

Propuesta de Diseño de Aplicación Funcional  
Registro y Mantenimiento de Extintores  
Team Brije

Shika Moriyama      Amaury Flores      Carol Torres  
Wilbert Arcila

27 de noviembre de 2023

# Índice general

<b>1. Propuesta uno</b>	<b>3</b>
1.1. Un Pequeño Resumen . . . . .	3
1.2. Requerimientos Funcionales . . . . .	3
1.3. Requerimientos No Funcionales . . . . .	3
1.4. Casos . . . . .	3
<b>2. Segunda Propuesta</b>	<b>5</b>
2.1. Una segunda opción . . . . .	5
2.2. Requerimientos funcionales . . . . .	5
2.3. Requerimientos no funcionales . . . . .	5
2.4. Casos . . . . .	6
<b>3. Diagramas</b>	<b>7</b>
3.1. Diagrama de componentes . . . . .	7
3.2. Diagrama de Entidad Relacion . . . . .	8
3.3. Diagrama de clases . . . . .	8
3.4. Diagrama de secuencia . . . . .	9
3.5. Diagramas de flujo . . . . .	10
3.6. Otro Diagrama de Clases . . . . .	13
<b>4. Links</b>	<b>14</b>

# Capítulo 1

## Propuesta uno

### 1.1. Un Pequeño Resumen

De forma breve, la aplicación que se piensa realizar es un generador y lector de códigos QR el cual almacene información de Extintores como su ubicación, fecha de mantenimiento y seguridad, entre otros. Para esto, se buscará crear una app que tenga tres principales elementos.

### 1.2. Requerimientos Funcionales

Estos son los requerimientos funcionales que deben de ser abarcados:

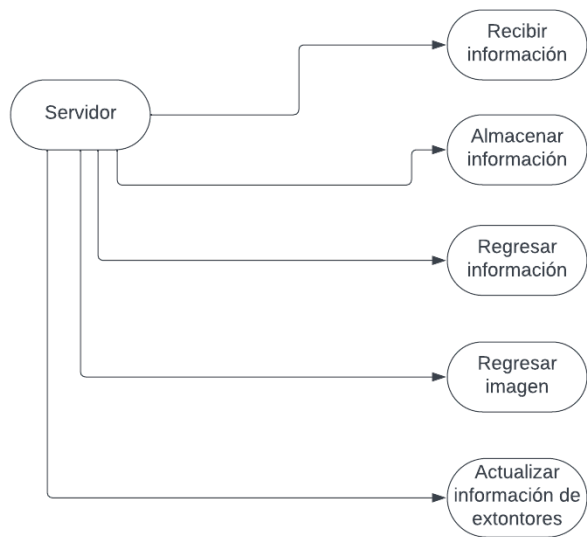
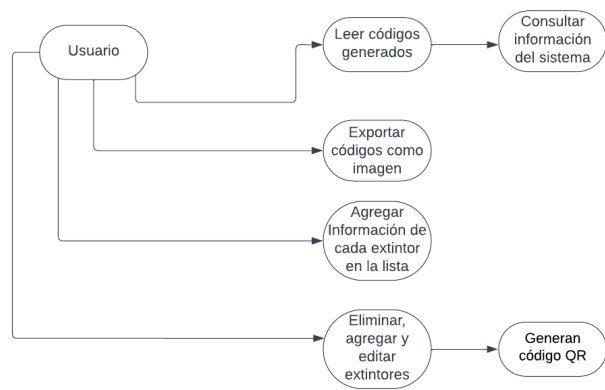
- Poder borrar TODOS los extintores de golpe.
- Poder leer estos códigos generados.
- Poder agregar, editar y eliminar extintores de una lista
- Poder agregar información a cada extintor en la lista

### 1.3. Requerimientos No Funcionales

Estos son los requerimientos no funcionales que deben de ser abarcados:

- Uso y gestión de base de datos
- Transformar IDs de diferentes elementos de la base de datos a códigos.
- Generar listas dinámicas según lo necesitado por el usuario
- Generación iterativa de cada elemento para la optimización de elementos.
- Optimización de archivos para consumir la menor cantidad de espacio en el almacenamiento del sistema.
- Creado en Android Studio

### 1.4. Casos



## Capítulo 2

# Segunda Propuesta

### 2.1. Una segunda opción

Como le llegamos a comentar en clase, también teníamos otra idea, la cual también se intentará efectuar, pero lo tomaremos como un reto extra, enfocándonos en el administrador de extintores primero. La segunda opción sería una calculadora de fusiones de el juego Persona 5, la cual estaría estilizada en la estética del juego mismo.

### 2.2. Requerimientos funcionales

Estos son los requerimientos funcionales que deben de ser abarcados:

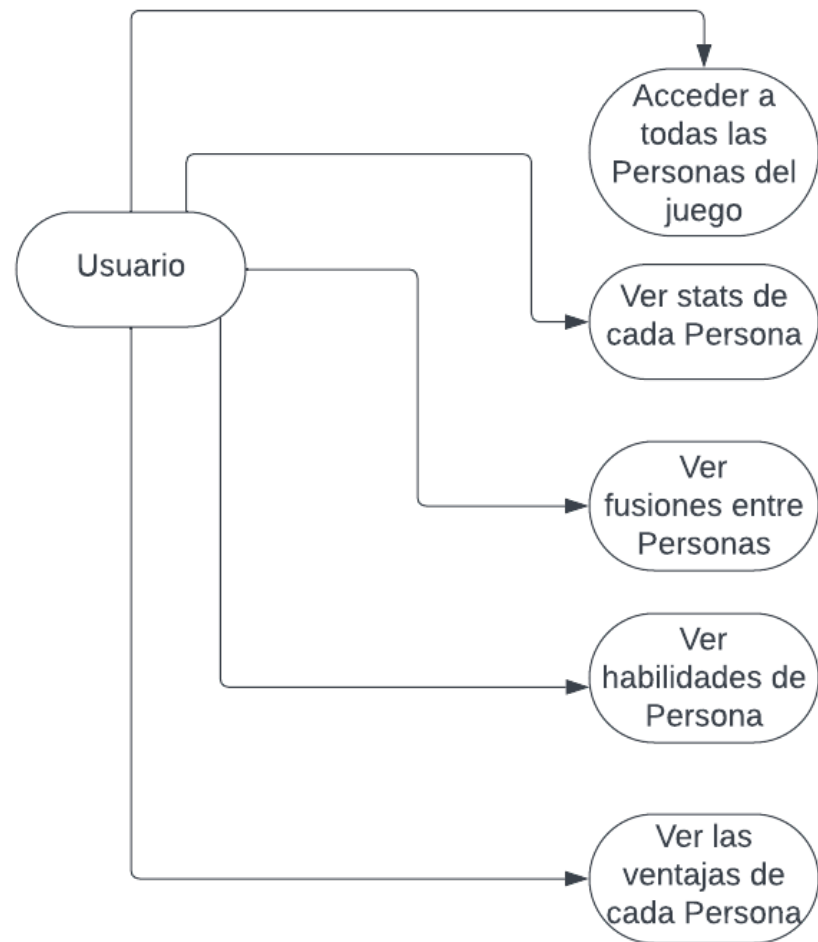
- Accesar a unas de las Personas del juego
- Meterse a cada Persona individual, poder ver sus Stats
- Ver las habilidades de cada Persona, y sus ventajas de tipo
- Igor Bailando Cumbiab

### 2.3. Requerimientos no funcionales

Estos son los requerimientos no funcionales que deben de ser abarcados:

- Será creado en UNITY
- Accesar a una base estatica de datos
- Iterar sobre una misma página

## 2.4. Casos

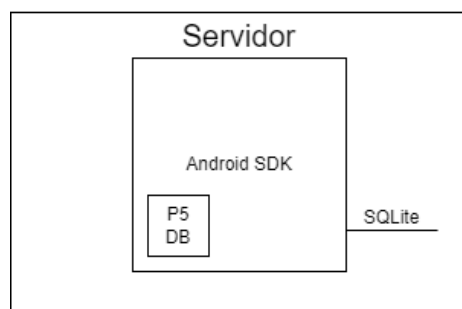
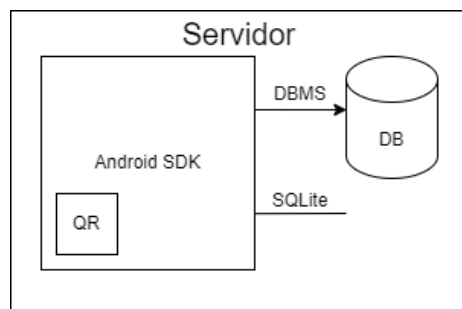


## Capítulo 3

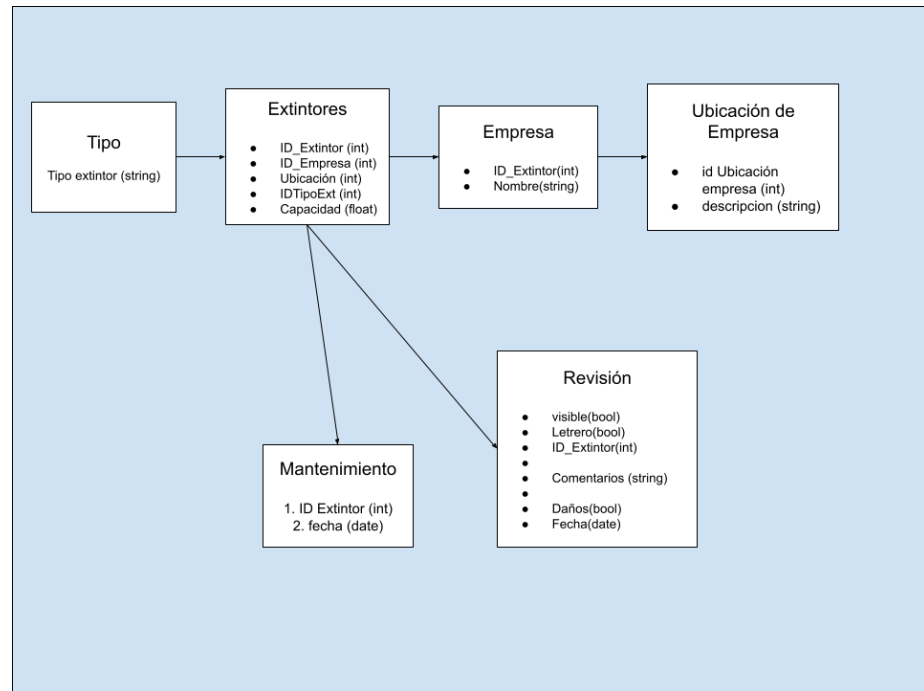
# Diagramas

Aqui vamos a poner los diagramas que hemos hecho para ambos juegos

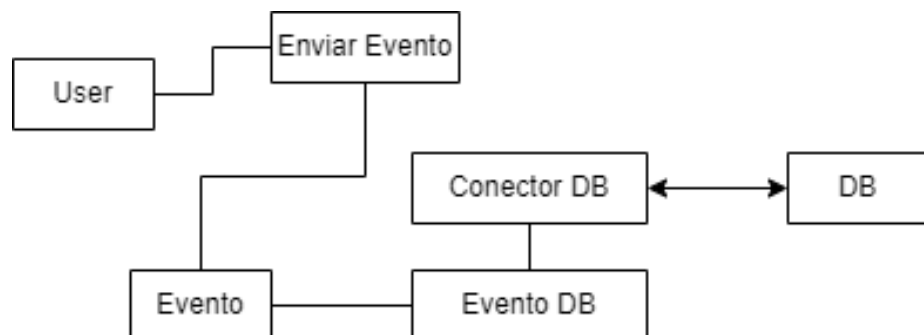
### 3.1. Diagrama de componentes



### 3.2. Diagrama de Entidad Relacion

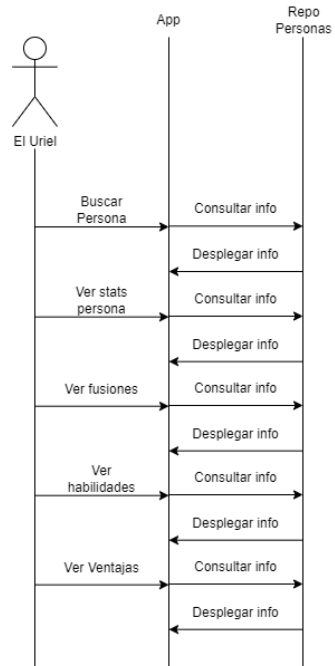
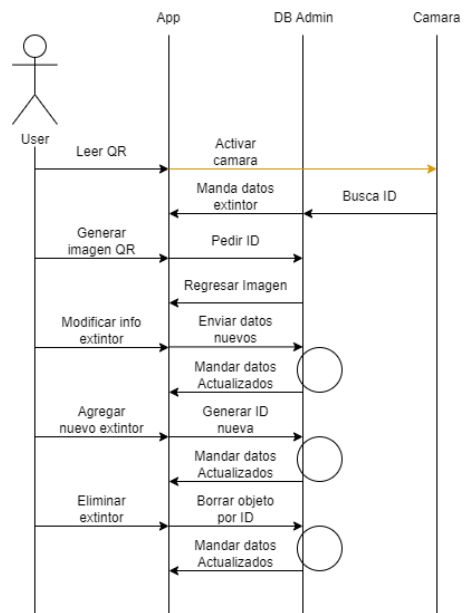


### 3.3. Diagrama de clases

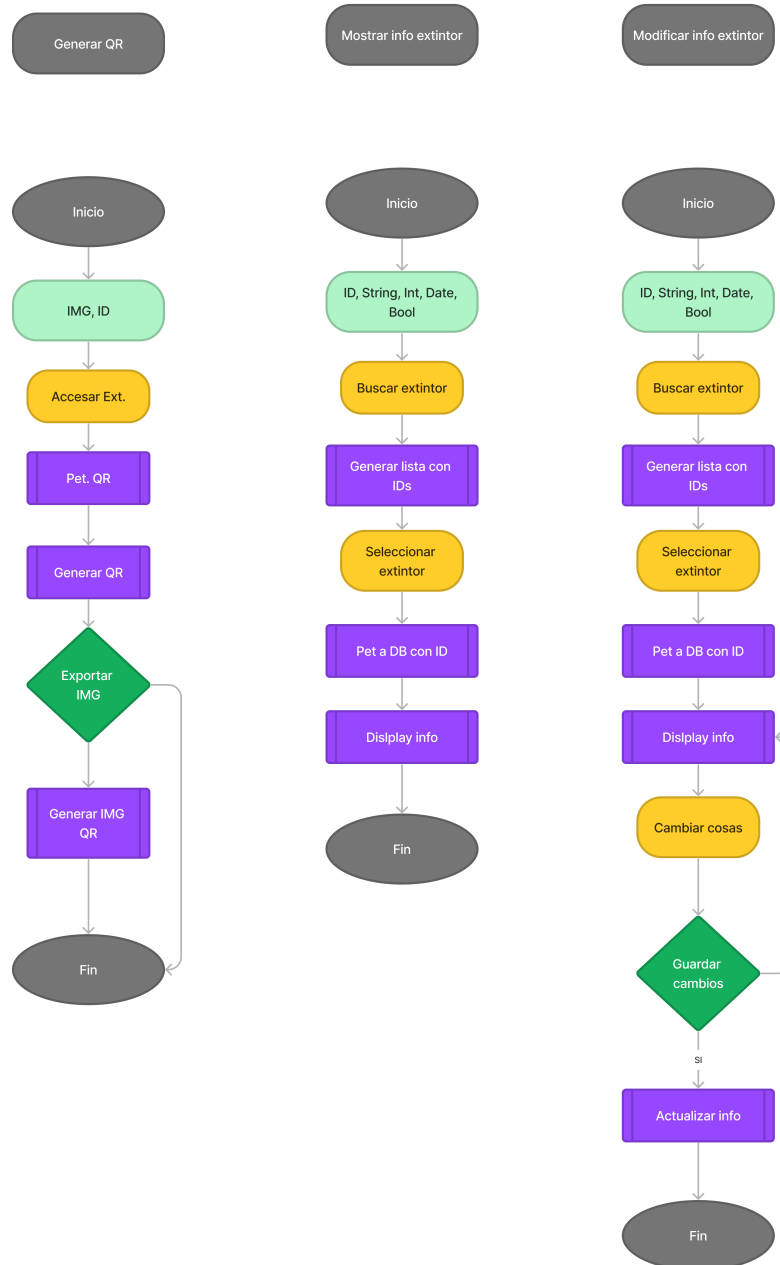


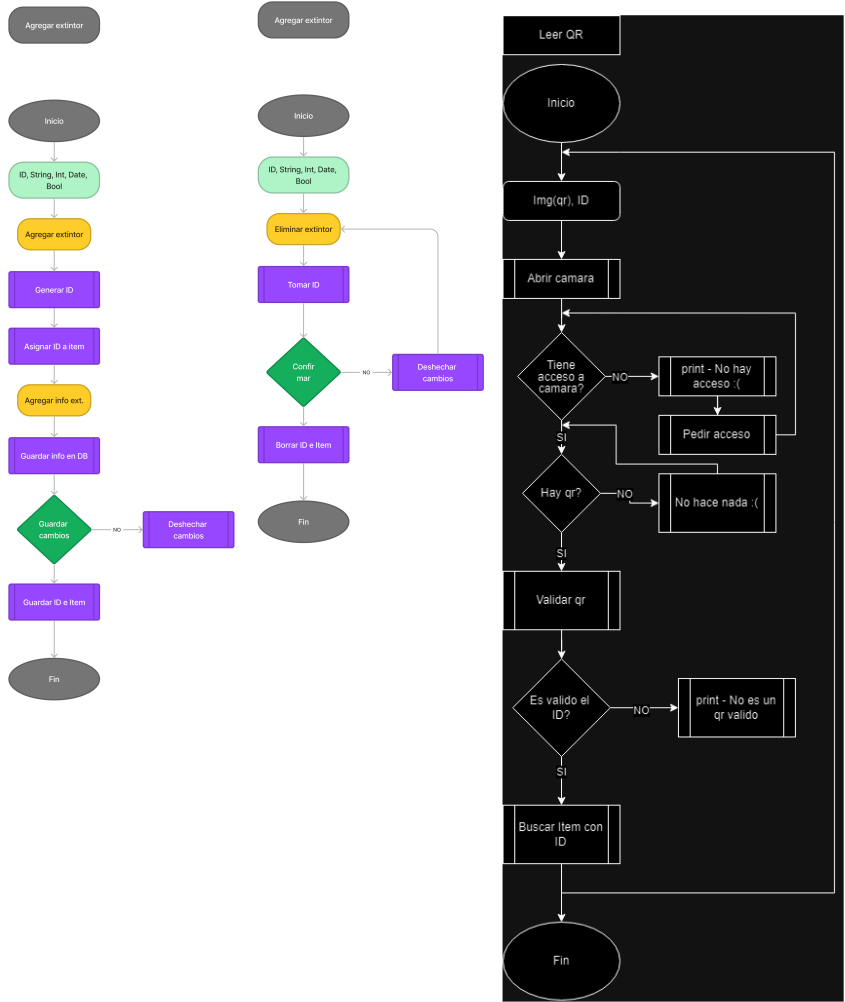


### 3.4. Diagrama de secuencia



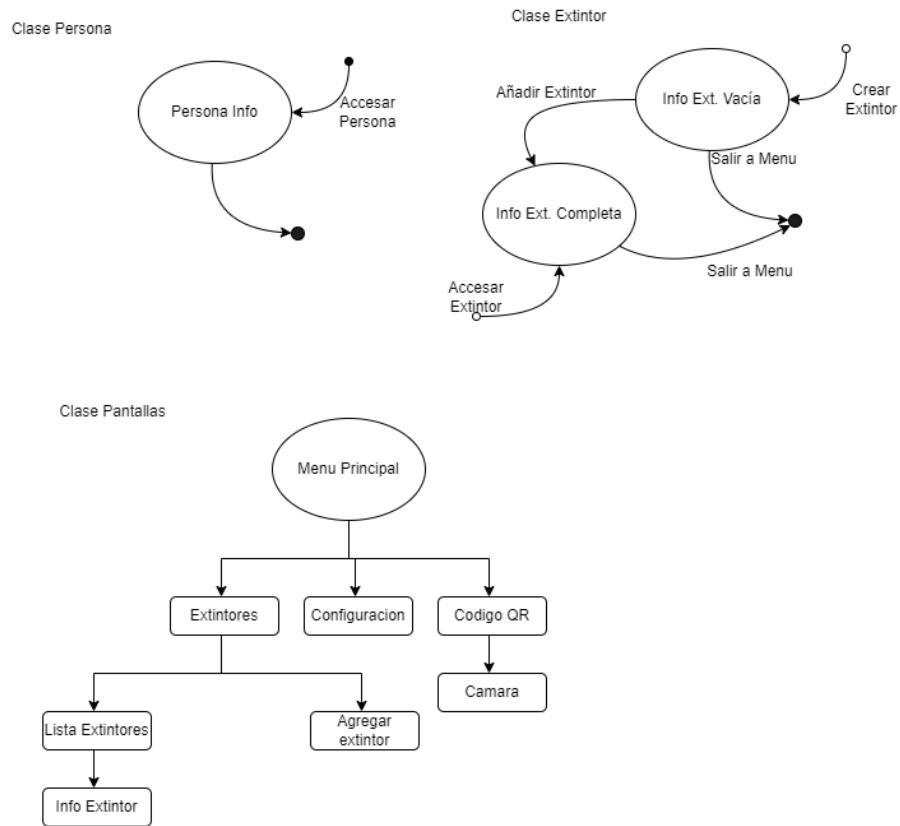
### 3.5. Diagramas de flujo







### 3.6. Otro Diagrama de Clases



## Capítulo 4

### Links

[Link a prototipado Figma Extintores](#)

[Link a prototipado Figma Persona 5](#)