

# Plan de gestión, análisis, diseño y memorial del proyecto KeyPaX

Bárbaros Software S.A. (Grupo 3)



Carlos Bellvis, 755452  
Jorge Bernal, 775695  
Jorge Borque, 777959  
Arturo Calvera, 776303  
Andoni Salcedo, 785649  
Javier Vela, 775593

Organización GitHub: <https://github.com/UNIZAR-30226-2021-03>

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Organización del proyecto</b>	<b>2</b>
<b>3. Plan de gestión del proyecto</b>	<b>2</b>
3.1. Procesos . . . . .	2
3.1.1. Procesos de inicio . . . . .	2
3.1.2. Procesos de ejecución y control . . . . .	2
3.1.3. Procesos técnicos . . . . .	2
3.2. Planes . . . . .	3
3.2.1. Plan de gestión de configuraciones . . . . .	3
3.2.2. Plan de construcción y despliegue del software . . . . .	3
3.2.3. Plan de aseguramiento de calidad . . . . .	4
3.2.4. Calendario del proyecto y división del trabajo . . . . .	4
<b>4. Análisis y diseño del sistema</b>	<b>4</b>
4.1. Análisis de requisitos . . . . .	4
4.2. Diseño del sistema . . . . .	4
<b>5. Memoria del proyecto</b>	<b>4</b>
5.1. Inicio del proyecto . . . . .	4
5.2. Ejecución y control del proyecto . . . . .	4
5.3. Cierre del proyecto . . . . .	4
<b>6. Conclusiones</b>	<b>4</b>
<b>Glosario</b>	<b>4</b>
<b>Anexo I. Actas de todas las reuniones realizadas</b>	<b>4</b>
<b>7. Conclusiones</b>	<b>4</b>
<b>8. Bibliografía</b>	<b>4</b>

## 1. Introducción

## 2. Organización del proyecto

El equipo del proyecto está formado por los 6 integrantes del grupo. Para dividir el trabajo y las responsabilidades se han formado 2 grupos iniciales, acorde con las capacidades y conocimientos individuales. Además, se ha designado como **director del proyecto** a **Arturo Calvera**, cuya responsabilidad abarca la gestión de los equipos de trabajo y sus responsabilidades.

### ■ Equipo *Backend*:

- Responsable: Arturo Calvera.
- Función: Desarrollo del *backend* del sistema.
- Integrantes: Arturo Calvera, Andoni Salcedo.

### ■ Equipo *Frontend*:

- Responsable: Javier Vela.
- Sub-equipos: El equipo dedicado a la vista del sistema se divide para el desarrollo de las dos interfaces.
  - **Equipo Web**:
    - ◊ Función: Desarrollo del *frontend* web del sistema.
    - ◊ Integrantes: Jorge Bernal, Javier Vela.
  - **Equipo Android**:
    - ◊ Función: Desarrollo de la aplicación cliente del sistema para dispositivos Android.
    - ◊ Integrantes: Carlos Bellvis, Jorge Borque.

- Se contempla la posibilidad de crear nuevos equipos para las necesidades que surjan durante el proyecto (*e.g.* despliegue, pruebas) o la reconfiguración de los actuales para adecuarse a la situación.

## 3. Plan de gestión del proyecto

### 3.1. Procesos

#### 3.1.1. Procesos de inicio

#### 3.1.2. Procesos de ejecución y control

#### 3.1.3. Procesos técnicos

Para implementar las vistas del sistema se utilizará por una parte Java (Android SDK) para construir el frontend para móviles Android y por otra JavaScript (React.JS) que corresponderá con los distintos navegadores web. En cuanto al backend, se desarrollará utilizando Node.JS y la framework Express. La base de datos utilizará el SGBD MongoDB. El despliegue del sistema se realizará en un entorno de contenedores Docker. Finalmente se ejecutarán unos scripts que llevarán a cabo las pruebas necesarias de las distintas funcionalidades para la comprobación del funcionamiento correcto.

## 3.2. Planes

### 3.2.1. Plan de gestión de configuraciones

### 3.2.2. Plan de construcción y despliegue del software

El sistema se fundamenta en el desarrollo de cuatro subsistemas que trabajan aislados de los otros, siendo su comunicación entre los módulos expuesta a través de interfaces que los relacionan, de este modo tanto las pruebas, compilación y dependencias son independientes al resto de subsistemas.

Dos de los cuatro subsistemas, el frontend web y el backend, estarán empujados en contenedores docker y desplegados en la nube utilizando los servicios de *Amazon Web Services*, se utilizarán scripts de *GitHub actions* para automatizar la compilación y testing y despliegue en AWS de los mismos, utilizando en el caso de entorno web librerías de testing como Jest y Postman en el caso de la *API REST*.

Para el subsistema que concierne a la capa de datos, se despliega en un cluster de MongoDB situado en Bélgica siendo este proporcionado por el equipo de *MongoDB Atlas*, donde se van a desarrollar una serie de diversos triggers que lleven un control de la consistencia de datos tanto tras el uso operaciones además se llevará un control periodico para evitar la replicación y la detección de anomalías a través del uso de scripts.

La interfaz móvil llevará por su cuenta la compilación y testing para cada versión funcional de la aplicación, se integrarán test de unidad, funcionalidad y sobrecarga del sistema.

El punto fuerte de utilizar el despliegue basado en contenedores es que el control de dependencias es indiferente al sistema operativo y al entorno de desarrollo de los programadores, de esta forma cada integrante del equipo podrá configurar y personalizar su entorno por cuenta propia. En el fronted móvil al ser independiente se fija una serie de requisitos para el equipo que trabaja en esta parte, se usará la versión Android 6.1 (*Marshmallow*) y se usará Java como lenguaje de programación, el control de dependencias es llevado por el propio Gradle de Android.

3.2.3. Plan de aseguramiento de calidad

3.2.4. Calendario del proyecto y división del trabajo

## 4. Análisis y diseño del sistema

4.1. Análisis de requisitos

4.2. Diseño del sistema

## 5. Memoria del proyecto

5.1. Inicio del proyecto

5.2. Ejecución y control del proyecto

5.3. Cierre del proyecto

## 6. Conclusiones

## Glosario

## Anexo I. Actas de todas las reuniones realizadas

## 7. Conclusiones

## 8. Bibliografía