



Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

Videojuego Serio:
Balanza DCorazon

Serious Game, Balanza DCorazon
Juvenal Santiso Hernández

La Laguna, 8 de Septiembre de 2015

D^a. **Carina Soledad González González**, con N.I.F. 54.064.251-Z profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

D. **Yeray del Cristo Barrios Feitas**, con N.I.F. 54.106.627-R, como cotutor.

C E R T I F I C A (N)

Que la presente memoria titulada:

"Videojuego Serio: Balanza DCorazon."

Ha sido realizada bajo su dirección por D. **Juvenal Santiso Hernández**, con N.I.F. 78.678.134-W.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 8 de Septiembre de 2015.

Agradecimientos

A Carina y Yeray, por apoyarme.

A mis profesores, por guiarme.

A mis amigos, por estar siempre.

Y a mi pareja por animarme.

Sin todos no hubiera sido posible nada.

Licencia



© Esta obra está bajo una licencia de
Creative Commons Reconocimiento 4.0
Internacional.

Resumen

La finalidad del presente trabajo ha sido la de crear un videojuego serio, para la integración en la plataforma de Provitao, cuyo objetivo es el de propiciar el aprendizaje de los niños, consiguiendo que sean capaces de preparar sus dietarios día a día. Esta clase de videojuegos comprende aquellas tareas que no buscan simplemente para la diversión del propio usuario, sino que pretenden que el usuario aprenda a medida de que se cumplan estas tareas ("juegos").

La aplicación consistirá en un juego con mecánicas de Drag&Drop con el objetivo de mezclar alimentos para el desarrollo del "minijuego", dando así la posibilidad de descubrir otros nuevos. Se cuenta con una base de datos en SQLite para el almacenaje de los objetos iniciales y con posibilidad de extensión a medida de que se vayan obteniendo alimentos como consecuencia de las mezclas.

Se intenta explotar el uso del móvil para su difusión, ya que se trata de un dispositivo enfocado a prácticamente cualquier edad y, en la actualidad, forma parte de la vida cotidiana de la mayoría de las personas. Se ha optado por utilizar la plataforma Unity3D para la creación de un videojuego para móviles. Su desarrollo ha requerido un estudio de su estructura (Mockups), así como acerca de que tecnologías usar para el cumplimiento de la misma, la gestión de bases de datos...

Este proyecto ha sido una gran oportunidad de conocer el motor de Unity3D, pues los conocimientos que se presentaban hasta el momento acerca del mismo eran escasos. Indagando en el motor y las funcionalidades ofrecidas se puede observar una cantidad infinita de posibilidades de crecimiento, gracias a todas las facilidades que suministra: el límite de los elementos que podrían integrarse en la aplicación, sólo se encuentra en la imaginación

Palabras clave: Provitao, APK, Drag&Drop, juego serio, Mockups, aplicación Unity3D, SQLite.

Abstract

The purpose of the current document has been to develop a serious game, which is integrated with Provitao and attempts to facilitate learning for the children, making them be able to build their daily dietary. This kind of games not only seek to the fun of playing a game but also to learn in the process while getting tasks done. This aims to maximize the use of this serious game in portable devices since is available at any age and this kind of devices are present in the people's daily lives. To do so, Unity3D platform will be used to develop the mobile game. To the development it has taken the study of the structure (Mockups), as well as the study of the possible alternative technologies in order to archive the objective sought.

This mobile application will consist of a game with mechanics as Drag&Drop with the goal of mix foods to the develop of the "mini game", this will offer the possibility to find out new food. An SQLite database will keep saved the initial objects among the new food created as a mixing elements consequence.

This project allows to delve into the knowledge of the cross-platform game engine Unity3D, which offers a great amount of possibilities and options to incorporate to the mobile application, this makes imagination the only possible limit.

Índice General

Capítulo 1. Introducción	1
1.1 Descripción	1
1.2 Juegos serios	1
1.3 Provitao	2
1.4 Aportación	3
Capítulo 2. Antecedentes y estado actual	4
2.1 Contexto de los juegos serios	4
2.2 Unity 3D	5
2.3 Videojuego propuesto	6
Capítulo 3. Objetivos	8
3.1 Actividades principales	8
3.1.1 Primera fase (Análisis)	8
3.1.2 Segunda fase (Desarrollo)	9
3.1.3 Tercera fase (Documentación)	10
3.2 Plan de trabajo	10
Capítulo 4. Desarrollo	11
4.1 Diseño	11
4.1.1 Estructura	11
4.1.2 Funcionalidades	14
4.2 Análisis	15
4.2.1 Aplicación (Mockups)	15
4.2.2 Mecánicas	20
4.2.3 SQLite.....	21
4.2.4 Resumen	22
4.3 Implementación	23
4.3.1 Esquema de escenas (Hierarchy)	23
4.3.2 Creación e interacción (UI)	26
4.3.3 Drag&Drop	27
4.3.4 SQLite.....	28
4.4 Documentación y testeo	29
4.5 Módulos de la aplicación	30

Capítulo 5. Conclusiones y líneas futuras	32
Capítulo 6. Summary and conclusions	33
6.1 Summary	33
6.2 Conclusions	33
Capítulo 7. Presupuesto	35
7.1 Coste del proyecto	35
Apéndice A. Enlaces del proyecto	37
Bibliografía	38

Índice de figuras

Figura 4.1. Estructura de directorios	
Figura 4.2. Mockup menu principal	
Figura 4.3. Mockup juego	
Figura 4.4. Mockup Dieta	
Figura 4.5. Mockup Inventario	
Figura 4.6. Esquema esqueleto app	
Figura 4.7. Jerarquía escena principal	
Figura 4.8. Jerarquía escena juego	
Figura 4.9. Jerarquía escena dieta	
Figura 4.10. SQLite Browser	

Índice de tablas

Tabla 3.1. Plan de trabajo	
Tabla 7.1. Estimación de tiempo y coste	
Tabla 7.2. Estimación del coste de materiales	

Capítulo 1.

Introducción

1.1 Descripción

Comenzaremos por introducirnos en materia con los antecedentes previos del proyecto, objetivos y alcance del mismo, para luego continuar con la explicación del desarrollo seguido estos meses. Para finalizar, se expondrán los resultados y conclusiones en vistas de mejorar con líneas futuras.

En primer lugar, se tratarán una serie de temas y cuestiones con el objetivo de entender el contexto que rodea al presente proyecto y la finalidad del mismo, de entrar en materia. Se continuará con la descripción de las actividades realizadas, junto a las decisiones tomadas a lo largo de todo el desarrollo.

Anexo a la memoria, se puede encontrar el material de apoyo utilizado, así como otros recursos de interés.

Finalizando, como no puede ser de otra manera, con las conclusiones y las referencias bibliográficas utilizadas a modo de apoyo.

1.2 Juegos serios

Juegos serios, una categoría cuyo reciente auge está cambiando la forma de percibir el concepto de 'juego' actual, busca ahondar en el uso de los juegos, tratando de servir de apoyo en el aprendizaje de los usuarios.

Esta vertiente trata a los juegos más a allá de la mera diversión que tratan de proporcionar al usuario, pretenden

además lograr otros objetivos, como pueden ser el de la educación, la mejora de la salud, la simulación, o la publicitación de productos. En el caso del presente trabajo, se pretende crear un juego serio que busca el aprendizaje por parte del jugador. Por tanto, las tareas que se van a crear serán destinadas a inculcar conocimientos mientras el usuario juega, para que sin él mismo darse cuenta pueda adquirir conocimiento de una forma divertida y ágil.

1.3 Provitao

El proyecto Provitao: PRograma de Videojuegos activos para el tratamiento de la Obesidad (PROVITAO), ha sido creado para abarcar uno de los grandes problemas de salud, la obesidad.

Un estudio¹ confirma que 44,5% de los niños de España sufren sobrepeso u obesidad, donde solo Italia y Chipre aumentan esta tasa.

Se puede declarar que este proyecto está dedicado a un programa de videojuegos serios para el tratamiento ambulatorio de la obesidad, que pretende atacar el tratamiento de la obesidad en edades tempranas y, por consecuente, prevenir el trastorno en edades adultas.

El centro del presente trabajo de fin de grado: poder ayudar a la causa, con la ayuda de una aplicación. Cuyo objetivo, consiste en un juego serio que permita a los niños reducir su sobrepeso, (mejorando así su salud), a través del aprendizaje relativo a la creación de dietarios cada día.



Logo Provitao

¹ Estudio IDEFICS (IDEFICS 2014)

1.4 Aportación

El resultado de este trabajo final de grado es el de crear un apk² que contenga el videojuego serio estructurado y funcional.

El proyecto estará firmado bajo la licencia de software libre, de modo que el código estará abierto a todos aquellos usuarios que deseen investigar, mejorar o reutilizar su contenido.

² Apk: Es un paquete del sistema operativo android que sirve para instalar aplicaciones.

Capítulo 2.

Antecedentes y estado actual

2.1 Contexto de los juegos serios

A pesar de que los juegos serios han existido mucho antes de la aparición de los dispositivos electrónicos en el mundo del entretenimiento, sobre todo con propósito educativo, el potencial de la tecnología actual ofrece un abanico extenso de posibilidades en cuanto a la creación de aplicaciones con funciones, como acudir a la solidaridad de las personas y potenciar las acciones sociales.

Unido al factor de gamificación³, permite generar un mayor atractivo, realzando su difusión y utilización. Son aspectos fundamentales, aún más en el caso de Provitao, tratándose este de un juego que pretende ser una fuente de diversión y con un gran potencial educativo para ofrecer una forma de apoyar a las futuras generaciones.

Aprovechando el auge de los dispositivos móviles y las posibilidades que ofrecen. Gracias tanto a sensores y nuevas tecnologías, como por su amplio alcance y aceptación en las nuevas generaciones. Cuya iniciación con los dispositivos electrónicos cada vez se produce a una edad más temprana.

³ Gamificación: Es el empleo de mecánicas de juego en entornos y aplicaciones no lúdicas con el fin de potenciar la motivación, la concentración, el esfuerzo, la fidelización y otros valores positivos comunes a todos los juegos. Se trata de una nueva y poderosa estrategia para influir y motivar a grupos de personas.

Hasta el momento se han centrado los esfuerzos en generar una aplicación funcional, con el objetivo de que el juego esté disponible desde sus inicios, y pueda ir creciendo a medida que lo hace su comunidad.

El juego propuesto tendrá dos módulos bastante diferenciados:

- 'Front-end'⁴:
 - Interfaz básica.
 - Minijuego de mezcla de alimentos.
- 'Back-end'⁵:
 - Base de datos funcional con información acerca de los Platos. (Nombre, Calorías, Valor)

2.2 Unity 3D

El corazón de este proyecto radica en su motor de juego, Unity3D. Esta plataforma de desarrollo de videojuegos posee un conjunto muy variado de herramientas que nos permiten, de una forma fácil y flexible, exportar el resultado del trabajo a distintas plataformas de juego simultáneamente. Esto significa que, usando el mismo código, podemos ejecutar el videojuego en plataformas tan distintas como son Android o IOS sin mayor esfuerzo.

Se trata entonces de una plataforma de desarrollo flexible y potente que optimiza los resultados del proyecto reduciendo sus costes y aumentando los beneficios. Esta ventaja permite exportar los proyectos creados a 22 plataformas (IOS, android, windowPhone, WiiU, aplicaciones de escritorio, WebGL, Oculus Rif, etc.) sin mayor esfuerzo.

⁴ Front-end: Se traduce como la interfaz que se comunica con el usuario.

⁵ Back-end: Se traduce como el motor o lógica de la aplicación que para el usuario es transparente.

Además Unity posee una tienda de activos completa con la que en un proyecto voluminoso podrías obtener poderosos activos para que cualquier proyecto. Ayuda a los mismos para que cuenten con utilidades y recursos potentes que faciliten su desarrollo. Ofrece unas características perfectas para el ámbito de los videojuegos ya que permite interactuar con muchos programas de diseño externos (3ds Studio Max, Maya, Blender, etc.). Unity cuenta además con herramientas internas como Mecanim, un módulo diseñado para llevar el movimiento fluido y natural de los personajes con una interfaz eficiente.

Resaltar que Unity cuenta con dos tipos de licencia, Unity free y Unity Pro. Unity Pro ofrece mejores sistemas de render, iluminación y gráficos, por otro lado la versión gratuita aparte de no tener esas mejoras de gráficos en nuestras aplicaciones aparecerá la marca de agua⁴ de Unity sin posibilidad de quitarla o modificarla.

Por último, ¿Por qué no usar Unreal Engine como motor? Es una versión potente pero en mi opinión Unity al utilizar el lenguaje C# es más fácil de usar que el C++, que es el que usa el Unreal, y no deja de seguir siendo potente. Apoyando también a que cuando vayamos a exportar a móviles dispone de una mayor facilidad Unity. En yuxtaposición la documentación y las librerías que nos ofrece C++ y Unreal Engine son muy buenas y amplias, podría decirse que mejor que la de Unity pero las de C# y las propias de la API de Unity son muy buenas así que esto no es un punto que haga un fuerte impacto en nuestra elección.

2.3 Videojuego propuesto

Tras un extenso análisis del mercado de los videojuegos y de las mejores mecánicas para lograr empatar con los objetivos de los usuarios dentro del proyecto, surge el nombre de "Alchemy".

Capítulo 3.

Objetivos

3.1 Actividades principales

Como principales acciones para llevar a cabo la idea propuesta por Provitaio, se pretende crear una apk⁶ para smartphones capaz de generar un ambiente de juego adecuado y entretenido. Es importante que esté correctamente estructurado y cuente con una buena organización de ficheros en directorios para un mejor manejo de los mismos a la hora de desarrollar la idea.

Para un adecuado desarrollo, se ha dividido en tres fases independientes entre sí: una primera fase de preparación y análisis, otra de desarrollo y por último una de documentación.

3.1.1 Primera fase (Análisis)

La primera fase consiste en ver las posibilidades que se tienen para llevar a cabo el proyecto, qué tecnologías usar, y cualquier otro aspecto necesario para la consecución de los objetivos. En esta fase lo que se busca como producto es un esquema basado en mockups⁷ de la aplicación. Dicho de otra manera, la primera fase aborda

⁶ Es un paquete del sistema operativo android que se usa para distribuir e instalar los componentes en la plataforma android.

⁷ Es un modelo a escala o tamaño real de un diseño o un dispositivo, utilizado para la demostración, evaluación del diseño, promoción, y otros fines.

el diseño y la planificación del proyecto, cuyo resultado es un prototipo conceptual de la aplicación.

3.1.2 Segunda fase (Desarrollo)

La segunda fase de desarrollo tiene como objetivo conseguir un producto mínimo viable que posea todas las mecánicas básicas y estructura. A su vez esta etapa se ha dividido en tres pequeñas tareas.

Un primer sprint⁸ para implementar la estructura de la aplicación y crear una base de datos⁹ con *SQLite*¹⁰.

Un segundo sprint para diseñar la mecánica principal del juego que es el Drag&Drop¹¹.

Para finalizar un tercer sprint para crear el ambiente de juego y sincronizar las partes.

Siendo el hito de esta fase la creación base funcional de la aplicación.

⁸ Es una etapa atómica donde se decide a realizar una serie de tareas y tenerlas listas en un espacio determinado de tiempo.

⁹ Bancos de información que contienen datos relativos a diversas temáticas y categorizados de distinta manera.

¹⁰ Es un sistema de gestión de bases de datos relacional compatible con ACID.

¹¹ Expresión informática que se refiere a la acción de mover objetos de una ventana a otra o entre partes de una misma ventana.

3.1.3 Tercera fase (Documentación)

La tercera y última fase del proyecto consiste en la generación de la documentación acerca del mismo, siendo el hito principal la creación de una memoria explicativa de la aplicación y una demo básica que muestre las funcionalidades del prototipo.

3.2 Plan de trabajo

Según se ha definido anteriormente, las tareas a realizar en el Trabajo de Fin de Grado se desglosarían:

Tarea	Nombre	Tiempo
1	Estudio de la propuesta	7 días
2	Desarrollo de la propuesta	7 días
3	PROTOTIPO A BAJO NIVEL (PANTALLAZOS)	HITO
4	Sprint 1: Estructura principal y flujo de la información	15 días
5	PLATAFORMA BASE	HITO
6	Sprint 2: Jugabilidad e Inteligencia Artificial	15 días
7	JUGABILIDAD	HITO
8	Sprint 3: Gamificación y Ludificación	15 días
9	GAMIFICACIÓN	HITO
10	Demo funcional	8 días
11	Guía de usuario	7 días
12	Memoria TFG	10 días
13	Video de Presentación	7 días
14	Presentación PowerPoint	7 días
15	DEFENSA TFG	HITO

Tabla 3.1 Plan de trabajo

También es importante tener en cuenta el testeo y la comprobación de que cada funcionalidad presenta el comportamiento esperado en dispositivos móviles (ya que, como es lógico, se desarrolla en un computador personal); así como la generación de documentación en cada fase, y de (dada la condición de software libre) una guía básica de instalación y uso.

Capítulo 4.

Desarrollo

4.1 Diseño

La primera fase del desarrollo del proyecto ha consistido en determinar exactamente con qué funcionalidades debe contar la aplicación, cómo crear una estructura de directorios lógicos y consistentes, y por último diseñar una serie de mockups a modo de apoyo para definir el esqueleto de la aplicación.

4.1.1 Estructura

Se ha seguido una estructura bastante legible y extensa para una mayor sensación de modulación y separación entre las diferentes partes del proyecto.

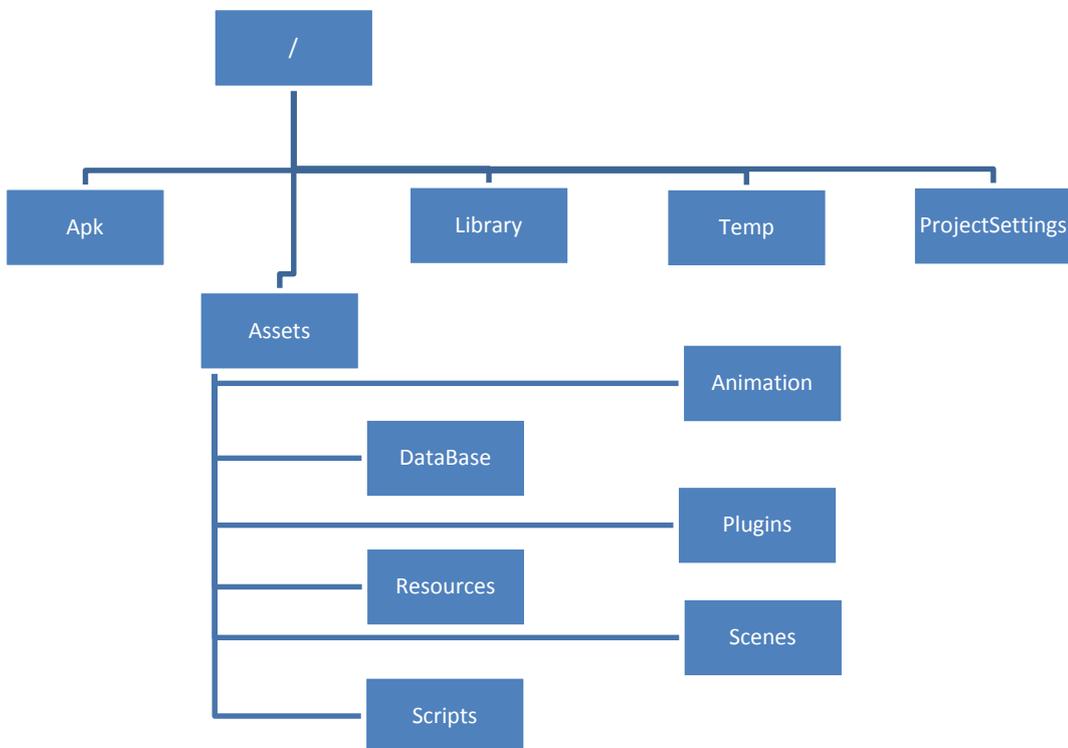


Figura 4.1 Estructura de directorios.

Como se ha decidido realizar el proyecto con Unity3D, la estructura inicial será la proporcionada por esta plataforma y la estructura de directorios donde vamos a navegar y la que realmente se cree será la que se encuentra dentro del directorio de Assets¹².

Dentro de esta carpeta se encuentra todos los recursos con los que vamos a interactuar, y todos los que vamos a necesitar para la finalización de las tareas. Una vez dentro de la carpeta de Assets existen carpetas con nombres especiales que Unity detecta automáticamente.

La carpeta Gizmos¹³ que sirve para agregar a la escena gráficos que ayuden a visualizar los detalles del diseño.

La carpeta Plugins¹⁴ es una carpeta donde Unity automáticamente busca complementos (DLL) normalmente escritos en C/C++ permitiendo a nuestra aplicación usar estas librerías o nuevas llamadas al sistema según el plugin que añadamos a esta carpeta.

La carpeta Resources, tiene como obligación contener las instancias que se requieran en las escenas, es decir, los objetos que se van a utilizar en la aplicación. Podemos cargar y modificar dichos objetos a través de un script y para lograr esto se utiliza la función Resources.Load y se necesita la existencia de dicha carpeta, ya que es una función del sistema y por defecto busca el objeto que le indiquemos pero con la condición de que esta carpeta sea la raíz.

StreamingAssets una carpeta que se encarga de persistir los archivos en crudo si exportas para algún dispositivo, es decir, cuando se exporta a una plataforma por ejemplo

¹² Es una fuente de recursos.

¹³ Marcadores para un objeto o posición.

¹⁴ Aplicación informática que se relaciona con otra para aportarle nuevas funcionalidades.

android como es el caso de este proyecto todos los archivos o imágenes que se utilicen en él se comprimirán para adaptarse al SO destino pero esta carpeta en el proceso de exportación se queda intacta y sirve para alojar en ella archivos que se necesita que no se toquen en el proceso de exportación y un ejemplo de archivo sería en este caso la base de datos local.

No se usarán todas las carpetas y los demás directorios se han creado para una mejor comprensión. La estructura queda de la siguiente manera:

Usamos la carpeta especial Resources y Plugins

- **Apk:** Se encarga de almacenar la aplicación exportada para Android.
- **DataBase:** Contiene el fichero que constituye la base de datos.
- **Animation:** Se encarga de almacenar todas las animaciones que existirán en la aplicación.
- **Scenes:** En la que estarán todas las escenas de la aplicación.
- **Scripts:** Tiene como objetivo encapsular todos los scripts que utilizaremos en la aplicación.
 - o Cabe destacar que para lograr una mejor organización dentro del directorio, los scripts se encuentran distribuidos en una serie de carpetas, asociada cada una a una escena distinta de la aplicación.

- Resources:
 - o Material: Texturas importadas.
 - o Prefabs¹⁵: Contiene los objetos que se visualizará en pantalla.
 - o Sprites¹⁶: Imágenes del juego.

4.1.2 Funcionalidades

En cuanto a las funcionalidades y su estructura, se decidió que la aplicación iba a contener una base de datos en *SQLite*, creada con *SQLite* browser en una primera instancia y se irá modificando a lo largo del proyecto.

Las funcionalidades que le dan vida a nuestro proyecto son las siguientes:

- Funciones para la consulta de la base de datos, para la modificación de valores, para añadir nuevos registros y las correspondientes a abrir y cerrar la base de datos.
- Poder asignar alimentos a las comidas principales del día ya sea el desayuno, el almuerzo o la cena.
- La mecánica principal Drag&Drop, que consiste en tocar el objeto que queremos arrastrar y soltar en otro sitio.
- La unión de elementos nuevos, es decir, la capacidad de reconocer que alimentos estamos intentando unir y si es una unión válida ponerla visible y en la base de datos.

¹⁵ Objetos pre-creados para cargarlos en escena.

¹⁶ Un tipo de mapa de bits dibujado en la pantalla.

Tras la implementación de todas las funcionalidades el juego está prácticamente listo para las creaciones de las escenas y la asignación de los objetos que dispondremos en las mismas.

4.2 Análisis

Posiblemente la fase de análisis puede considerarse como la más costosa y extensa llevada a cabo en el proyecto. Aquí se han tomado las decisiones acerca del diseño (mediante el uso de mockups) y sobre cómo implementar una base de datos en Android para llevar a cabo este proyecto.

4.2.1 Aplicación (Mockups)

Se ha pretendido crear una app¹⁷ para Android¹⁸. El mecanismo utilizado es el de exportar a través del programa Unity3D y crear una apk que posteriormente será instalada en el dispositivo.

En este apartado se explicarán los mockups que se diseñaron y se realizará un esquema con las conexiones que hay entre cada pantalla.

¹⁷ Aplicación móvil, una aplicación informática diseñada para ser ejecutada en dispositivos móviles (smartphones, tablets,...).

¹⁸ Sistema operativo diseñado por la empresa Google, para dispositivos móviles con pantalla táctil.

Se diseñó una interfaz muy básica por no decir primitiva para la parte del menú inicial del juego:

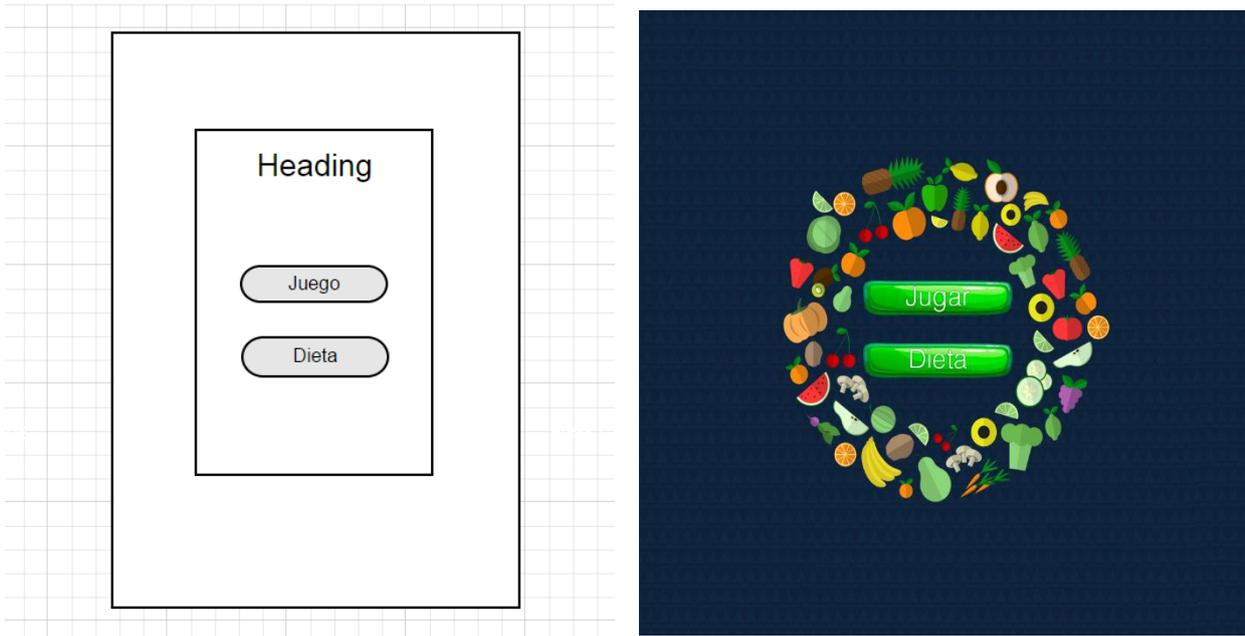


Figura 4.2 Mockup Menu principal

Para la pantalla del juego se ha creado otra interfaz bastante sencilla e intuitiva de cómo mezcla. Una zona en el medio para dejar caer los ítems que estarán organizados en la parte inferior de la pantalla para poder arrastrarlos con facilidad.

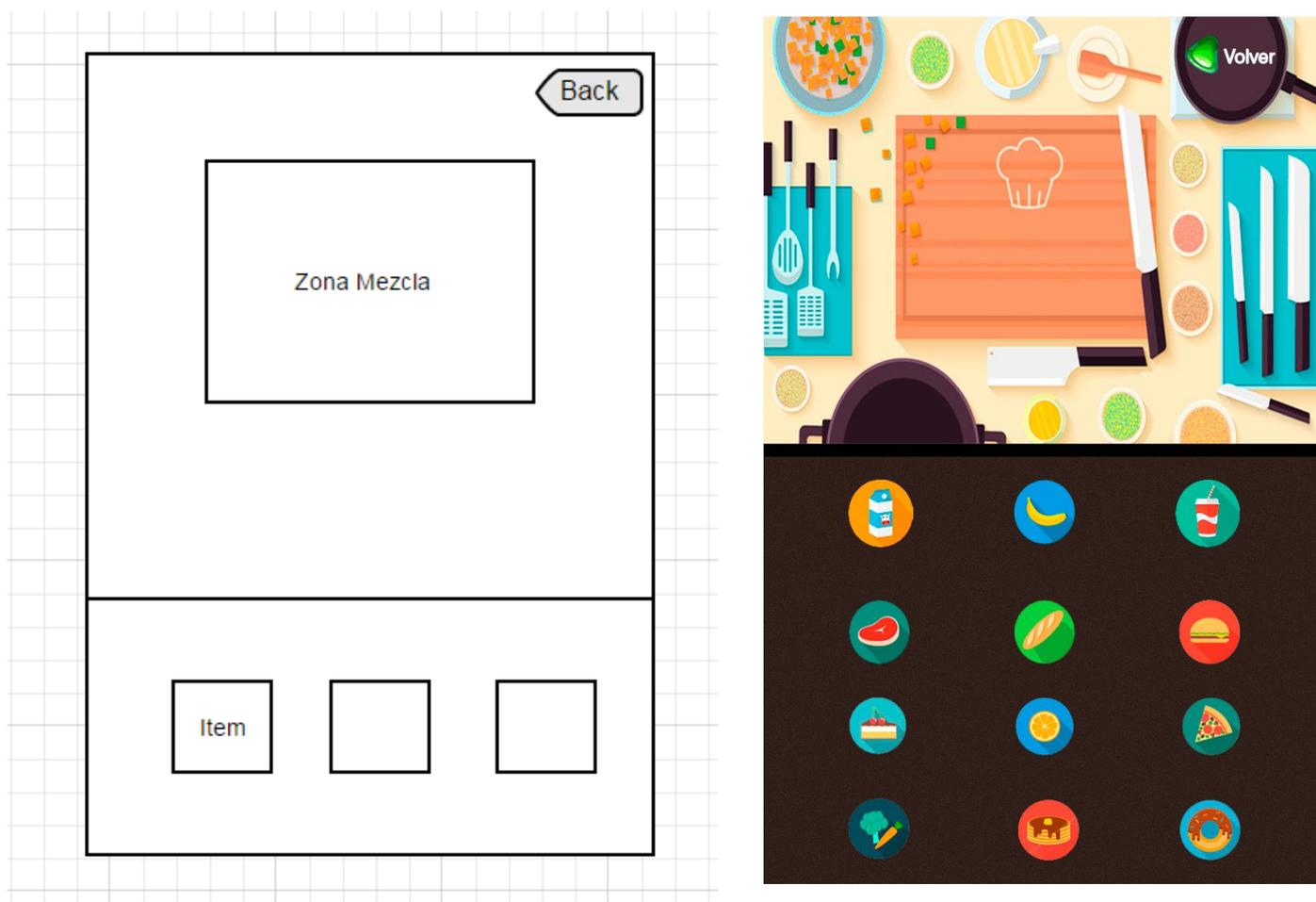
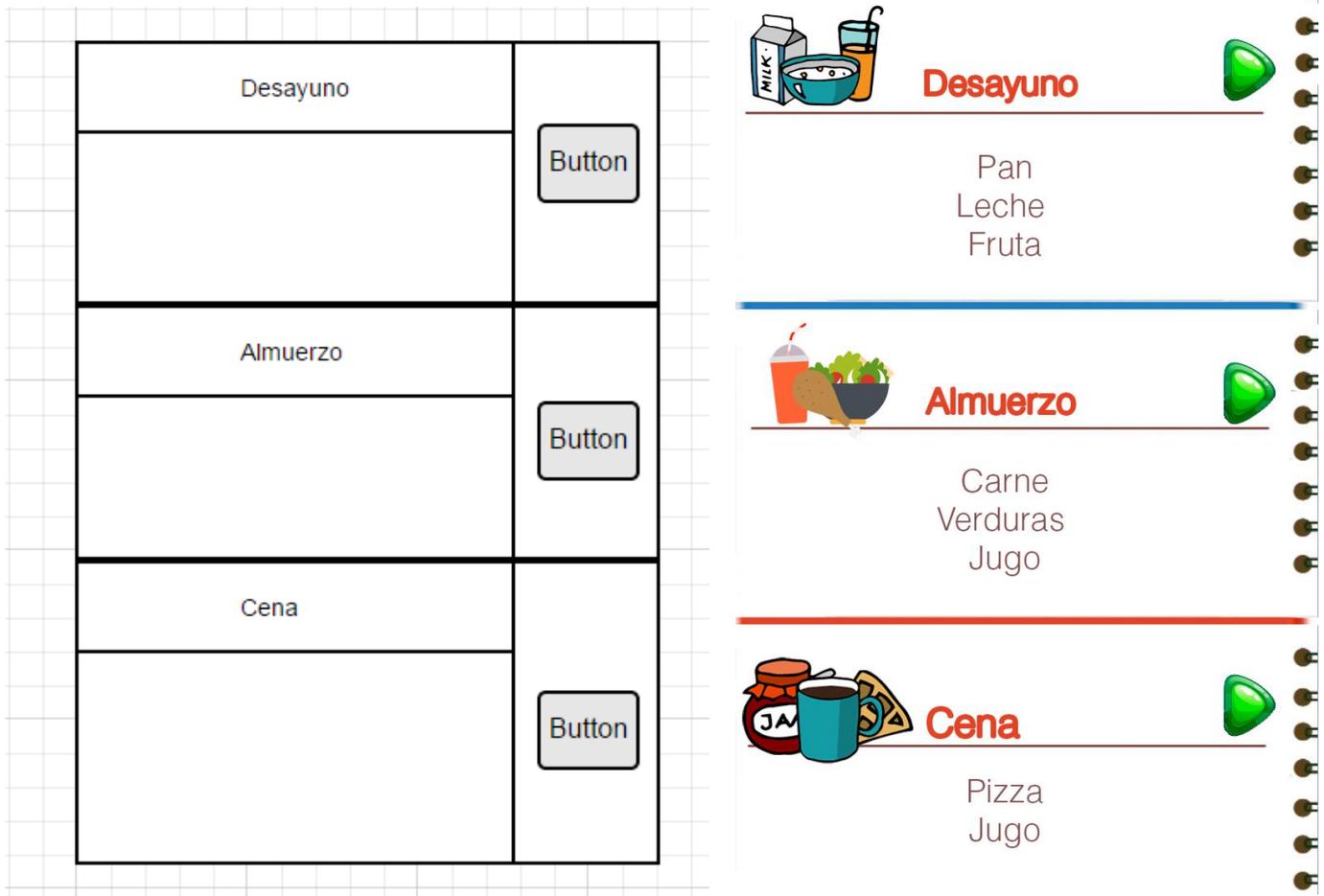


Figura 4.3 Mockup Juego

La pantalla de dieta se divide en tres sencillas zonas para asignar los alimentos al desayuno, al almuerzo o a la cena.

Figura 4.4 Mockup Dieta



Por último la pantalla donde estará el inventario donde podremos asignar las comidas y con un botón de back:

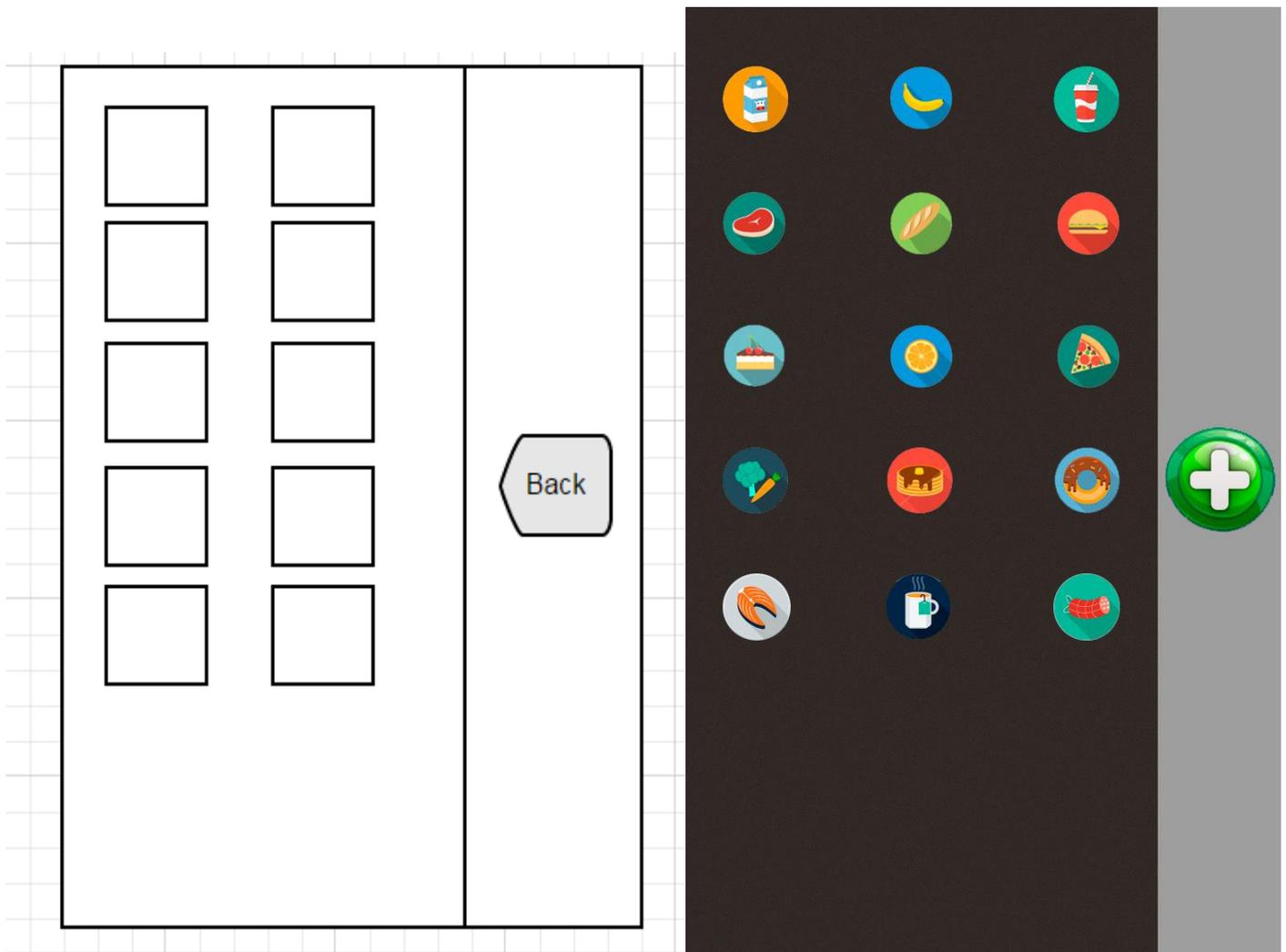


Figura 4.5 Mockup inventario

Teniendo explicada cada pantalla por separado, deberán unirse que unirlas a través de acciones o con los botones de los que se dispone en cada pantalla. Se usan los botones del menú principal para ir con el botón de juego a la figura 4.3 y con el botón de dieta a la figura 4.4. Una vez en la pantalla de la dieta, con los botones que se tienen, se puede editar cada comida de forma individual y pasar a la figura 4.5.

Ahora el esquema que relaciona dichos pantallazos, dejando el esqueleto del diseño de la aplicación de la siguiente forma:

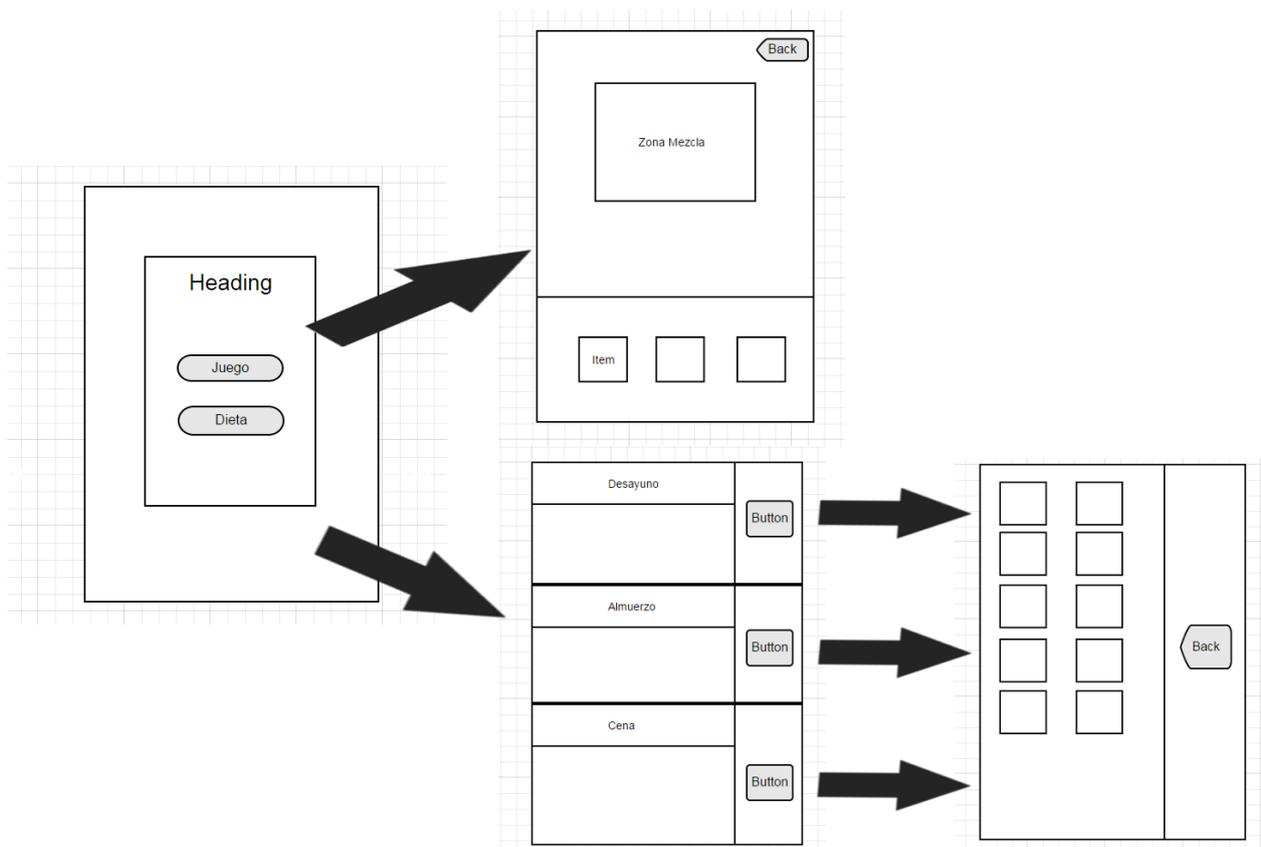


Figura 4.6 Esquema esqueleto app

4.2.2 Mecánicas

Teniendo en cuenta el rango de edad de los usuarios del videojuego, se ha diseñado la típica mecánica de Drag&Drop para la mezcla de alimentos.

En la escena de los alimentos se podrán arrastrar hacia la zona de mezcla, hasta un máximo de dos objetos. Como veíamos antes en los mockups los ítems (platos) estarán dispuestos en el inferior de la pantalla y podremos arrastrarlos hasta el centro donde está la zona que se encarga de recoger esos objetos e identificar si forman una unión válida.

Para mostrar todos los objetos de la base de datos a la hora de la mezcla se diseñó para que al deslizar el dedo por la pantalla se vayan mostrando los objetos. Se podría concebir que así creamos una interfaz natural¹⁹ y por tanto intuitiva para cualquier tipo de persona.

4.2.3 SQLite

Hasta hace muy poco resultaba muy costoso y complejo incorporar bases de datos a nuestras aplicaciones. No obstante, Android incorpora la librería SQLite que nos permitirá utilizar bases de datos mediante el lenguaje SQL, de una forma sencilla y utilizando muy pocos recursos del sistema. Almacenar tu información en una base de datos no es mucho más complejo que almacenarlos en un fichero, por tanto además resulta mucho más potente ya que nos ofrece la estabilidad de una base de datos.

SQLite es un sistema open source de bases de datos que combina una interfaz muy limpia de SQL y que nos permite trabajar con poca memoria y con una velocidad bastante decente, características que son necesarias cuando hablamos de entornos móviles.

SQLite soporta las características estándar de las bases de datos relacionales como la sintaxis que se basa en SQL, transacciones y la elaboración de consultas. Debido a esto, cualquier desarrollador que haya trabajado con bases de datos sin importar el entorno, no encontrará una dificultad especial en trabajar con bases de datos locales en Android. Por otro lado, hay que dejar en claro que la API nativa de SQLite no es compatible con JDBC contrario a lo que podría pensarse tomando en cuenta el uso de Java como lenguaje nativo en aplicaciones Android, pero a

¹⁹ Uso de movimientos gestuales para comunicarse con el sistema

nosotros que vamos a utilizar unity como intermediario para exportar a android no debemos preocuparnos por este detalle.

Por último, sobre SQLite, cabe destacar que SQLite, como su nombre indica, utiliza el lenguaje SQL para las consultas (SELECT), manipulación de datos (INSERT, DELETE, etc.), y de definición de datos (CREATE TABLE, etc). SQLite presenta unas pequeñas variaciones donde se desvía del estándar SQL-92, que aplica para la mayoría de bases de datos SQL.

Entre ellas, el uso de FOREIGN KEY, transacciones anidadas, RIGHT OUTER JOIN, FULL OUTER JOIN y algunos usos de ALTER TABLE no son válidas en SQLite.

Para utilizar SQLite no se requiere de ninguna configuración o administrador de base de datos, por lo que únicamente deberemos preocuparnos por el manejo de SQL.

Por su sencillez, su compatibilidad con android y con la potencia y pocos recursos que necesita se decidió utilizar SQLite para crear una base de datos, en lugar de otro método de almacenamiento de datos como pudo haber sido hojas XML.

4.2.4 Resumen

Por tanto al final hemos decidido que nuestra app dispondrá de dos mecánicas básicas e intuitivas, para que sea fácil para todos los públicos. Por último, la persistencia a datos se implementará con la ayuda de SQLite, dada su potencia y escalabilidad, lo que nos asegura una versatilidad y seguridad necesaria para un proyecto que puede crecer rápidamente.

4.3 Implementación

En el presente apartado se describirán las tareas realizadas, para el cumplimiento de la aplicación. Se contemplarán todos los aspectos del desarrollo, desde la creación de las escenas y de la jerarquía de objetos de las mismas hasta la integración de SQLite con Unity3D.

4.3.1 Esquema de escenas (Hierarchy)

En Unity las escenas siguen un sistema jerárquico de objetos y esto será lo que redactemos en este apartado para describir las escenas que se han creado.

Se ha decidido crear tres escenas para toda la aplicación encargándose cada una de ellas de una parte distinta y bien diferenciada.

La primera escena es la encargada de hacernos de menú principal de nuestro juego. Por tanto la única obligación que tiene es llevarnos a una escena u otra según el usuario haya elegido.

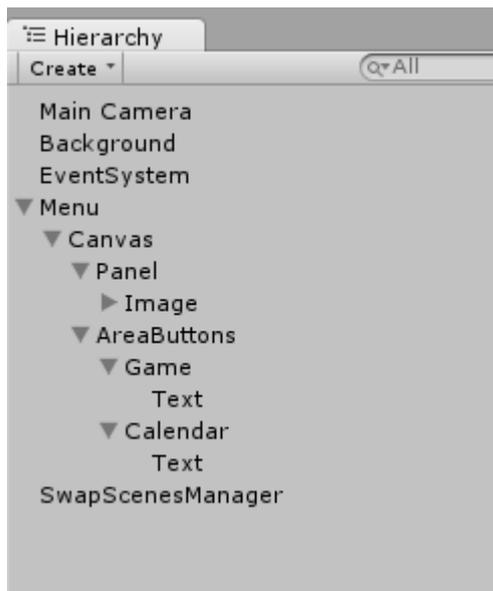


Figura 4.7 Jerarquía escena principal

La jerarquía²⁰ de esta escena está organizada en una cámara²¹, un objeto que servirá de fondo, un objeto que recoge los eventos de la escena (pulsar un botón...), un objeto que hemos creado nosotros vacío "SwapScenesManager" que tendrá los scripts de control de la escena y un objeto que será el encargado de contener a todos los objetos del menú, es decir la UI, pero explicaremos la UI y sus características en un apartado posterior.

La segunda escena que creamos fue la escena que contendrá el juego. Tendrá la principal mecánica de Drag&Drop y toda la información para crear los nuevos objetos e introducirlos en la base de datos.

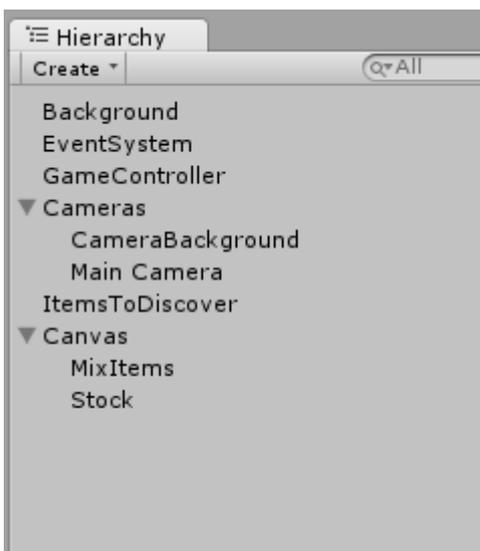


Figura 4.8 Jerarquía escena juego

En esta ocasión la jerarquía tiene un contenedor de cámaras que se encargarán de renderizar la escena desde dos perspectivas distintas para que el fondo se adapte

²⁰ Estructura que establece su orden en una escala ordenante y subordinante.

²¹ Objeto de Unity que sirve para renderizar la escena en pantalla

mejor a la resolución, un canvas²² que será el contenedor de los paneles de mezcla y stock, es decir los paneles donde se depositarán los elementos a mezclar y todos los elementos de los que disponemos respectivamente, un GameObject²³ llamado itemsToDiscover que tendrá una lista de todos los objetos que podemos crear en nuestro juego y un objeto (GameController) para almacenar los scripts de control de la escena (script de manejo de base de datos ...).

La tercera escena que creamos fue la escena que contendrá la asignación de platos a las comidas del día. Se encarga de que se pueda elegir cualquiera de los platos descubiertos a las comidas.

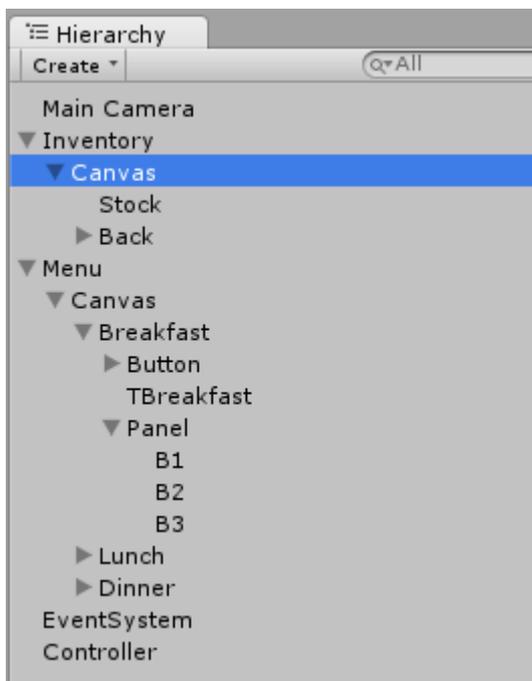


Figura 4.9 Jerarquía escena dieta

Se puede observar un objeto Inventory que será el que se encargue de mostrar a través del canvas todos los objetos

²² Área donde se encontrarán todos los objetos de interfaz de usuario (UI) en Unity.

²³ Son los objetos fundamentales de Unity, y componen las escenas representando cualquier posible figura renderizable.

que tenemos en nuestra base de datos, y un menú en el para el desayuno, el almuerzo y la cena sigue la misma estructura, un botón de editar, un título y un panel donde estarán los nombres de las comidas seleccionadas, con un máximo de tres platos por comida.

4.3.2 Creación e interacción (UI)

Este sistema de UI lo proporciona Unity para una creación rápida e intuitiva de interfaces de usuario.

En la sección anterior cuando se comentó la jerarquía se pudo observar algunos elementos de la UI como lo eran el canvas, los botones...

Todos los elementos de la UI tienen que estar dentro de un canvas. Cabe destacar que Unity proporciona sus propios eventos para saber si un botón ha sido pulsado y por tanto nos proporciona una función 'OnClick()' para asignarle lo que sucede si el evento se dispara.

Cuando se creaban las interfaces lo que se llevaba a cabo seguía unos pasos estructurados:

- Crear canvas
- Cerciorarnos si algún layout²⁴ nos convenía, ya que controlan mejor el posicionamiento de los objetos en pantalla con el cambio de resolución, sino se tiene

²⁴ Esquema de distribución de los elementos dentro un diseño.

que controlar manualmente los anclajes de los elementos.

- En el caso de que un layout no sirviera pues ajustar los anclajes de cada elemento en la pantalla para la escala de resoluciones.
- Introducir los paneles, botones... que se habían apuntado cuando se hizo el mockup de la escena.

4.3.3 Drag&Drop

Esta mecánica tuvo varias fases de desarrollo, en una primera instancia se programó directamente, pero cuando estuvo terminado se pudo comprobar que hacía demasiadas operaciones y se encontró con que Unity disponía de eventos de drag y eventos de drop.

Con las interfaces, `IBeginDragHandler`, `IDragHandler`, `IEndDragHandler` y `IDropHandler`, estas interfaces nos proporcionaban los métodos abstractos para introducir el código que queremos que se ejecute según el evento lanzado.

Con `IBeginDragHandler` el evento que recoge el método abstracto que nos proporciona esta interfaz, es justo cuando tocamos el objeto que tiene este script, nos sirve para guardar la posición en la que estaba y para en una variable estática²⁵ recoger que objeto es el que se va a arrastrar.

Con `IDragHandler` el evento que recoge el método abstracto que nos proporciona esta interfaz, ocurre cuando estamos

²⁵ Variable cuyo valor perdura a lo largo de la ejecución del programa.

arrastrando el objeto y la operación que realizamos aquí es actualizar la posición de nuestro objeto.

Con `IEndDragHandler` el método que nos da esta interfaz recoge el evento que se lanza cuando soltamos el objeto que estábamos arrastrando. Aquí reseteamos las variables y situamos el objeto en el lugar destino si se ha dejado en una zona válida o en el lugar de origen (lugar desde donde se empezó a arrastrar el objeto) si no es así.

Con `IDropHandler` nos proporciona la función que recoge el evento responsabilizado de detectar si un objeto que se está arrastrando esta sobre una zona de drop, válida para su depósito. Por tanto, aquí actualizamos la posición a donde tendría que volver y el evento de `IEndDragHandler` se encargará de colocarlo en su sitio.

4.3.4 SQLite

En esta sección se pondrá en manifiesto como se ha creado la base de datos y como se ha integrado con el programa. Para la creación del fichero se ha utilizado la herramienta `sqliteBrowser`, ya que al empezar a utilizar la aplicación la base de datos siempre será la misma.

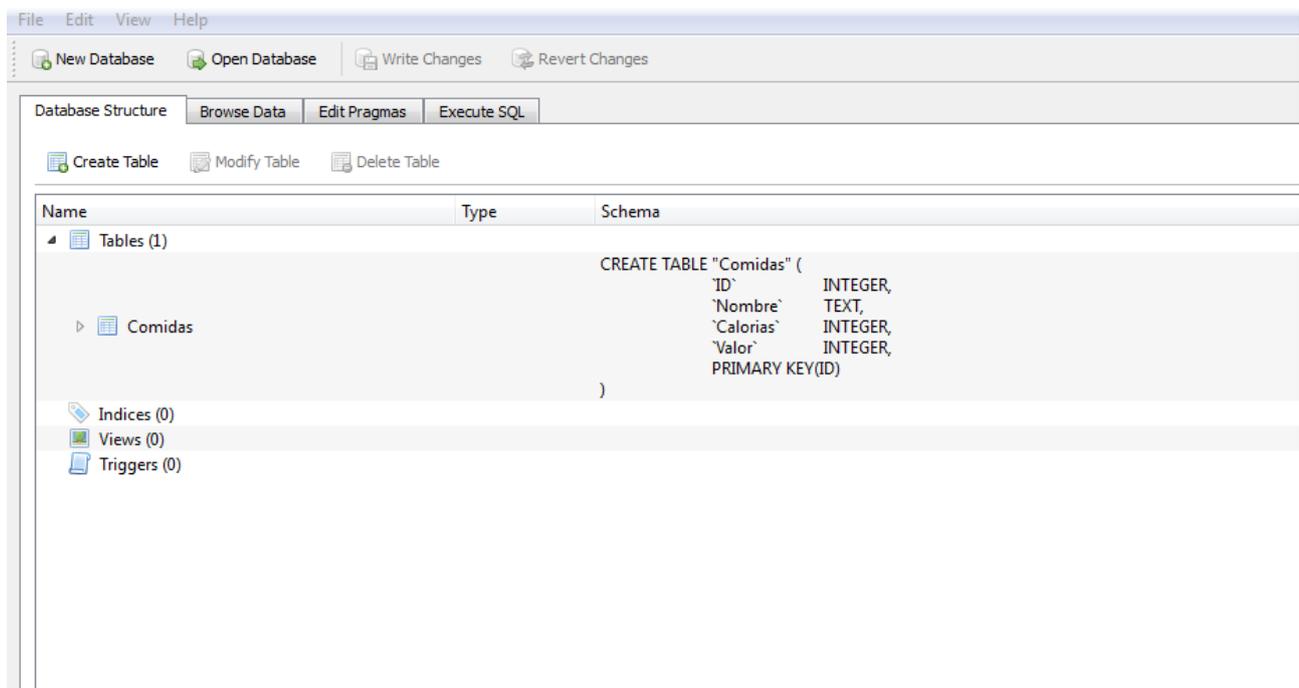


Figura 4.10 SQLite Browser

Se creó una tabla llamada "Comidas" con un ID para identificar cada plato, un nombre, calorías y un valor. Con la PRIMARY KEY en el atributo ID.

Una vez en la aplicación se ha creado un script que ya se ha mencionado "DataBaseManager.cs". El cual se encarga de todas las tareas relacionadas con la base de datos, con funciones para cada interacción. Desde tareas simples de búsqueda:

```
Select id From Comidas where Nombre = 'Fruta'
```

Hasta inserciones en la misma de la forma:

```
INSERT INTO Comidas(Nombre, Calorias, Valor) VALUES ('Tortilla', 650, 15)
```

Una vez dentro de nuestra aplicación solo tenemos que asignar dicho script a la escena y acceder a las funciones que nos ofrece. Las mencionadas anteriormente son las que realmente necesitaremos, ya que una sirve para saber si ya existe una entrada en nuestra base de datos y la otra para insertarlos nuevos platos que vamos descubriendo en la misma, respectivamente.

4.4 Documentación y testeo

La documentación que se presenta con este proyecto es un código bastante autodocumentado y una guía de usuario. Siendo más conciso, se ha creado una estructura de directorios bastante clara, por tanto bastante fácil de entender y cada segmento de los scripts y en la cabecera de cada script redactando correctamente sus funcionalidades.

En lo que respecta al testeo de la aplicación, dado que es una aplicación para móviles se ha probado en distintos tipos de móviles, según la arquitectura del móvil la base de datos presenta inconsistencia, pero la aplicación actúa según lo esperado.

4.5 Módulos de la aplicación



El proyecto tiene tres partes bien diferenciadas y gestionadas de forma distinta. Aunque el módulo del juego y de la dieta consulta la base de datos para su correcto funcionamiento pero se pueden desacoplar.

El módulo de la base de datos tiene como obligación acceder a la base de datos y realizar las consultas pertinentes para el las peticiones que se le hagan.

El módulo de juego contiene todas las mecánicas del mismo, desde como unir dos objetos hasta cual será el resultado de la unión.

Por último el módulo de la dieta se encargará de seleccionar las comidas que deseemos a distintos momentos del día.

Capítulo 5.

Conclusiones y líneas futuras

Con el presente proyecto sobre la mesa podemos pensar un poco más hacia el futuro aunque ya lo hayamos estado haciendo a medida que realizábamos el proyecto, siempre mirábamos a un funcionamiento posterior, ya que lo más importante de todo gran proyecto es una buena base, y cuando se empieza desde cero algo es muy costoso conseguir un buen diseño a la primera.

Las líneas futuras que nos deja este trabajo son infinitas, ya que podemos ir ampliando el conocimiento de la base de datos, podemos añadir más tablas categorizando los platos, un sistema de logros para poder tener una mejor sensación de juego, sistema de recompensas...

Otra línea por donde podríamos ir sería una mejor documentación orientada hacia el desarrollador para que otras personas que quieran acoger la idea tengan una base y sepan cómo usarla para mejorarla y añadir nuevas funcionalidades.

Ahora bien para dar por concluido la presente memoria no puede faltar mencionar lo novedoso e intrépido que me ha parecido este proyecto y la temática sobre la que gira. A primera instancia este trabajo me ha dado la oportunidad de conocer el motor de Unity3D, aprender con él y de él, también con un lenguaje C# que no había conocido anteriormente y el tema de los videojuegos serios para aprovechar el 'boom' tecnológico por el que estamos pasando y los niños tengan una mejor experiencia con la educación.

Por tanto me gustaría aclarar dos puntos con los que me quedaré de este proyecto. Por un lado el tema de los videojuegos serios es un tanto joven y tiene que madurar mucho para poder estirar sus ramas lo más alto y más extenso que se pueda y el otro punto a destacar la potencia evidente de la plataforma de Unity para crear videojuegos para diversas plataformas, aprovechándose así de un mercado más amplio.

Para finalizar esperar que el presente documento les haya permitido ver con más profundidad el tema sobre el que se ha hablado y animar a seguir con la investigación a futuros nuevos usos sobre los juegos serios.

Capítulo 6.

Summary and conclusions

6.1 Summary

Serious games is a special kind of game whose objective is that the player gets to have fun but learn with it. This has emerge to get them to learn more about topics such as health or education without seeming so tedious.

This project focuses on creating a serious game for smartphone whose aim was for children to learn to create their own diaries, each day.

In order to build a prototype with basic functions, the required steps were: a design phase, an analysis phase, a development phase and a phase of testing and documentation. Following these phases we can see a prototype about our app.

The application interacts with a database and has a game of mixed meals; the interaction is due to know that objects have to add the scene, you query the database to find out which objects are existing, and that when we mix two

objects and finds a new one we have to insert the corresponding record to the database data.

6.2 Conclusions

I can't forget to mention the novelty and fearless that I found this project and the theme on which around. This job has given me the opportunity to know the engine Unity3D, learn from it, also with a C# language that I hadn't previously known and the subject of serious games to take advantage of the technology to reach children.

So I would like to clarify two points that I will stay in this project. First the subject of serious games is rather young and has to mature and second the apparent power of the Unity platform to create various platforms video games, taking advantage of a wider market.

To conclude I hope that this document has allowed you to learn about this topic and encourage the research into new uses for future serious games.

Capítulo 7.

Presupuesto

7.1 Coste del proyecto

El cumplimiento del trabajo se ha seccionado en una serie de fases bien distinguidas, que se han llevado a cabo en un lapso de tres meses. A cada se les ha dedicado un número de horas que coinciden con las detalladas a continuación, con un coste económico, considerando que la remuneración asciende a 15 €/h.

Fases	Duración (h)	Coste (€)
Análisis	80 h	1200 €
Diseño	30 h	450 €
Implementación	160 h	2400 €
Total	270 h	4050 €

Tabla 7.1. Estimación de tiempo y coste.

También es importante, dadas las características del proyecto, considerar una serie de materiales necesarios para el desarrollo del mismo:

Materiales	Coste (€)
Smartphone de gama media	190 €
Total	190 €

Tabla 7.2. Estimación del coste de materiales.

Apéndice A.

Enlaces del proyecto

[Manual del desarrollador](#)

Bibliografía

Provitao: <http://provitao.ull.es/web/index.html>

API Unity: <http://docs.unity3d.com/ScriptReference/>

Drag&Drop: https://en.wikipedia.org/wiki/Drag_and_drop

Sqlite: <https://www.sqlite.org/download.html>

SqliteBrowser: <http://sqlitebrowser.org/>

IDEFICS: <http://www.eufic.org/article/es/artid/Como-prevenir-obesidad-infantil-nuevos-datos-estudio-IDEFICS/>