

## 6.2. Sprint 2

### 6.2.1. Actualización pila del producto

Tras la realización de cada uno de los *sprint* se actualiza la **pila del producto** o *Product Backlog*. Esta tarea consiste en volver a estimar las historias de usuario, añadir nuevas historias de usuario, eliminar de existentes o cambiar prioridades para el cliente. A continuación se muestra la pila del producto actualizada tras la finalización del primer *sprint*.

*Las historias de usuario de color gris están terminadas y validadas al 100% y se muestran en naranja los valores de la pila del producto que han cambiado respecto a la última versión.*

#### 1. Registrarse como jugador

Como jugador quiero registrarme como usuario para poder jugar en el juego.

Estimación: 2 puntos de historia — Valor: 80

#### 2. Registrarse con Facebook

Como jugador quiero identificarme como usuario a través de mi cuenta de Facebook para no tener que crearme un usuario.

Estimación: 4 puntos de historia — Valor: 80

#### 3. Identificarte como usuario

Como jugador quiero identificarme en la aplicación como usuario registrado para poder jugar.

Estimación: 4 puntos de historia — Valor: 80

#### 4. Recordar contraseña

Como jugador quiero recordar mi contraseña a través del correo electrónico para poder identificarte en caso de olvidarla.

Estimación: 2 puntos de historia — Valor: 70

#### 5. Menú

Como jugador quiero que la pantalla del menú esté formada por 3 pantallas(tabs) que contengan la información del juego, preguntas, temas sugeridos y amigos para poder ver de forma resumida las principales estadísticas y puntuaciones.

Estimación: 4 puntos de historia — Valor: 70

#### 6. Ver información general

Como jugador quiero acceder a una pantalla de perfil de usuario donde pueda ver toda la información relativa a mi usuario como el nombre de usuario, correo electrónico y foto de perfil, y todos los resultados y estadísticas generales del juego (nivel, logros, puntos de usuario, progreso y preguntas sugeridas o corregidas) para conocer mi progreso en el juego.

Estimación: 4 puntos de historia — Valor: 70

7. **✓ Jugar individualmente**  
Como jugador quiero jugar al juego de forma individual para mejorar mis habilidades.  
Estimación: **4** puntos de historia — Valor: **70**
8. **Añadir amigos contactos**  
Como jugador quiero buscar a mis amigos a través de los números de teléfono almacenados en mi agenda para jugar contra ellos.  
Estimación: **2** puntos de historia — Valor: **60**
9. **Añadir amigos contactos (email)**  
Como jugador quiero buscar a mis amigos a través de los correos electrónicos almacenados en mi agenda para jugar contra ellos.  
Estimación: **2** puntos de historia — Valor: **60**
10. **Añadir amigos *Facebook***  
Como jugador quiero buscar a mis amigos a través del Facebook para jugar contra ellos.  
Estimación: **2** puntos de historia — Valor: **60**
11. **Jugar Duelo**  
Como jugador quiero retar a mis amigos a un 'Duelo' para obtener puntos extra.  
Estimación: **4** puntos de historia — Valor: **60**
12. **Añadir nueva pregunta**  
Como jugador quiero añadir una nueva pregunta en el juego para obtener más puntos de usuario.  
Estimación: **4** puntos de historia — Valor: **50**
13. **Corregir pregunta**  
Como jugador quiero corregir preguntas propuestas por otros usuarios para obtener puntos de usuario.  
Estimación: **2** puntos de historia — Valor: **50**
14. **Jugar Reto**  
Como jugador quiero medir mis habilidades en un 'Reto' para obtener logros dentro del juego.  
Estimación: **8** puntos de historia — Valor: **40**
15. **Visualizar *Ranking***  
Como jugador quiero ver un *ranking* de usuarios junto a mis amigos para saber el nivel de mis amigos.  
Estimación: **4** puntos de historia — Valor: **30**

Durante la actualización de la pila del producto se han producido pocos cambios significativos. El único cambio resaltado es el cambio de prioridad de la historia de usuario **Ver información general** que ocupaba anteriormente la posición 7 y sube una posición respecto a la versión anterior. En la pila del producto se hace referencia a este cambio mediante una

flecha verde ascendente en la historia de usuario que asciende una posición y una flecha roja descendente en las historia de usuario que ocupa una posición inferior en la pila del producto.

### 6.2.2. Planificación del Sprint

Para el *sprint* 2 se ha decidido coger solamente 2 historias de usuario de estimación 4 ph cada una de ellas. Por lo tanto, la suma de la estimación es de 8 ph, lo equivalente a dos semanas de trabajo ( 8 ph \* 5 h/ph = 40 horas ). A continuación se muestran todas las historias de usuario descompuestas en tareas estimadas.

**Yo como jugador quiero que la pantalla del menú contengan la información del juego, preguntas, temas sugeridos y amigos para poder ver de forma resumida las principales estadísticas y puntuaciones.**

- Crear vista del menú. [2 horas].
- Crear el controlador del menú y adaptadores. [3 horas]
- Crear **Fragment** (Controlador) para el *tab* de juego. [2 horas]
- Crear vista para el **Fragment** de juego. [3 horas]
- Crear **Fragment** (Controlador) para el *tab* de WikiQuizzy. [2 horas]
- Crear vista para el **Fragment** de WikiQuizzy. [3 horas]
- Crear **Fragment** (Controlador) para el *tab* de amigos. [2 horas]
- Crear vista para el **Fragment** de amigos. [3 horas]

Tiempo total: [20h] ⇒ 4 puntos historia (20 horas)

**Yo Como jugador quiero acceder a una pantalla de perfil de usuario donde pueda ver toda la información relativa a mi usuario para conocer mi progreso en el juego.**

- Crear la vista del perfil de usuario. (Vista dinámica y muy compleja) [10 horas].
- Crear Controlador del perfil. [3 horas]

Tiempo total: [13 h] ⇒ 4 puntos de historia

### 6.2.3. Pruebas de aceptación

Tras la finalización del segundo sprint la aplicación debe cumplir con las pruebas de aceptación definidas. A continuación se muestra una lista con las pruebas de aceptación definidas y los resultados obtenidos tras este segundo *sprint*. Las pruebas marcadas de color verde se tratan de pruebas que tras el sprint se satisfacen correctamente y en rojo las pruebas de aceptación que no satisfacen la prueba o no pueden ser validadas por otros motivos.

- **DADO** que un usuario accede a la aplicación, **CUANDO** se identifica correctamente **ENTONCES** accede al menú de la aplicación.
- **DADO** un usuario que se encuentra en el menú de la aplicación **CUANDO** accede al *tab* de juego **ENTONCES** tendrá la opción de jugar individualmente, duelo o formular una nueva pregunta.
- **DADO** un usuario que se encuentra en el menú de la aplicación **CUANDO** accede al *tab* de juego **ENTONCES** podrá visualizar su nivel, número total de tokens, y la suma de puntos obtenida hasta el momento.
- **DADO** un usuario que se encuentra en el menú de la aplicación **CUANDO** accede al *tab* de WikiQuizzy **ENTONCES** podrá acceder a formular una nueva pregunta o corregir una pregunta.
- **DADO** un usuario que se encuentra en el menú de la aplicación **CUANDO** accede al *tab* de WikiQuizzy **ENTONCES** podrá visualizar su nivel, número total de tokens, y la cantidad de pregunta sugeridas y corregidas.
- **DADO** un usuario que se encuentra en el menú de la aplicación **CUANDO** accede al *tab* de WikiQuizzy **ENTONCES** podrá visualizar las categorías recientes de las cuales se han añadido nuevas preguntas.
- **DADO** un usuario que se encuentra en el menú de la aplicación **CUANDO** accede al *tab* de amigos **ENTONCES** podrá acceder a buscar nuevos amigos en su lista de contactos, en facebook o en el juego.
- **DADO** un usuario que se encuentra en el menú de la aplicación **CUANDO** accede al *tab* de amigos **ENTONCES** podrá visualizar todos los amigos que tenga en el juego.
- **DADO** un usuario que se encuentra en el menú de la aplicación **CUANDO** accede al perfil de usuario **ENTONCES** podrá visualizar toda la información relativa al usuario como nombre de usuario, email, nivel, puntos obtenidos (desglosados por categorías) y partidas ganadas, empatadas y perdidas.

#### 6.2.4. Resultados obtenidos

Tras la realización de la segunda iteración hemos vuelto a obtener un producto funcional, al cual, hemos incrementado la funcionalidad obtenida en el *sprint* 1. Por lo tanto, en la figura 6.2 se muestran las pantallas finalizadas con respecto a la aplicación final. Este gráfico simplemente tiene como intención mostrar el avance del proyecto en respecto a las pantallas implementadas.

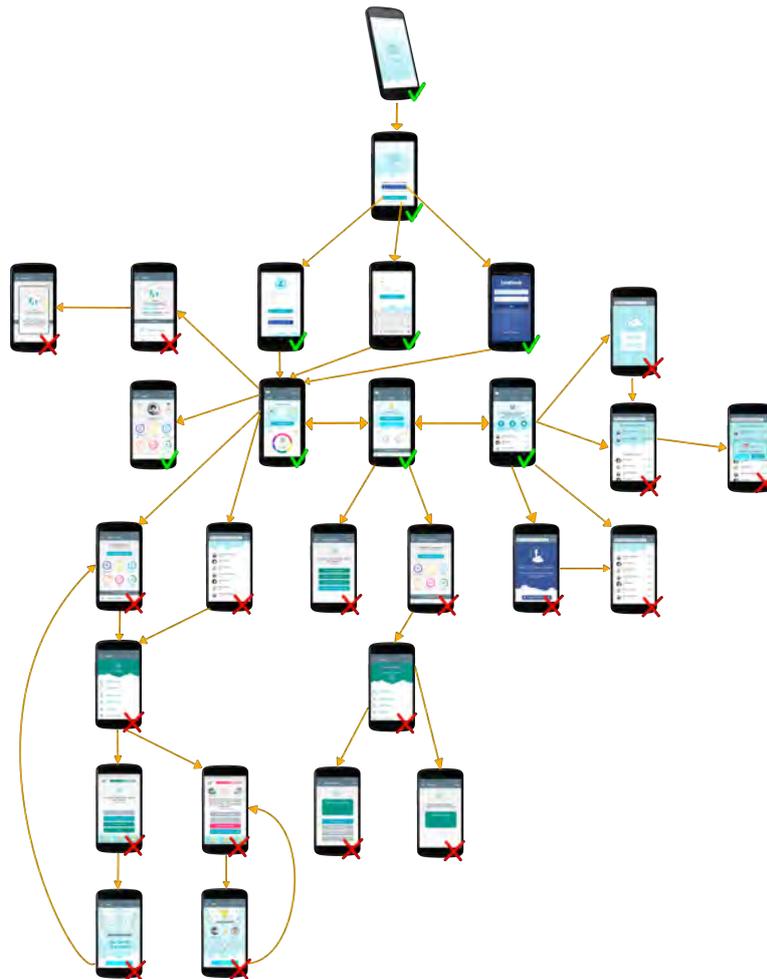


Figura 6.2: Jerarquía de pantallas completada Sprint 2.

Además de esto, la cantidad de puntos de historia implementados es de 18, teniendo en cuenta que la suma total de puntos de historia es de 50 podemos concluir que durante la primera iteración tenemos completado alrededor del 36% de sistema.

## 6.3. Sprint 3

### 6.3.1. Actualización pila del producto

*Las historias de usuario de color gris están completadas al 100 %, las historias de color verde se han añadido en esta revisión de la pila y se muestran en naranja los valores de la pila del producto que han cambiado respecto a la última versión.*

#### 1. Registrarse como jugador

Como jugador quiero registrarme como usuario para poder jugar en el juego.

Estimación: **2** puntos de historia — Valor: **80**

#### 2. Registrarse con Facebook

Como jugador quiero identificarme como usuario a través de mi cuenta de Facebook para no tener que crearme un usuario.

Estimación: **4** puntos de historia — Valor: **80**

#### 3. Identificarte como usuario

Como jugador quiero identificarme en la aplicación como usuario registrado para poder jugar.

Estimación: **4** puntos de historia — Valor: **80**

#### 4. Recordar contraseña

Como jugador quiero recordar mi contraseña a través del correo electrónico para poder identificarte en caso de olvidarla.

Estimación: **2** puntos de historia — Valor: **70**

#### 5. Menú

Como jugador quiero que la pantalla del menú esté formada por 3 pantallas(tabs) que contengan la información del juego, preguntas, temas sugeridos y amigos para poder ver de forma resumida las principales estadísticas y puntuaciones.

Estimación: **4** puntos de historia — Valor: **70**

#### 6. Ver información general

Como jugador quiero acceder a una pantalla de perfil de usuario donde pueda ver toda la información relativa a mi usuario como el nombre de usuario, correo electrónico y foto de perfil, y todos los resultados y estadísticas generales del juego (nivel, logros, puntos de usuario, progreso y preguntas sugeridas o corregidas) para conocer mi progreso en el juego.

Estimación: **4** puntos de historia — Valor: **70**

#### 7. Mostrar Categorías

Como usuario quiero que antes de jugar individualmente me muestre una pantalla con todas las categorías generales del juego y las categorías más recientes a la que he jugado para seleccionar la categoría a la que jugar.

Estimación: **2** puntos de historia — Valor: **80**

8. **Mostrar subcategorías**  
Como jugador quiero que al seleccionar la categoría general sobre la que quiero jugar individualmente me muestre todas las subcategorías que tiene esta categoría para seleccionar una para jugar.  
Estimación: **2** puntos de historia — Valor: **80**
9. **Contestar preguntas individualmente**  
Como jugador quiero que me aparezca un conjunto de preguntas por pantalla de una en una e ir contestándolas.  
Estimación: **4** puntos de historia — Valor: **75**
10. **Jugar individualmente**  
Como jugador quiero que mientras esté respondiendo una pregunta pueda obtener más tiempo para responder, simplificar las respuestas o pasar para ganar la partida más fácilmente.  
Estimación: **1** puntos de historia — Valor: **70**
11. **Añadir amigos contactos**  
Como jugador quiero buscar a mis amigos a través de los números de teléfono almacenados en mi agenda para jugar contra ellos.  
Estimación: **2** puntos de historia — Valor: **60**
12. **Añadir amigos contactos (email)**  
Como jugador quiero buscar a mis amigos a través de los correos electrónicos almacenados en mi agenda para jugar contra ellos.  
Estimación: **2** puntos de historia — Valor: **60**
13. **Añadir amigos *Facebook***  
Como jugador quiero buscar a mis amigos a través del Facebook para jugar contra ellos.  
Estimación: **2** puntos de historia — Valor: **60**
14. **Jugar Duelo**  
Como jugador quiero retar a mis amigos a un ‘Duelo’ para obtener puntos extra.  
Estimación: **4** puntos de historia — Valor: **60**
15. **Mostrar categorías para añadir pregunta**  
Como usuario quiero que antes de formular una pregunta me muestre una pantalla con todas las categorías generales del juego y las últimas categorías más recientes que he sugerido para seleccionar la categoría sobre la cual quiero añadir una nueva categoría.  
Estimación: **2** puntos de historia — Valor: **50**
16. **Añadir nueva pregunta**  
Como jugador quiero añadir una nueva pregunta en el juego para obtener más puntos de usuario.  
Estimación: **4** puntos de historia — Valor: **50**
17. **Corregir pregunta**  
Como jugador quiero corregir preguntas propuestas por otros usuarios para obtener puntos

de usuario.

Estimación: **2** puntos de historia — Valor: **50**

#### 18. **Jugar Reto**

Como jugador quiero medir mis habilidades en un 'Reto' para obtener logros dentro del juego.

Estimación: **8** puntos de historia — Valor: **40**

#### 19. **Visualizar *Ranking***

Como jugador quiero ver un *ranking* de usuarios junto a mis amigos para saber el nivel de mis amigos.

Estimación: **4** puntos de historia — Valor: **30**

En esta actualización de la pila del producto se ha identificado un problema en la historia de usuario **Jugar individualmente**. El problema surge de la gran dificultad de estimar la tarea debido a su gran dimensión. Por lo tanto, para solucionar este problema se ha descompuesto la historia de usuario en 4 historias de usuario que nos permiten estimar más concretamente el trabajo a realizar. Por lo tanto, ahora se han añadido 4 nuevas historias de usuario a la pila del producto.

Además de estas 4 historias de usuario se ha añadido una nueva historia de usuario, con una prioridad baja y que se ha de implementar antes que la historia de usuario **Añadir una nueva pregunta**. La nueva historia de usuario implica seleccionar la categoría de la nueva historia de usuario.

### 6.3.2. Planificación del sprint

Antes de empezar el *sprint* 3 se han planificado un conjunto de 4 historias de usuario, que hacen un total de 9 ph ( $9 \text{ ph} * 5\text{h/ph} = 45 \text{ horas}$ ). En este caso se puede comprobar que la cantidad de horas necesarias para la ejecución del *sprint* es superior a la cantidad de tiempo disponible. Este hecho se debe al aumento de velocidad de desarrollo gracias a la adquisición de experiencia con la plataforma de desarrollo. Para evitar que esto vuelva a suceder en la próxima actualización de la pila del producto se volverán a planificar todas y cada una de las historias de usuario con el fin de estimar de nuevo las tareas de acuerdo a la nueva velocidad de desarrollo.

A continuación se muestran las historias de usuario y su desglose en tareas.

**Yo como usuario quiero que antes de jugar individualmente me muestre una pantalla con todas las categorías generales del juego y las categorías más recientes a la que he jugado para seleccionar la categoría a la que jugar.**

- Crear vista de las categorías. [2 horas]

- Crear el controlador de la vista. [3 horas]
- Construir un servicio para recuperar los datos de las categorías y de las últimas categorías jugadas. [3 horas]
- Construir métodos para obtener los datos. [2 horas]

Tiempo total: [10h] ⇒ 2 puntos historia (10 horas)

**Yo como jugador quiero que al seleccionar la categoría general sobre la que quiero jugar individualmente me muestre todas las subcategorías que tiene esta categoría para seleccionar una para jugar.**

- Crear vista de subcategorías. [2 horas]
- Crear el controlador de la vista. [3 horas]
- Construir un servicio para recuperar las subcategorías. [2 horas]

Tiempo total: [7h] ⇒ 2 puntos historia (10 horas)

**Yo como jugador quiero que me aparezca un conjunto de preguntas por pantalla de una en una para ir contestándolas.**

- Crear vista de juego individual. [3 horas]
- Crear el controlador de la vista. [5 horas]
- Programar la lógica de negocio del juego individual. [4 horas]
- Construir un servicio para recuperar las preguntas del juego. [1 hora]
- Construir un servicio para enviar los resultados del juego. [1 hora]

Tiempo total: [14h] ⇒ 4 puntos historia (20 horas)

**Yo como jugador quiero que mientras esté respondiendo una pregunta pueda obtener más tiempo para responder, simplificar las respuestas o pasar para ganar la partida más fácilmente.**

- Añadir a la vista del juego los nuevos botones. [2 horas]
- Añadir al controlador la nueva lógica de negocio del juego individual. [2 horas]

Tiempo total: [4h] ⇒ 2 puntos historia (10 horas)

### 6.3.3. Pruebas de aceptación

Además de realizar la planificación, también se han definido un conjunto de pruebas de aceptación que la aplicación debe cumplir al finalizar la iteración. Hasta el momento la aplicación había superado todas las pruebas de aceptación definidas. Pero, al realizar las pruebas de aceptación se ha detectado un problema gráfico en la representación de la vista del juego individual. Por lo tanto, al no superar las pruebas de aceptación, esta historia de usuario se mantendrá en la pila del sprint para realizar los cambios oportunos y realizar de nuevo la validación.

A continuación se listan las pruebas de aceptación propuestas para este *sprint* y si se han realizado de forma satisfactoria o errónea.

- **DADO** un usuario que se encuentra en el menú de la aplicación **CUANDO** accede a la opción de jugar individual **ENTONCES** la aplicación le mostrará todas las categorías del juego y la cantidad de veces que ha jugado a cada una de ellas.
- **DADO** un usuario que se encuentra en el menú de la aplicación **CUANDO** accede a la opción de jugar individual **ENTONCES** la aplicación le mostrará las categorías recientes a las cuales el usuario ha jugado recientemente.
- **DADO** un usuario que se encuentra en la pantalla de categorías **CUANDO** presiona sobre una categoría **ENTONCES** accede a la pantalla de subcategorías.
- **DADO** un usuario que se encuentra en la pantalla de categorías **CUANDO** presiona sobre una subcategoría reciente **ENTONCES** empezará el juego individual.
- **DADO** un usuario que empieza un juego individual **CUANDO** se muestra una pregunta **ENTONCES** se representará el título de la pregunta y sus posibles respuestas en cualquier dispositivo.
- **DADO** un usuario que se encuentra jugando individualmente **CUANDO** pasan 30 segundos sin responder **ENTONCES** la pregunta se declara como fallida y se presenta la siguiente.
- **DADO** un usuario que se encuentra jugando individualmente **CUANDO** presiona sobre la opción de tiempo extra **ENTONCES** se le añaden 15 segundos al cronómetro.
- **DADO** un usuario que se encuentra jugando individualmente **CUANDO** presiona sobre la opción de simplificar **ENTONCES** se desactivan dos respuestas incorrectas.
- **DADO** un usuario que se encuentra jugando individualmente **CUANDO** presiona sobre la opción de pasar preguntas **ENTONCES** la pregunta actual se declara como acertada y se muestra la siguiente pregunta.

### 6.3.4. Resultados obtenidos

Tras la finalización del tercer *sprint* hemos obtenido como resultado una versión mejorada de la aplicación obtenida tras el segundo *sprint*. En la figura 6.3 se muestra la jerarquía de pantallas y cuáles de ellas están implementadas.

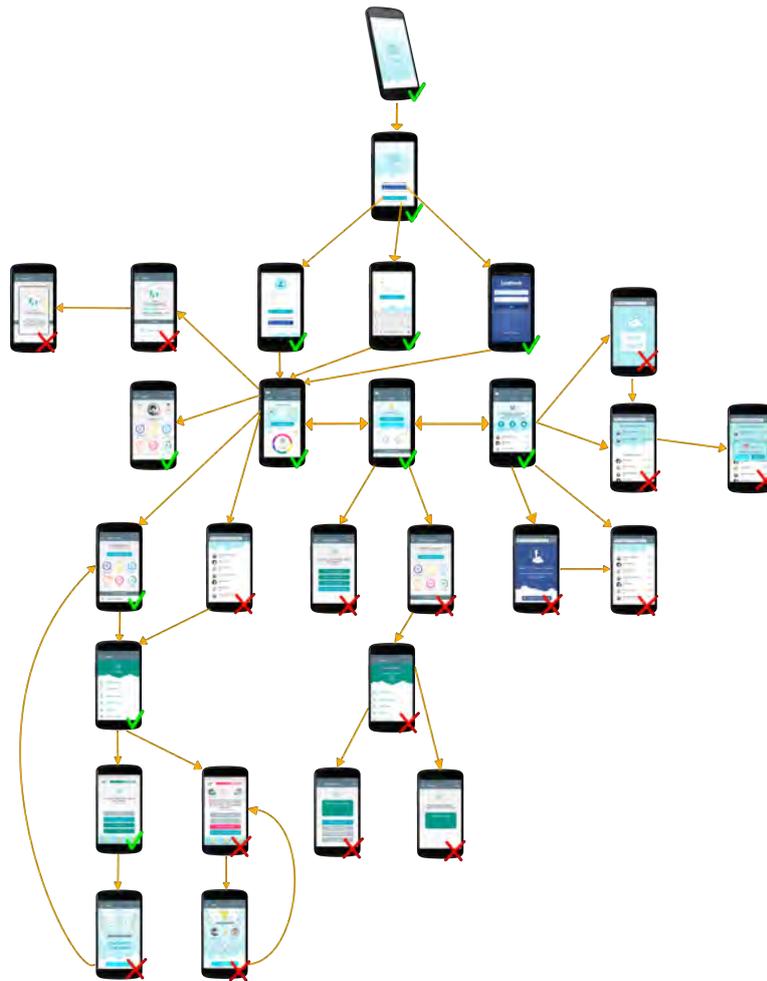


Figura 6.3: Jerarquía de pantallas completada Sprint 2.

La cantidad de puntos de historia completados es de 25, teniendo en cuenta que la cantidad total es de 59 podemos concluir que durante la primera iteración tenemos completado alrededor del 42 % de sistema.

## 6.4. Sprint 4

### 6.4.1. Actualización pila del producto

Tal y como se ha comentado en la sección anterior, tras aumentar la velocidad de desarrollo, la capacidad de trabajo es mayor y por lo tanto, las estimaciones son muy pesimistas. De este modo, antes de realizar la planificación de la cuarta iteración se ha vuelto a estimar las historias de usuario que ocupan la pila del producto. Cabe destacar que actualmente la velocidad de desarrollo ha aumentado con respecto a la velocidad inicial. Además que tras conocer el sistema y tener una mayor experiencia en esta tecnología nos permitirá estimar con mayor precisión.

*Las historias de usuario de color gris están completadas al 100 %, las historias de color verde se han añadido en esta revisión de la pila y se muestran en naranja los valores de la pila del producto que han cambiado respecto a la última versión.*

#### 1. Registrarse como jugador

Como jugador quiero registrarme como usuario para poder jugar en el juego.

Estimación: **2** puntos de historia — Valor: **80**

#### 2. Registrarse con Facebook

Como jugador quiero identificarme como usuario a través de mi cuenta de Facebook para no tener que crearme un usuario.

Estimación: **4** puntos de historia — Valor: **80**

#### 3. Identificarte como usuario

Como jugador quiero identificarme en la aplicación como usuario registrado para poder jugar.

Estimación: **4** puntos de historia — Valor: **80**

#### 4. Recordar contraseña

Como jugador quiero recordar mi contraseña a través del correo electrónico para poder identificarte en caso de olvidarla.

Estimación: **2** puntos de historia — Valor: **70**

#### 5. Menú

Como jugador quiero que la pantalla del menú esté formada por 3 pantallas(tabs) que contengan la información del juego, preguntas, temas sugeridos y amigos para poder ver de forma resumida las principales estadísticas y puntuaciones.

Estimación: **4** puntos de historia — Valor: **70**

#### 6. Ver información general

Como jugador quiero acceder a una pantalla de perfil de usuario donde pueda ver toda la información relativa a mi usuario como el nombre de usuario, correo electrónico y foto de perfil, y todos los resultados y estadísticas generales del juego (nivel, logros, puntos de usuario, progreso y preguntas sugeridas o corregidas) para conocer mi progreso en el

juego.

Estimación: 4 puntos de historia — Valor: 70

#### 7. **Mostrar Categorías**

Como usuario quiero que antes de jugar individualmente me muestre una pantalla con todas las categorías generales del juego y las categorías más recientes a la que he jugado para seleccionar la categoría a la que jugar.

Estimación: 2 puntos de historia — Valor: 80

#### 8. **Mostrar subcategorías**

Como jugador quiero que al seleccionar la categoría general sobre la que quiero jugar individualmente me muestre todas las subcategorías que tiene esta categoría para seleccionar una para jugar.

Estimación: 2 puntos de historia — Valor: 80

#### 9. **Contestar preguntas individualmente**

Como jugador quiero que me aparezca un conjunto de preguntas por pantalla de una en una e ir contestándolas.

Estimación: 4 puntos de historia — Valor: 75

#### 10. **Jugar individualmente**

Como jugador quiero que mientras esté respondiendo una pregunta pueda obtener más tiempo para responder, simplificar las respuestas o pasar para ganar la partida más fácilmente.

Estimación: 1 puntos de historia — Valor: 70

#### 11. **Añadir amigos contactos**

Como jugador quiero buscar a mis amigos a través de los números de teléfono almacenados en mi agenda para jugar contra ellos.

Estimación: 2 puntos de historia — Valor: 60

#### 12. **Añadir amigos contactos (email)**

Como jugador quiero buscar a mis amigos a través de los correos electrónicos almacenados en mi agenda para jugar contra ellos.

Estimación: 2 puntos de historia — Valor: 60

#### 13. **Añadir amigos *Facebook***

Como jugador quiero buscar a mis amigos a través del Facebook para jugar contra ellos.

Estimación: 2 puntos de historia — Valor: 60

#### 14. **Jugar Duelo**

Como jugador quiero retar a mis amigos a un 'Duelo' para obtener puntos extra.

Estimación: 4 puntos de historia — Valor: 60

#### 15. **Mostrar categorías para añadir pregunta**

Como usuario quiero que antes de formular una pregunta me muestre una pantalla con todas las categorías generales del juego y las últimas categorías más recientes que he sugerido para seleccionar la categoría sobre la cual quiero añadir una nueva categoría.

Estimación: 2 puntos de historia — Valor: 50

#### 16. **Añadir nueva pregunta**

Como jugador quiero añadir una nueva pregunta en el juego para obtener más puntos de usuario.

Estimación: **2** puntos de historia — Valor: **50**

#### 17. **Corregir pregunta**

Como jugador quiero corregir preguntas propuestas por otros usuarios para obtener puntos de usuario.

Estimación: **1** puntos de historia — Valor: **50**

#### 18. **Jugar Reto**

Como jugador quiero medir mis habilidades en un 'Reto' para obtener logros dentro del juego.

Estimación: **4** puntos de historia — Valor: **40**

#### 19. **Visualizar *Ranking***

Como jugador quiero ver un *ranking* de usuarios junto a mis amigos para saber el nivel de mis amigos.

Estimación: **2** puntos de historia — Valor: **30**

Muchas de las historias de usuario que actualmente se encuentran en la pila del producto se han estimado de nuevo. Con esto hemos conseguido actualizar las estimaciones con un valor realista y no realizando una estimación basándonos en nuestra experiencia previa.

### 6.4.2. Planificación del sprint

Antes de empezar el cuarto *sprint* se ha planificado una serie de 4 historias de usuario sumando un total de 10 ph ( $10 \text{ ph} * 5\text{h/ph} = 50 \text{ horas}$ ). Cabe destacar que una de las historias de usuario tiene alrededor de 3 horas de trabajo y su estimación es de 4 puntos de historia (20 horas). A continuación se muestran las historias de usuario y su desglose en tareas.

**Yo como jugador quiero que me aparezca un conjunto de preguntas por pantalla de una en una para ir contestándolas.**

- Solucionar problema gráfico en la vista de juego individual. [3 horas]

Tiempo total: [3h]  $\Rightarrow$  1 punto historia (5 horas)

**Yo como jugador quiero buscar a mis amigos a través de los números de teléfono almacenados en mi agenda para jugar contra ellos.**

- Crear vista para mostrar el listado de amigos. [2 horas]

- Añadir el controlador para representar amigos. [2 horas]
- Crear servicio web para obtener amigos registrados en el juego. [3 horas]

Tiempo total: [7h] ⇒ 2 puntos historia (10 horas)

**Yo como jugador quiero buscar a mis amigos a través de los correos electrónicos almacenados en mi agenda para jugar contra ellos.**

- Crear vista para mostrar el listado de amigos. [2 horas]
- Añadir el controlador para representar amigos. [2 horas]
- Crear servicio web para obtener amigos registrados en el juego. [3 horas]

Tiempo total: [7h] ⇒ 2 puntos historia (10 horas)

**Yo como jugador quiero buscar a mis amigos a través de Facebook para jugar contra ellos.**

- Crear vista para mostrar el listado de amigos de Facebook. [2 horas]
- Añadir el controlador para representar amigos. [2 horas]
- Crear servicio web para obtener amigos registrados en el juego a través de Facebook. [3 horas]

Tiempo total: [7h] ⇒ 2 puntos historia (10 horas)

### 6.4.3. Pruebas de aceptación

Como sucede en cada iteración se han definido un conjunto de pruebas de aceptación que la aplicación debe cumplir al finalizar la iteración. Cabe destacar que en este *sprint* se han superado correctamente todas las pruebas de aceptación, incluyendo la prueba de aceptación que no se había sido satisfecha en el sprint anterior.

A continuación se listan las pruebas de aceptación propuestas para este *sprint* y si se han realizado de forma satisfactoria o errónea.

- **DADO** un usuario que empieza un juego individual **CUANDO** se muestra una pregunta **ENTONCES** se representará el título de la pregunta y sus posibles respuestas en cualquier dispositivo.

- **DADO** un usuario que se encuentra en la sección de amigos **CUANDO** accede a la opción de buscar amigos en el juego a través de sus contactos **ENTONCES** la aplicación le mostrará los usuarios almacenados en contactos (teléfono) y que además sean usuarios en el juego.
- **DADO** un usuario que se encuentra en la sección de amigos **CUANDO** accede a la opción de buscar amigos en el juego a través de sus emails **ENTONCES** la aplicación le mostrará los usuarios almacenados en contactos (email) y que además sean usuarios en el juego.
- **DADO** un usuario que se encuentra en la sección de amigos **CUANDO** accede a la opción de buscar amigos en el juego a través de Facebook y no está registrado a través de Facebook **ENTONCES** la aplicación le mostrará una pantalla informativa y un mecanismo para vincular su cuenta a la cuenta de Facebook.
- **DADO** un usuario que se encuentra en la sección de amigos **CUANDO** accede a la opción de buscar amigos en el juego a través de Facebook **ENTONCES** la aplicación le mostrará los jugadores de la app que son amigos suyos en Facebook.
- **DADO** un usuario que se encuentra en la sección de amigos **CUANDO** accede a la opción de buscar amigos en el juego y aún no ha jugado contra nadie **ENTONCES** la aplicación le mostrará una ventana informativa.
- **DADO** un usuario que se encuentra en la sección de amigos **CUANDO** accede a la opción de buscar amigos en el juego y realizado partidas contra jugadores desconocidos **ENTONCES** se mostrará un listado de todos los jugadores contra los que ha jugado y no son amigos.

#### 6.4.4. Resultados obtenidos

Tras la la cuarta iteración sobre la aplicación hemos obtenido de nuevo una aplicación parcial y completamente funcional. En la figura 6.4 se muestra la jerarquía de pantallas y cuáles de ellas están implementadas.

La cantidad de puntos de historia completados es de 35, teniendo en cuenta que la cantidad total es de 50 podemos concluir que durante la primera iteración tenemos completado alrededor del 70 % de sistema.



## 6.5. Sprint 5

### 6.5.1. Actualización pila del producto

Es esta revisión de la pila del producto se ha añadido una nueva historia de usuario debido a petición del responsable del proyecto. Se ha añadido esta historia ya que debido al estado avanzado del proyecto se puede permitir añadir nuevas historias de usuario.

*Las historias de usuario de color gris están completadas al 100 %, las historias de color verde se han añadido en esta revisión de la pila y se muestran en naranja los valores de la pila del producto que han cambiado respecto a la última versión.*

#### 1. Registrarse como jugador

Como jugador quiero registrarme como usuario para poder jugar en el juego.

Estimación: **2** puntos de historia — Valor: **80**

#### 2. Registrarse con Facebook

Como jugador quiero identificarme como usuario a través de mi cuenta de Facebook para no tener que crearme un usuario.

Estimación: **4** puntos de historia — Valor: **80**

#### 3. Identificarte como usuario

Como jugador quiero identificarme en la aplicación como usuario registrado para poder jugar.

Estimación: **4** puntos de historia — Valor: **80**

#### 4. Recordar contraseña

Como jugador quiero recordar mi contraseña a través del correo electrónico para poder identificarte en caso de olvidarla.

Estimación: **2** puntos de historia — Valor: **70**

#### 5. Menú

Como jugador quiero que la pantalla del menú esté formada por 3 pantallas(tabs) que contengan la información del juego, preguntas, temas sugeridos y amigos para poder ver de forma resumida las principales estadísticas y puntuaciones.

Estimación: **4** puntos de historia — Valor: **70**

#### 6. Ver información general

Como jugador quiero acceder a una pantalla de perfil de usuario donde pueda ver toda la información relativa a mi usuario como el nombre de usuario, correo electrónico y foto de perfil, y todos los resultados y estadísticas generales del juego (nivel, logros, puntos de usuario, progreso y preguntas sugeridas o corregidas) para conocer mi progreso en el juego.

Estimación: **4** puntos de historia — Valor: **70**

7. **Mostrar Categorías**  
Como usuario quiero que antes de jugar individualmente me muestre una pantalla con todas las categorías generales del juego y las categorías más recientes a la que he jugado para seleccionar la categoría a la que jugar.  
Estimación: 2 puntos de historia — Valor: 80
8. **Mostrar subcategorías**  
Como jugador quiero que al seleccionar la categoría general sobre la que quiero jugar individualmente me muestre todas las subcategorías que tiene esta categoría para seleccionar una para jugar.  
Estimación: 2 puntos de historia — Valor: 80
9. **Contestar preguntas individualmente**  
Como jugador quiero que me aparezca un conjunto de preguntas por pantalla de una en una e ir contestándolas.  
Estimación: 4 puntos de historia — Valor: 75
10. **Jugar individualmente**  
Como jugador quiero que mientras esté respondiendo una pregunta pueda obtener más tiempo para responder, simplificar las respuestas o pasar para ganar la partida más fácilmente.  
Estimación: 1 puntos de historia — Valor: 70
11. **Añadir amigos contactos**  
Como jugador quiero buscar a mis amigos a través de los números de teléfono almacenados en mi agenda para jugar contra ellos.  
Estimación: 2 puntos de historia — Valor: 60
12. **Añadir amigos contactos (email)**  
Como jugador quiero buscar a mis amigos a través de los correos electrónicos almacenados en mi agenda para jugar contra ellos.  
Estimación: 2 puntos de historia — Valor: 60
13. **Añadir amigos *Facebook***  
Como jugador quiero buscar a mis amigos a través del Facebook para jugar contra ellos.  
Estimación: 2 puntos de historia — Valor: 60
14. **Resumen juego individual**  
Como jugador quiero que tras jugar un juego individual me muestre la cantidad de preguntas que he acertado.  
Estimación: 1 puntos de historia — Valor: 60
15. **Jugar Duelo**  
Como jugador quiero retar a mis amigos a un 'Duelo' para obtener puntos extra.  
Estimación: 4 puntos de historia — Valor: 60
16. **Mostrar categorías para añadir pregunta**  
Como usuario quiero que antes de formular una pregunta me muestre una pantalla con

todas las categorías generales del juego y las últimas categorías más recientes que he sugerido para seleccionar la categoría sobre la cual quiero añadir una nueva categoría.

Estimación: **2** puntos de historia — Valor: **50**

**17. Añadir nueva pregunta**

Como jugador quiero añadir una nueva pregunta en el juego para obtener más puntos de usuario.

Estimación: **2** puntos de historia — Valor: **50**

**18. Corregir pregunta**

Como jugador quiero corregir preguntas propuestas por otros usuarios para obtener puntos de usuario.

Estimación: **1** puntos de historia — Valor: **50**

**19. Jugar Reto**

Como jugador quiero medir mis habilidades en un 'Reto' para obtener logros dentro del juego.

Estimación: **4** puntos de historia — Valor: **40**

**20. Visualizar *Ranking***

Como jugador quiero ver un *ranking* de usuarios junto a mis amigos para saber el nivel de mis amigos.

Estimación: **2** puntos de historia — Valor: **30**

### 6.5.2. Planificación del sprint

Las historias de usuario seleccionadas para realizar en el quinto *sprint* son las siguientes:

**Yo como jugador quiero que tras jugar un juego individual me muestre la cantidad de preguntas que he acertado..**

- Crear la vista de la pantalla. [1 horas]
- Crear controlador para la vista. [2 horas]
- Crear servicio para obtener los datos a representar. [1 horas]

Tiempo total: [4h] ⇒ 1 punto historia (5 horas)

**Yo como jugador quiero retar a mis amigos a un 'Reto' para obtener puntos extra..**

- Crear vista de selección de usuario del reto. [1 horas]

- Crear controlador para la vista de selección de jugador. [1 horas]
- Crear servicio web para obtener jugadores del juego. [1 horas]
- Adaptar pantalla de selección de categoría para seleccionar una categoría. [1 horas]
- Adaptar pantalla de selección de subcategoría para seleccionar una categoría. [1 horas]
- Crear servicios de comunicación (servicios). [8 horas]
- Crear vista del duelo. [2 horas]
- Crear controlador y lógica del duelo. [3 horas]
- Crear vista de resumen de duelo. [1 horas]
- Crear controlador para resumen del duelo. [1 horas]

Tiempo total: [20h]  $\Rightarrow$  4 puntos historia (20 horas)

**Yo como jugador quiero corregir preguntas propuestas por otros usuarios para obtener puntos de usuario.**

- Crear vista para corregir pregunta. [2 horas]
- Crear controlador de la vista. [1 horas]
- Crear servicio para obtener pregunta a corregir y enviar la respuesta. [2 horas]

Tiempo total: [5h]  $\Rightarrow$  1 puntos historia (5 horas)

**Yo como usuario quiero que antes de formular una pregunta me muestre una pantalla con todas las categorías generales del juego y las últimas categorías más recientes que he sugerido para seleccionar la categoría sobre la cual quiero añadir una nueva categoría..**

- Crear vista para mostrar categorías y subcategorías. [2 horas]
- Crear controlador de la vista. [2 horas]
- Crear servicio para obtener categorías y subcategorías recientes. [3 horas]

Tiempo total: [7h]  $\Rightarrow$  2 puntos historia (10 horas)

Para esta iteración se han planificado 4 historias de usuario. La suma de sus estimaciones es de 8 ph (7 ph \* 5h/ph = 40 horas).

### 6.5.3. Pruebas de aceptación

Para esta iteración en concreto se han definido el siguiente conjunto de pruebas de aceptación:

- **DADO** un usuario que realiza un juego individual **CUANDO** termina de contestar todas las preguntas **ENTONCES** se muestra en la pantalla la cantidad de preguntas que ha respondido bien.
- **DADO** un usuario que se encuentra en el menú de la aplicación **CUANDO** pulsa el botón de duelo **ENTONCES** se muestra una lista de los usuarios disponibles para realizar un duelo.
- **DADO** un usuario que se encuentra en la pantalla para seleccionar un jugador para el duelo **CUANDO** selecciona un jugador **ENTONCES** se muestran las posibles categorías para jugar y las subcategorías más recientes.
- **DADO** un usuario que se encuentra en la pantalla de selección de categoría **CUANDO** selecciona una categoría **ENTONCES** se muestran las posibles subcategorías para jugar.
- **DADO** un usuario que se encuentra en la pantalla de selección de categoría **CUANDO** selecciona una categoría **ENTONCES** se muestran las posibles subcategorías para jugar.
- **DADO** un usuario que ha creado un duelo **CUANDO** el oponente acepta el duelo **ENTONCES** pregunta al usuario cuando desea jugar.
- **DADO** un usuario retado a un duelo **CUANDO** el oponente envía petición del duelo **ENTONCES** llega una notificación.
- **DADO** un usuario que quiere corregir nuevas preguntas **CUANDO** pulsa el botón de corregir pregunta **ENTONCES** se muestra una pregunta a corregir.
- **DADO** un usuario que está corrigiendo una pregunta y pulsa **CUANDO** pulsa en la respuesta correcta **ENTONCES** se muestra diálogo informado que la opción es la correcta.
- **DADO** un usuario que está corrigiendo una pregunta y pulsa **CUANDO** pulsa en la respuesta incorrecta **ENTONCES** se muestra diálogo informado que la opción es incorrecta y la opción de reportar error de la pregunta.
- **DADO** un usuario se encuentra en el menú de la aplicación **CUANDO** accede a añadir una nueva pregunta **ENTONCES** aparece una pantalla para seleccionar la categoría de la pregunta que desea añadir.

### 6.5.4. Resultados obtenidos

Como se muestra en la lista de pruebas de aceptación, solamente se han cumplido 7 de las 11 pruebas de aceptación diseñadas al inicio de la iteración. En principal motivo de no

alcanzar los objetivos de esta iteración se debe a la gran cantidad de problemas acaecidos a la hora de construir el sistema de comunicación del duelo. Estos problemas han visto reflejados ocasionando un retraso del resto de historias de usuario e impidiendo la completa realización de estas. Es necesario resaltar que la historia de usuario **Jugar Duelo** es la historia de usuario más compleja y crítica del sistema.

Tras la quinta iteración sobre la aplicación hemos obtenido de nuevo una aplicación completamente funcional, aunque incompleta. En la figura 6.5 se muestra la jerarquía de pantallas y cuáles de ellas están implementadas.

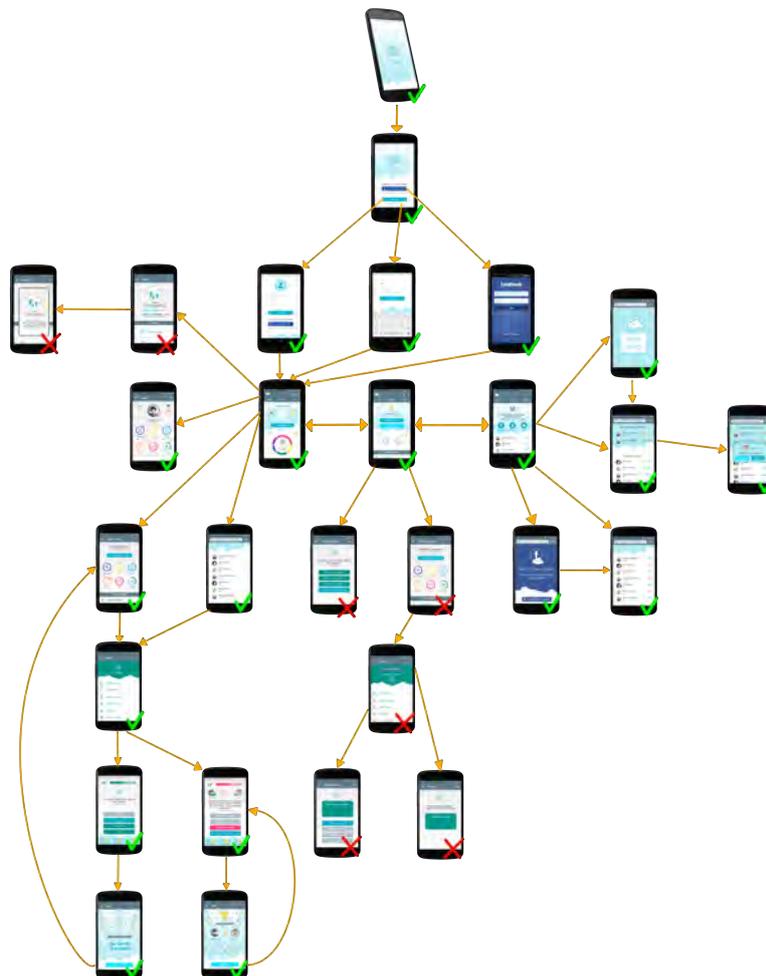


Figura 6.5: Jerarquía de pantallas completada Sprint 5.

La cantidad de puntos de historia completados es de 35, teniendo en cuenta que la cantidad total es de 48 podemos concluir que durante la primera iteración tenemos completado alrededor del 72% de sistema.

## 6.6. Sprint 6

### 6.6.1. Actualización pila del producto

A continuación se muestra la pila del producto tras la última revisión. Cabe destacar que se han eliminado dos historias de usuario que no aportan valor para el usuario y que quedaban fuera del alcance temporal del proyecto. Por lo tanto se ha decidido desecharlas y añadir dos nuevas historia de usuario que aporta mucho más valor para el usuario.

*Las historias de usuario de color gris están completadas al 100 %, las historias de color verde se han añadido en esta revisión de la pila y se muestran en naranja los valores de la pila del producto que han cambiado respecto a la última versión.*

#### 1. Registrarse como jugador

Como jugador quiero registrarme como usuario para poder jugar en el juego.

Estimación: 2 puntos de historia — Valor: 80

#### 2. Registrarse con Facebook

Como jugador quiero identificarme como usuario a través de mi cuenta de Facebook para no tener que crearme un usuario.

Estimación: 4 puntos de historia — Valor: 80

#### 3. Identificarte como usuario

Como jugador quiero identificarme en la aplicación como usuario registrado para poder jugar.

Estimación: 4 puntos de historia — Valor: 80

#### 4. Recordar contraseña

Como jugador quiero recordar mi contraseña a través del correo electrónico para poder identificarte en caso de olvidarla.

Estimación: 2 puntos de historia — Valor: 70

#### 5. Menú

Como jugador quiero que la pantalla del menú esté formada por 3 pantallas(tabs) que contengan la información del juego, preguntas, temas sugeridos y amigos para poder ver de forma resumida las principales estadísticas y puntuaciones.

Estimación: 4 puntos de historia — Valor: 70

#### 6. Ver información general

Como jugador quiero acceder a una pantalla de perfil de usuario donde pueda ver toda la información relativa a mi usuario como el nombre de usuario, correo electrónico y foto de perfil, y todos los resultados y estadísticas generales del juego (nivel, logros, puntos de usuario, progreso y preguntas sugeridas o corregidas) para conocer mi progreso en el juego.

Estimación: 4 puntos de historia — Valor: 70

7. **Mostrar Categorías**  
Como usuario quiero que antes de jugar individualmente me muestre una pantalla con todas las categorías generales del juego y las categorías más recientes a la que he jugado para seleccionar la categoría a la que jugar.  
Estimación: **2** puntos de historia — Valor: **80**
8. **Mostrar subcategorías**  
Como jugador quiero que al seleccionar la categoría general sobre la que quiero jugar individualmente me muestre todas las subcategorías que tiene esta categoría para seleccionar una para jugar.  
Estimación: **2** puntos de historia — Valor: **80**
9. **Contestar preguntas individualmente**  
Como jugador quiero que me aparezca un conjunto de preguntas por pantalla de una en una e ir contestándolas.  
Estimación: **4** puntos de historia — Valor: **75**
10. **Jugar individualmente**  
Como jugador quiero que mientras esté respondiendo una pregunta pueda obtener más tiempo para responder, simplificar las respuestas o pasar para ganar la partida más fácilmente.  
Estimación: **1** puntos de historia — Valor: **70**
11. **Añadir amigos contactos**  
Como jugador quiero buscar a mis amigos a través de los números de teléfono almacenados en mi agenda para jugar contra ellos.  
Estimación: **2** puntos de historia — Valor: **60**
12. **Añadir amigos contactos (email)**  
Como jugador quiero buscar a mis amigos a través de los correos electrónicos almacenados en mi agenda para jugar contra ellos.  
Estimación: **2** puntos de historia — Valor: **60**
13. **Añadir amigos *Facebook***  
Como jugador quiero buscar a mis amigos a través del Facebook para jugar contra ellos.  
Estimación: **2** puntos de historia — Valor: **60**
14. **Resumen juego individual**  
Como jugador quiero que tras jugar un juego individual me muestre la cantidad de preguntas que he acertado.  
Estimación: **1** puntos de historia — Valor: **60**
15. **Jugar Duelo**  
Como jugador quiero retar a mis amigos a un ‘Duelo’ para obtener puntos extra.  
Estimación: **4** puntos de historia — Valor: **60**
16. **Mostrar categorías para añadir pregunta**  
Como usuario quiero que antes de formular una pregunta me muestre una pantalla con

todas las categorías generales del juego y las últimas categorías más recientes que he sugerido para seleccionar la categoría sobre la cual quiero añadir una nueva categoría.

Estimación: **2** puntos de historia — Valor: **50**

**17. Añadir nueva pregunta**

Como jugador quiero añadir una nueva pregunta en el juego para obtener más puntos de usuario.

Estimación: **2** puntos de historia — Valor: **50**

**18. Corregir pregunta**

Como jugador quiero corregir preguntas propuestas por otros usuarios para obtener puntos de usuario.

Estimación: **1** puntos de historia — Valor: **50**

**19. Añadir nueva categoría**

Como jugador quiero poder añadir una nueva subcategoría dentro de una categoría ya existente.

Estimación: **2** puntos de historia — Valor: **50**

**20. Mostrar información niveles de usuario**

Como jugador quiero conocer la información a cerca de todos los niveles de usuario que existen en la aplicación.

Estimación: **2** puntos de historia — Valor: **50**

**21. Jugar Reto**

Como jugador quiero medir mis habilidades en un 'Reto' para obtener logros dentro del juego.

Estimación: **4** puntos de historia — Valor: **40**

**22. Visualizar *Ranking***

Como jugador quiero ver un *ranking* de usuarios junto a mis amigos para saber el nivel de mis amigos.

Estimación: **2** puntos de historia — Valor: **30**

### 6.6.2. Planificación del sprint

Las historias de usuario seleccionadas para realizar en el quinto *sprint* son las siguientes:

**Yo como usuario quiero que antes de formular una pregunta me muestre una pantalla con todas las categorías generales del juego y las últimas categorías más recientes que he sugerido para seleccionar la categoría sobre la cual quiero añadir una nueva categoría.**

- Crear vista para mostrar categorías y subcategorías. [2 horas]

- Crear controlador de la vista. [2 horas]
- Crear servicio para obtener categorías y subcategorías recientes. [3 horas]

Tiempo total: [7h] ⇒ 2 puntos historia (10 horas)

**Yo como jugador quiero corregir preguntas propuestas por otros usuarios para obtener puntos de usuario.**

- Crear vista para corregir pregunta. [2 horas]
- Crear controlador de la vista. [1 horas]
- Crear servicio para obtener pregunta a corregir y enviar la respuesta. [2 horas]

Tiempo total: [5h] ⇒ 1 puntos historia (5 horas)

**Yo como jugador quiero añadir una nueva pregunta en el juego para obtener más puntos de usuario.**

- Crear vista para añadir nueva pregunta. [2 horas]
- Crear controlador de la vista y validador de datos. [3 horas]
- Crear servicio para enviar la pregunta y recibir resultado. [2 horas]

Tiempo total: [7h] ⇒ 2 puntos historia (10 horas)

**Yo como jugador quiero poder añadir una nueva subcategoría dentro de una categoría ya existente.**

- Crear vista para añadir nueva pregunta. [2 horas]
- Crear controlador de la vista y validador de datos. [3 horas]
- Crear servicio para enviar la pregunta y recibir resultado. [2 horas]

Tiempo total: [7h] ⇒ 2 puntos historia (10 horas)

**Yo como jugador quiero conocer la información a cerca de todos los niveles de usuario que existen en la aplicación.**

- Crear vista para mostrar todos los niveles. [1 horas]

- Crear controlador de la vista. [1 horas]
- Crear *pop-up* para mostrar en detalle el nivel de usuario. [2 horas]

Tiempo total: [4h]  $\Rightarrow$  1 punto historia (5 horas)

Para esta iteración se han planificado 4 historias de usuario. La suma de sus estimaciones es de 8 ph ( $8 \text{ ph} * 5\text{h/ph} = 40 \text{ horas}$ ).

### 6.6.3. Pruebas de aceptación

Para esta iteración en concreto se han definido el siguiente conjunto de pruebas de aceptación:

- **DADO** un usuario que quiere corregir nuevas preguntas **CUANDO** pulsa el botón de corregir pregunta **ENTONCES** se muestra una pregunta a corregir.
- **DADO** un usuario que está corrigiendo una pregunta y pulsa **CUANDO** pulsa en la respuesta correcta **ENTONCES** se muestra diálogo informado que la opción es la correcta.
- **DADO** un usuario que está corrigiendo una pregunta y pulsa **CUANDO** pulsa en la respuesta incorrecta **ENTONCES** se muestra diálogo informado que la opción es incorrecta y la opción de reportar error de la pregunta.
- **DADO** un usuario se encuentra en el menú de la aplicación **CUANDO** accede a añadir una nueva pregunta **ENTONCES** aparece una pantalla para seleccionar la categoría de la pregunta que desea añadir.
- **DADO** un usuario se encuentra en la pantalla de añadir categoría e introduce un nombre para la nueva categoría **CUANDO** presiona el botón de añadir **ENTONCES** añade la nueva categoría y muestra un mensaje.
- **DADO** un usuario se encuentra en la pantalla de añadir categoría y no ha introducido un nombre para la nueva categoría **CUANDO** presiona el botón de añadir **ENTONCES** muestra un mensaje de error y no se añade ninguna categoría.
- **DADO** un usuario se encuentra en la pantalla de añadir pregunta e introduce los datos de la nueva pregunta **CUANDO** presiona el botón de añadir **ENTONCES** se añade la nueva pregunta y muestra un mensaje de confirmación.
- **DADO** un usuario se encuentra en la pantalla de añadir pregunta no introduce los datos correctos de la nueva pregunta **CUANDO** presiona el botón de añadir **ENTONCES** no se añade ninguna pregunta y muestra un mensaje de error.

- **DADO** un usuario se encuentra en el menú de la aplicación o en la pantalla de perfil **CUANDO** presiona en la categoría **ENTONCES** se muestra la pantalla de niveles de usuario indicando en el nivel que se encuentra el usuario.
- **DADO** un usuario se encuentra en el listado de niveles de usuario **CUANDO** presiona sobre un nivel **ENTONCES** se muestra un mensaje indicando las características de ese nivel en concreto.

#### 6.6.4. Resultados obtenidos

Finalmente, tras finalizar la iteración la aplicación cumple con todas y cada una de las pruebas de aceptación, no solo las diseñadas para las historias de usuario de este *sprint*, sino todas las pruebas de aceptación diseñadas a lo largo de la construcción de la aplicación.

En la figura 6.6 se muestra la jerarquía de pantallas completa tras la finalización del sexto y último *sprint*.

Llegados a este punto hemos obtenido una aplicación móvil completa y funcional similar a alguna de las aplicaciones disponibles en los principales canales de distribución de aplicaciones móviles como **Google Play**.

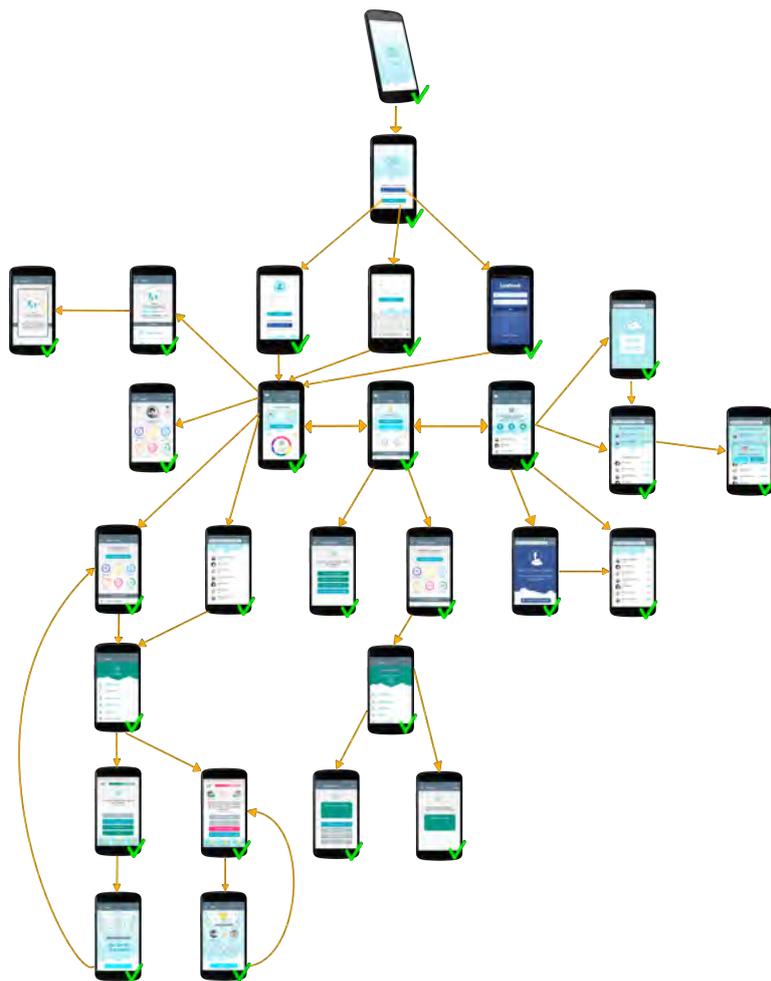


Figura 6.6: Jerarquía de pantallas completada Sprint 6.



## Capítulo 7

# Documentación

A lo largo del código escrito para la construcción de la aplicación se han desarrollado un componentes reutilizable que se presenta en la siguiente sección.

### 7.1. SocketIOUtils

`SocketIOUtils` es una clase de utilidad construida utilizando el patrón de diseño **Singleton**, detallado en la sección 5.1.

La gran mayoría de las comunicaciones que se realizan en el sistema entre la aplicación y el servidor se realizan mediante **Sockets I/O**(mecanismo de transporte) y **Node.js** (servidor). Por lo tanto, para realizar una conexión de sockets o *Socket Connection* hemos utilizado una librería de terceros llamada **Socket.IO** [[1]]. Esta librería nos permite crear un objeto de tipo **Socket** a través del cual podemos mantener la comunicación.

El problema de utilizar esta librería aparece porque es necesario mantener una única referencia de este objeto **Socket**. Esto sucede por varios motivos. En primer lugar, debemos de proporcionar un único punto de comunicación entre la aplicación y el servidor. En segundo lugar, solo podemos crear una única conexión, ya que el servidor necesita conocer en cada momento los clientes (dispositivo móvil) que establecen comunicación directamente con él.

El principal motivo por el cual se ha decidido utilizar este tipo de implementación es por la necesidad de almacenar una única instancia del objeto **Socket**. Es muy importante mantener una **única** instancia de este objeto por los siguientes motivos. En primer lugar, porque el coste de crear la instancia es muy alto, de este modo, si tenemos que crear constantemente instancias de este objeto el rendimiento será muy bajo. En segundo lugar, porque debemos de proporcionar un único punto de acceso a la referencia creada y accesible desde cualquier punto de la aplicación, es decir, a lo largo de toda la aplicación solamente de crear una instancia de este objeto y es

utilizado en una gran cantidad de clases.

A continuación se muestra el código de este objeto:

```
public class SocketIOUtils {

    private static volatile SocketIOUtils instance;

    private Socket socket;

    private SocketIOUtils(){
        super();
        // Create socket
        try {
            this.socket = IO.socket(URL_SOCKET);
            this.socket.connect();
        } catch (URISyntaxException e) {
            e.printStackTrace();
        }
    }

    public static SocketIOUtils getInstance(){
        if ( instance == null ){
            synchronized ( SocketIOUtils.class ){
                if ( instance == null ){
                    instance = new SocketIOUtils();
                }
            }
        }
        return instance;
    }

    public Socket getSocket() {
        return this.socket;
    }

    public boolean closeSocket(){
        if( this.socket != null ){
            this.socket.close();
            return true;
        }
        return false;
    }
}
```

}

Existe una amplia variedad de alternativas para solucionar nuestro problema con este objeto. Aunque en nuestro contexto es la solución más óptima, sencilla y elegante.

## 7.2. DeviceUtils y GenericUtils

Además de la clase de utilidad descrita en la sección anterior, también se ha desarrollado un conjunto de clases de utilidad que **exponen métodos estáticos** para facilitar la vida a los desarrolladores. Estas clases nos facilitan la realización de tareas complejas e independientes.

La clase `DeviceUtils` nos proporciona 2 métodos estáticos:

`getNumberPhonesToContacts()` : permite obtener los números de teléfono de la agenda de nuestro dispositivo en un `Array<String>`.

`getEmailsToContacts()` : permite obtener los emails de la agenda de nuestro dispositivo en un `Array<String>`.

La clase `GenericUtils` nos proporciona un método estático llamado `forceKeyboardClose()` que se encarga de ocultar el teclado del dispositivo en el caso que esté abierto. En cualquier otro caso no realiza ninguna acción.

Este tipo de clases de utilidad pueden ser llamadas desde cualquier punto de la aplicación sin la necesidad de instanciar ningún objeto.



## Capítulo 8

# Futuras extensiones y aplicación

### Índice

---

8.1. Futuras extensiones del sistema . . . . .	91
8.2. Aplicaciones . . . . .	92
8.3. Estrategia comercial . . . . .	93

---

### 8.1. Futuras extensiones del sistema

Actualmente está construida la aplicación Android, *Web Services* y *Sockets IO* en Node.js para realizar la comunicación. Por lo tanto, a corto plazo se pretende construir la aplicación equivalente para dispositivos **iOS** y una **página web** para mostrar información relativa del sistema y una zona de usuarios donde puedan sugerir preguntas, crear temas nuevos y consultar su puntuación.

Por lo que respecta a las futuras extensiones de la aplicación Android ya construida, se pretende construir un **ranking** de usuarios, retos de usuarios, un sistema de reconocimiento de logros y una tienda virtual para poder obtener *tokens* a cambio de una pequeña cantidad de dinero.

Además de estas extensiones, existen centenares de nuevas ideas que se pueden añadir a la funcionalidad de la aplicación y que gracias a la arquitectura escalable, modificable y extensible que hemos construido nos ofrece la posibilidad de seguir desarrollando la aplicación y su sistema sin la necesidad de rediseñar la arquitectura o cualquier otro módulo ya implementado.

## 8.2. Aplicaciones

Hoy en día, las grandes compañías tienen como objetivo lanzar al mercado nuevos productos y servicios que revolucionaran el mundo y les aporten una gran cantidad de beneficios económicos. Dejando de lado el aspecto económico, una aplicación de estas características puede causar un gran impacto social potenciando la cultura y el conocimiento en los más jóvenes y en aquellos no tan jóvenes.

Existen una amplia variedad de utilidades para esta aplicación enumeradas a continuación.

### Educación

Existe una gran cantidad de artículos científicos que demuestran que los humanos y en especial los niños demuestran una mayor capacidad de absorción de conocimientos cuando interactúan con sistemas de estas características. Aunque actualmente solo sea una minoría, algunos centros de educación primaria y secundaria está empezando a implantar nuevos métodos de enseñanza basados en las nuevas tecnologías, utilizando aplicaciones informáticas semejantes a esta.

### Entretenimiento

Actualmente existe una gran cantidad de aplicaciones similares a la implementada. Este tipo de aplicaciones se desarrollan con fines comerciales y con un único objetivo, entretener a sus clientes. Muchos son los usuarios que aprovechan cualquier momento del día para consultar su dispositivo y utilizar alguna aplicación de este tipo. Las situaciones más comunes suelen ser en una sala de espera, esperando a un amigo, patio de un instituto, ...

### Sanitaria

Cada vez más son las personas afectadas por enfermedades como la amnesia o cualquier otra enfermedad que produzca pérdidas de memoria o deficiencia mental. Este tipo de aplicaciones pueden llegar a ser extremadamente beneficiosas con este tipo de pacientes. El principal motivo es que este tipo de aplicaciones nos **obligan a pensar** y estar atentos a ciertos eventos, aumentando el nivel de concentración y provocando resultados realmente increíbles.

## Social

El carácter social de este tipo de juegos también puede resultar de ayuda a personas con trastornos mentales caracterizados por alteración de la interacción social y la comunicación verbal como el autismo. Gracias a este tipo de aplicaciones se les proporciona un nuevo mecanismo de comunicación e interacción con el resto de civilización. Provocando en ellos un sentimiento social e integrando en la sociedad a este tipo de personas.

### 8.3. Estrategia comercial

A pesar de la gran cantidad de beneficios sociales y culturales que puede aportar la aplicación se ha de buscar una estrategia o posicionamiento comercial. El abanico de posibilidades para monetizar la aplicación nos ofrece las siguientes posibilidades.

- **Vender el sistema completo** a otra compañía.
- Ofrecer a los actuales clientes la posibilidad de **añadir el juego en su aplicación comercial**.
- **Publicar la aplicación** en los principales canales de distribución. Para esto sería necesario monetizar la aplicación con una de las siguientes opciones:
  - **Aplicación gratuita con publicidad.**
  - **Aplicación Freemium.** Ofreciendo la aplicación gratuitamente y con opción a abonar una pequeña cantidad de dinero para ofrecer servicios exclusivos. *Spotify* o *Dropbox* siguen esta política.
  - **Aplicación de pago.**
- Con el sistema ya construido es posible **obtener varias aplicaciones** y publicar juegos similares pero con diferente temática.

Como se puede comprobar, la estrategia comercial de las aplicaciones puede llegar a ser un gran problema para las *StartUp* o jóvenes empresas que quieren lanzar su aplicación o servicio al mercado. El motivo de este gran problema es debido por la gran incertidumbre e inestabilidad que presenta actualmente el sector de las aplicaciones móviles y el sector tecnológico en general.



## Capítulo 9

# Conclusiones

### Índice

---

9.1. Resumen . . . . .	95
9.2. Objetivos logrados . . . . .	96
9.3. Valoración personal . . . . .	96

---

### 9.1. Resumen

Como resumen de la ejecución del proyecto se puede concluir que la utilización de una metodología ágil como Scrum no nos soluciona todos los problemas de la gestión de proyectos. Los problemas siguen existiendo, aunque este tipo de procesos proporcionan una serie de herramientas y técnicas que nos ayudan a gestionarlos con efectividad.

Otro aspecto fundamental a destacar es acerca de la documentación. Muchos programadores piensan que la adopción de una metodología ágil le permitirá desarrollar un sistema informático más rápido y sin necesidad de documentar el sistema. Esta afirmación es completamente falsa. A la hora de afrontar un proyecto utilizando una metodología ágil, bien es cierto que la documentación a realizar es menor, aunque no nos hemos de olvidar de realizar un análisis y un diseño a nuestro software.

La única diferencia que presenta *Scrum* con el desarrollo tradicional en cascada con respecto a la documentación del análisis y el diseño de software es el modo de realizarla. **El desarrollo tradicional en cascada** se ha de realizar una documentación exhaustiva del análisis del sistema completo y posteriormente con esta documentación realizar el diseño de software. En cambio, *Scrum* propone ir construyendo el software gradualmente (o por etapas), por lo tanto, lo mismo sucede con el análisis y el diseño de software, se van realizando a medida que es necesario analizar un requisito y posteriormente modificar el diseño actual añadiendo las nuevas características.

Para el correcto desarrollo del proyecto de software ha sido necesaria la formación e investigación en aspectos de Android y SocketIO (Node.js). Se puede llegar a la conclusión que antes de realizar una aplicación móvil o cualquier otro tipo de sistema informático todos y cada uno de los miembros del equipo deben de estar bien formados y ser conocedores de las tecnologías que se van a emplear durante todo este proceso.

Además de esto, el proyecto se ha desarrollado en un entorno de trabajo profesional y en un grupo de trabajo joven, multidisciplinar, muy motivado e innovador. Estos son los ingredientes necesarios para conseguir el éxito de un proyecto.

Y por supuesto, el progreso del proyecto ha sido documentado tal y como ha sucedido. En ningún momento se ha tratado de alterar la realidad o de estilizar alguna de las etapas o procesos obtenidos. De este modo se han reflejado los problemas que surgen como ocurre en todos los proyectos independientemente de la metodología utilizada.

## 9.2. Objetivos logrados

Tras finalizar el proyecto, los responsables de la aplicación han concluido que se han logrado los objetivos marcados. Se ha obtenido una primera versión funcional del sistema descrito y analizado a lo largo de todo el documento, proporcionando una experiencia de usuario agradable, ofreciendo una amplia variedad de posibilidades, fundamentándose en la participación activa de los jugadores para la construcción del contenido y con el aliciente de tratarse de una aplicación social donde los usuarios pueden competir con sus propios amigos.

## 9.3. Valoración personal

A nivel personal, este ha sido el primer proyecto a nivel profesional en el cual he trabajado. La colaboración activa en el desarrollo de este proyecto me ha servido para formarme como desarrollador de aplicaciones móviles, en especial **Android**, muy demandadas en la actualidad. Además de esto, me ha servido para asentar conocimientos a cerca de muchos otros aspectos como *web services*, iniciarme en el desarrollo con *Socket IO* y Node.js, diseño e implementación de bases de datos, y por supuesto conocer de primera mano las metodologías ágiles.

Personalmente, he podido experimentar la aplicación de una metodología ágil, la cual nos ha permitido ser mucho más flexibles al cambio, entregar software funcional desde casi el inicio del proyecto, gestionar el proyecto con mucha más flexibilidad, más independencia y auto organización del equipo de trabajo.

Me gustaría agradecer profundamente al equipo de desarrollo de *CuatroOchenta* por el acogimiento con el que me recibieron y me han tratado hasta el último momento.

## Anexo A

# Selección de API para Android

### A.1. El problema

Actualmente existen una gran cantidad de versiones instaladas en los dispositivos móviles Android. La correcta elección del API para desarrollar nuestra aplicación puede llegar a ser clave para que la compatibilidad con todos los dispositivos sea completa y conseguir el éxito de nuestra aplicación.

Existe un problema si desarrollamos con un **API inferior** al que utilizaran la mayoría de nuestros usuarios. El principal problema es no poder ofrecer a nuestros usuarios un gran abanico de novedades introducidas a partir de la versión (API) que estamos desarrollando. Al contrario sucede si utilizamos un **API superior**. En este caso, impedimos a nuestro cliente a utilizar la aplicación desarrollada.

Por lo tanto, la elección del API de desarrollo en las aplicaciones Android es tan importante como la elección del paradigma general de la arquitectura. Una mala elección podría hacernos perder una gran cantidad de usuarios.

### A.2. Valoración

Google nos presenta a través del *Informe de versiones Android* [2], cómo se encuentra la distribución de las diferentes versiones que componen Android entre los millones de dispositivos repartidos por todo el mundo. En este último informe de Marzo de 2015 cabe destacar el descenso de todas las versiones excepto **Lollipop** (API 19) y **KitKat** (API 21) que siguen creciendo tímidamente hasta alcanzar 5% y 41,4% respectivamente.

En la Figura A.1 se pueden observar los datos oficiales del informe de versiones de Android

publicado el día 6 de Marzo de 2015.

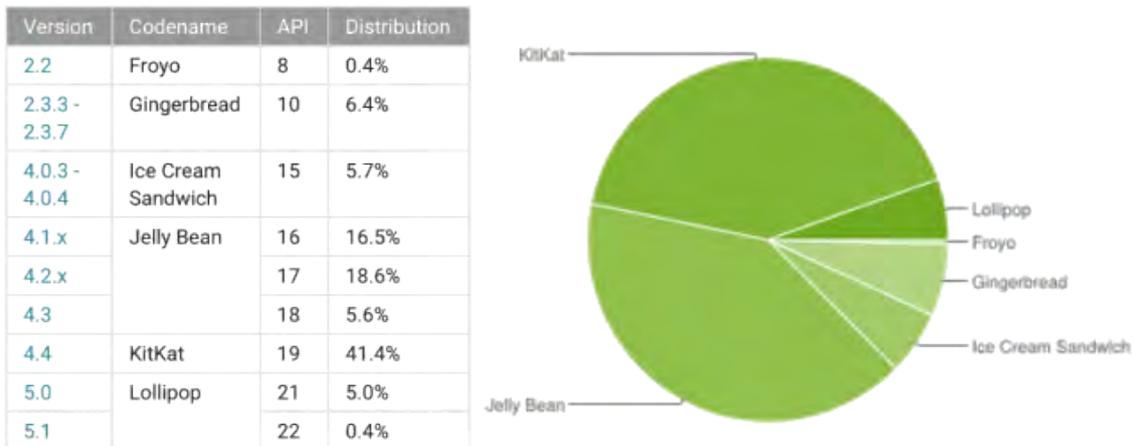


Figura A.1: Distribución de versiones Android.

En esta tabla podemos apreciar que aún existen dispositivos con versiones 2.2 *Froyo* y 2.3.3-2.3.7 *Gingerbread* aunque cada vez son inferiores. Estos dispositivos aparecieron en un momento en el que Android tuvo un crecimiento muy elevado, saliendo al mercado millones de modelos y por desgracia no todos acabaron actualizando a nuevas versiones, es por tanto un lastre que ha durado mucho tiempo y que se está solucionando progresivamente a medida que los usuarios van renovando sus terminales.

Es interesante destacar que por motivos de eficiencia los dispositivos no es recomendable actualizar su versión. Tras la actualización la gran cantidad de dispositivos se vuelve inestable e ineficiente produciendo errores que hacen el teléfono inservible.

### A.3. Decisión

Para el desarrollo de esta aplicación se ha decidido utilizar el **API 15** (versión 4.0.3), debido a que cantidad total de posibles usuarios es del 93,2% del total. Posiblemente, en los próximos meses esta cantidad de usuarios irá en aumento debido a la renovación de los terminales y las actualizaciones.

Por lo tanto, se ha decidido sacrificar un 6,8% de los usuarios a cambio de ofrecer a los usuarios una mayor calidad del producto.

## Anexo B

### *Scrum*

*Scrum* es un modelo de desarrollo ágil caracterizado por adoptar una estrategia de desarrollo incremental basada en la calidad del resultado a través de equipos multidisciplinares y autoorganizados solapando las diferentes fases del desarrollo.

La correcta aplicación de esta metodología proporciona:

- Desarrollo incremental y evolutivo
- Producto funcional temprano
- Revisión de software por interacciones (*Sprints*)

#### B.1. *Sprints*

Un desarrollo incremental consiste en planificar el proyecto en diferentes bloques temporales (en nuestro caso de 1 a 2 semanas) llamados iteraciones o *Sprints*. Por lo tanto, un esprint es:

Periodo de tiempo en el cual se crea un incremento de software.

Después de cada *Sprints* se obtiene un producto completamente funcional con valor. Gracias a esto podemos obtener *feedback* del proyecto y obtener el porcentaje de proyecto completado. En todas las iteraciones se repite un proceso de trabajo similar al realizado en un modelo de desarrollo tradicional.

## B.2. Requisitos funcionales

Una de las ventajas que nos aporta la utilización de las metodologías ágiles y en concreto *Scrum* es la administración y gestión de los requisitos funcionales del nuevo sistema.

En un sistema tradicional los requisitos se especifican y documentan al inicio del proyecto y su posterior modificación puede implicar una gran cantidad de cambios en el trabajo realizado hasta el momento.

*Scrum* en este caso realiza una gestión de los requisitos mucho más flexible que las metodologías tradicionales. En este caso, al inicio del proyecto se define una **pila del producto**.

La **pila del producto** es el conjunto de todos los requisitos de usuario que ha de cumplir el producto final. Este elemento consta de una lista que está en continua evolución y abierta a todos los implicados del producto como los desarrolladores, clientes, dueños del sistema, etc. Cabe destacar que la pila del producto se trata de una lista de requisitos de usuario priorizados en función del valor que aportan al cliente.

Una vez se ha especificado una pila del producto completa y priorizada se ha de planificar un *sprint*. En este caso se crea una **pila del *sprint*** para cada uno de ellos que contiene todas las tareas (descomposición de historias de usuario) que se realizan durante el *sprint*.

Al finalizar el *sprint* se obtiene un incremento del sistema que proporciona un valor añadido para el cliente. Es decir, el cliente obtendrá una pequeña parte del sistema en completo funcionamiento.

Finalmente tras concluir el *sprint* se actualiza la **pila del producto** y se eliminan de ella los requisitos implementados en el *sprint* realizado. Por lo tanto, siempre que los requisitos aún se encuentren en la pila del producto pueden ser modificados o eliminados. Además también se puede añadir más requisitos a la pila del producto sin grandes esfuerzos.

Gracias a este mecanismo *Scrum* nos proporciona una gran flexibilidad a la hora de gestionar los cambios en los requisitos.

## B.3. Reuniones

Durante el transcurso de un proyecto que aplique la metodología de desarrollo *Scrum* se realizan un conjunto de reuniones que se detallan a continuación:

- **Planificación del *sprint***: esta reunión se realiza al inicio del proyecto y en el inicio de cada *sprint*. En ella se toma como punto de partida la **pila del producto** y se obtiene la

**pila del *sprint***. A lo largo de la reunión se especifican todos los objetivos del *sprint*, se selecciona un subconjunto de requisitos de la **pila del producto** y se descomponen por tareas. A partir de estas tareas se realiza una estimación de todas las tareas para calcular la duración del *sprint*. Finalmente se obtienen todas las tareas y se forma la **pila del *sprint***.

- **Dayly *Scrum***: reunión diaria en la cual se hablan temas del día a día. Por ejemplo, los problemas del día anterior, reestimar las tareas realizadas. La principal función de esta tarea es actualizar la pila del *sprint*.
- **Revisión del *sprint***: esta reunión se realiza al terminar el *sprint*. Tiene como objetivo obtener *feedback* del cliente o del equipo de desarrollo. En ella se presentan los éxitos, el trabajo no completado durante el *sprint*, se presenta al cliente el incremento del sistema realizando demos de las nuevas características implementadas y se anuncia el nuevo *sprint*.



## Anexo C

# Gestión de datos

### C.1. Base de datos SQLite

Android incorpora la librería **SQLite** que nos permitirá utilizar bases de datos mediante el lenguaje SQL, de una forma sencilla y utilizando muy pocos recursos del sistema. SQLite es un **sistema de gestión de bases de datos relacional** multiplataforma. Las características más interesantes de esta herramienta son su pequeño tamaño, no necesitar servidor, poca configuración, transaccional y por supuesto ser de código libre.

Android incorpora todas las herramientas necesarias para la creación y gestión de bases de datos SQLite para llevar a cabo de manera sencilla todas las tareas necesarias.

### C.2. MDF (*Model Data Framework*)

MDF es un ORM (*Object-Relational Mapping*) construido por la empresa Soluciones CuatroOchenta con el fin de construir una base de datos orientada a objetos utilizando una base de datos relacional. Esta herramienta nos proporciona soporte para crear la base de datos física y las clases Java de forma automática a través de un fichero XML. Es decir, en nuestro proyecto definimos un fichero XML llamado *schema* como el siguiente:

```
<application>
  <database name="databaseQuizy">
    <table name="Question" sqlTableName="question">
      <column name="oid" sqlColumnName="id" type="PK" primaryKey="true"
        autoIncrement="true"/>
      <column name="title" type="VARCHAR" size="255" />
    </table>
  </database>
</application>
```

```

        <column name="correct_answer" type="VARCHAR" size="255" />
        <column name="answer2" type="VARCHAR" size="255" />
        <column name="answer3" type="VARCHAR" size="255" />
        <column name="answer4" type="VARCHAR" size="255" />
        <column name="user_id" type="PK" />
        <column name="question_state_id" type="PK" />
    </table>
</database>
</application>

```

Y gracias a esta anotación, el *framework* se encarga de generar un conjunto de clases Java y el esquema físico de la base de datos.

Dentro del conjunto de clases Java auto generadas por esta herramienta podemos encontrar:

- Clases del modelo o entidades.
- Clases de utilidad para persistir y obtener las entidades de la base de datos.
- Clase que representa la base de datos SQLite de Android.

Además de esto, MDF también nos genera el esquema físico de la base de datos SQLite, en este caso, nos ha generado el siguiente esquema:

```

CREATE TABLE app_info (
    id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
    num_game_questions integer,
    time_question integer,
    time_extra_question integer,
    tokens_time_extra integer,
    tokens_simply integer,
    tokens_pass integer,
);

```

```

CREATE TABLE level (
    id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
    name varchar,
    min_points integer,
    max_points integer,
    image varchar
);

```

```
CREATE TABLE game_result (  
  id integer PRIMARY KEY AUTOINCREMENT NOT NULL,  
  name varchar  
);
```

```
CREATE TABLE answer_type (  
  id integer PRIMARY KEY AUTOINCREMENT NOT NULL,  
  name varchar  
);
```

```
CREATE TABLE state (  
  id integer PRIMARY KEY AUTOINCREMENT NOT NULL,  
  name varchar  
);
```

```
CREATE TABLE user (  
  id integer PRIMARY KEY AUTOINCREMENT NOT NULL,  
  username varchar,  
  name varchar,  
  email varchar,  
  password varchar,  
  phone varchar,  
  facebook_id varchar,  
  last_name varchar,  
  icon varchar,  
  iconLocalPath varchar,  
  tokens integer  
);
```

```
CREATE TABLE question (  
  id integer PRIMARY KEY AUTOINCREMENT NOT NULL,  
  title varchar,  
  correct_answer varchar,  
  answer2 varchar,  
  answer3 varchar,  
  answer4 varchar,  
  user_id integer,  
  state_id integer,  
  FOREIGN KEY(user_id) REFERENCES user(id),  
  FOREIGN KEY(state_id) REFERENCES state(id)  
);
```

```
CREATE TABLE question_category (  
  id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
```

```

name varchar,
icon varchar,
icon_list varchar,
icon_detail varchar,
background_color varchar,
progress_color varchar,
question_category_parent_id integer,
user_creator_id integer,
FOREIGN KEY(question_category_parent_id) REFERENCES question_category(id),
FOREIGN KEY(user_creator_id) REFERENCES user(id)
);

```

```

CREATE TABLE question_question_category (
id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
question_id integer,
question_category_id integer,
FOREIGN KEY(question_id) REFERENCES question(id),
FOREIGN KEY(question_category_id) REFERENCES question_category(id)
);

```

```

CREATE TABLE question_review (
id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
date integer,
correct bool,
review text,
user_id integer,
question_id integer,
FOREIGN KEY(user_id) REFERENCES user(id),
FOREIGN KEY(question_id) REFERENCES question(id)
);

```

```

CREATE TABLE user_friend_request (
id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
request_date integer,
accepted_date integer,
cancellation_date integer,
is_ignore_request bool,
is_reject_request bool,
user_id integer,
friend_id integer,
FOREIGN KEY(user_id) REFERENCES user(id),
FOREIGN KEY(friend_id) REFERENCES user(id)
);

```

```

CREATE TABLE game (
  id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
  start_date integer,
  num_questions integer,
  is_accepted bool,
  user_creator_id integer,
  FOREIGN KEY(user_creator_id) REFERENCES user(id)
);

CREATE TABLE game_category (
  id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
  game_id integer,
  question_category_id integer,
  FOREIGN KEY(game_id) REFERENCES game(id),
  FOREIGN KEY(game_category_id) REFERENCES game_category(id)
);

CREATE TABLE game_user (
  id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
  game_id integer,
  user_id integer,
  result_id integer,
  response_date integer,
  FOREIGN KEY(game_id) REFERENCES game(id),
  FOREIGN KEY(user_id) REFERENCES user(id),
  FOREIGN KEY(result_id) REFERENCES game_result(id)
);

CREATE TABLE game_question (
  id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
  game_id integer,
  question_id integer,
  FOREIGN KEY(game_id) REFERENCES game(id),
  FOREIGN KEY(question_id) REFERENCES question(id)
);

CREATE TABLE game_question_user_answer (
  id integer PRIMARY KEY AUTOINCREMENT NOT NULL,
  datetime integer,
  response_time integer,
  game_user_id integer,
  game_question_id integer,
  answer_type_id integer,
  FOREIGN KEY(game_user_id) REFERENCES game_user(id),

```

```
FOREIGN KEY(game_question_id) REFERENCES game_question(id),  
FOREIGN KEY(answer_type_id) REFERENCES answer_type(id)  
);
```

```
CREATE TABLE user_point (  
  id integer PRIMARY KEY AUTOINCREMENT NOT NULL,  
  num_points integer,  
  user_id integer,  
  question_created_id integer,  
  game_id integer,  
  FOREIGN KEY(user_id) REFERENCES user(id),  
  FOREIGN KEY(question_created_id) REFERENCES question(id),  
  FOREIGN KEY(game_id) REFERENCES game(id)  
);
```

# Bibliografía

- [1] *Librería Socket.IO.*
- [2] Android Libre. *Informe versiones Android en Marzo 2015.*
- [3] M Cohn. *User stories applied: For agile software development.* Addison-Wesley, 2004.
- [4] Fundación telefónica. *Informe de la Sociedad de la Información en España 2014.* Informe del uso de smartphones e internet en España durante el 2014.