

# TP Integrador

## Soporte de Gestión de Datos para la Programación Visual

### Integrantes:

Alés, Nicolás (L. 44916)

Nicola, Francisco (L. 45090)

### Resumen del Documento

En el presente documento detallaremos cómo desarrollaremos una aplicación móvil utilizando Python. El objetivo es crear una plataforma para unir jugadores que se encuentren cerca, para que puedan comunicarse y organizar juegos de rol o de mesa de manera sencilla.

## Narrativa

Nuestro proyecto para la cátedra consiste en crear una plataforma online en el que jugadores, mediante una aplicación de celular, puedan comunicarse con otros jugadores cercanos de la misma localidad y organizar juegos de rol o de mesa.

Inicialmente los jugadores deben registrarse, detallando sus datos como nombre, edad, localidad, juegos favoritos, etc. Una vez dados de alta podrán realizar búsquedas en el sistema sobre gente en la misma región interesada en jugar. También podrán crear salas virtuales donde pueden comunicarse entre sí y organizar una reunión.

Cada jugador tendrá una reputación que indica su actitud personal en los juegos, ya sea si es amistoso, competitivo, etc.

## Especificaciones Técnicas

Para el desarrollo de la aplicación utilizaremos como tecnología principal Python 3.7.9. También haremos uso de los siguientes módulos y tecnologías:

- Kivy: Permite crear interfaces gráficas de usuario multiplataforma (Linux, Windows, OS X, Android, iOS, Raspberry Pi). Incluye un lenguaje de marcado, conocido como KvLang, que simplifica la creación de árboles de widgets de una manera declarativa.
- SQLAlchemy: Es un ORM que proporciona un conjunto completo de patrones de persistencia de nivel empresarial bien conocidos, diseñados para un acceso eficiente y de alto rendimiento a la base de datos.
- SQLite: Es un motor de bases de datos relacional ligero compatible con ACID.

- Passlib: Es una biblioteca de hash de contraseñas que proporciona implementaciones multiplataforma de más de 30 algoritmos de hash de contraseñas, así como un marco para administrar hashes de contraseñas existentes.

## Casos de Uso

### CU 001 - Registrar usuario nuevo

Descripción: Usuario se registra en el sistema.

Actores: Usuario, Sistema.

Flujo normal:

1. Usuario ingresa datos requeridos. Estos incluyen nombre y email.
2. Sistema valida información.
3. Sistema da usuario de alta.

Flujo alternativo:

- 2.a. Nombre y/o email ya se encuentran registrados.
  - 2.a.1. Usuario re ingresa datos distintos.

Postcondiciones: El usuario se registró exitosamente.

### CU 002 - Crear lobby

Descripción: Usuario crea un lobby personal.

Actores: Usuario, Sistema.

Precondiciones: El usuario se ha logueado.

Flujo normal:

1. Usuario ingresa nombre del nuevo lobby, descripción y cantidad jugadores máximos.
2. Sistema da lobby de alta.
3. Sistema genera código identificador del lobby y lo muestra al usuario.

Postcondiciones: El lobby se creó exitosamente.

## CU 003 - Unirse a lobby

Descripción: Usuario se une a un lobby.

Actores: Usuario, Sistema.

Precondiciones: El usuario se ha logueado.

Flujo normal:

1. Usuario accede al buscador de lobbies.
2. Usuario busca por nombre, descripción o etiqueta el lobby.
3. Sistema muestra información de los lobbys encontrados.
4. Usuario solicita unirse.

Flujo alternativo:

- 3.a. Sistema no encontró ningún lobby que se corresponda con datos ingresados.
  - 3.a.1. Usuario busca por otras palabras clave.

Postcondiciones: El usuario envió la solicitud de unión al lobby exitosamente.

## CU 004 - Administrar lobby

Descripción: Usuario actualiza información del lobby.

Actores: Usuario, Sistema.

Precondiciones: El usuario se ha logueado. El lobby ya ha sido creado.

Flujo normal:

1. Usuario accede a configuración del lobby.
2. Usuario edita los parámetros deseados.
3. Usuario confirma cambios.
4. Sistema actualiza cambios.

Postcondiciones: Los cambios del lobby se actualizaron exitosamente.

## CU 005 - Editar perfil de usuario

Descripción: Usuario actualiza información de su perfil.

Actores: Usuario, Sistema.

Precondiciones: El usuario se ha logueado.

Flujo normal:

1. Usuario accede a configuración de perfil.
2. Usuario edita los parámetros deseados.
3. Usuario confirma cambios.
4. Sistema actualiza cambios.

Postcondiciones: Los cambios del perfil se actualizaron exitosamente.

## CU 006 - Administrar miembros del lobby

Descripción: Usuario actualiza miembros del lobby y sus roles.

Actores: Usuario, Sistema.

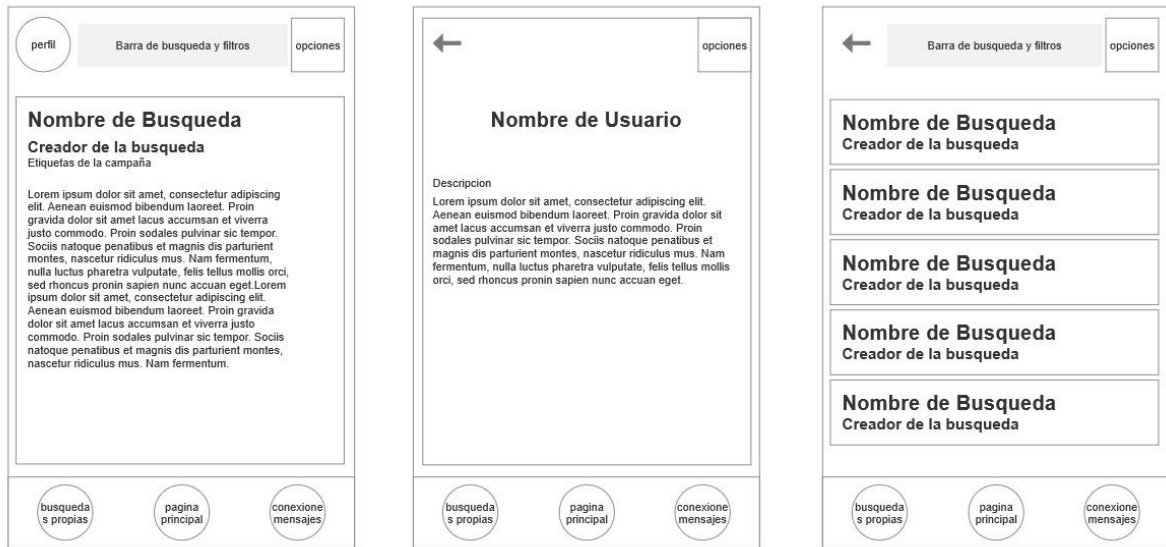
Precondiciones: El usuario se ha logueado. El lobby ya ha sido creado, El usuario tiene los permisos necesarios para administrar miembros.

Flujo normal:

1. Usuario accede a configuración de miembros del lobby.
2. Usuario realiza los cambios deseados.
3. Sistema actualiza cambios.

Postcondiciones: Los cambios del lobby se actualizaron exitosamente.

# UI Mockup



# Modelo de Dominio

