



GUESSIT!

APP PARA APRENDIZAJE COMUNITARIO DE IDIOMAS CON MONITORIZACION

v1

20 DE OCTUBRE DE 2014

AUTOR: ALBERTO CEJAS SÁNCHEZ



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA INFORMÁTICA

Guess It!, APP para aprendizaje comunitario de idiomas con monitorizacion

- Departamento: Ingeniería Informática y Filología Francesa e Inglesa
- Directores del proyecto: Manuel Palomo Duarte, Anke Berns
- Autor del proyecto: Alberto Cejas Sánchez

Cádiz, 20 de octubre de 2014

Fdo: Alberto Cejas Sánchez

Agradecimientos

Gracias a mi padre, fuente de motivación y orgullo. Gracias a mi madre por aguantar las consecuencias de las mañanas poco productivas.

Este proyecto no se podría haber llevado adelante sin la ambición, el apoyo y el impulso constante de Manuel Palomo y Anke Berns. Gracias también a los colaboradores por su inestimable ayuda.

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright (c) 2014 Alberto Cejas Sánchez.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Notación y formato

Cuando nos refiramos a un programa o biblioteca en concreto, utilizaremos la notación:

Eclipse.

Cuando nos refiramos a un fragmento de código, usaremos la notación:

```
System.out.println("Hola mundo");
```

Cuando nos refiramos a algún comando introducido en la terminal, usaremos la notación:

```
sudo apt-get install
```


Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Estructura del documento	2
2. Descripción general del proyecto	3
2.1. Descripción	3
2.2. Características del videojuego	3
2.2.1. Componentes del juego	4
2.2.2. Vision general del sistema	4
2.3. Colaboradores	5
3. Planificación	7
3.1. Fase inicial	7
3.2. Fase de análisis	7
3.3. Fase de aprendizaje	7
3.4. Fase de desarrollo	8
3.5. Pruebas y correcciones	8
3.6. Redacción de la memoria	9
3.7. Diagrama de Gantt	9
4. Análisis	13
4.1. Especificación de requisitos del sistema	13
4.1.1. Requisitos de interfaces externas	13
4.1.2. Requisitos funcionales	15
4.1.3. Requisitos de rendimiento	15
4.1.4. Restricciones de diseño	16
4.1.5. Requisitos del sistema software	16
4.2. Modelo de casos de uso	16
4.2.1. Diagrama de los casos de uso	16
4.2.2. Descripción de los casos de uso	18
4.3. Modelo conceptual de datos	23
4.3.1. Diagrama de Entidad-Relación	23
4.4. Modelo de comportamiento del sistema	24

4.4.1. Diagramas de secuencia y contrato de las operaciones del sistema.	24
5. Diseño	55
5.1. Interfaz gráfica	55
6. Implementación	73
6.1. Layout	73
6.2. Multi-idioma 100 %	73
6.3. Carga de paquetes de idiomas	74
6.4. Tareas automatizadas	74
6.5. Selección aleatoria de definiciones	74
6.6. Estadísticas dinámicas	74
7. Pruebas	77
7.1. Pruebas unitarias	78
7.2. Pruebas de integración	78
7.3. Pruebas de jugabilidad	78
7.4. Pruebas de interfaz	79
8. Camino Recorrido	81
8.1. Análisis científico del caso de estudio	81
8.2. Difusión	82
9. Conclusiones	87
9.1. Resumen de objetivos	87
9.2. Conclusiones personales	87
9.3. Mejoras y ampliaciones	88
A. Herramientas utilizadas	89
A.1. Lenguaje de programación del cliente	89
A.2. Lenguaje de programación de los scrpits	89
A.3. Base de datos	89
A.4. Frontend	90
A.5. Backend	90
A.6. Adobe Illustrator y Adobe Photoshop	91
A.7. Sistema de control de versiones	91
A.8. Documentación del código	91
A.9. Redacción de la memoria	92
A.10. Realización de diagramas: Dia	92
B. Manual de instalación	93
B.1. Cliente	93
B.2. Interfaz web del profesor	93
C. Manual de usuario	95
Bibliografía y referencias	99
GNU Free Documentation License	101

1. APPLICABILITY AND DEFINITIONS	101
2. VERBATIM COPYING	103
3. COPYING IN QUANTITY	103
4. MODIFICATIONS	103
5. COMBINING DOCUMENTS	105
6. COLLECTIONS OF DOCUMENTS	105
7. AGGREGATION WITH INDEPENDENT WORKS	105
8. TRANSLATION	106
9. TERMINATION	106
10. FUTURE REVISIONS OF THIS LICENSE	106
11. RELICENSING	106
ADDENDUM: How to use this License for your documents	107

Indice de figuras

2.1. Logo GuessIt! : Language Trainer	3
2.2. Diagrama visión general del sistema	5
3.1. Diagrama de Gantt, parte 1.	10
3.2. Diagrama de Gantt, parte 2.	11
4.1. Análisis: Diagrama de casos de uso App	17
4.2. Análisis: Diagrama Entidad-Relación	24
4.3. Análisis: Diagrama de secuencia Selección idioma	25
4.4. Análisis: Diagrama de secuencia Login (escenario principal)	26
4.5. Análisis: Diagrama de secuencia Login (escenario 4a)	27
4.6. Análisis: Diagrama de secuencia Registro (escenario principal)	28
4.7. Análisis: Diagrama de secuencia Registro (escenario 3a)	29
4.8. Análisis: Diagrama de secuencia Menu Principal(escenario principal)	30
4.9. Análisis: Diagrama de secuencia Menu Principal(escenario 3a)	30
4.10. Análisis: Diagrama de secuencia Menu Principal(escenario 3b)	31
4.11. Análisis: Diagrama de secuencia Menu Principal(escenario 3c)	32
4.12. Análisis: Diagrama de secuencia Menu Principal(escenario 3d)	32
4.13. Análisis: Diagrama de secuencia Jugar (escenario principal)	33
4.14. Análisis: Diagrama de secuencia Jugar (escenario principal)	34
4.15. Análisis: Diagrama de secuencia Jugar (escenario 6a)	38
4.16. Análisis: Diagrama de secuencia Jugar (escenario 7a)	39
4.17. Análisis: Diagrama de secuencia Jugar (escenario 7b)	40
4.18. Análisis: Diagrama de secuencia Jugar (escenario 8a)	41
4.19. Análisis: Diagrama de secuencia Jugar (escenario 8b)	43
4.20. Análisis: Diagrama de secuencia Jugar (escenario 8c)	45
4.21. Análisis: Diagrama de secuencia Jugar (escenario 9a)	47
4.22. Análisis: Diagrama de secuencia Ver créditos (escenario principal)	48
4.23. Análisis: Diagrama de secuencia Añadir Definición (escenario principal)	49
4.24. Análisis: Diagrama de secuencia Bloc de Notas (escenario principal)	50
4.25. Análisis: Diagrama de secuencia Estadísticas juego (escenario principal)	51
4.26. Análisis: Diagrama de secuencia Estadísticas de definiciones (escenario principal)	51
4.27. Análisis: Diagrama de secuencia Revisar última partida (escenario principal)	52

5.1. Navegación entre pantallas	56
5.2. Pantalla de selección de idioma	57
5.3. Pantalla de login	58
5.4. Pantalla de registro	59
5.5. Pantalla menú principal	60
5.6. Pantalla de opciones de juego	61
5.7. Pantalla de juego	62
5.8. Pantalla de evaluación	63
5.9. Pantalla de Última Partida	64
5.10. Pantalla de Bloc de Notas	65
5.11. Pantalla de Nueva Definición	66
5.12. Pantalla de Créditos	67
5.13. Pantalla de Estadísticas	68
5.14. Pantalla de Estadísticas de Juego	69
5.15. Pantalla de Estadísticas de Definiciones	70
5.16. Interfaz del profesor	70
5.17. Diseño: Diagrama lógico BD	71
8.1. Ratio de acierto por porcentaje de definiciones	82
8.2. I Concurso de Proyectos Científicos de la UCA	82
8.3. VIII Concurso Universitario de Software Libre	83
8.4. Feria de las Ciencias de Granada	83
8.5. Escuela Oficial de Idiomas (San Fernando)	84
8.6. AtrÉBT!	84
8.7. EC-TEL 2014	85
8.8. Noche Europea de los Investigadores	85
8.9. TEEM Conference 2014	86
A.1. Herramientas utilizadas: Logo de Python	89
A.2. Herramientas utilizadas: Logo de Mysql	90
A.3. Herramientas utilizadas: Logo de Codeigniter	90
A.4. Herramientas utilizadas: Logo de PHP	91
A.5. Herramientas utilizadas: Logo de Adobe Illustrator	91
A.6. Herramientas utilizadas: Logo de L ^A T _E X	92
A.7. Herramientas utilizadas: Logo de Dia	92
B.1. GuessIt! : Language Trainer	93

Introducción

1.1. Motivación

Tras el largo proceso de desarrollo de un videojuego de fútbol en 3D para el PFC de la Ingeniería Técnica Informática, mi objetivo en la Ingeniería Informática seguía siendo mantenerme en el mismo campo pero en una línea más simplista y a la vez perfeccionista, sin dejar pasar la oportunidad de aprender nuevas tecnologías y plataformas que me pudiesen servir en un futuro no muy lejano para dar el salto a ciertos sectores concretos del mundo laboral.

El vacío de ideas sobre qué juego llevar a cabo duró poco, ya que Manuel Palomo me puso en contacto con Anke Berns y entre los tres, confeccionamos un proyecto innovador para plataformas móviles. El reto era interesante porque cumplía todos los requisitos anteriormente expuestos, así que decidí aceptarlo.

Además de lo que ya sabía que encontraría, descubrí otros factores que me ayudarían a mantener el mismo nivel de motivación del primer día. Algunos ejemplos son el hecho de aunar distintas disciplinas, la ambición del equipo por aportar valor añadido a mi PFC a través de publicaciones y la calidad humana.

1.2. Objetivos

El objetivo principal del proyecto es realizar un videojuego sencillo que permita aprender idiomas, enfocado al vocabulario, en cualquier lugar y sin mínimos de tiempo. Además, el jugador puede ser partícipe de su propio aprendizaje seleccionando el nivel y las categorías que quiere jugar, evaluando las deficiencias ya existentes y aportando nuevas.

Los destinatarios principales del videojuego son estudiantes de idiomas dentro de cualquier centro docente supervisado por un profesor. Sin descartar a los jugadores casuales autodidactas, quienes suelen dedicar a jugar pequeños intervalos de tiempo libre.

1.3. Estructura del documento

Este documento esta compuesto por las siguientes partes:

- **Introducción:** pequeña descripción del proyecto, así como los objetivos y estructura del documento.
- **Descripción general:** descripción más amplia sobre el proyecto, así como todas las características relevantes que tendrá.
- **Planificación:** exposición de la planificación del proyecto y las distintas etapas que esta compuesto el mismo.
- **Análisis:** fase de análisis del sistema, empleando la metodología seleccionada. Se definirán los requisitos funcionales del sistema, diagramas de caso de uso, diagramas de secuencia y contrato de las operaciones.
- **Diseño:** realización del diseño del sistema, diagramas de secuencia y clases aplicadas al diseño.
- **Implementación:** aspectos más relevantes durante la implementación del proyecto. Y problemas que han aparecido durante el desarrollo de este.
- **Pruebas y validaciones:** pruebas realizada a la aplicación, con el fin de comprobar su correcto funcionamiento y cumplimiento de las expectativas.
- **Camino recorrido:** reconocimientos y publicaciones relacionadas con la aplicación.
- **Conclusiones:** conclusiones obtenidas tras el desarrollo de la aplicación.
- **Apéndices:**
 - **Herramientas utilizadas:** explicación de todas las herramientas usadas a lo largo del desarrollo del proyecto.
 - **Manual de instalación:** manual para la correcta instalación del proyecto en el sistema.
 - **Manual de usuario:** manual de usuario para el correcto uso de la aplicación.
- **Bibliografía:** libros y referencias consultadas durante el desarrollo del proyecto.
- **Licencia GNU GFDL:** texto completo sobre la licencia GNU GFDL en inglés.

Capítulo 2

Descripción general del proyecto

2.1. Descripción

El juego consiste en adivinar palabras a partir de su definición y una pista opcional. Se trata de un juego clásico adaptado a las nuevas tecnologías móviles con el añadido de que los jugadores pueden aportar contenido nuevo sin la necesidad de que tenga que ser estrictamente supervisado por el docente. Está implementado para dispositivos Android aunque sería portable a iOS o Windows Phone.



Figura 2.1: Logo GuessIt! : Language Trainer

Paralelamente al cliente Android, encontramos el backend situado en el servidor y que cumple la misión de proveer a los usuarios con la información (definiciones, estadísticas, etc) requeridas.

Por último, el profesor podrá tener control absoluto del contenido del juego así como acceso exclusivo a estadísticas que le ayudarán a detectar fallos comunes en un grupo de alumnos a través de una plataforma web.

2.2. Características del videojuego

El videojuego ofrece una alternativa libre y gratuita para aprender idiomas en cualquier lugar y momento. Sigue una filosofía consistente en girar en torno a la comunidad tanto para darle servicio como para retroalimentarse. Esto la hace el eje de la aplicación y el factor diferencial de las ya existentes en el mercado.

2.2.1. Componentes del juego

En esta sección se hará una pequeña descripción de los distintos componentes que conforman el juego.

Definiciones

Son cadenas de texto que describen una o varias palabras concretas, las cuales en el caso en que estén presentes en la propia definición dentro de la pantalla de juego, se sustituirán por huecos.

Categorías

Son las distintas temáticas en las que se agrupan las definiciones.

Niveles

Establecen la dificultad de entender la definición o de la palabra a adivinar y van del uno al cuatro, cubriendo el nivel A1 del Marco Común Europeo de Referencia para las lenguas (MCER).

Usuarios

Los jugadores tendrán una cuenta personal con su información de login para poder acceder desde cualquier dispositivo. Esto también permite identificar inequívocamente las partidas que hayan jugado cada uno.

Estadísticas

A partir de la información almacenada por cada partida jugada, se extraen datos estadísticos para mostrarlos tanto en el cliente como en la interfaz web del profesor.

Bloc de notas

Los jugadores tendrán una sección donde recopilar las palabras que quieran revisar posteriormente.

2.2.2. Vision general del sistema

El principal objetivo es crear un ecosistema donde los jugadores(estudiantes) y los docentes interactúen directamente entre ellos a través de la información almacenada en el servidor. El siguiente diagrama lo ejemplifica visualmente:

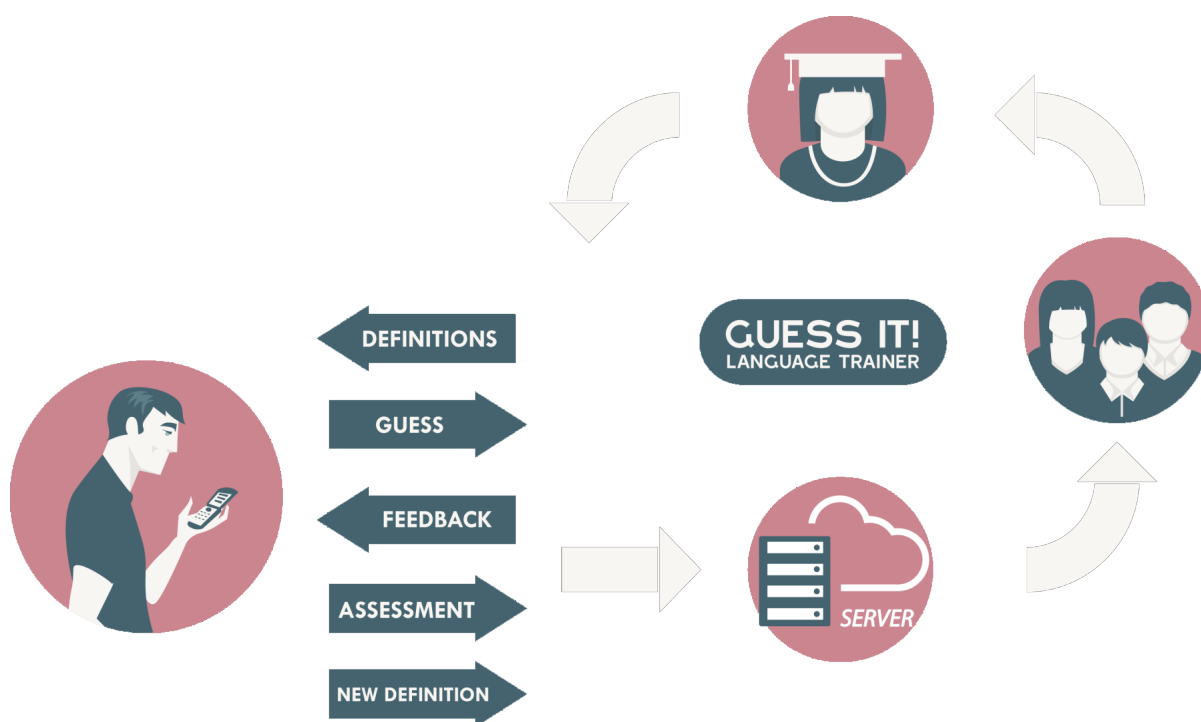


Figura 2.2: Diagrama visión general del sistema

2.3. Colaboradores

Todo el apartado del proyecto referente a la programación del mismo se ha realizado de forma individual. En cambio, en cuanto al diseño gráfico, se ha llevado a cabo tanto individualmente como con ayuda externa. Por último, los paquetes de idiomas están realizados por nativos o expertos. Los créditos se pueden encontrar dentro del juego y son los siguientes:

Diseño y Concepto Manuel Palomo, Anke Berns, Alberto Cejas Sánchez.

Programación Alberto Cejas Sánchez.

Diseño Gráfico

- Iconos categorías: <http://www.flaticon.com>.
- Icono app: José Carlos Cejas Sánchez.
- Interfaz web: Bootstrap 3.
- Resto: Alberto Cejas Sánchez.

Paquetes de Idiomas

- Alemán: Anke Berns, Sebastian Pitula, Andrea Calderón Márquez, Ignacio Calleja Olmedo y Pilar Romero Sevilla.

- Inglés: Anke Berns, Sebastian Pitula, Andrea Calderón Márquez e Ignacio Calleja Olmedo.
- Ruso: Andrés Santana Arribas, Mariya Maiseyenka y Anna Kazakova.

Capítulo 3

Planificación

La planificación realizada para el desarrollo del proyecto, está dividida en varias partes:

3.1. Fase inicial

La primera fase consistió en plantear la idea del proyecto, con la ayuda de Manuel Palomo y Anke Berns. Tras plantear varios enfoques, se decidió realizar este proyecto debido a las motivaciones escritas anteriormente.

También se pensó en que lenguaje se desarrollaría el proyecto, así como las principales bibliotecas que se usarían durante la realización del mismo, priorizando siempre opciones libres. Finalmente se optó por utilizar el SDK oficial de Android (Java) por su documentación, comunidad y robustez.

3.2. Fase de análisis

Esta etapa está dividida principalmente en las dos partes siguiente:

- **Especificación de los requisitos:** estudio de los requisitos que deberá cumplir el juego.
- **Recursos necesarios:** recursos necesarios que deberemos usar durante el desarrollo del proyecto.

3.3. Fase de aprendizaje

Dado que hasta ahora no me había embarcado en el desarrollo de una aplicación Android ni tampoco había implementado un servidor web que se comunicara con ésta, tuve que dedicarle tiempo a aprender algunas reglas de diseño, sintaxis del lenguaje y API de librerías.

Esta fase se caracterizó por intentar entender código ya escrito así como leer tutoriales y documentación oficial. Se podría dividir en tres etapas principales:

- **Aprendizaje del SDK de Android:** Las aplicaciones Android siguen un diseño específico basado en Actividades así como un sistema de layout propio. Sin previa experiencia, el código evoluciona desastrosamente. Los principales recursos que utilicé fueron <http://stackoverflow.com> y la documentación oficial.
- **Aprendizaje del arte:** Hasta ahora estaba familiarizado con el retoque fotográfico pero nunca había ilustrado. A partir de videotutoriales y diversas pruebas *ensayo-error* aprendí el manejo básico de los programas de ilustración vectorial.
- **Aprendizaje de otras bibliotecas:** Para algunas funcionalidades del juego requería incluir bibliotecas externas como *Android-support-v7*, *Gson* y *Guava*.
- **Recordar SQL:** Hacía bastante tiempo que no utilizaba un lenguaje de consulta estructurado por lo que tuve que refrescar algunos aspectos básicos como los tipos de *join*.
- **Aprendizaje de lenguajes y tecnologías típicas del desarrollo web:** Para llevar adelante el lado del servidor tuve que aprender conceptos básicos de *PHP* así como el framework *CodeIgniter*. Además, para el estilo visual me inicié en *Bootstrap 3* y para añadirle mayor dinamismo a la interfaz web hice uso de *Javascript*, concretamente *jQuery* y *Ajax*. Por último, para los gráficos aprendí *HighCharts*.

3.4. Fase de desarrollo

Tras la consecución de las etapas anteriores, se comenzó el desarrollo del proyecto. Esta etapa del desarrollo es la más extensa de todas, como es comprensible. Y me fue posible llevarla a cabo gracias a los prototipos que iba implementando conforme aprendía.

- **Base de datos:** Fue lo primero que realicé para tener una idea inicial de la información que manejaría dentro de la aplicación.
- **Cliente:** Es la aplicación Android y funciona con una arquitectura donde la lógica está separada de las vistas. Es el escaparate para el jugador donde se muestra toda la información.
- **Servidor web:** Su misión es funcionar como el intermediario entre el cliente Android y la base de datos. Para ello se encarga de responder a las peticiones del cliente y almacenar la información requerida.
- **Plataforma web del profesor:** Interfaz mediante la cual el docente puede supervisar y modificar la información de la base de datos y al mismo tiempo obtener información estadística acerca del juego.

3.5. Pruebas y correcciones

Una de las etapas más importantes del desarrollo de cualquier proyecto. Esta etapa se realizó en paralelo a la de desarrollo, ya que conforme se implementaron nuevas funcionalidades, se iban probando exhaustivamente bajo el contexto de las distintas situaciones que pudiera darse, hasta obtener el comportamiento esperado.

Además, conforme le iba entregando una versión funcional a Anke Berns o Manuel Palomo, también ejercían de testers, dándome un feedback muy importante para mejorar y corregir fallos.

3.6. Redacción de la memoria

La redacción de la memoria se ha redactado conforme se iba avanzando en el desarrollo del proyecto. Pero tras la finalización de éste, se le ha dedicado más tiempo a su finalización, corrigiendo puntos que finalmente no se han adecuado al producto final.

3.7. Diagrama de Gantt

A continuación se muestra la planificación anteriormente comentada, en su correspondiente diagrama de Gantt:

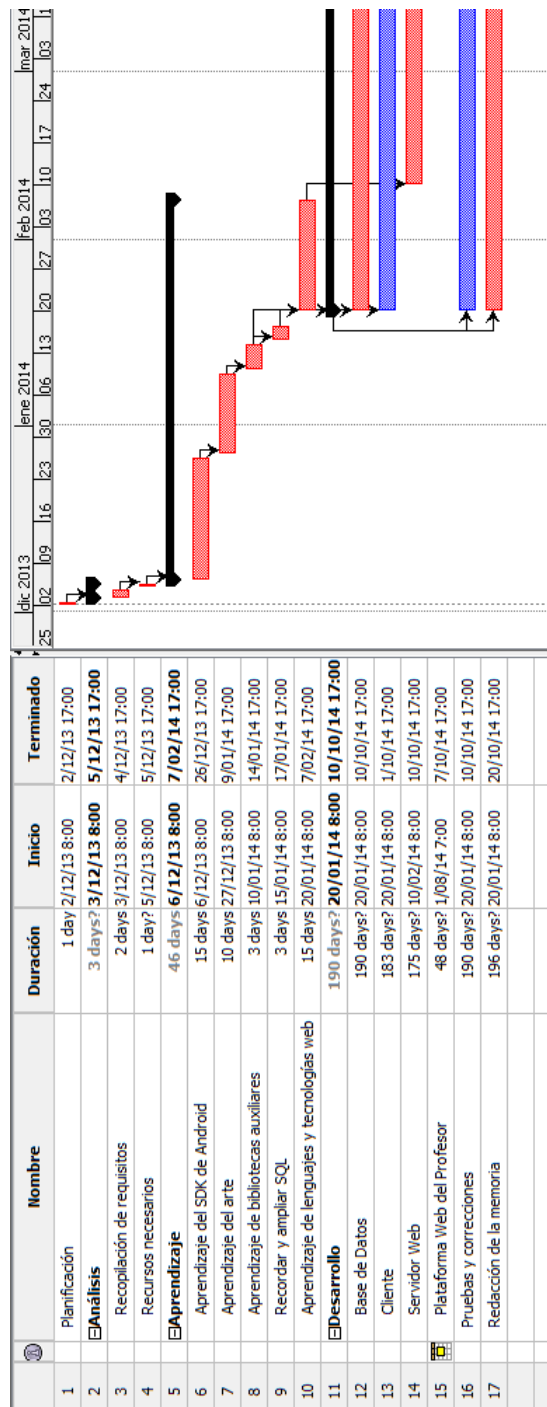


Figura 3.1: Diagrama de Gantt, parte 1.

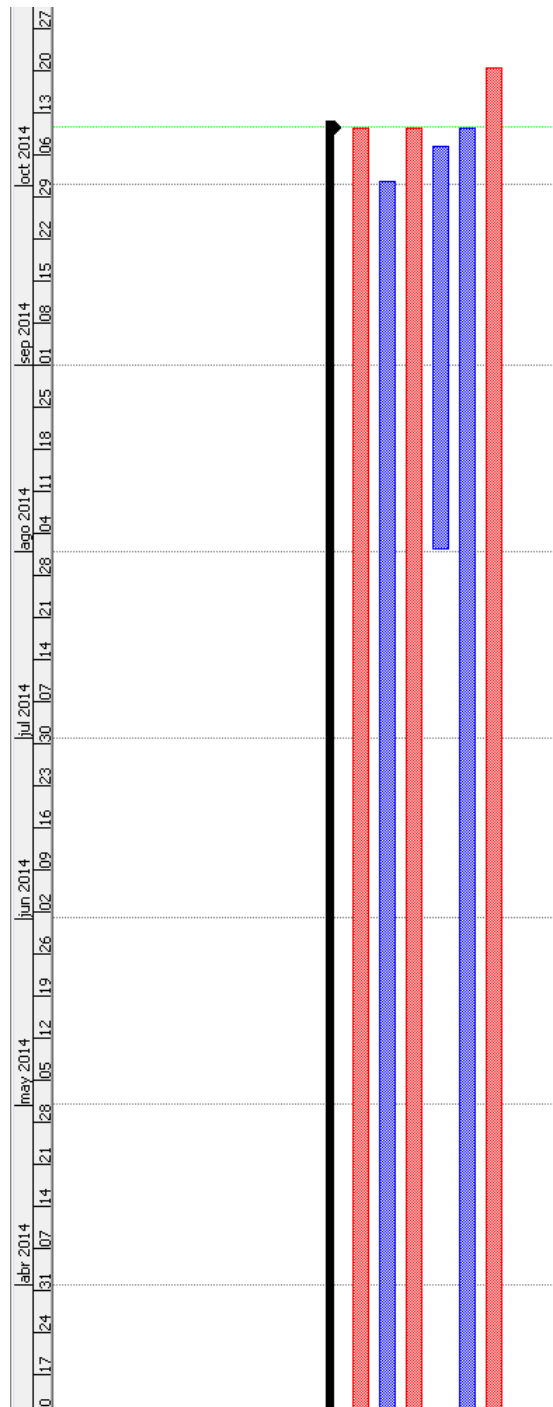


Figura 3.2: Diagrama de Gantt, parte 2.

Capítulo 4

Análisis

4.1. Especificación de requisitos del sistema

Para la creación de cualquier producto software, es necesario establecer las distintas condiciones y necesidades que ha de satisfacer. Seguiremos un esquema que nos permita describir los requisitos de una forma metódica y racional.

4.1.1. Requisitos de interfaces externas

En este apartado se describirá los requisitos de conexión del software y el hardware, así como la interfaz de usuario.

De la conexión entre el software y el hardware se encarga el SDK de Android, por lo que al ser un sistema preestablecido, no será necesario realizar el diseño, ni el análisis, sólo haremos uso de él.

A continuación, pasamos a definir la interfaz del videojuego. Todas las ventanas de la aplicación se adaptarán a la resolución nativa del dispositivo en que se ejecute.

El usuario utilizará la pantalla táctil para interactuar con la interfaz. Por defecto, bastará con una pulsación típica para activar el evento correspondiente. En caso contrario, se mencionará explícitamente.

Las diferentes pantallas que modelarán la aplicación serán:

Selección Idioma Es la pantalla inicial y tiene como función ofrecer al usuario la capacidad de elegir qué idioma quiere jugar.

Login Permite al usuario acceder al contenido de la aplicación ingresando sus credenciales. En caso de que no tenga cuenta, hay un enlace a la pantalla de registro.

Registro Permite al usuario crearse una cuenta en *GuessIt!*. La información introducida será validada y almacenada en la base de datos.

Menú Principal La ventana del menú principal muestra el abanico de opciones disponibles dentro del juego, que son las siguientes:

- Jugar
- Estadísticas
- Bloc de notas
- Nueva definición
- Créditos

Opciones Juego Se accede al pulsar *Jugar* en el menú principal y muestra las distintas categorías a seleccionar con una simple pulsación, así como el/los nivel(es) y el número de preguntas a jugar a través de dos selectores en rueda.

Juego Es la pantalla más importante ya que es donde tiene lugar el juego como tal y muestra la definición de la(s) palabra(s) a adivinar dejando un hueco para escribir dicha palabra. Además también cuenta con un botón rojo para obtener una pista. Por último, en caso de que el idioma tenga diversidad de artículos, se colocará un recordatorio en la parte inferior al que se le añadirá dinámicamente la palabra introducida.

Evaluación Definición Una vez jugada una definición, ésta debe ser corregida por el sistema y evaluada por el propio jugador cuya puntuación oscilará entre uno y tres y se utilizará un selector por estrellas tradicional. Desde esta pantalla también podrá reportar la definición a través de un dedo rojo hacia abajo, si dicha definición contiene errores o el contenido no es adecuado. Por último, se le permitirá al usuario escuchar la pronunciación de la definición entera (con la palabra incluida) y se le ofrecerá la posibilidad de añadir palabras a su bloc de notas a través de una pulsación larga sobre la palabra concreta y posteriormente otra pulsación corta sobre un botón representativo. Si no se ha alcanzado la última pregunta, esta pantalla redirigirá al usuario de nuevo a la pantalla de juego con la próxima definición.

Última Partida Muestra una visión global de la última partida enumerando las definiciones jugadas y colocándoles una marca de acierto o error. Además se puede pulsar en cada una de ellas para volver a la pantalla de Evaluación de Definición (Figura: 5.8)

Bloc de notas Contiene la lista de palabras que el usuario ha añadido a su bloc de notas personal, ordenadas alfabéticamente. Además se puede acceder rápidamente por letra a cualquier entrada del bloc.

Nueva Definición Tras un cierto número de definiciones jugadas, el usuario podrá introducir una definición para una palabra propuesta por el sistema a través de esta pantalla.

Créditos En esta pantalla se mostrarán los creadores y colaboradores de *GuessIt!* desglosados por función.

Estadísticas Recoge información del paso de jugador por *GuessIt!*. Concretamente es un submenú que da acceso a Última Partida (Figura: 5.9), y a las dos próximas pantallas.

Estadísticas de Juego Muestra información acerca del ratio de acierto del jugador, en qué categorías es mejor/peor, los niveles que suele jugar, etc.

Estadísticas de Definiciones Otorga al jugador una visión muy general de la calidad de las definiciones que ha aportado al juego.

Interfaz Web del Profesor Por último, el docente tendrá acceso a:

- Consultar el avance en la última semana de una forma muy general.
- Consultar y gestionar los idiomas, categorías, usuarios, definiciones y estadísticas.
- Importar definiciones automáticamente a partir de un Excel que siga una estructura dada.
- Visualizar diversos tipos de estadísticas que permitan conocer el avance de los jugadores.

4.1.2. Requisitos funcionales

Los requisitos funcionales que el ecosistema debe cumplir son los siguientes:

- Salir de la aplicación desde cualquier ventana.
- Seleccionar idioma a jugar.
- Login.
- Registro.
- Seleccionar categoría a jugar.
- Seleccionar niveles a jugar.
- Seleccionar número de preguntas a jugar.
- Adivinar palabra(s).
- Puntuar una definición.
- Escuchar una definición.
- Añadir una palabra al bloc de notas.
- Revisar la lista general de aciertos/fallos.
- Consultar el bloc de notas.
- Añadir una palabra nueva por cada X jugadas.
- Ver los créditos.
- Mostrar el progreso de los alumnos en la última semana.
- Mostrar datos específicos de juego desgranados por alumno.

Existe un único modo de juego, jugador vs máquina en el que el primero de ellos debe acertar las definiciones planteadas por el segundo.

4.1.3. Requisitos de rendimiento

El rendimiento de la aplicación debe ser tal que permita un desempeño agradable de juego. Los tiempos de respuesta del servidor no deben ser muy largos al transmitir sólo caracteres de texto. Debe cachearse en el cliente toda la información posible para recurrir en las mínimas llamadas al servidor.

Además, debe minimizarse la utilización de recursos cuando sea posible para ahorrar batería.

4.1.4. Restricciones de diseño

Como comento los puntos del apartado anterior el tiempo de respuesta y minimizar la utilización de recursos tiene que primar sobre cualquier factor. Esta será la principal restricción de diseño que tendrá nuestra aplicación.

Será de valor añadido que el paquete .APK final tenga el menor tamaño posible para abarcar aquellos teléfonos con menos memoria secundaria.

Los videojuegos están pensados como aplicación principal, de forma que no tenga que compartir recursos con otros procesos, por lo que se permitirá que consuma gran parte de los recursos del sistema. Sin embargo, debe tener en cuenta que los dispositivos móviles Android (Java) pueden perder el contexto de la aplicación que ejecutan por lo que el contenido debe ser recuperable.

4.1.5. Requisitos del sistema software

La aplicación deberá cumplir los siguientes requisitos del sistema:

- Deberá funcionar como mínimo para versiones de Android superiores a 2.3.6 dando soporte así a más del 88 % del mercado actual según Google el 9 de Septiembre de 2014.
- El código debe ser mantenible y fácilmente ampliable para futuras versiones.

4.2. Modelo de casos de uso

Para describir los distintos comportamientos que tendrá el sistema, usaremos el lenguaje de modelado de sistemas *UML*; que representa los requisitos funcionales del sistema, centrándose en que hace y no cómo lo hace.

4.2.1. Diagrama de los casos de uso

En primer lugar mostramos el modelo de casos de uso (figura 4.1), que representa la funcionalidad completa de la aplicación Android. Se ha usado el siguiente esquema:

1. Identificar los usuarios del sistema y los roles que pueden tener.
2. Para cada rol, identificar las distintas formas de interactuar en el sistema. En el caso de *GuessIt!* existen dos roles de acceso distintos. Sin embargo, en el cliente Android siempre se logueará con funcionalidades de estudiante mientras que en la interfaz web como profesor.
3. Creación de los casos de uso para todos los objetivos que queramos cumplir.
4. Estructurar dichos casos de uso.

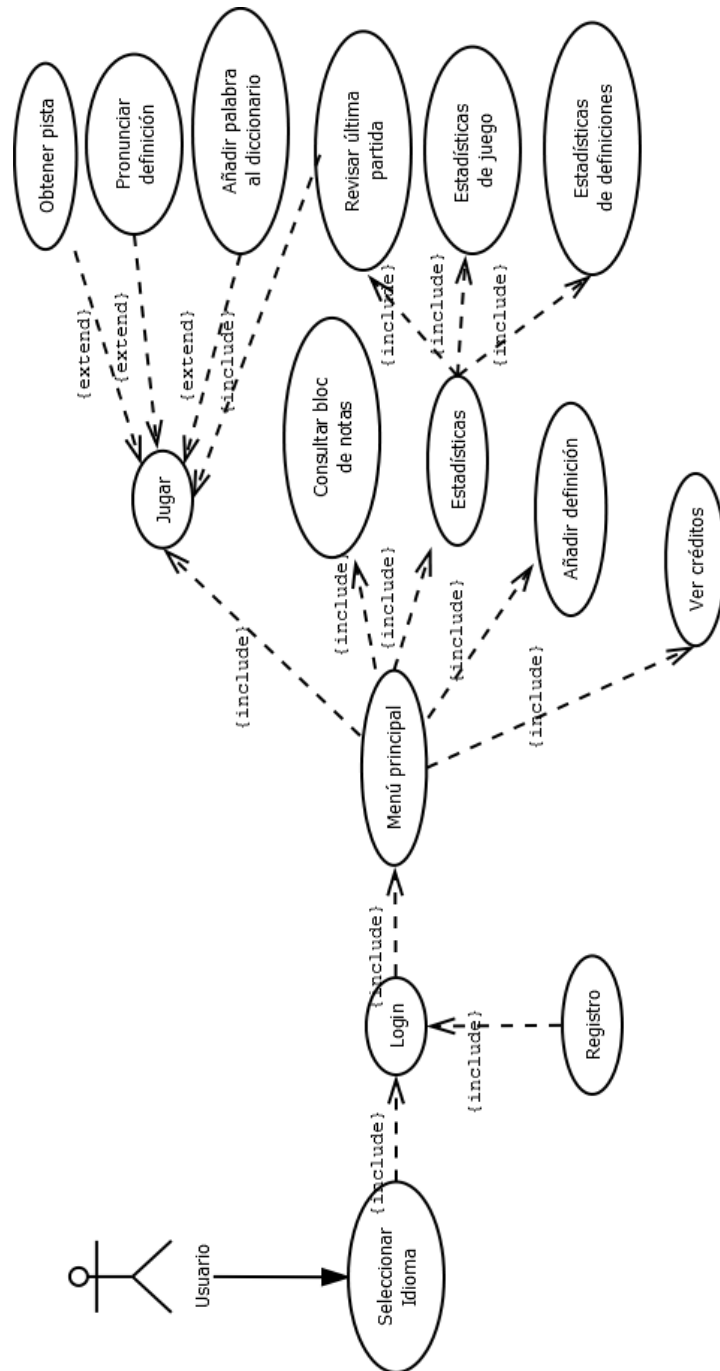


Figura 4.1: Análisis: Diagrama de casos de uso App

4.2.2. Descripción de los casos de uso

A continuación pasamos a la descripción de cada uno de los casos de uso, para la cual usaremos una notación forma usando plantillas. El texto debe ser legible y comprendido por un usuario que no sea experto.

Caso de uso: Seleccionar Idioma

Caso de uso Seleccionar Idioma

Descripción El usuario elige con qué idioma quiere iniciar la aplicación: Inglés, Alemán o Ruso.

Actores Usuario

Precondiciones Ninguna

Postcondiciones La aplicación se inicia para el idioma seleccionado.

Escenario principal

1. El sistema muestra una pantalla con las distintas banderas de los idiomas disponibles.
2. El usuario selecciona una bandera.

Caso de uso: Login

Caso de uso Login

Descripción El usuario se identifica inequívocamente en el sistema.

Actores Usuario

Precondiciones Ninguna.

Postcondiciones El usuario accede al menú principal de la aplicación.

Escenario principal

1. El sistema muestra la pantalla de login.
2. El usuario introduce email y contraseña.
3. El sistema valida los datos.
4. El usuario accede al menú principal.

Extensiones — flujo alternativo

- 4a** El usuario no existe en el sistema y se pide que reintroduzca los datos.

Caso de uso: Registro

Caso de uso Registro

Descripción El usuario se registra en el sistema

Actores Usuario

Precondiciones Ninguna.

Postcondiciones Las credenciales del usuario quedan registradas en el sistema y el usuario logueado en la aplicación.

Escenario principal

1. El sistema muestra la pantalla de registro.
2. El usuario introduce nombre, email y contraseña.
3. El sistema valida los datos.
4. El sistema registra los datos y redirecciona al usuario al menú principal.

Extensiones — flujo alternativo

3a Los datos no son válidos y se pide que se reinserten.

Caso de uso: Menú principal

Caso de uso Menú principal

Descripción Se muestra el menú principal de la aplicación, donde es posible elegir entre jugar, añadir una nueva definición, ver las estadísticas, ver su bloc de notas o ver los créditos.

Actores Usuario

Precondiciones El usuario está previamente logueado en el sistema.

Postcondiciones Ninguna

Escenario principal

1. El sistema muestra el menú principal del juego en pantalla.
2. El usuario selecciona la opción **jugar**.
3. El sistema inicia el juego.

Extensiones — flujo alternativo

3a El usuario selección la opción **estadísticas**.

1. El sistema inicia la opción estadísticas

3b El usuario selección la opción **créditos**.

1. El sistema inicia la opción créditos

3c El usuario selección la opción **añadir nueva definición**.

1. El sistema inicia la opción añadir nueva definición.

3d El usuario selección la opción **bloc de notas**.

1. El sistema inicia la opción bloc de notas.

Caso de uso: Jugar

Caso de uso Jugar

Descripción El usuario juega una partida

Actores Usuario

Precondiciones El usuario está previamente logueado en el sistema.

Postcondiciones El usuario completa una partida.

Escenario principal

1. El sistema carga las categorías y el número de preguntas disponibles para cada una de ellas divididas por dificultad.
2. El sistema muestra la pantalla de opciones de juego.
3. El usuario selecciona las categorías que quiere jugar, la dificultad y el número de preguntas.
4. El sistema carga aleatoriamente el número solicitado de preguntas que cumplan los requisitos.
5. El sistema muestra una definición con un hueco a completar.
6. El usuario escribe la(s) palabra(s) a adivinar.
7. El sistema da por válida la respuesta y muestra el feedback, conjuntamente con tres estrellas y un dedo hacia abajo.
8. El usuario puntúa la definición de 1 a 3 estrellas.
9. El sistema pasa a la siguiente pregunta.

Extensiones — flujo alternativo

- 6.1a** El usuario selecciona el botón de solicitar pista
- 6.2a** El usuario escribe la(s) palabra(s) a adivinar.
- 7a** El sistema rechaza la respuesta y vuelve a 5.
- 7b** El sistema rechaza la respuesta por tercera vez, muestra el feedback y las tres estrellas con el dedo. Posteriormente va a 8.
- 8.1a** El usuario presiona el dedo hacia abajo.
- 8.2a** El sistema abre un diálogo con los distintos motivos por los que reportar la definición.
- 8.3a** El usuario selecciona uno de ellos y a continuación el botón de reportar.
- 8.1b** El usuario selecciona el botón de escuchar la pronunciación.
- 8.2b** El sistema reproduce la definición con acento nativo generado por Google Speech.
- 8.3b** El usuario puntúa la definición de 1 a 3 estrellas.
- 8.1c** El usuario añade una palabra al bloc de notas.
- 8.2c** El usuario puntúa la definición de 1 a 3 estrellas.

9a Se llega al límite de preguntas y el sistema muestra la pantalla de revisión de partida.

Caso de uso: Ver créditos

Caso de uso Ver créditos

Descripción El sistema muestra los autores y colaboradores de la aplicación.

Actores Usuario

Precondiciones El usuario está previamente logueado en el sistema y situado en el menú principal.

Postcondiciones Ninguna.

Escenario principal

1. El usuario pulsa sobre el botón de créditos del menú principal.
2. El sistema muestra los créditos.

Caso de uso: Añadir definición

Caso de uso Añadir definición

Descripción El usuario puede añadir una definición cada X definiciones jugadas para el idioma seleccionado.

Actores Usuario

Precondiciones Se han jugado X definiciones antes para el idioma seleccionado y se está logueado en el sistema.

Postcondiciones Se registra una nueva definición que entra en juego para el resto de usuarios.

Escenario principal

1. El usuario selecciona la opción de Añadir una nueva definición del menú principal.
2. El sistema propone una palabra a definir, enmarcada en una categoría y una dificultad basada en la experiencia de juego del usuario.
3. El usuario introduce la definición, pudiendo recurrir a la propia palabra.
4. El usuario selecciona el artículo si procede e introduce la pista.
5. El sistema registra la nueva definición sustituyendo la(s) palabra(s) a definir por el hueco en blanco si fuese necesario.

Caso de uso: Consultar bloc de notas

Caso de uso Consultar bloc de notas

Descripción Se muestra el contenido del bloc de notas del usuario para el idioma seleccionado.

Actores Usuario

Precondiciones El usuario está previamente logueado en el sistema.

Postcondiciones Ninguna.

Escenario principal

1. El usuario pulsa sobre el botón de Bloc de notas del menú principal.
2. El sistema muestra el bloc de notas personal del usuario para el idioma seleccionado.
3. El usuario puede acceder rápidamente a una palabra por su primera letra a través de un selector invisible en rueda en la zona derecha.

Caso de uso: Estadísticas de juego

Caso de uso Estadísticas de juego

Descripción El sistema muestra información sobre la experiencia de juego del usuario logueado.

Actores Usuario

Precondiciones El usuario está previamente logueado en el sistema y hay un idioma seleccionado.

Postcondiciones Ninguna.

Escenario principal

1. El usuario pulsa sobre el botón de Estadísticas del menú principal y posteriormente en estadísticas de juego.
2. El sistema muestra el ratio de acierto, el número de palabras jugadas, la mejor/peor categoría, el nivel más jugado y los niveles jugados.

Caso de uso: Estadísticas de definiciones

Caso de uso Estadísticas de definiciones

Descripción El sistema muestra información sobre las definiciones aportadas por el usuario logueado.

Actores Usuario

Precondiciones El usuario está previamente logueado en el sistema y hay un idioma seleccionado.

Postcondiciones Ninguna.

Escenario principal

1. El usuario pulsa sobre el botón de Estadísticas del menú principal y posteriormente en estadísticas de definiciones.
2. El sistema muestra la media de puntuación recibida para sus definiciones, el porcentaje de acierto y el número de reportes.

Caso de uso: Revisar última partida

Caso de uso Revisar última partida

Descripción Se muestra toda la información referente a la última partida del usuario.

Actores Usuario, Sistema

Precondiciones El jugador está logueado en el sistema y ha jugado al menos una partida en el idioma seleccionado.

Postcondiciones Ninguna.

Escenario principal

1. El sistema muestra una pantalla resumen de la última partida con los aciertos y los fallos, así como las pistas usadas.
2. El usuario pulsa sobre la pregunta que desea revisar.
3. El sistema muestra la información de juego de esa pregunta, tal y como se vería en la pantalla de juego.

4.3. Modelo conceptual de datos

Este apartado del análisis sirve para especificar los requisitos del sistema y las relaciones estáticas que existen entre ellos.

Para este fin se utiliza como herramienta los diagramas de clase. En estos diagramas se representan las clases de objetos, las asociaciones entre dichas clases, los atributos que componen las clases y las relaciones de integridad.

4.3.1. Diagrama de Entidad-Relación

En la siguiente imagen podemos ver el diagrama ER que representa las entidades relevantes del sistema de información implementado, así como sus interrelaciones y propiedades:

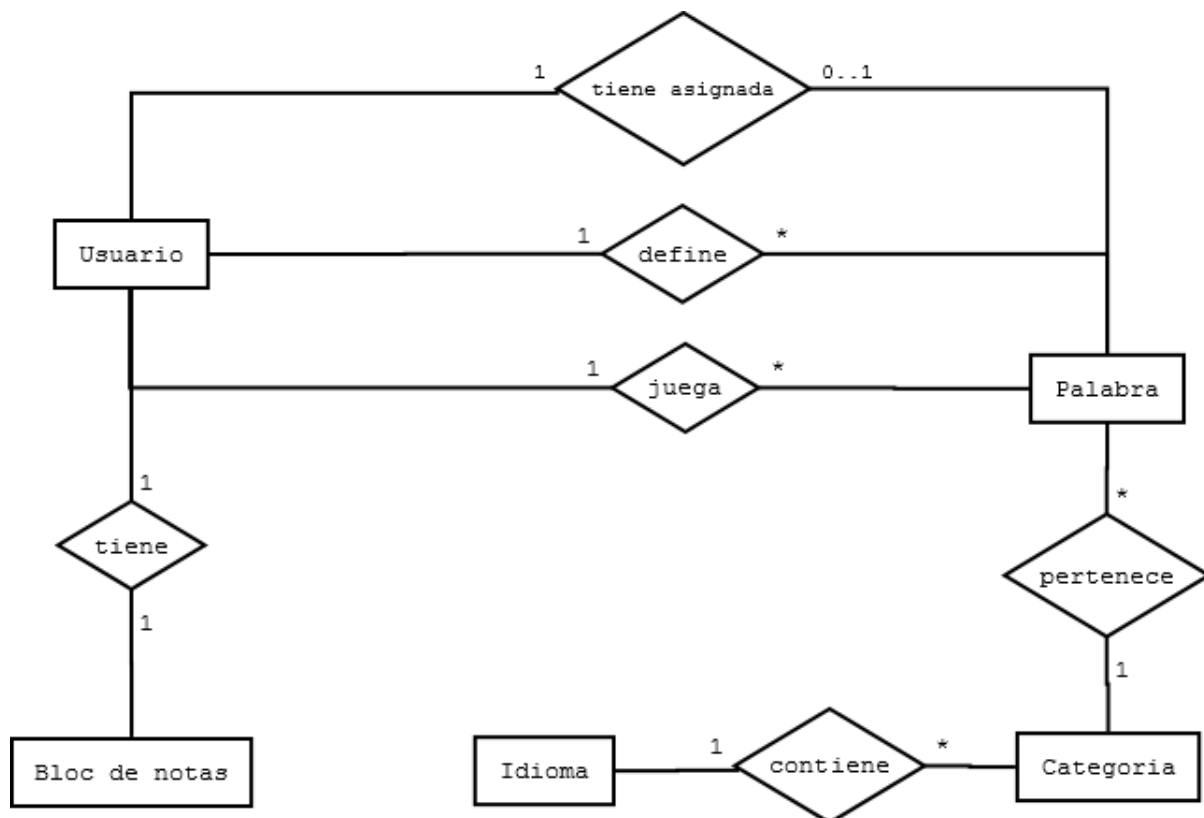


Figura 4.2: Análisis: Diagrama Entidad-Relación

4.4. Modelo de comportamiento del sistema

El modelo de comportamiento especifica como debe actuar el sistema. El sistema es el que engloba todos los objetos, y el modelo consta de dos partes:

- Diagramas de secuencias del sistema: muestran la secuencia de eventos entre el usuario y el sistema.
- Contrato de las operaciones del sistema: describen el efecto que producen las operaciones en el sistema.

4.4.1. Diagramas de secuencia y contrato de las operaciones del sistema.

No todos los posibles diagramas de secuencia aparecerán, nos centraremos en los más importantes, los que implican algún tipo de cambio en el sistema. Para evitar contenidos duplicados, se omitirán las operaciones que hayan sido explicadas con anterioridad.

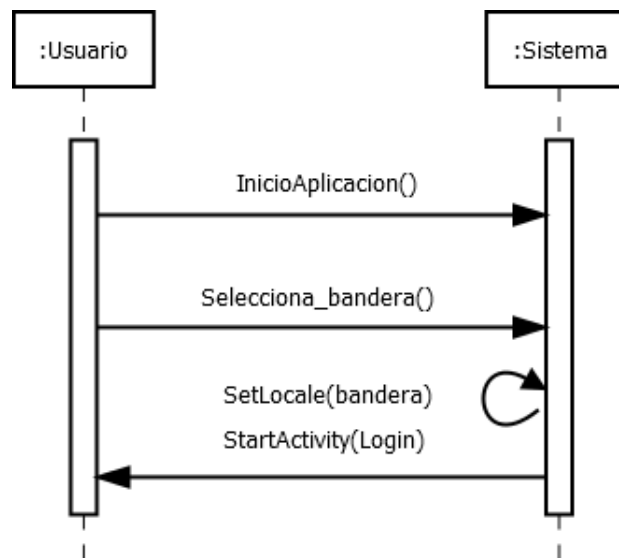
Caso de uso: Selección de idioma

Figura 4.3: Análisis: Diagrama de secuencia Selección idioma

Operación InicioAplicacion()**Actores** Usuario, sistema.**Responsabilidades** inicia la aplicación y muestra el menú de selección de idioma con banderas.**Precondiciones** Ninguna**Postcondiciones** El juego arranca y con ello la posibilidad de interacción con el usuario.**Operación** Selecciona_bandera()**Actores** Usuario.**Responsabilidades** establecer el idioma activo para las actividades esenciales del sistema.**Precondiciones** Las banderas están cargadas en pantalla**Postcondiciones** Se registra la bandera seleccionada**Operación** SetLocale(bandera)**Actores** Sistema.**Responsabilidades** Establecer un localización para el sistema**Precondiciones**

- El idioma por defecto es inglés de Reino Unido.

Postcondiciones

- Se cambia el idioma por defecto al seleccionado.

Operación StartActivity(actividad)

Actores Sistema.

Responsabilidades Carga la actividad recibida, tanto la parte lógica como el layout.

Precondiciones Ninguna

Postcondiciones Se muestra la actividad recibida

Caso de uso: Login

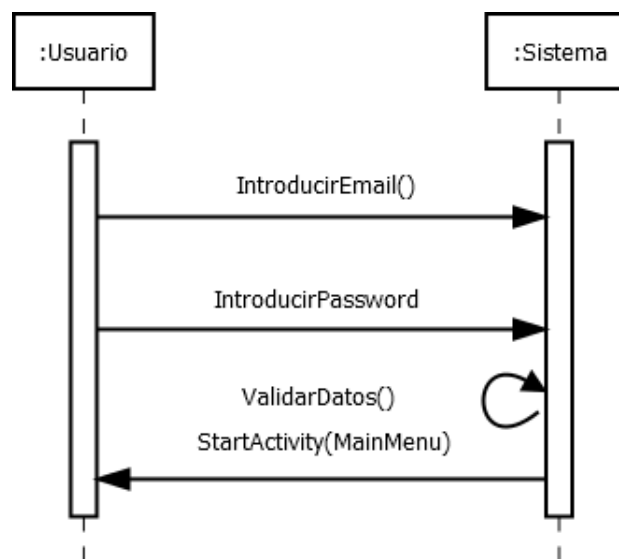


Figura 4.4: Análisis: Diagrama de secuencia Login (escenario principal)

Operación IntroducirEmail()

Actores Usuario.

Responsabilidades Ninguna.

Precondiciones El usuario posee una dirección de correo.

Postcondiciones Ninguna.

Operación IntroducirPassword()

Actores Usuario.

Responsabilidades Ninguna.

Precondiciones Ninguna.

Postcondiciones Ninguna.

Operación Validardatos()

Actores Sistema.

Responsabilidades Comprobar que el email y el password tienen una sintaxis correcta.

Precondiciones El usuario ha introducido un email y un password.

Postcondiciones El sistema registra un inicio de sesión para el usuario identificado.

Caso de uso: Login(escenario 4a)

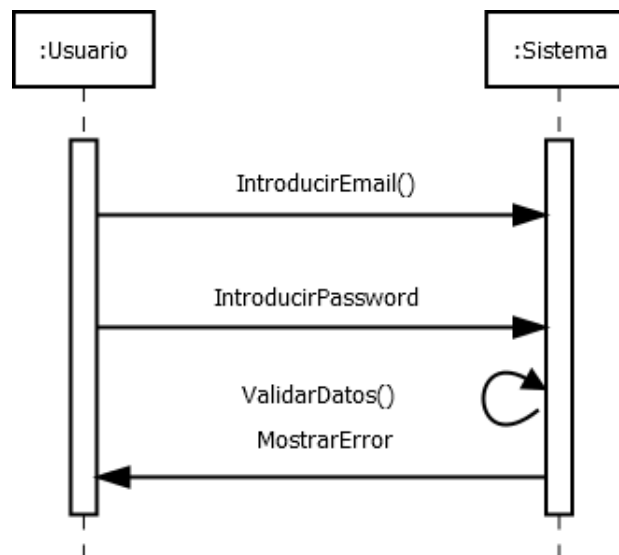


Figura 4.5: Análisis: Diagrama de secuencia Login (escenario 4a)

Operación ValidarDatos()

Actores Sistema.

Responsabilidades Comprobar que el email y el password tienen una sintaxis correcta.

Precondiciones El usuario ha introducido un email y un password.

Postcondiciones Ninguna.

Operación MostrarError()

Actores Sistema.

Responsabilidades Informar al usuario de que los datos introducidos no son válidos

Precondiciones

- El usuario ha introducido email y password.

Postcondiciones

- Se pide al usuario que vuelva a introducir los datos.

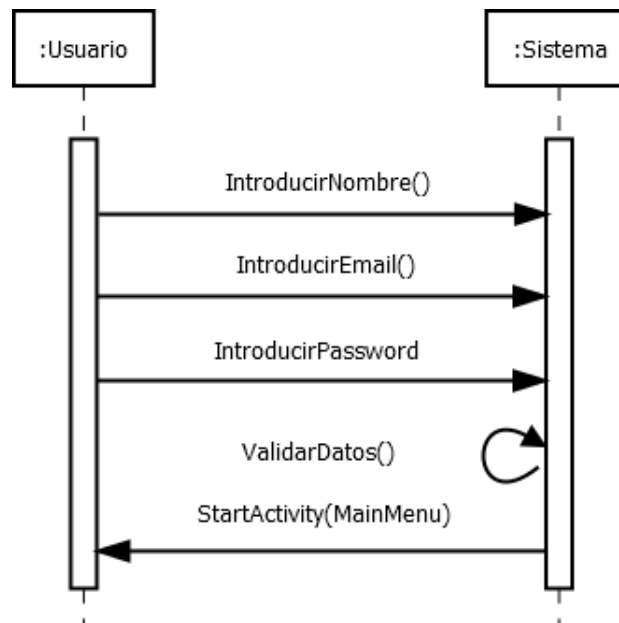
Caso de uso: Registro (escenario principal)

Figura 4.6: Análisis: Diagrama de secuencia Registro (escenario principal)

Operación IntroducirNombre()

Actores Usuario.

Responsabilidades Ninguna.

Precondiciones Ninguna.

Postcondiciones Ninguna.

Operación ValidarDatos()

Actores Sistema.

Responsabilidades Comprobar que el nombre, el email y el password tienen una sintaxis correcta.

Precondiciones El usuario ha introducido un nombre, un email y un password.

Postcondiciones El sistema registra un inicio de sesión para el usuario identificado.

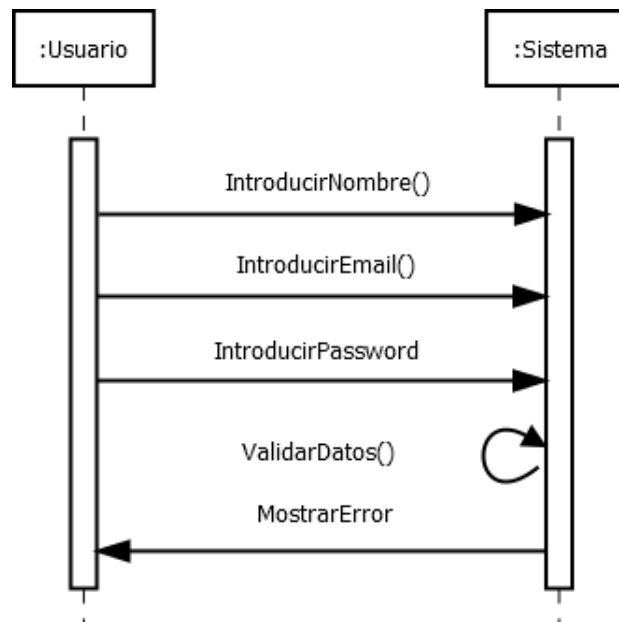
Caso de uso: Registro (escenario 3a)

Figura 4.7: Análisis: Diagrama de secuencia Registro (escenario 3a)

Operación ValidarDatos()**Actores** Sistema.**Responsabilidades** Comprobar que el nombre, el email y el password tienen una sintaxis correcta.**Precondiciones** El usuario ha introducido un nombre, un email y un password.**Postcondiciones** Ninguna.**Operación** MostrarError()**Actores** Sistema.**Responsabilidades** Informar al usuario de que los datos introducidos no son válidos**Precondiciones**

- El usuario ha introducido un nombre, un email y un password.

Postcondiciones

- Se pide al usuario que vuelva a introducir los datos.

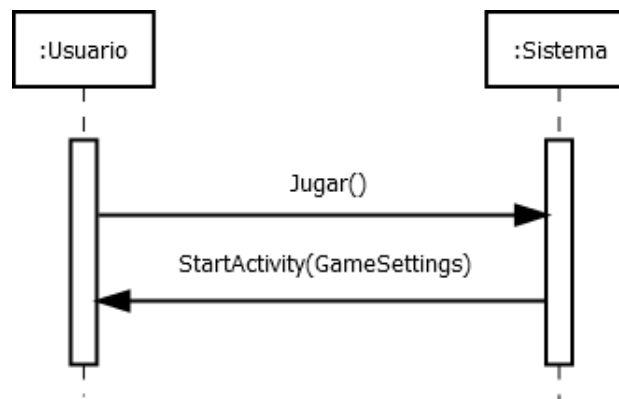
Caso de uso: Menú principal (escenario principal)

Figura 4.8: Análisis: Diagrama de secuencia Menu Principal(escenario principal)

Operación Jugar()**Actores** Jugador, sistema.**Responsabilidades** Ordenar la carga del menú de opciones de juego.**Precondiciones**

- El juego está en el menú principal.

Postcondiciones

- El sistema carga el menú de opciones de juego.

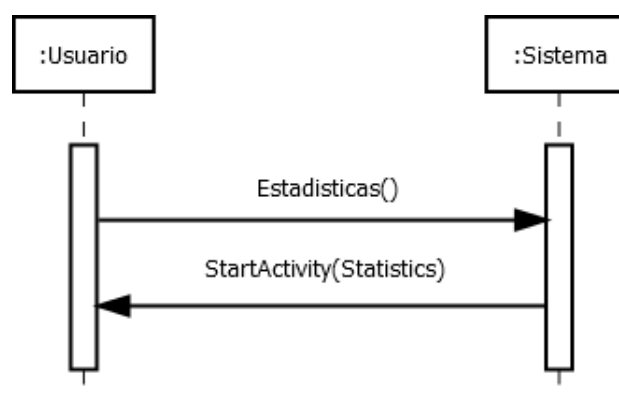
Caso de uso: Menú principal (escenario 3a)

Figura 4.9: Análisis: Diagrama de secuencia Menu Principal(escenario 3a)

Operación Estadísticas()**Actores** Jugador, sistema.

Responsabilidades Ordenar la carga del menú de estadísticas.

Precondiciones

- El juego está en el menú principal.

Postcondiciones

- El sistema carga el menú de estadísticas.

Caso de uso: Menú principal (escenario 3b)

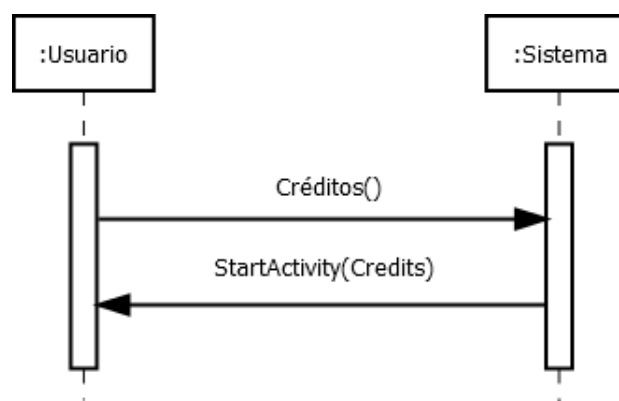


Figura 4.10: Análisis: Diagrama de secuencia Menu Principal(escenario 3b)

Operación Credits()

Actores Jugador, sistema.

Responsabilidades Ordenar la carga de los créditos.

Precondiciones

- El juego está en el menú principal.

Postcondiciones

- El sistema carga los créditos.

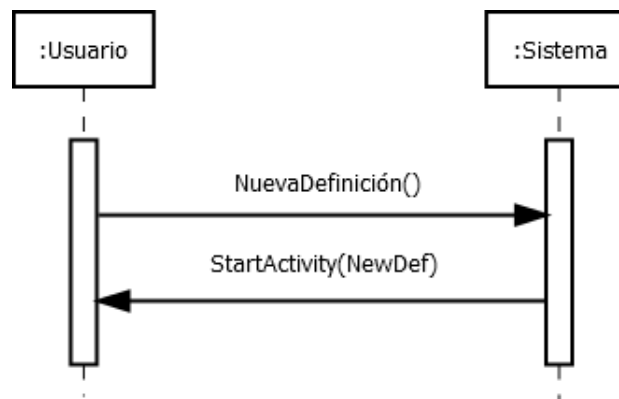
Caso de uso: Menú principal (escenario 3c)

Figura 4.11: Análisis: Diagrama de secuencia Menu Principal(escenario 3c)

Operación NuevaDefinicion()**Actores** Jugador, sistema.**Responsabilidades** Ordenar la carga de la pantalla de nueva definición.**Precondiciones**

- El juego está en el menú principal.

Postcondiciones

- El sistema carga la pantalla de nueva definición.

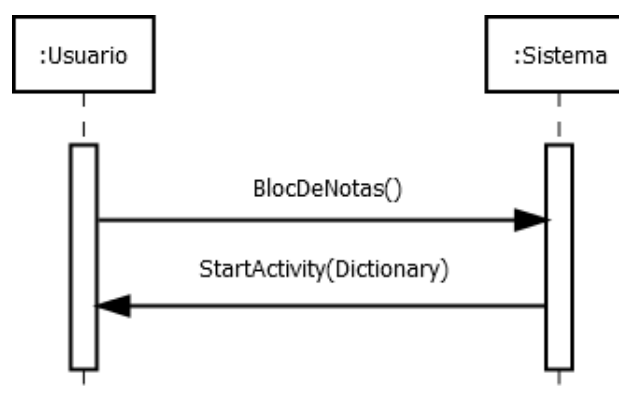
Caso de uso: Menú principal (escenario 3d)

Figura 4.12: Análisis: Diagrama de secuencia Menu Principal(escenario 3d)

Operación BlocDeNotas()**Actores** Jugador, sistema.

Responsabilidades Ordenar la carga de la pantalla del bloc de notas

Precondiciones

- El juego está en el menú principal.

Postcondiciones

- El sistema carga la pantalla del bloc de notas.

Caso de uso: Jugar (escenario principal)

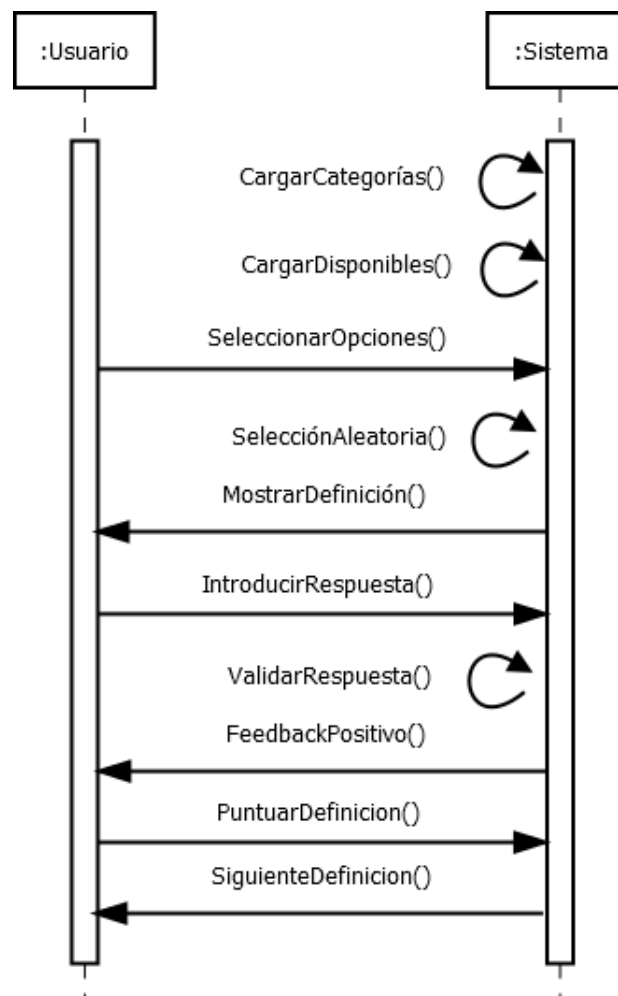


Figura 4.13: Análisis: Diagrama de secuencia Jugar (escenario principal)

Operación CargarCategorías()

Actores Sistema.

Responsabilidades Cargar las categorías disponibles y su correspondiente imagen.

Precondiciones

- El juego está en la pantalla de opciones de juego.
- Hay un idioma seleccionado.
- Existen categorías para el idioma seleccionado en la base de datos.

Postcondiciones

- Aparecen en pantalla las imágenes de las categorías, guardando su modelo lógico.

Caso de uso: Jugar (escenario 6a)

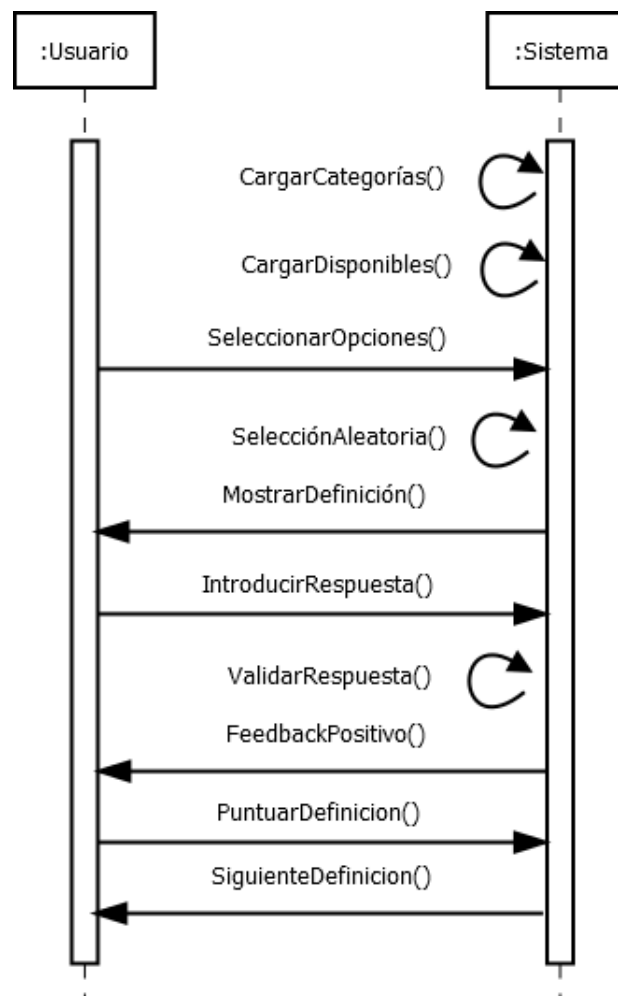


Figura 4.14: Análisis: Diagrama de secuencia Jugar (escenario principal)

Operación CargarCategorías()

Actores Sistema.

Responsabilidades Cargar las categorías disponibles y su correspondiente imagen.

Precondiciones

- El juego está en la pantalla de opciones de juego.
- Hay un idioma seleccionado.
- Existen categorías para el idioma seleccionado en la base de datos.

Postcondiciones

- Aparecen en pantalla las imágenes de las categorías, guardando su modelo lógico.

Operación CargarDisponibles()

Actores Sistema.

Responsabilidades Cargar el número de preguntas disponibles por categoría y dificultad.

Precondiciones

- El juego está en la pantalla de opciones de juego.
- Hay un idioma seleccionado.

Postcondiciones

- Aparecen las ruedas de selección de dificultad y preguntas a jugar.

Operación SeleccionarOpciones()

Actores Usuario, Sistema.

Responsabilidades Selecciona las opciones de juego antes de cargar la partida.

Precondiciones

- El juego está en la pantalla de opciones de juego.
- Hay un idioma seleccionado.

Postcondiciones

- Las rueda de selección de número de preguntas varía el valor que contiene según las categorías y niveles seleccionados.

Operación SelecciónAleatoria()

Actores Sistema.

Responsabilidades Selecciona un subconjunto aleatorio de preguntas según los criterios de selección.

Precondiciones

- El juego está en la pantalla de opciones de juego.
- Hay un idioma seleccionado.
- Hay un subconjunto de categorías seleccionado.
- Hay una dificultad seleccionada.
- Hay un número de preguntas seleccionado.

Postcondiciones

- El sistema carga un subconjunto aleatorio de preguntas que reúnen las condiciones seleccionadas por el usuario.

Operación MostrarDefinición()

Actores Sistema.

Responsabilidades Cargar la pantalla de juego

Precondiciones

- Hay un subconjunto de preguntas seleccionado.

Postcondiciones

- El sistema carga la pantalla de juego que contiene la primera definición a jugar.

Operación IntroducirRespuesta()

Actores Usuario.

Responsabilidades El usuario intenta adivinar la definición

Precondiciones

- Hay una pantalla de juego con una definición activa.

Postcondiciones

- El sistema registra la respuesta del usuario.

Operación ValidarRespuesta()

Actores Sistema.

Responsabilidades El sistema comprueba si la respuesta del usuario coincide con la correcta.

Precondiciones

- El usuario ha introducido una respuesta.

Postcondiciones

- El sistema comprueba si la respuesta es correcta.

Operación FeedbackPositivo()

Actores Sistema.

Responsabilidades El sistema informa al usuario de que la respuesta es correcta.

Precondiciones

- El sistema ha validado la respuesta.

Postcondiciones

- El sistema muestra la palabra acertada en verde, tres estrellas para puntuar y un dedo hacia abajo para reportar.

Operación PuntuarDefinición()

Actores Usuario.

Responsabilidades El usuario puntúa o reporta una definición.

Precondiciones

- El sistema ha cargado la pantalla de puntuación.

Postcondiciones

- El sistema registra la puntuación/report del usuario para esa definición.

Operación SiguienteDefinición()

Actores Sistema.

Responsabilidades El sistema muestra la siguiente definición a jugar.

Precondiciones

- El jugador ya ha puntuado.
- Quedan preguntas por jugar.

Postcondiciones

- El sistema muestra la pantalla de juego con la siguiente palabra a jugar.

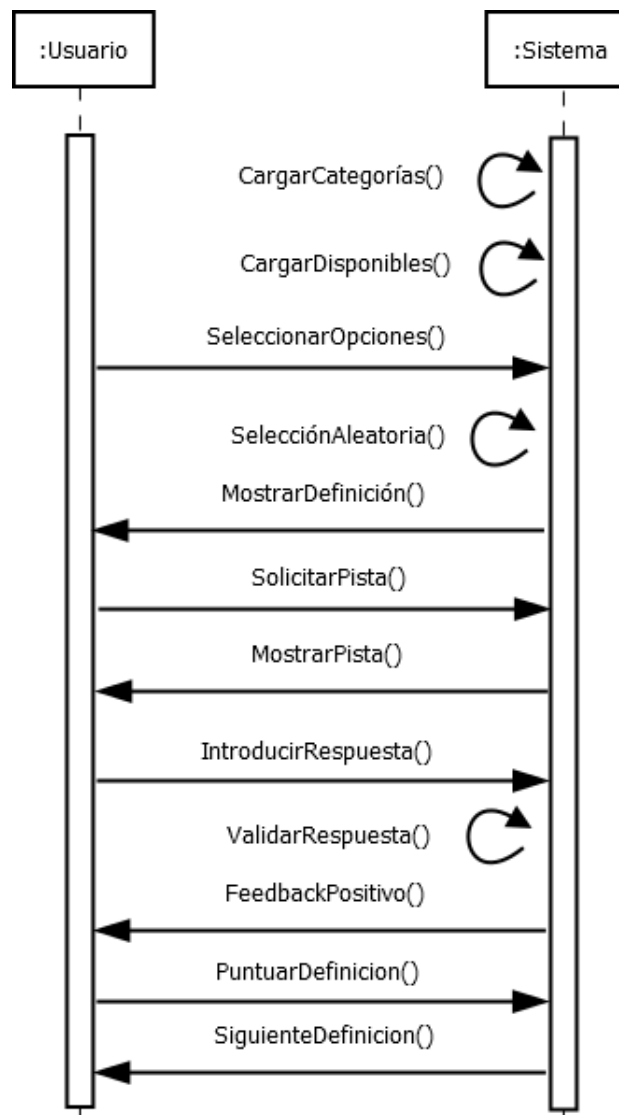
Caso de uso: Jugar (escenario 6a)

Figura 4.15: Análisis: Diagrama de secuencia Jugar (escenario 6a)

Operación SolicitarPista()**Actores** Jugador.**Responsabilidades** El usuario solicita una pista.**Precondiciones**

- La pantalla actual es la de juego.

Postcondiciones

- Ninguna.

Operación MostrarPista()

Actores Sistema.

Responsabilidades Mostrar una pista para adivinar la definición.

Precondiciones

- La pantalla actual es la de juego.
- El usuario ha solicitado una pista.

Postcondiciones

- Se muestra la pista en un diálogo.

Caso de uso: Jugar (escenario 7a)

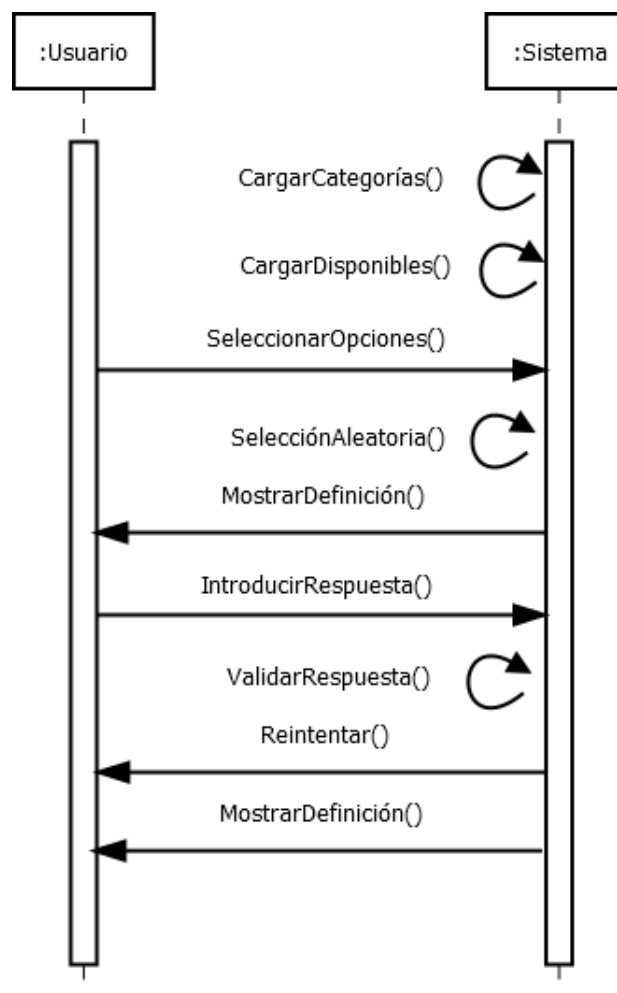


Figura 4.16: Análisis: Diagrama de secuencia Jugar (escenario 7a)

Operación Reintentar()

Actores Sistema.

Responsabilidades El sistema informa del fallo.

Precondiciones

- El usuario ha introducido una respuesta incorrecta.

Postcondiciones

- El sistema informa al jugador que tiene 3 intentos y el intento actual.

Caso de uso: Jugar (escenario 7b)

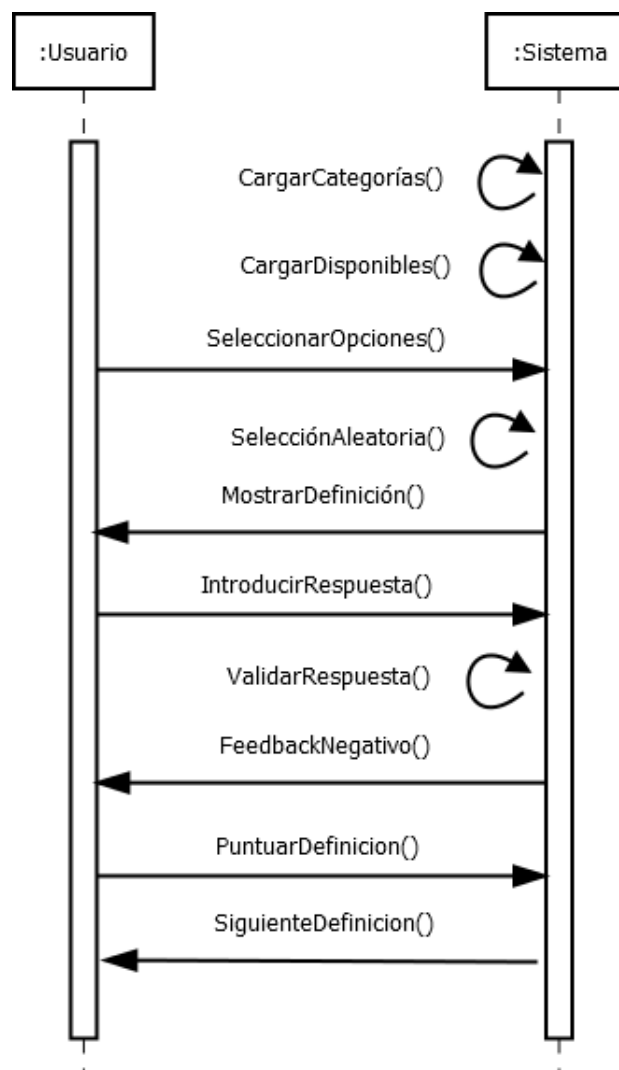


Figura 4.17: Análisis: Diagrama de secuencia Jugar (escenario 7b)

Operación FeedbackNegativo()

Actores Sistema.

Responsabilidades El sistema informa del fallo y da feedback de la respuesta correcta.

Precondiciones

- El usuario ha introducido una respuesta incorrecta 3 veces.

Postcondiciones

- El sistema muestra la respuesta correcta y abre la pantalla de puntuación/report.

Caso de uso: Jugar (escenario 8a)

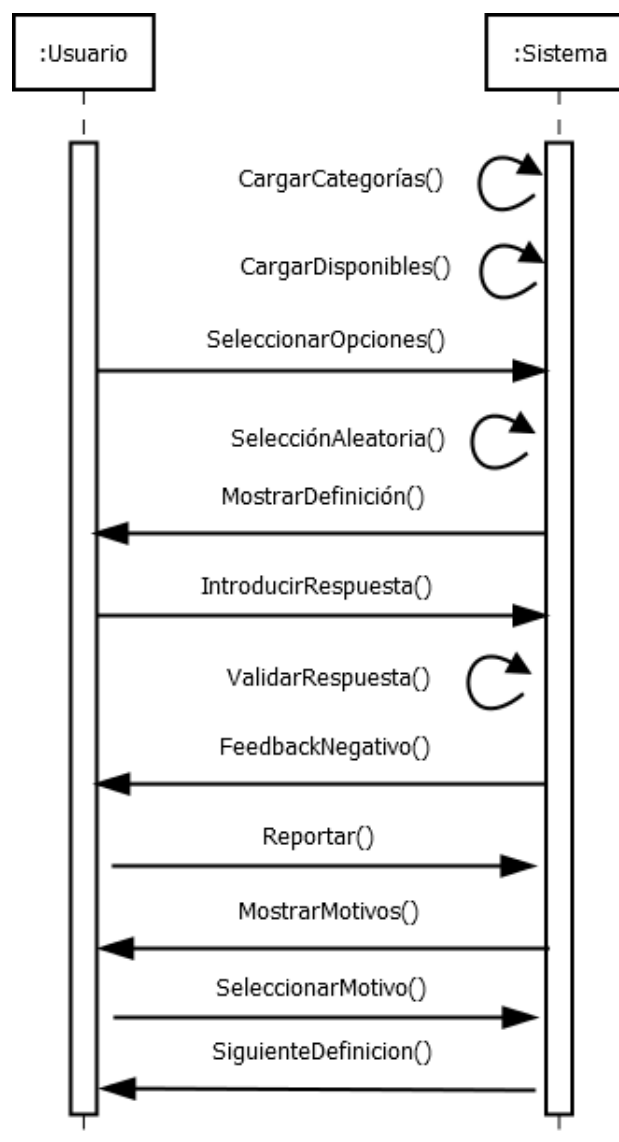


Figura 4.18: Análisis: Diagrama de secuencia Jugar (escenario 8a)

Operación Reportar()

Actores Usuario.

Responsabilidades Informar al sistema de la intención por parte del usuario de reportar la definición actual.

Precondiciones

- El usuario ha jugado una definición.
- El usuario ha seleccionado el botón de reportar.

Postcondiciones Ninguna.

Operación MostrarMotivos()

Actores Sistema.

Responsabilidades Muestra los motivos disponibles para reportar.

Precondiciones

- El usuario ha pulsado sobre el botón de reportar definición.

Postcondiciones Se muestra un diálogo con los motivos disponibles para reportar.

Operación SeleccionarMotivos()

Actores Sistema, Usuario.

Responsabilidades Registrar el motivo seleccionado por el usuario

Precondiciones

- El usuario ha seleccionado un motivo.

Postcondiciones El sistema registra el motivo de reporte seleccionado para esa definición.

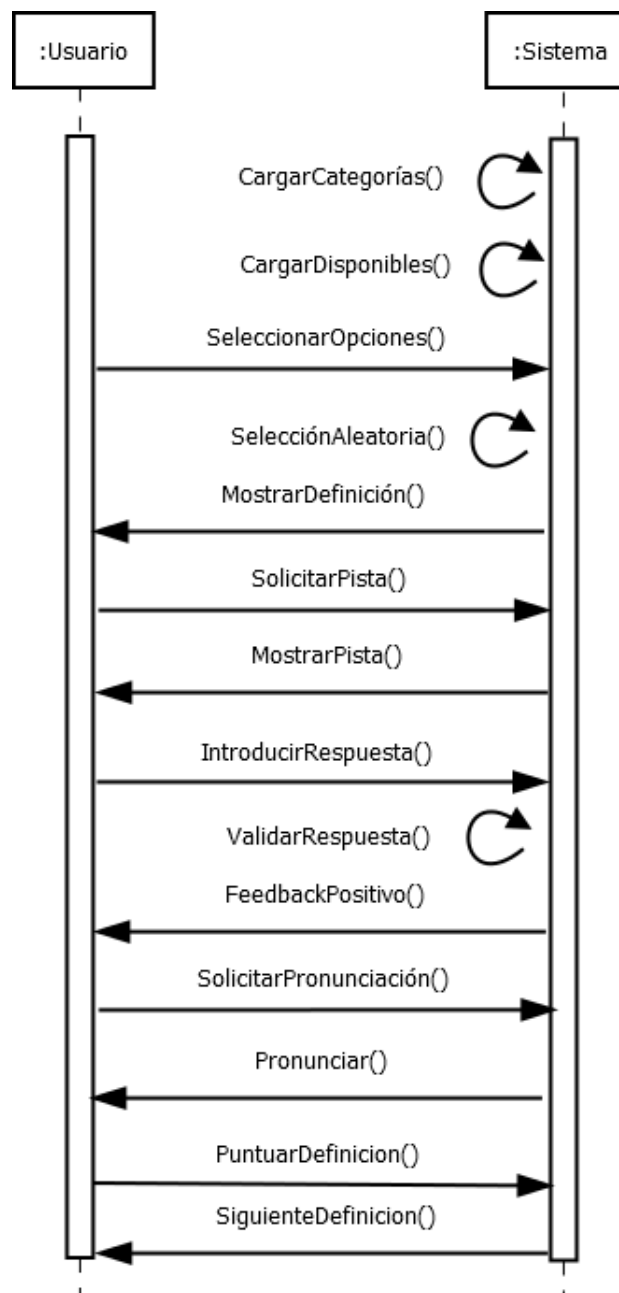
Caso de uso: Jugar (escenario 8b)

Figura 4.19: Análisis: Diagrama de secuencia Jugar (escenario 8b)

Operación SolicitarPronunciación()**Actores** Usuario.**Responsabilidades** Informar al sistema de la intención por parte del usuario de escuchar la pronunciación de la definición.**Precondiciones**

- El usuario ha pulsado sobre el botón de escuchar definición.

Postcondiciones Ninguna

Operación Pronunciar()

Actores Sistema.

Responsabilidades Pronunciar nativamente la definición.

Precondiciones

- Google speech ha sido cargado con anterioridad.

Postcondiciones Se reproduce la pronunciación de la definición actual.

Caso de uso: Jugar (escenario 8c)

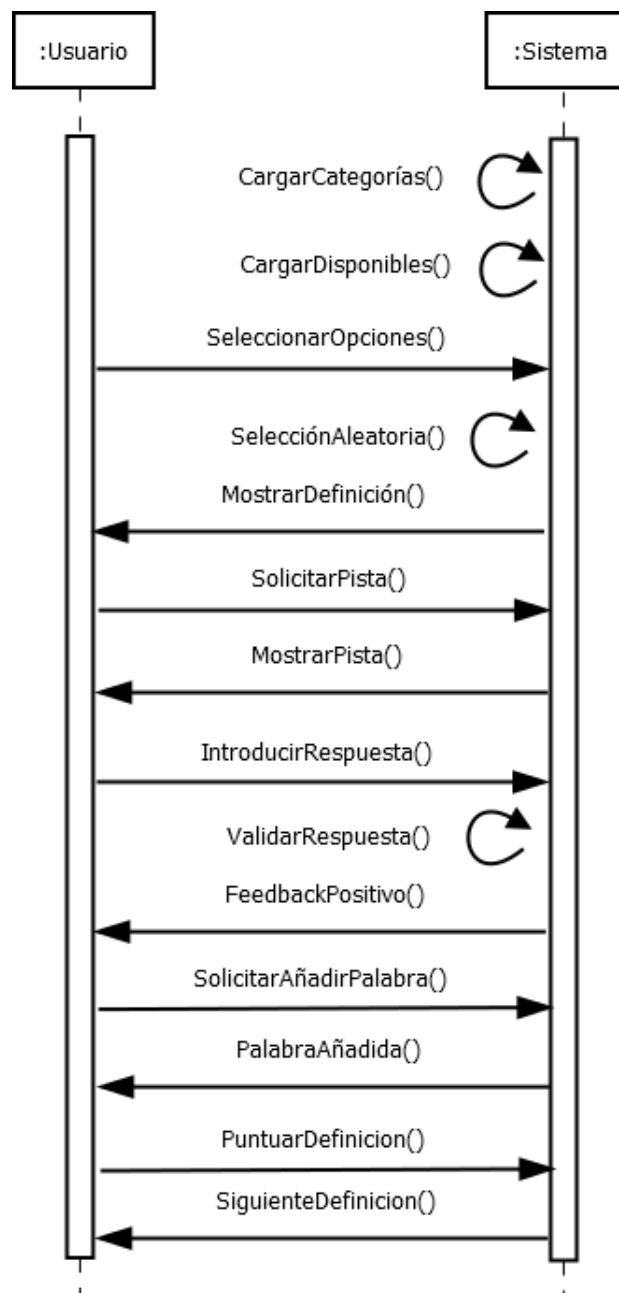


Figura 4.20: Análisis: Diagrama de secuencia Jugar (escenario 8c)

Operación SolicitarAñadirPalabra()

Actores Usuario.

Responsabilidades Informar al sistema de la intención por parte del usuario de añadir la palabra seleccionada a su bloc de notas personal.

Precondiciones

- El usuario ha seleccionado una palabra.

- El juego está en la pantalla de puntuar/reportar.

Postcondiciones Ninguna

Operación PalabraAñadida()

Actores Sistema.

Responsabilidades Informar al usuario de que la palabra ha sido añadida.

Precondiciones

- El usuario ha solicitado añadir una palabra a su bloc de notas.

Postcondiciones La palabra seleccionada queda guardada en el bloc de notas del jugador.

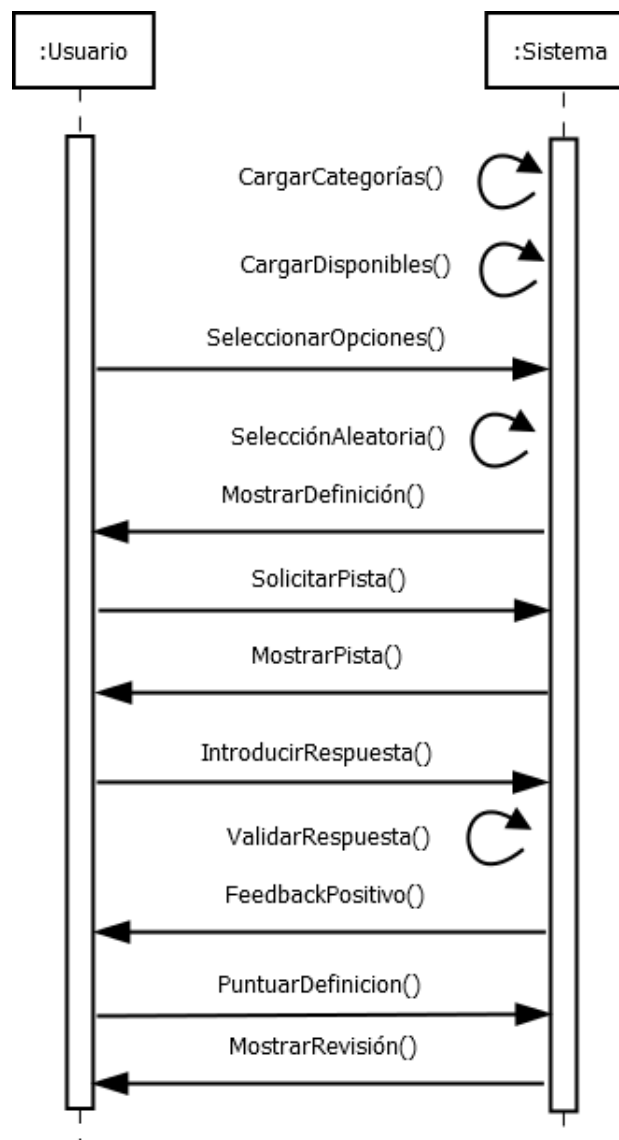
Caso de uso: Jugar (escenario 9a)

Figura 4.21: Análisis: Diagrama de secuencia Jugar (escenario 9a)

Operación MostrarRevisión()**Actores** Sistema.**Responsabilidades** Mostrar una pantalla con la información de juego de la partida que acaba de finalizar.**Precondiciones**

- No quedan más preguntas por jugar en la partida.

Postcondiciones El usuario puede revisar sus aciertos/errores y modificar puntuaciones en la pantalla de revisión generada.

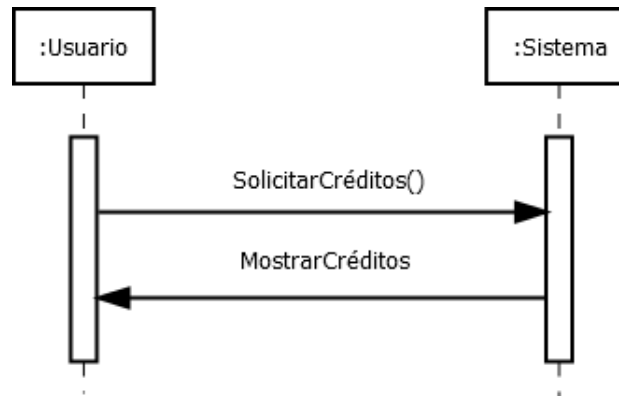
Caso de uso: Ver créditos(escenario principal)

Figura 4.22: Análisis: Diagrama de secuencia Ver créditos (escenario principal)

Operación SolicitarCréditos()

Actores Usuario.

Responsabilidades Informa al sistema de la intención del usuario por ver los créditos.

Precondiciones

- El jugador está en el menú principal.

Postcondiciones Ninguna

Operación MostrarCréditos()

Actores Sistema.

Responsabilidades Muestra la pantalla de créditos.

Precondiciones

- El jugador ha solicitado ver los créditos.

Postcondiciones

- Se muestra la pantalla de créditos.

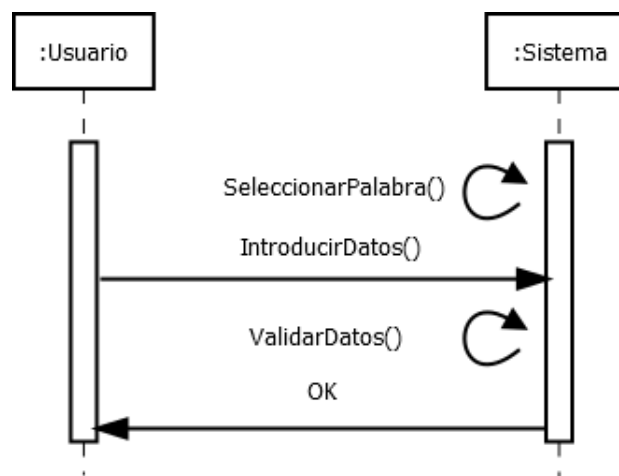
Caso de uso: Añadir definición (escenario principal)

Figura 4.23: Análisis: Diagrama de secuencia Añadir Definición (escenario principal)

Operación SeleccionarPalabra()**Actores** Sistema.**Responsabilidades** Selecciona una palabra a definir por el usuario acorde a su nivel.**Precondiciones**

- El jugador quiere definir una nueva palabra.
- El jugador ha jugado al menos X palabras.

Postcondiciones

- El sistema le asigna una palabra para definir.

Operación IntroducirDatos()**Actores** Usuario.**Responsabilidades** El usuario introduce la definición y la pista.**Precondiciones**

- El sistema le ha asignado una palabra a definir

Postcondiciones

- El sistema registra los datos que el usuario quiere introducir.

Operación ValidarDatos()**Actores** Sistema.**Responsabilidades** El sistema valida que la pista y la definición siguen el formato correcto.**Precondiciones**

- El usuario ha introducido una pista y una definición.

Postcondiciones

- El sistema da por válidos los datos.

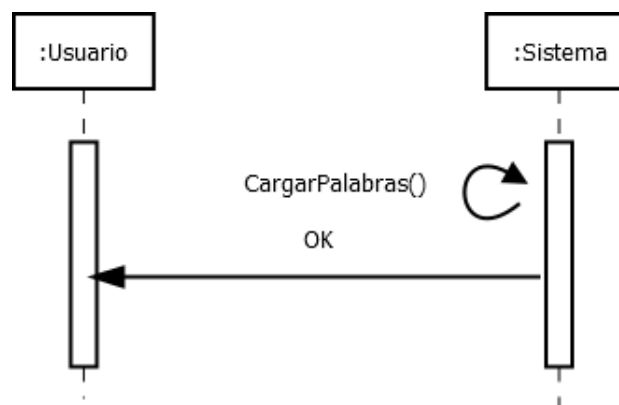
Caso de uso: Consultar bloc de notas (escenario principal)

Figura 4.24: Análisis: Diagrama de secuencia Bloc de Notas (escenario principal)

Operación CargarPalabras()

Actores Sistema.

Responsabilidades Carga el bloc de notas personal del usuario para el idioma seleccionado.

Precondiciones

- Existe un objeto p1 de Player perteneciente al SoccerTeam s que posea el balón.
- Existe un objeto p2 de Player perteneciente al SoccerTeam s que no posea el balón.
- El objeto juego está en estado Playing.

Postcondiciones

- El objeto player p1 deja de poseer el balón.
- El objeto player p2 es el nuevo poseedor del balón.

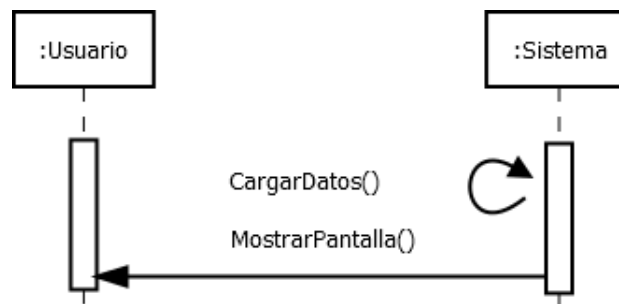
Caso de uso: Estadísticas de juego (escenario principal)

Figura 4.25: Análisis: Diagrama de secuencia Estadísticas juego (escenario principal)

Operación CargarDatos()**Actores** Sistema.**Responsabilidades** Muestra información relevante de acerca de la experiencia de juego del jugador.**Precondiciones**

- Hay un idioma seleccionado.
- Hay un usuario logueado.
- El usuario ha seleccionado la opción de estadísticas en el menú principal.

Postcondiciones

- Se muestra la pantalla de estadísticas que contiene los siguientes datos: ratio de acierto, mejor / peor categoría, niveles jugados y niveles más jugados.

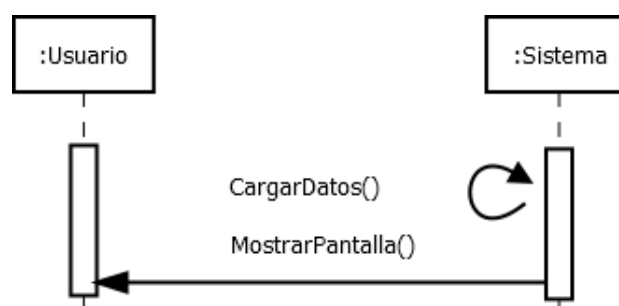
Caso de uso: Estadísticas de definiciones (escenario principal)

Figura 4.26: Análisis: Diagrama de secuencia Estadísticas de definiciones (escenario principal)

Operación CargarDatos()**Actores** Sistema.

Responsabilidades Muestra información relevante de acerca de las definiciones aportadas por el jugador.

Precondiciones

- Hay un idioma seleccionado.
- Hay un usuario logueado.
- El usuario ha seleccionado la opción de estadísticas en el menú principal.

Postcondiciones

- Se muestra la pantalla de estadísticas que contiene los siguientes datos: puntuación media recibida, porcentaje de acierto y número de reports.

Caso de uso: Revisar última partida (escenario principal)

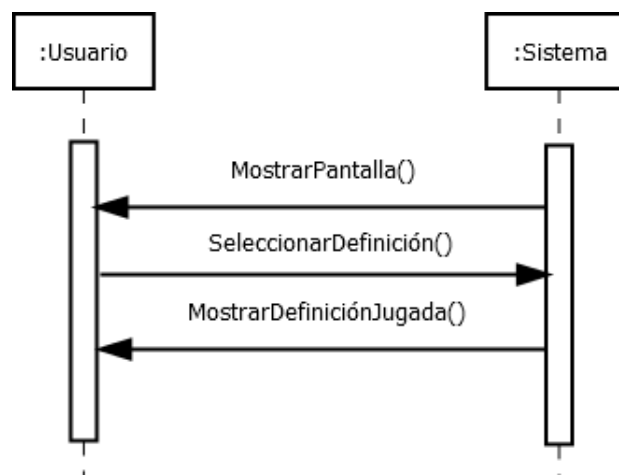


Figura 4.27: Análisis: Diagrama de secuencia Revisar última partida (escenario principal)

Operación MostrarPantalla()

Actores Sistema.

Responsabilidades permite al usuario revisar la última partida jugada.

Precondiciones

- Hay un usuario logueado.
- Hay un idioma seleccionado.
- Se ha jugado al menos una partida.

Postcondiciones

- Se muestra la pantalla de revisión dando una idea general de cómo fue la última partida.

Operación SeleccionarDefinición()

Actores Usuario.

Responsabilidades Informa al sistema de la intención del usuario de revisar una definición concreta ya jugada en la última partida.

Precondiciones

- El usuario ha seleccionado una definición jugada.

Postcondiciones Ninguna

Operación MostrarDefiniciónJugada()

Actores Sistema.

Responsabilidades Mostrar la información referenta a una definición jugada en la última partida.

Precondiciones

- El usuario ha seleccionado una definición jugada.

Postcondiciones

- Se muestra la pantalla puntuación con el feedback negativo/positivo.

Capítulo 5

Diseño

El proceso de descripción del diseño es mucho más sencillo una vez que ya se ha especificado que hace el sistema en el capítulo anterior. También decir, que al igual que el análisis del sistema, el diseño no contempla muchos detalles del sistema final, sólo una idea orientativa de como se implementará el sistema.

5.1. Interfaz gráfica

Tras los resultado que se han obtenido en la fase de análisis del sistema, es necesario desarrollar una interfaz sencilla y agradable para el usuario de la misma. En el diseño de estos, se intentará en todo momento que el flujo de ejecución sea lo más intuitivo posible.

En el siguiente diagrama podemos ver las distintas opciones de navegación dentro del juego:

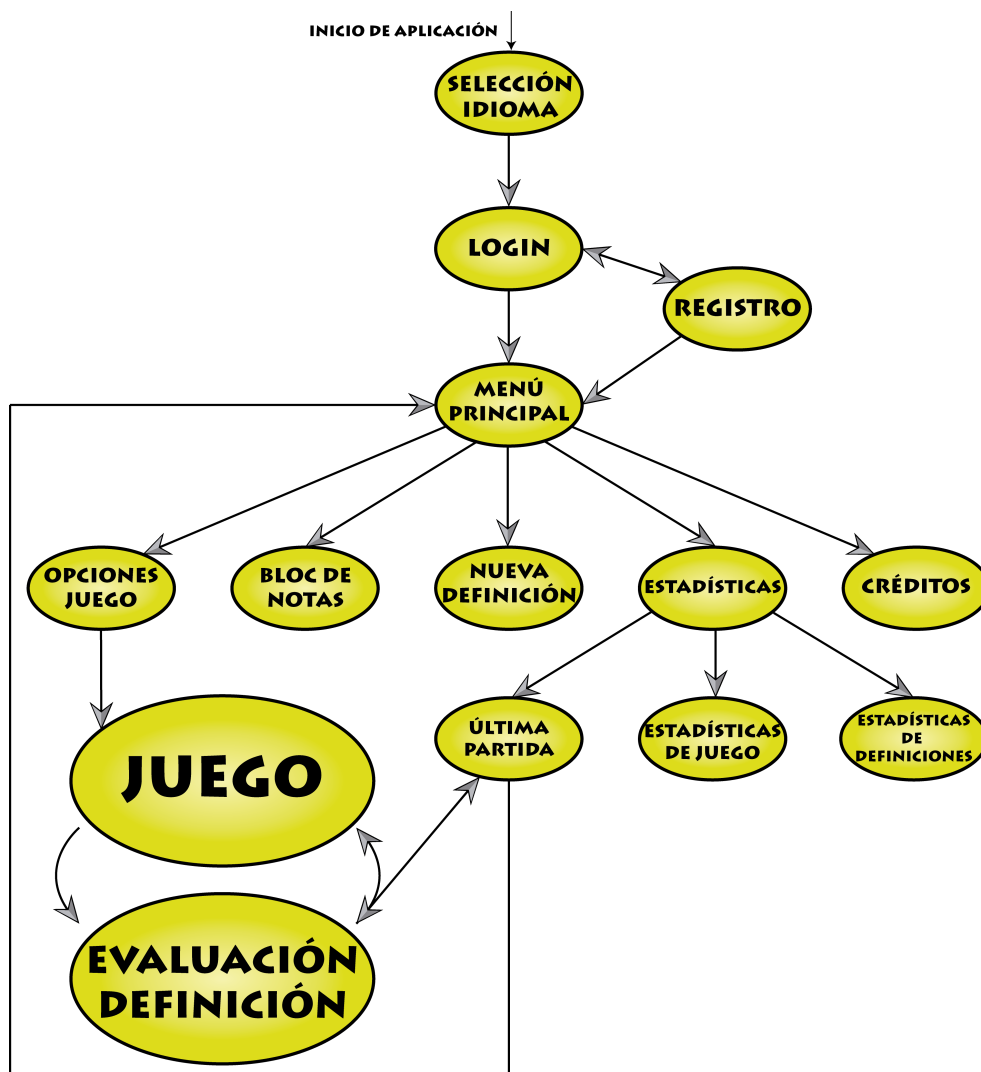


Figura 5.1: Navegación entre pantallas

La visión final de las pantallas sería la siguiente:



Figura 5.2: Pantalla de selección de idioma

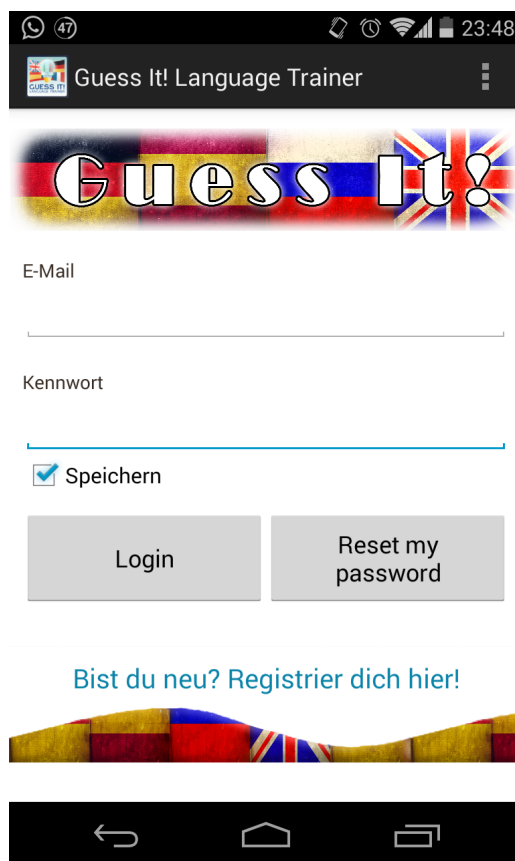


Figura 5.3: Pantalla de login

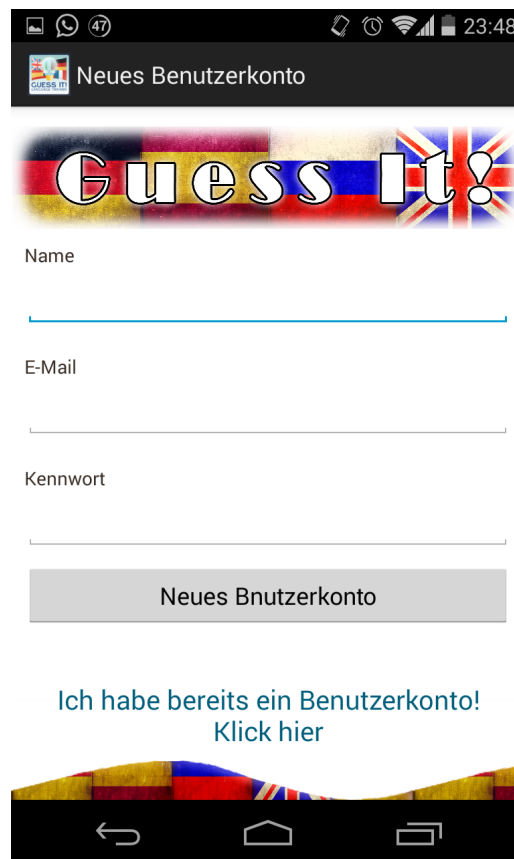


Figura 5.4: Pantalla de registro



Figura 5.5: Pantalla menú principal



Figura 5.6: Pantalla de opciones de juego

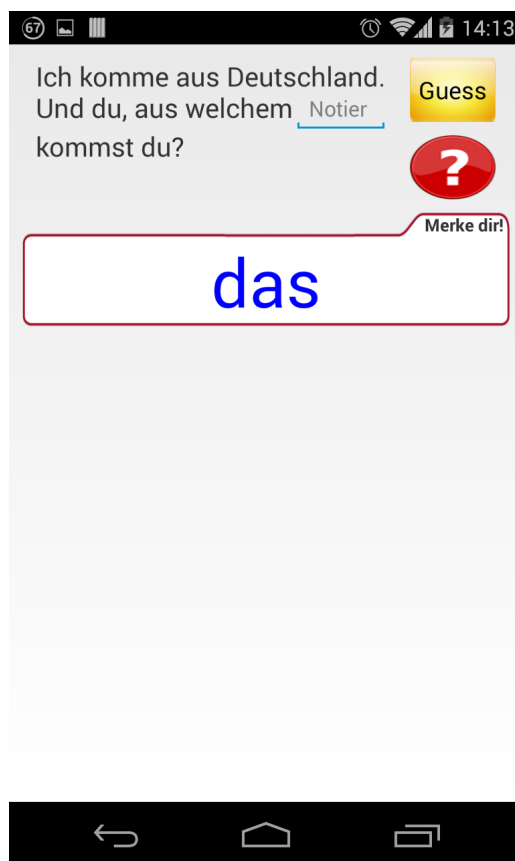


Figura 5.7: Pantalla de juego

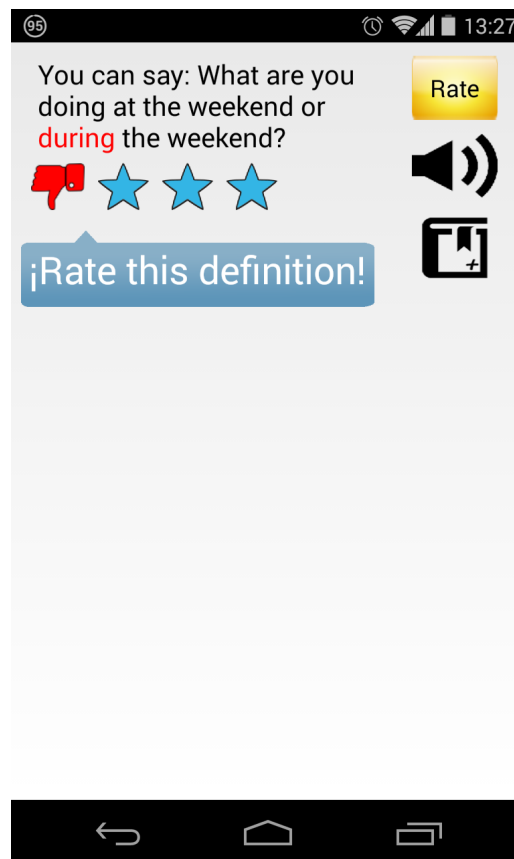


Figura 5.8: Pantalla de evaluación

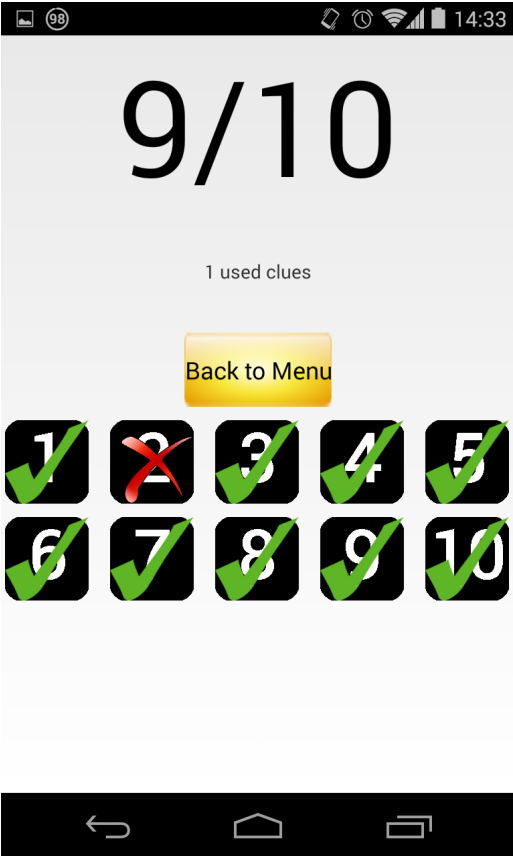


Figura 5.9: Pantalla de Última Partida

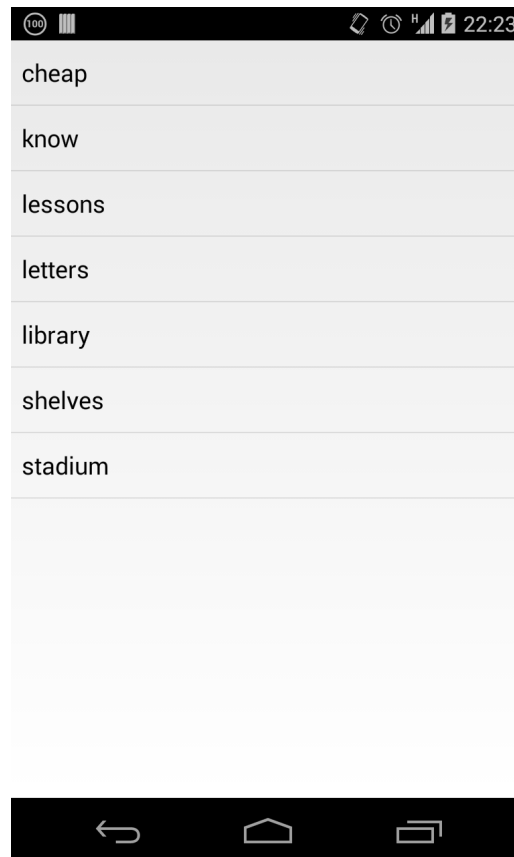


Figura 5.10: Pantalla de Bloc de Notas

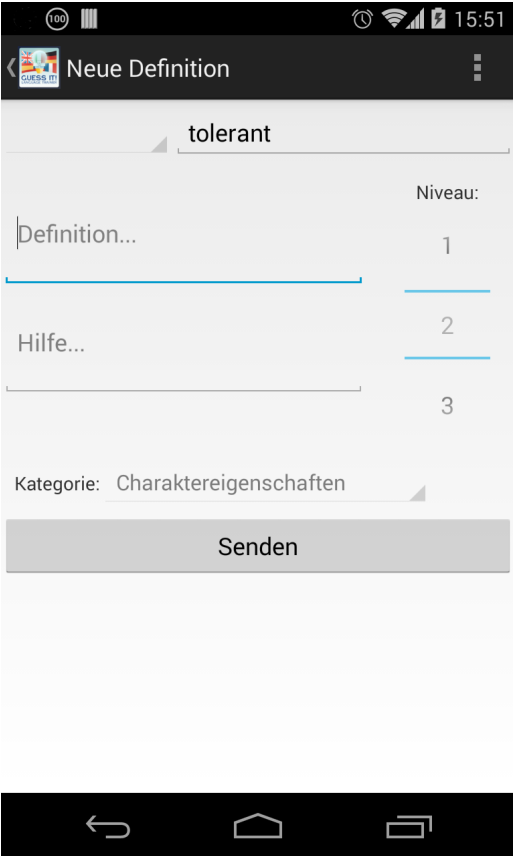


Figura 5.11: Pantalla de Nueva Definición

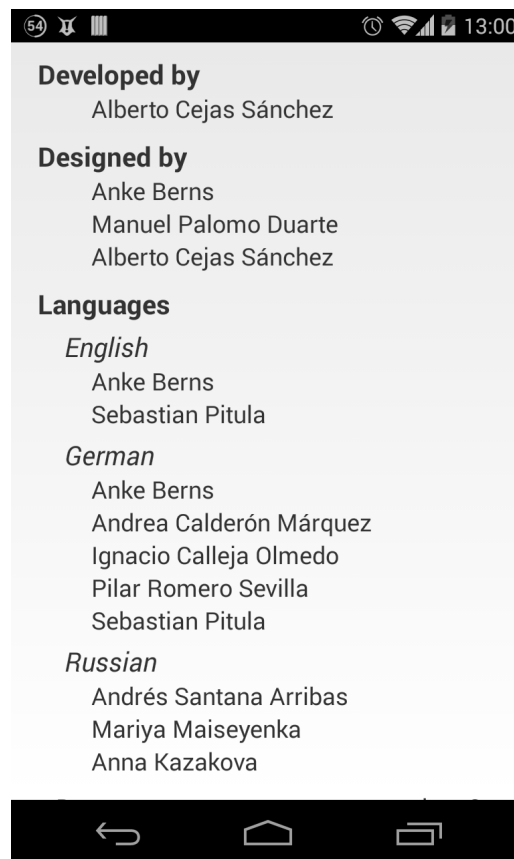


Figura 5.12: Pantalla de Créditos



Figura 5.13: Pantalla de Estadísticas

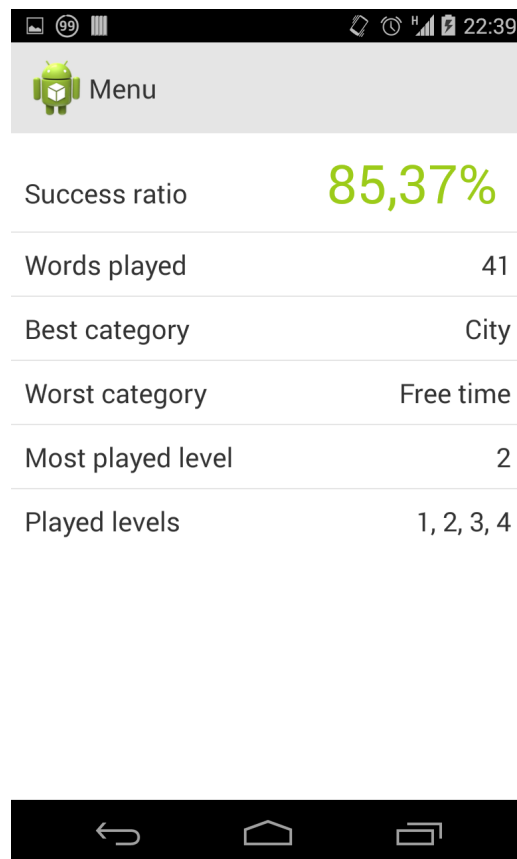


Figura 5.14: Pantalla de Estadísticas de Juego

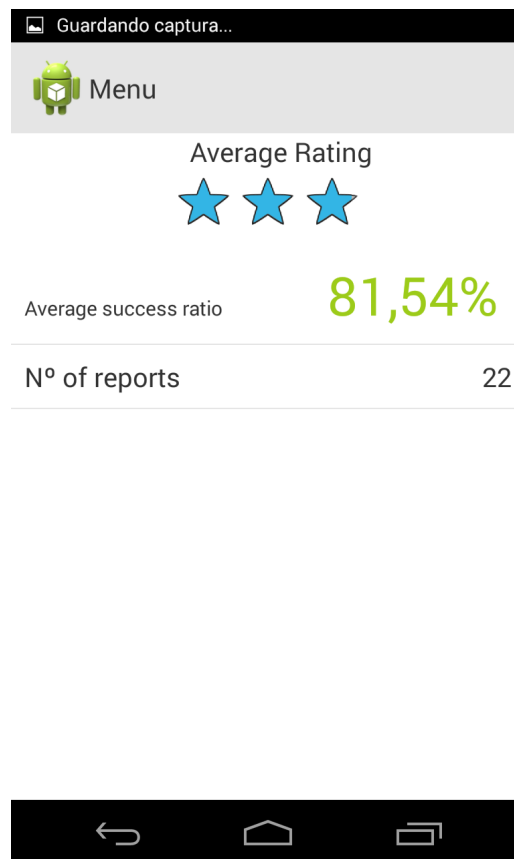


Figura 5.15: Pantalla de Estadísticas de Definiciones

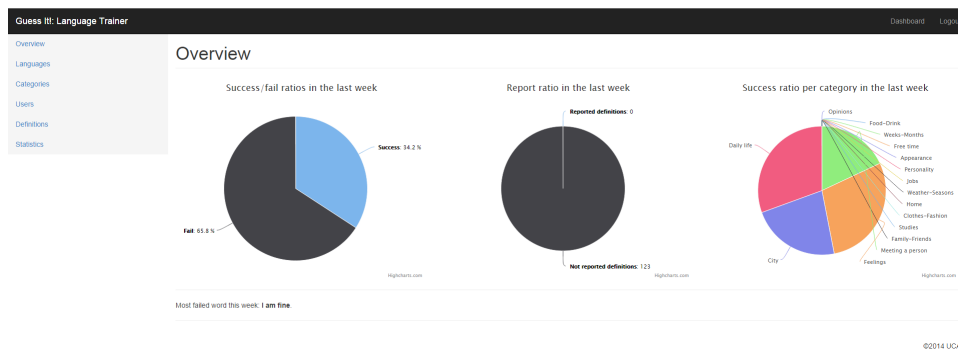


Figura 5.16: Interfaz del profesor

Tras el primer planteamiento sobre el contenido que almacenará la aplicación, se llega al siguiente diagrama lógico para la base de datos:

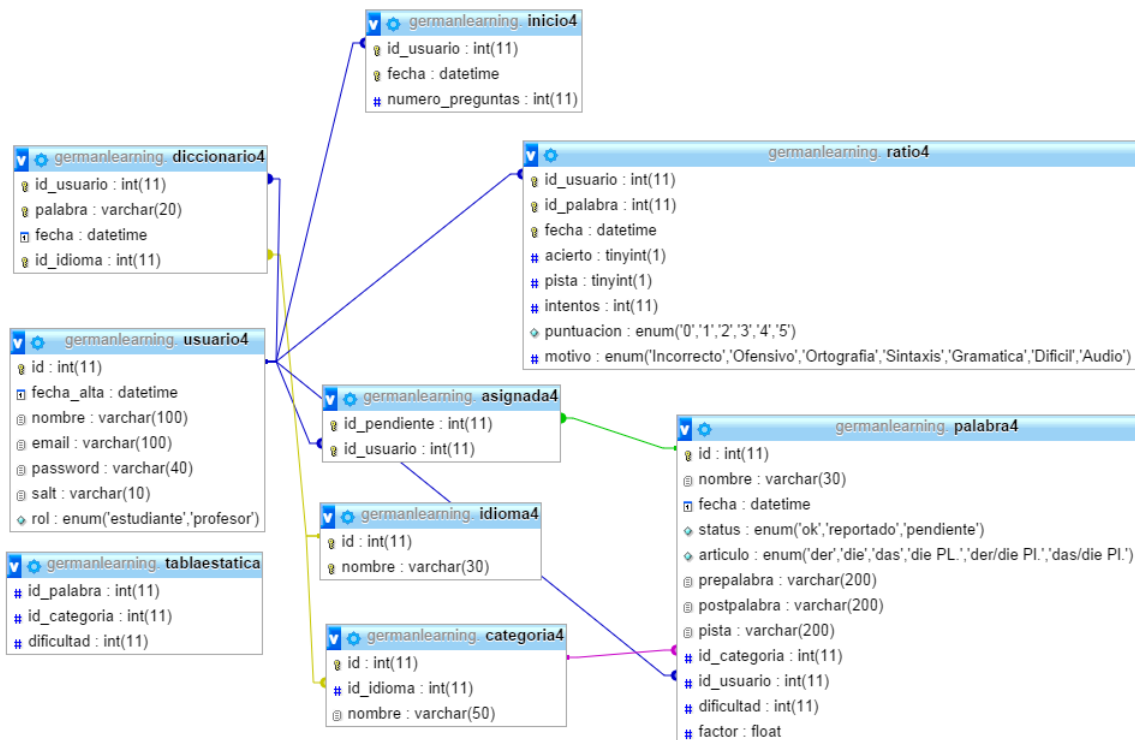


Figura 5.17: Diseño: Diagrama lógico BD

A continuación, se muestra una lista con las diferentes clases Java necesarias para la realización del cliente. Junto a cada una de las clases habrá una pequeña descripción sobre la labor que desempeña cada una.

LanguageSelectionActivity Actividad Android lanzadora de la aplicación, encargada de captar el idioma que el usuario seleccione y establecerlo como el predeterminado para la aplicación.

LoginActivity Actividad Android que cumple como fachada de la aplicación, dejando pasar únicamente a usuarios logueados.

RegisterActivity Actividad Android encargada de registrar a nuevos usuarios.

MainMenuActivity Actividad Android que implementa el menú principal.

AboutActivity Actividad Android encargada de cargar los créditos.

NewDefinitionActivity Actividad Android que implementa la pantalla para añadir nuevas definiciones desde el cliente.

DictionaryActivity Actividad Android encargada de mostrar el bloc de notas del usuario para el idioma seleccionado.

IndexableListView Clase Java que contiene la estructura de datos que implementa la lista de acceso rápido para el bloc de notas.

IndexScroller Clase Java auxiliar para modelar el índice de la estructura de datos anterior.

StringMatcher Clase para soportar UTF-8 en el acceso rápido del bloc de notas.

StatisticsActivity Actividad Android que contiene la fachada de las estadísticas.

HorizontalScrollView Clase Java encargada de modelar el menú horizontal pensado para ampliar secciones tales como estadísticas colectiva, rankings, etc.

ViewUtils Clase Java encargada de modelar cada una de las opciones situadas dentro del menú anterior.

StatisticsManager Clase Java encargada de guardar los datos estadísticos.

IndividualStatistics . Actividad Android encargada de mostrar el menú de opciones en estadísticas individuales.

CustomMenuList Clase Java contenedora de la estructura de datos que modela la fachada de las estadísticas.

MenuAdapter Clase Java que da forma a cada una de las opciones en la fachada de estadísticas.

ResultActivity Es la primera opción en el menú de estadísticas. Muestra la información referente a la última partida jugada.

NumberImageAdapter Clase Java que implementa una estructura de datos consistente en un adaptador para una lista de elementos donde cada uno está compuesto de un fondo y un número.

MarkableImageView Añade a una imagen otra imagen de aprobación(tick verde) o error (cruz roja).

MarkableButton Similar a la anterior pero para botones.

GameManager Clase Java encargada de almacenar la información del juego actual o del último jugado.

ReviewQuestionActivity Actividad Android que muestra la interfaz para puntuar/reporta, escuchar la pronunciación y añadir palabras al bloc.

BalloonHint Clase Java que implementa los bocadillos de ayuda. Usado para el mensaje Rate me!".

PlayStatistics Actividad Android que muestra la información contenida en el StatisticsManager referente a la experiencia de juego.

DefStatistics Actividad Android que muestra la información contenida en el StatisticsManager referente a las definiciones de un usuario.

PlayMenuActivity ActividadAndroid que se encarga de mostrar la pantalla de configuración de juego.

ImageCategoriesAdapter Clase Java a cargo de cargar las categorías.

GameActivity Actividad Android donde se desarrolla el juego en sí.

Question Clase Java que contiene la información de cada definición preguntada en el juego.

TabuUtils Clase Java estática con funciones auxiliares comunes dentro de la aplicación.

Environment Clase estática con información sobre el entorno de ejecución de la aplicación.

JSONParser Clase a cargo de parsear contenido serializado en formato JSON.

ConnectionManager Clase central en la aplicación. Su misión es funcionar de intermediario entre las demás clases y el servidor.

Implementación

Durante todo el desarrollo del proyecto, han ido apareciendo diversas dificultades y problemas que se debieron ir resolviendo para el correcto y continuo avance del mismo.

Por lo que en este capítulo se explicarán las distintas soluciones para los principales problemas encontrados.

También esta disponible toda la documentación del cliente generada con Javadoc. Aquí un enlace al directorio de dicha documentación:

<https://dl.dropboxusercontent.com/u/17018559/guessIt/doc.zip>

A continuación se detallan los aspectos de la implementación que no han sido reflejados en los apartados anteriores de esta memoria.

6.1. Layout

Existen multitud de dispositivos Android que cumplen los requisitos de ejecución. Cada uno de ellos tiene especificaciones distintas tales como tamaño y densidad de pantalla, memoria principal y secundaria, etc. Esto implica que el aspecto visual variará de dispositivo a dispositivo. Sin embargo, se han puesto esfuerzos en conseguir minimizar dicho impacto.

6.2. Multi-idioma 100 %

El juego es 100 % y para demostrarlo trae la versión de ruso que supone un nuevo alfabeto con la complejidad que ello conlleva.

6.3. Carga de paquetes de idiomas

Las definiciones se podrán cargar fácilmente a través de un formulario de la interfaz web, el cual solicita un archivo Excel que siga una estructura determinada y lo cargará en la base de datos instantáneamente.

Esto permite a usuarios no familiarizados con la tecnología, añadir nuevo contenido.

6.4. Tareas automatizadas

Para facilitar la creación de la base de datos se implementó un script en Python que se encarga de ello. Simplemente se le debe suministrar un usuario y un password con privilegios.

Al trabajar con distintos colaboradores las versiones de paquetes de idiomas Excel fueron teniendo matices por lo que se implementó un script que lo adaptase a la estructura exacta que la aplicación requería. Por último, para facilitarme el trabajo de hacer funcionar la aplicación en cualquier dispositivo Android compatible, decidí realizar un script que me escalase una imagen a las distintas resoluciones objetivo.

6.5. Selección aleatoria de definiciones

La puntuación de las definiciones tiene una importancia vital en el transcurso del juego y es que mantiene las mejores definiciones vivas mientras que las malas caen en desuso, sin desaparecer del todo porque siguen suponiendo un reto para comprobar que el usuario es capaz de detectar fallos. Para implementar esto, decidí que todas las noches generaría a través de un procedimiento PL/SQL una tabla estática de definiciones donde el número de veces que aparecía cada definición venía determinada por la media de puntuación recibida escalada para otorgarle mayor relevancia. Dándole valor doble a las puntuaciones de los últimos 15 días ya que el usuario debe tener un mejor criterio a medida que avanza en el aprendizaje. A la hora de jugar, el servidor elegiría de manera aleatoria y sin repetición las definiciones de la tabla estática generada la noche antes.

6.6. Estadísticas dinámicas

Las estadísticas que aparecen en la interfaz del profesor se calculan en tiempo real y se muestran dinámicamente. Esto dota al profesor de una potente herramienta donde consultar el avance y los fallos más comunes de los alumnos para así plantear mejor sus clases. A continuación se describen las gráficas ofrecidas:

Número de definiciones jugadas por un alumno en cada semana	X	Semanas	<p>Select a student:</p> <ul style="list-style-type: none"> maria maria Maria del Carmen Maria Escobar Maria Mercedes Garcia maria mesa maria mesa Maria Nimo Maria Noelia Lago Trujillo Maria Oliva Maria Pizano Galindo Maria Roldán mariaamorales 	
	Y	Definiciones jugadas		
	Criterio	Un alumno		
Número de definiciones jugadas de un nivel por un alumno en cada semana	X	Semanas	<p>Select a level:</p> <p>1</p> <p>Select a student:</p> <ul style="list-style-type: none"> Maria Garcia Alias Maria Romero Jimenez MariaRaf Mary marcagrava Mayte Mercedes Merche Meruyert miham Miriam Fopiani monica 	
	Y	Definiciones jugadas		
	Criterio	Alumno/nivel		
Ratio de aciertos de un alumno por semana	X	Semanas	<p>Select a student:</p> <ul style="list-style-type: none"> Lucia Ramos II Lucy Manuel Marco Antonio Núñez Peto maria maria Maria del Carmen Maria Escobar Maria Mercedes Garcia maria mesa maria mesa Maria Nimo Maria Noelia Lago Trujillo 	
	Y	Ratio de aciertos		
	Criterio	Un alumno		
Ratio de aciertos de definiciones de un nivel para un alumno por semana	X	Semanas	<p>Select a level:</p> <p>1</p> <p>Select a student:</p> <ul style="list-style-type: none"> Rafael Rafael Alfonso Ruiz Puyón Raquel Cifredo Ricardo Rocio Rocio Homigo Flores Rubén Payán Sandra sandra sandra sandra Sara 	
	Y	Ratio de aciertos		
	Criterio	Alumno/nivel		

Definiciones jugadas de cada nivel	X	Semanas	
	Y	Definiciones jugadas por nivel	
	Criterio	Alumno	
Ratio de definiciones que ha acertado de cada nivel	X	Semanas	
	Y	Ratio de acierto por nivel	
	Criterio	Alumno	
Tabla con los alumnos que más juegan, indicando porcentaje de acierto total y desglosado por nivel	X	-	
	Y	-	
	Criterio	Comunidad	
Número de definiciones de un alumno jugadas en cada semana	X	Semanas	
	Y	Definiciones jugadas por nivel	
	Criterio	Alumno	
	Y	Ratio de acierto por nivel	
	Criterio	Alumno	
	Criterio	Comunidad	

Pruebas

El diseño de casos de pruebas para un videojuego es algo complicado, no sólo para *GuessIt!*, si no para la mayoría de los distintos tipos de videojuegos existentes. Esto se debe a que estamos en un escenario con una alta interacción externa e interna. Sin embargo, las pruebas son algo necesario si deseamos un software con una calidad aceptable.

Todos los módulos que componen la aplicación han sido probados individualmente, tanto por mi como por los usuarios de la aplicación y los tutores. Esto significa que ha sido usado por más de 200 personas.

Además, en los distintos eventos en los que presentamos el proyecto, hemos recogido feedback de los asistentes que se tradujo en modificaciones (sobre todo de usabilidad) en la APP.

También se realizaron muchas pruebas de integración, ya que había módulos, una vez probado en solitario, debían realizar distintas acciones junto con otros módulos. Como por ejemplo al entrar en contacto con el servidor. Otras veces se requería usar librerías externas.

Otras pruebas que se realizaron fueron las de jugabilidad, tanto yo con los tutores, los alumnos colaboradores, los alumnos que lo probaron, gente de congresos, etc ofrecieron sus opiniones sobre aspectos que deberían ser modificados o errores que aparecían a lo largo de la ejecución del juego.

Se hicieron pruebas sobre la interfaz a medida que se implementaba el menú de juego, donde se probaban la reacción de esto ante situaciones para las que no estaban pensado su uso. Hubo muchos cambios en este ámbito.

En definitiva, la organización de los casos de pruebas fue la siguiente:

1. Tras finalizar la implementación de cada módulo, se realizaban pruebas unitarias sobre estos.
2. A medida que distintos módulo que anteriormente probados individualmente, debían colaborar entre ellos, se llevaban a cabo pruebas de integración.
3. Con las distintas versiones jugables se realizaban pruebas de jugabilidad.
4. Pruebas de interfaz sobre el menú que se implementó.

7.1. Pruebas unitarias

Estas pruebas se realizaron junto a la fase de implementación, conforme se implementaban nuevos módulos necesarios para la aplicación se realizaban pruebas individuales sobre estos módulos. De esta forma se buscaban todos los caminos posibles que podría dar cada módulo, teniendo en cuenta aquellos que fueran más predispuesto a fallos.

De esta forma todas las sentencias se ejecutaban como mínimo una vez y los posibles fallos se encontraban de una forma más sencilla. Por lo que también eran más fácil localizar donde estaba el problema y afrontar la solución de este.

7.2. Pruebas de integración

Conforme aparecían nuevos módulos, cuya implementación era necesaria y a su vez estos requerían el uso de otros módulos que posteriormente habían sido probados individualmente, se realizaban pruebas de integración entre dichos módulos.

Conforme se avanzaba en el desarrollo de *GuessIt!* se realizaban pruebas de integración a mayor escala. No solo entre módulos del mismo sistema, sino con el servidor y la interfaz web del profesor.

En este apartado los principales problemas se encontraron a la hora de formar el ecosistema completo entre la base de datos, el servidor y el cliente.

7.3. Pruebas de jugabilidad

Cada vez que estaban disponibles nuevas versiones jugables del juego, se pedían la ayuda a personas, ligadas y ajenas al desarrollo de la aplicación, que probaran las distintas demos disponibles. De esta forma cada uno de los colaboradores daba su opinión sobre distintos aspectos como la jugabilidad, dificultad, respuesta del juego o el propio contenido lingüístico. Tras recopilar la información que todos ellos proporcionaron, se procedía a realizar los ajustes necesarios a los distintos parámetros requeridos.

Entre los distintos aspectos a probar que se le recomendaban a los colaboradores que hicieran especial hincapié, son los siguientes:

- **Interfaz:** ¿requería formación previa el uso de la aplicación?
- **Innovación:** ¿Suponía una idea nueva? ¿Se diferenciaba del resto?
- **Docencia:** ¿Sería útil si se aplica en el ámbito docente?
- **Ampliación:** Aspectos limitados que faltaban por enriquecer.

La mayor parte de las personas que probaron el juego, además de encontrar fallos, opinaron que era una idea innovadora sobre un juego clásico, que preferían adivinar X definiciones en sus ratos libres que hacer tarea en casa y que estaría bien poder añadir más juegos. La mayoría de funcionalidades recomendadas fueron implementadas.

7.4. Pruebas de interfaz

Conforme se realizaban las distintas pantallas referentes a los menús que podemos encontrar en el juego, se realizaban pruebas exhaustivas sobre la interacción entre éstas. Sobre todo se comprobaban que las interfaces fueran intuitivas y claras. Así como comprobar la coherencia de elementos cambiantes como la posición de las palabras a adivinar, la interfaz en los distintos dispositivos, etc.

Se comprobó que la interfaz en ocasiones requería de una explicación previa y se decidió redactar el manual de usuario.

Capítulo 8

Camino Recorrido

8.1. Análisis científico del caso de estudio

Los primeros pasos se dieron poniendo la app en práctica con los alumnos de Alemán 2 de la Universidad de Cádiz (A1.1 CEFR, 6 ECTS) gracias a la perseverancia de Anke Berns. Únicamente contenía la versión alemana del juego con **282 definiciones**.

Sirvió como un entorno de pruebas brutal ya que contabamos con más de **120 alumnos** dándonos constante feedback explícito e implícito.

Al final del experimento, acabamos con **826 definiciones**, lo que significa que los alumnos añadieron más de **500 nuevas definiciones** (más de 4 por estudiante).

Los alumnos hicieron pre y post-test para evaluar su aprendizaje. El 50 % de las preguntas provenían del profesor mientras que la otra mitad de los propios alumnos en la app. La diferencia entre el primer test y el último fue demoledora: **5.35 puntos**.

Se jugaron más de **165.178 definiciones**. Considerando 1 minuto por definición(lectura, respuesta, evaluación/reporte, copiado), cada alumno jugó más de **20 horas**.

En el siguiente gráfico se puede apreciar como la mayoría de las definiciones eran acertadas por los alumnos:

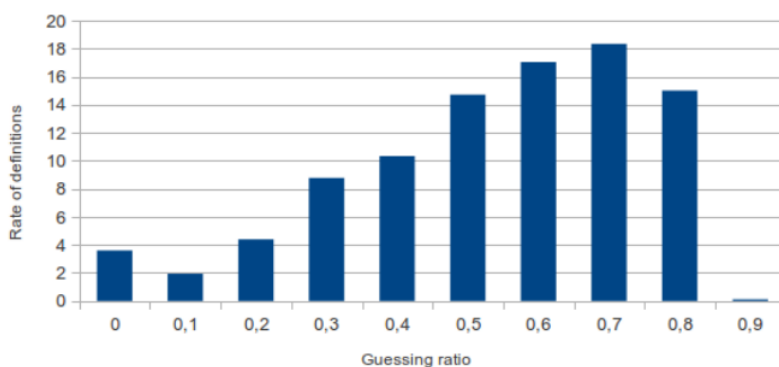


Figura 8.1: Ratio de acierto por porcentaje de definiciones

Se analizó también la diferencia en la evaluación entre los estudiantes y el profesor y se llegó a la conclusión de que no era tan dispar, concretamente **1.28 puntos** de 5 posibles.

Más de un **20 %** de las evaluaciones fueron exáctamente las mismas que las del profesor.

Un **2.5 %** de los estudiantes dieron una puntuación radicalmente opuesta(0-5) y el 97 % de las valoraciones fueron superiores a los 3 puntos (de 5 posibles).

8.2. Difusión

A lo largo de casi un año de desarrollo se ha recorrido un camino bastante largo de pruebas pero también de reconocimientos. El primero de ellos fue el I Concurso de Proyectos Científicos de la UCA, donde todavía teníamos un prototipo muy temprano:



Figura 8.2: I Concurso de Proyectos Científicos de la UCA

Posteriormente lo presentamos al VIII Concurso Universitario de Software Libre, consiguiendo el reconocimiento al *Mejor Proyecto Comunitario de Software Libre* de la fase local gaditana.



Figura 8.3: VIII Concurso Universitario de Software Libre

El I Concurso de Proyectos Científicos sirvió de pasarela para llegar a la Feria de las Ciencias de Granada donde nos ofrecieron un stand y pudimos presentar la aplicación en un entorno totalmente abierto y de esta manera, continuar recibiendo feedback:



Figura 8.4: Feria de las Ciencias de Granada

La UCA hacía eco de la noticia en su portal web:

<http://www.uca.es/es/cargarAplicacionNoticia.do?identificador=6713>

Sin embargo, aun nos faltaba llegar a otro entorno docente y decidimos presentarlo también a los profesores de la Escuela Oficial de Idiomas de San Fernando(Cádiz).



Figura 8.5: Escuela Oficial de Idiomas (San Fernando)

La aplicación se encontraba ya en una fase estable y pensamos que podríamos tener alguna oportunidad en *AtrÉBT!*, un concurso de ideas de empresa de la UCA. Finalmente, conseguimos el primer premio de la modalidad.



Figura 8.6: AtrÉBT!

Es entonces cuando decidimos ampliar fronteras y dar un pasito adelante presentando el proyecto a *Ninth European Conference on Technology Enhanced Learning (EC-TEL)*. Allí nos aceptaron un *demo paper* con la publicación *Guess it! Using Gamificated Apps to Support Students Foreign Language Learning by Organic Driven Community Peer Assessment:*

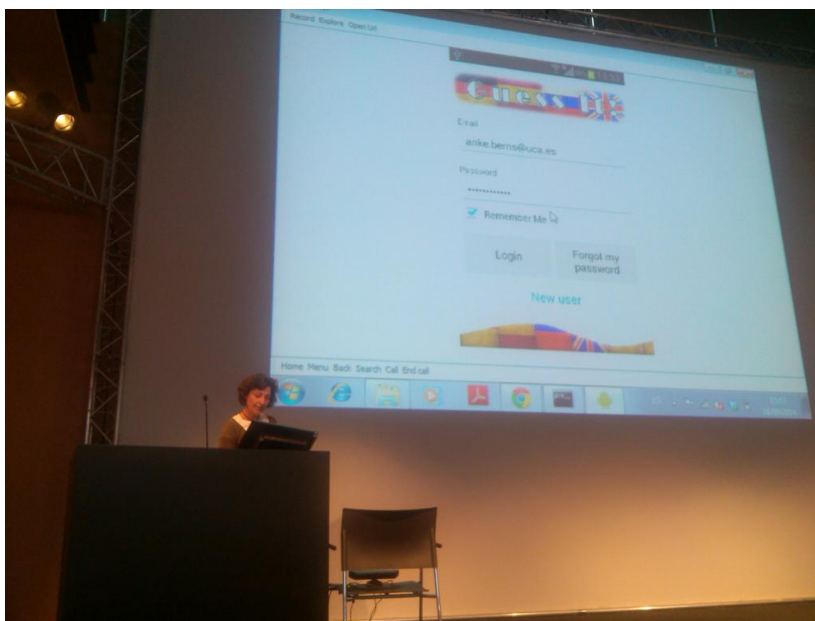


Figura 8.7: EC-TEL 2014

También nos invitaron a formar parte de la Noche Europea de los Investigadores en Jerez.



Figura 8.8: Noche Europea de los Investigadores

Por último, asistimos al *Second International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2014)* con un paper llamado *Foreign Language Learning using a gamified APP to support peer-assessment*.



Figura 8.9: TEEM Conference 2014

Conclusiones

En esta sección se comentarán las distintas conclusiones que se han obtenido tras la finalización del proyecto *GuessIt! : Language Trainer*.

9.1. Resumen de objetivos

Es evidente que su realización de este proyecto no me ha dejado indiferente. No ha sido fácil enfrentarse a las distintas tecnologías que lo envuelven, además de solucionar los distintos problemas que han ido apareciendo a lo largo de su desarrollo.

También decir que el proyecto me ha ocupado más tiempo del esperado en un principio debido a que los directores consiguieron añadir motivaciones y retos. Tuve muchos problemas y dudas en algunas fases del desarrollo de proyecto, sobre todo al recibir una gran cantidad de feedback de errores y bug y compatibilizarlo con las mejoras y ampliaciones.

Se puede decir que con este proyecto he aprendido muchísimo en todas y cada una de los aspectos y fases que lo componen, destacando sobre el resto el diseño de este tipo de sistema, cómo interactúan sus componentes y cómo implementar esto de manera elegante y flexible.

9.2. Conclusiones personales

Durante el desarrollo del proyecto se han aprendido muchísimas cosas: como hacer distintas ramas de desarrollo, plantear y crear calendarios, usar las herramientas adecuadas, hacer decisiones importantes para el desarrollo de este, documentación del código, organización, etc. Ya que durante la carrera se han realizado distintas prácticas y trabajos de complejidad, pero nada con el tamaño y duración que requiere un Proyecto de fin de carrera. Una vez finalizado éste creo que tengo la experiencia necesaria para afrontar otro proyecto con un resultado mucho más brillante.

Además añadiría que este proyecto me ha preparado para enfrentarme al mundo laboral donde seguramente encararé estas tecnologías. Ahora considero que tengo una idea previa de la arquitectura, herramientas y patrones que se utilizan a menudo para el desarrollo profesional.

Puedo decir que he profundizado y consolidado bastante en el lenguaje de programación Java y más concretamente el SDK de Android. Además he obtenido más práctica a la hora de manejar bibliotecas externas, así como entender su documentación e integrarlas en un proyecto.

En definitiva, este proyecto me ha hecho madurar como persona, estudiante y profesional. He aprendido a buscar opiniones en otras personas, compartir ideas, seguir un horario. Es reconfortante enfrentarme a un proyecto de estas características.

9.3. Mejoras y ampliaciones

Las posibles mejoras y ampliaciones que se podrían añadir al proyecto en futuras versiones, se comentan a continuación:

- **Modo multijugador:** Añadir la posibilidad de que jueguen dos jugadores en la misma partida.
- **Sinónimos:** Añadir soporte para sinónimos de modo que haya más palabras válidas para una misma definición.
- **Estadísticas:** Añadir mayor información estadística, concretamente rankings y detalles de los reports.
- **Sonidos:** Añadir efectos de sonido cuando se producen eventos.
- **Gamificación:** Aumentar el nivel de gamificación de la aplicación mediante medallas y logros.

Herramientas utilizadas

A.1. Lenguaje de programación del cliente

El lenguaje escogido fue Java debido a que es el que utiliza el SDK de Android.

1. Lenguaje orientado a objetos con el que estaba familiarizado.
2. Lenguaje compilado con gran comunidad detrás
3. Conjunto de bibliotecas estándar muy amplias y muy bien documentadas.

A.2. Lenguaje de programación de los scrpits

Se decidió Python para realizar los scripts para aprenderlo un poco más. Además de su capacidad de resolver problemas en pocas líneas.



Figura A.1: Herramientas utilizadas: Logo de Python

A.3. Base de datos

Se decidió utilizar MYSQL por ser libre, sencilla y potente.



Figura A.2: Herramientas utilizadas: Logo de Mysql

A.4. Frontend

Se decidió utilizar PHP con Codeigniter como framework. El motivo principal es la facilidad de utilizar y aprender.



Figura A.3: Herramientas utilizadas: Logo de Codeigniter

A.5. Backend

Se decidió utilizar PHP por su fiabilidad en entornos web.



Figura A.4: Herramientas utilizadas: Logo de PHP

A.6. Adobe Illustrator y Adobe Photoshop

Son programas de ilustración y retoque fotográfico respectivamente. Con él se han realizado texturas, botones, y todo el contenido 2D de elaboración propia.



Figura A.5: Herramientas utilizadas: Logo de Adobe Illustrator

A.7. Sistema de control de versiones

Todo el código y recursos de *GuessIt!: Language Trainer* está alojado en el sistema que proporciona GitHub, que consiste en un entorno completo usando el sistema de control de versiones *Git*.

Nos permite llevar todas las versiones y visualizar todos los cambios que se han producido durante todo el desarrollo del proyecto, entre los distintos archivos del mismo, así como poder volver a versiones anteriores en caso de necesidad y cualquier operación que permita cualquier sistema de control de versiones.

A.8. Documentación del código

Para la documentación del código me decanté por usar *Javadoc*, esta permite la documentación sencilla y legible de todo el código, generando la documentación en *HTML*.

También comentar que dicha herramienta se usó en proyectos anteriores con resultados muy satisfactorios.

A.9. Redacción de la memoria

Para la completa realización de la memoria se ha usado \LaTeX . Es un sistema de composición de textos, orientado especialmente a la creación de libros, documentos científicos y técnicos que contengan fórmulas matemáticas.



Figura A.6: Herramientas utilizadas: Logo de \LaTeX

\LaTeX es un sistema de composición de textos que está formado mayoritariamente por órdenes (macros) construidas a partir de comandos de TeX. \LaTeX es una herramienta práctica y útil pues, a su facilidad de uso, se une toda la potencia de TeX.

A.10. Realización de diagramas: Dia

Para la realización de todos los diagramas necesarios que aparecen a lo largo de toda la memoria se ha usado el creador de diagramas Dia.

Dia es un programa de creación de diagramas en GNU/Linux, MacOS X, Unix y Windows, bajo la licencia GPL. Puede ser utilizado para dibujar diferentes tipos de diagramas. Actualmente cuenta con herramientas para dibujar diagramas entidad relación, diagramas UML, diagramas de flujo, diagramas de red, y muchos otros diagramas.



Figura A.7: Herramientas utilizadas: Logo de Dia

Apéndice **B**

Manual de instalación

B.1. Cliente

El cliente Android se puede descargar fácilmente desde el *Google Play Store* concretamente en la siguiente dirección:

<https://play.google.com/store/apps/details?id=es.uca.tabu>

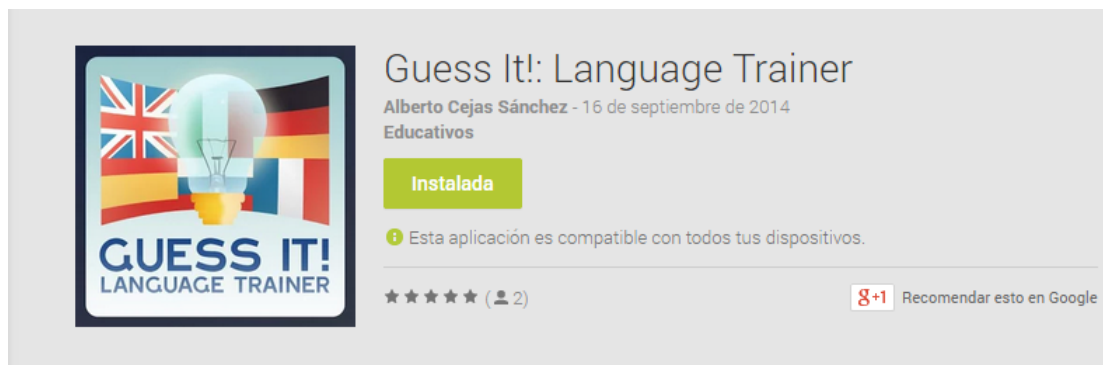


Figura B.1: GuessIt! : Language Trainer

B.2. Interfaz web del profesor

La interfaz web del profesor no requiere instalación, basta con estar registrado como docente en la plataforma y visitar la siguiente dirección:

<http://94.247.31.212/tabu/guessit/>

Apéndice **C**

Manual de usuario

GUESS IT

Alberto Cejas Sánchez – 18/10/2014

Manual de Usuario

V1

Índice

1. Antes de empezar...
2. Registro
3. Login
4. Jugar
5. Estadísticas
6. Bloc de notas
7. Nueva definición

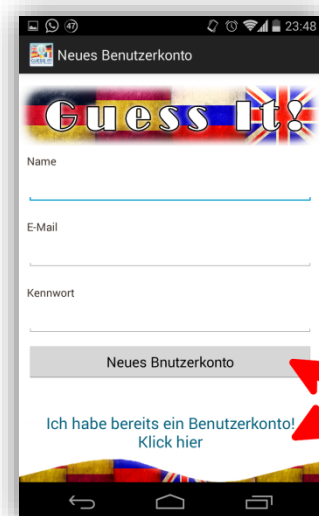
Antes de empezar...

Esta aplicación no funciona actualmente para versiones de Android igual o por debajo de 2.3.6. Puedes consultar tu versión desde **Ajustes** → **Información del teléfono** → **Versión de Android***.

Registro

Lo primero que debemos hacer al ejecutar la aplicación por primera vez es registrarnos en la siguiente pantalla.

Se deben introducir nombre, email y contraseña, que podrán contener exclusivamente letras y números.

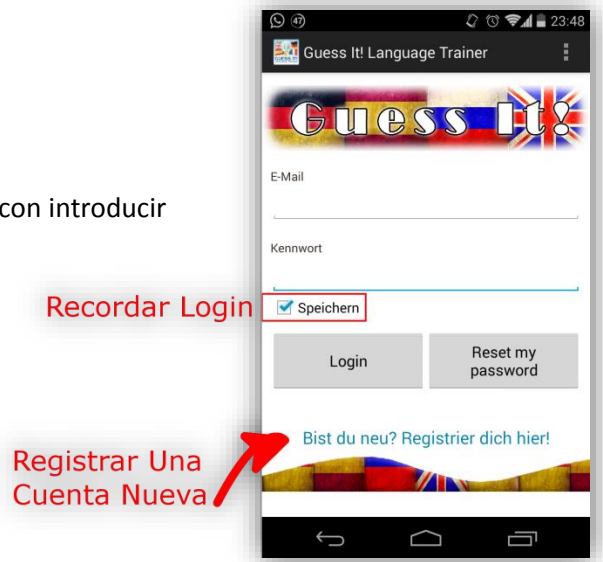


Registrarse

Pantalla De Login

Login

En caso de que ya dispongamos de una cuenta bastará con introducir email y contraseña y pulsar el botón de login.



Jugar

Una vez estás logueado, selecciona **Spielen** en el menú principal para jugar. Y a continuación **Single Player Mode**. En esta pantalla podrás personalizar tu partida:



Y aquí el flujo desde la pantalla de juego a la pantalla de evaluación:



Al finalizar todas las preguntas te aparecerá una pantalla resumen con la información de preguntas acertadas, preguntas totales y pistas usadas, pudiendo acceder a cada pregunta particular y revisándola particularmente.

Estadísticas

Pulsa **Statistiken** para acceder. Desde esta pantalla podrás ver la última partida que hayas jugado, redirigiéndote a la pantalla de revisión anteriormente nombrada.



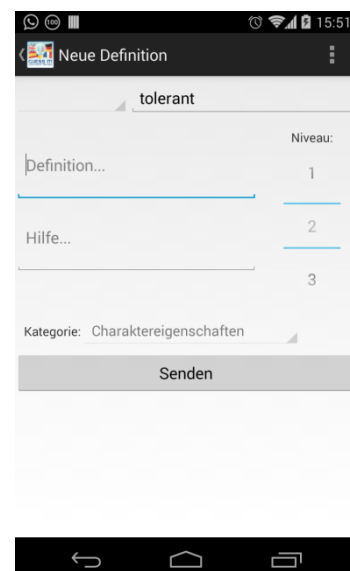
Bloc de notas

En esta sección del menú principal podrás reestudiar las palabras que hayas seleccionado manualmente con el botón de bloc de la pantalla de revisión.

Nueva definición

Por último, por cada 20 preguntas que juegues se te asignará una palabra a definir. Para llevar a cabo dicha tarea, accede a **Neue Definition** desde el menú principal. En el campo definición podrás incluir o no la palabra en cuestión según tu preferencia. A continuación se detallan ejemplos de cómo escribir las definiciones:

Palabra	Definición	Pista
loyal	Eine Person, die nicht korrupt ist, ist loyal	Das Wort beginnt mit dem Buchstaben "l"
(der) Rum	Legendario ist ein Rum	Am Wochenende trinke ich Whisky oder ...
(die PL.) Augen	Die Augen von B. Pitt sind blau	Jede Person hat 2 Beine, 2 Arme, 2 Hände, 2 Ohren und 2 ...
(das) Bier	Cruzcampo ist ein Bier	Die Deutschen und Engländer trinken viel ...



Como puedes observar, **NO** debes incluir la palabra en la pista.

Bibliografía

- [1] Preguntas y respuestas
<http://stackoverflow.com>.
- [2] Documentación oficial Java
<http://docs.oracle.com/javase/7/docs/>
- [3] Documentación oficial Android.
<http://developer.android.com/reference/>.
- [4] Referencia del lenguaje de programación Python.
<https://docs.python.org/3.3/reference/>
- [5] Documentación oficial de PHP.
<http://php.net/docs.php>
- [6] Tutorial de como construir un webservice paso a paso.
<http://www.androidhive.info/2012/01/android-login-and-registration-with-php-mysql-and-sqlite/>
- [7] Ayuda oficial Adobe Illustrator.
<http://helpx.adobe.com/es/illustrator/topics.html>
- [8] Beginning Android Games. ISBN-13:978-1430246770.

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License,

but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
 - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
 - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
 - K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
 - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
 - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
 - N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
 - O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.