MANUAL DE COORDINACIÓN

Redacción:

José Rubén Céspedes Heredia

Revisión:

Diego Alfonso Candelaria Rodríguez

Versión: 2

14 de noviembre de 2018

VERSIONES

Versión	Fecha de modificación	Cambios
1	17-10-2018	Versión inicial del proyecto
2	07-11-2018	➔ <Pág 3> En el apartado "*Ciclo de vida*" se ha modificado la especificación de la entrega al cliente en la primera iteración ➔ <Pág 3> En el apartado "*Recursos software de desarrollo*": ◆ se ha modificado la plataforma de la aplicación (de web a dispositivos *Android*), el framework y el *IDE* de desarrollo. ➔ <Pág 4> En el apartado "Herramientas para las comunicaciones con el equipo de trabajo" se ha añadido el correo electrónico como una nueva herramienta de comunicación

Índice

Ciclo de vida

Como ciclo de vida del software se ha optado por un modelo de proceso incremental, donde en la primera iteración se entregará al cliente la documentación inicial del proyecto; en la segunda iteración se entregará un producto funcional que cubra los requisitos más relevantes (se dejarán a un lado funcionalidades relacionadas con los comentarios, valoraciones y sugerencias); y en la tercera iteración se mejorará el producto intentado cubrir todos los requisitos posibles (ordenados por prioridad).

Metodología de desarrollo

Debido a la variabilidad y la alta probabilidad de cambios en estos tipos de proyectos se va a seguir una metodología de desarrollo ágil, en concreto, la metodología **SCRUM**.

Por lo tanto, se van a realizar reuniones semanales, donde se revisará el trabajo realizado por el equipo y se especificará lo siguiente:

- ¿Qué se ha hecho desde la última reunión de equipo?
- ¿Qué obstáculos se han encontrado?
- ¿Qué se planea hacer mientras llega la siguiente reunión de equipo?

Aunque no se realizarán reuniones diarias, se mantendrá una comunicación abierta mediante un grupo de Telegram, donde cualquier miembro del equipo pueda expresar en cualquier momento los obstáculos o dificultades que pueda tener, así como los avances que lleve hasta el momento.

Recursos software de desarrollo

Para la realización de nuestro proyecto hemos optado por el desarrollo de una app para dispositivos Android. Por ello, vamos a hacer uso del software conocido como **Android Studio**, ya que es una herramienta que nos va a permitir desarrollar una aplicación para cualquier dispositivo Android sea cual sea su versión de Android.

Por otro lado, nos proporciona una IDE de desarrollo basado en IntelliJ IDEA, un editor visual del *layout* de nuestra app, un analizador para nuestra APK (que nos ayudará a reducir el tamaño de la misma), un emulador virtual de dispositivos (donde podremos comprobar el funcionamiento de nuestra app sobre un dispositivo virtual) y un sistema de compilación de Gradle.

Además del framework de desarrollo Android Studio, haremos uso de algunas APIs como soporte de nuestro sistema, como puede ser la API de *Google Maps*.

Organización del equipo de trabajo

A cada miembro del equipo se le ha asignado un rol de entre los siguientes:

- **Gestor de proyecto**. Encargado de supervisar y controlar el trabajo de cada miembro del equipo.
- **Auditor de accesibilidad**. Encargado de adquirir y transmitir conocimientos acerca de usabilidad y accesibilidad. Además, deberá supervisar que se cumplen con los requisitos de usabilidad y accesibilidad exigidos por el cliente.
- **Diseñador**. Encargado de realizar la documentación necesaria para la gestión del proyecto así como de esbozar el propio diseño del producto software que se pretende desarrollar.
- **Programador**. Encargado de desarrollar el producto software.

Cabe mencionar que los roles asignados a cada componente se irán rotando (exceptuando el *Auditor de accesibilidad*) en cada iteración. Además, el número de componentes por cada rol también variará dependiendo del estado del proyecto y de la iteración en la que nos encontremos. De este modo, en la primera iteración tenemos una auditora de accesibilidad, dos gestores de proyecto, cuatro diseñadores y un programador, pero en iteraciones posteriores podrá variar.

Herramientas para las comunicaciones con el equipo de trabajo

Mayormente usaremos dos herramientas:

- **Telegram**. Para la comunicación diaria y directa entre todos los miembros del grupo.
- **Trello**. Para la comunicación indirecta y la organización y gestión de las tareas.

Además de las herramientas anteriores, se mantendrá una comunicación vía correo electrónico donde se acordarán plazos intermedios de entrega así como reuniones para revisar el trabajo elaborado hasta el momento. También servirá como método para el intercambio de documentación.

Relaciones con el cliente

Inicialmente, se ha realizado una **entrevista** con el cliente a partir de la cual se han extraído los requisitos iniciales del proyecto. Posteriormente, se realizarán **reuniones semanales**

donde se mostrará el estado actual del proyecto. Además, se realizarán **revisiones** del trabajo desarrollado cada cierto tiempo (especificado en el documento de planificación temporal del proyecto) y se mantendrá una comunicación activa mediante **correo electrónico**.

Por último, el cliente tendrá acceso en todo momento a la documentación desarrollada mediante acceso a una **carpeta compartida** en *Google Drive*.

Estándares de código

Para que el código desarrollado sea aceptado, debe cumplir las siguientes **reglas**:

- Todos los nombres de variable deben ser representativos.
- Las variables constantes o de ámbito global se deben escribir en mayúsculas.
- Las variables, funciones y clases con nombres compuestos se escribirán siguiendo la tipografía CamelCase.
- Los corchetes y llaves iniciales se escribirán en la misma línea que la estructura que encierra.
- Todas las funciones deberán estar comentadas con la funcionalidad que se pretende desarrollar. Del mismo modo, se comentarán todos los trozos de código que sean difíciles de entender.

Todos los documentos tendrán la siguiente estructura:

- **Portada**. La portada tendrá los siguientes elementos con los siguientes estilos:
    - **Logo** de la empresa.
    - **Título** del documento. Fuente: Oswald; Tamaño: 52; Color: personalizado (RGB 66 66 66); Centrado e interlineado sencillo.
    - **Subtítulo**. Fuente: Económica; Tamaño: 24; Color: Gris oscuro 3; Alineado a la izquierda; Interlineado: 1.3
    - **Fechas**. Fuente: Oswald; Tamaño: 14; Color: Gris oscuro 3; Centrado; Interlineado: 1.15
- **Índice**. El índice tendrá los siguientes elementos y estilos:
    - **Título**. Fuente: Roboto Slab; Tamaño: 30; Color: Negro; Centrado e interlineado doble.
    - **Componentes** del índice. Fuente: Roboto Slab; Tamaño: 14; Negrita; Color:Negro; Alineado a la izquierda e interlineado doble.

- **Resto del documento**. Serán apartados con la siguiente estructura y estilos:
    - **Título 1**. Fuente: Roboto Slab; Tamaño: 18; Negrita; Color: Negro; Justificado; Interlineado: 1.15
    - **Texto normal**. Fuente: Roboto Slab; Tamaño: 11; Color: Negro; Justificado; Interlineado: 1.15

Control de versiones

Para el control de versiones se utilizará la herramienta **GitHub**, tanto su aplicación web como las distintas herramientas asociadas (*Git*, *GitKraken*, etc…).

La metodología a seguir es que tendremos una rama principal, donde se encontrará la versión más reciente del proyecto. Además, cada miembro del equipo tendrá una rama secundaria donde realizará las tareas de desarrollo que se le hayan asignado. Además de controlar las versiones del código desarrollado también se realizará un control de versiones de la documentación generada a lo largo del proyecto, ya que esta se irá completando con cada iteración.

Sólo los gestores de proyecto podrán actualizar la rama principal.

Gestión de calidad

Para asegurar que nuestro proyecto cumple con unos estándares mínimos de calidad, seguiremos el siguiente proceso de desarrollo:

- En primer lugar, se realizará una reunión donde seleccionaremos los requisitos y funcionalidades que vamos a cubrir en la iteración que corresponda.
- Seguidamente, los diseñadores pasarán a diseñar y modelar todos los componentes software necesarios para alcanzar nuestros objetivos.
- Mientras tanto, los programadores buscarán documentación, videos y tutoriales acerca de las herramientas y lenguajes que vayamos a usar en la implementación.
- Una vez realizado el diseño, los gestores, junto con la ayuda de algún componente del grupo que tenga mayores conocimientos en la materia revisarán y verificarán que el diseño es correcto y no falta ningún detalle.
- A continuación, los programadores ya podrán repartirse las tareas de desarrollo y comenzar con las implementaciones. Cada programador dispondrá de su propia rama donde desarrollará la funcionalidad que se le haya asignado. Una vez terminada y probada por el propio programador, los gestores comprobarán de nuevo su correcto funcionamiento y actualizarán la rama principal.

- Una vez terminada la iteración y obtenido un producto funcional se procederá a la prueba junto con el cliente.

Para la etapa de diseño usaremos el lenguaje de modelado UML (diagramas de clase, de secuencia, de comunicación, etc…). Además, haremos uso de todas las herramientas que creamos oportunas y nos ayuden a realizar de manera digital los diagramas anteriormente mencionados (como puede ser el caso de *draw.io*).

Para la gestión de calidad de la accesibilidad se va a intentar seguir las pautas de accesibilidad de la WCAG (*Web Content Accessibility Guidelines*) con el objetivo de conseguir una aplicación web lo más accesible posible. Además, se usarán diversas herramientas integradas en el navegador que nos informarán de puntos clave de nuestro sitio web a tener en cuenta para mejorar la accesibilidad. Algunas de estas herramientas son:

- ***Complex Table Inspector***
- ***Favelets For The Validator***
- ***NCAM Accessibility QA Favelet***

Además de todo lo anterior, los gestores de proyecto supervisarán que cada tarea asignada se ha realizado correctamente, así como cerciorarse de que se ha realizado en el tiempo establecido para ello. En caso contrario, deberán tomar las medidas adecuadas con el compañero encargado de dicha tarea.