengine.org/>, 12/07/2023.
- [10] Xiaomi, Mi-A2, <https://www.mi.com/es/mi-a2/>, 09/03/2023.
- [11] Samuel Inkilainen, Visual Novel Backgrounds – Lanoratory & Control Rooms, <https://samuelinkilainen.com/projects/zOYKzD>, 13/04/2023.
- [12] Trello, Welcome to Trello, <https://trello.com/es>, 03/07/2023.
- [13] Google, Google Meet videollamadas premium, <https://meet.google.com/>, 12/07/2023.
- [14] Universidad de Granada, Grado en Farmacia y Nutrición Humana y Dietética, <https://www.ugr.es/estudiantes/grados/grado-farmacia-nutricion-humana-dietetica>, 7/07/2023.
- [15] AULA EN RED, Espacio de educación virtual, <https://aulaenred.edu20.org/>, 04/03/2023.
- [16] Biomodel, Laboratorios virtuales, <https://biomodel.uah.es/lab/inicio.htm>,

03/03/2023.

- [17] ChemCollective, Chemical Reaction Virtual Lab, <https://chemcollective.org/vlabs>, 05/03/2023.
- [18] Softonic, CHEMLAB, <https://model-chemlab.softonic.com/?ex=RAMP-1097.2>, 07/03/2023.
- [19] Kahoot, Welcome to Kahoot!, <https://kahoot.it/>, 11/03/2023.
- [20] CloudLabs, Virtual STEM, <https://cloudlabs.us/es/home-es/>, 11/03/2023.
- [21] Unreal Engine, Todo el poder de UE, <https://www.unrealengine.com/es-ES>, 07/03/2023.
- [22] Unity, Expanding community, <https://unity.com/es>, 07/03/2023.
- [23] GameMaker Studio 2, Todo lo que necesitas para crear videojuegos, <https://gamemaker.io/es>, 08/03/2023.
- [24] Stencyl, Create amazing games without code, <https://www.stencyl.com/>, 09/03/2023.
- [25] Armory3D, Build games in blender, <https://armory3d.org/>, 13/03/2023.
- [26] Defold, The ultimate game engine for web and mobile, <https://defold.com/>, 13/03/2023.
- [27] ProyectosAgiles, Desarrollo incremental, <https://proyectosagiles.org/desarrollo-iterativo-incremental/>, 17/05/2023.
- [28] Godot-Engine, Asset Library, <https://godotengine.org/asset-library/asset>, 30/06/2023.
- [29] MAcademy, Logotipo-Godot-Engine, <https://mmmacademy.es/godot-que-es-ventajas-funcionamiento/>, 22/05/2023.
- [30] Wikimedia Commons, Iterative Process Diagram, https://commons.wikimedia.org/wiki/File:Iterative_Process_Diagram.svg, 24/03/2023.
- [31] Deloitte, Burndown diagram, <https://www2.deloitte.com/es/es/pages/technology/articles/diagramas-burndown-herramienta-o-arma-de-destruccion.html>, 30/05/2023.
- [32] Software-Fish, Gantt-chart, <https://software.fish/project-management-software/gantt-chart-template>, 27/04/2023.

