Gestión de instalaciones deportivas

# **DESCRIPCIÓN DEL PROBLEMA**

Se quiere realizar el control de reservas de las instalaciones que tiene un polideportivo (futbol, tenis, pádel, etc.). El programa dispone de una parte pública a la que pueden acceder los socios para realizar reservas. A determinadas partes de la aplicación solo pueden acceder los administradores de las instalaciones deportivas. Al realizar una reserva, los socios tienen determinadas restricciones:

* Un socio no puede mantener activa más de una reserva de una misma instalación.
* Solo se puede reservar en los 15 días siguientes a la fecha actual.  La duración de la reserva tendrá un tiempo máximo, que estará en función de la instalación reservada.  Un socio puede cancelar sus reservas hasta una hora antes del comienzo de la misma, en caso de que no suceda se le cobrará el importe de la reserva. Los administradores de las instalaciones deportivas no tienen estas restricciones a la hora de realizar y cancelar reservas. El horario de reservas estará pautado de hora en hora.

# **SOLUCIÓN PROPORCIONADA**

## Introducción

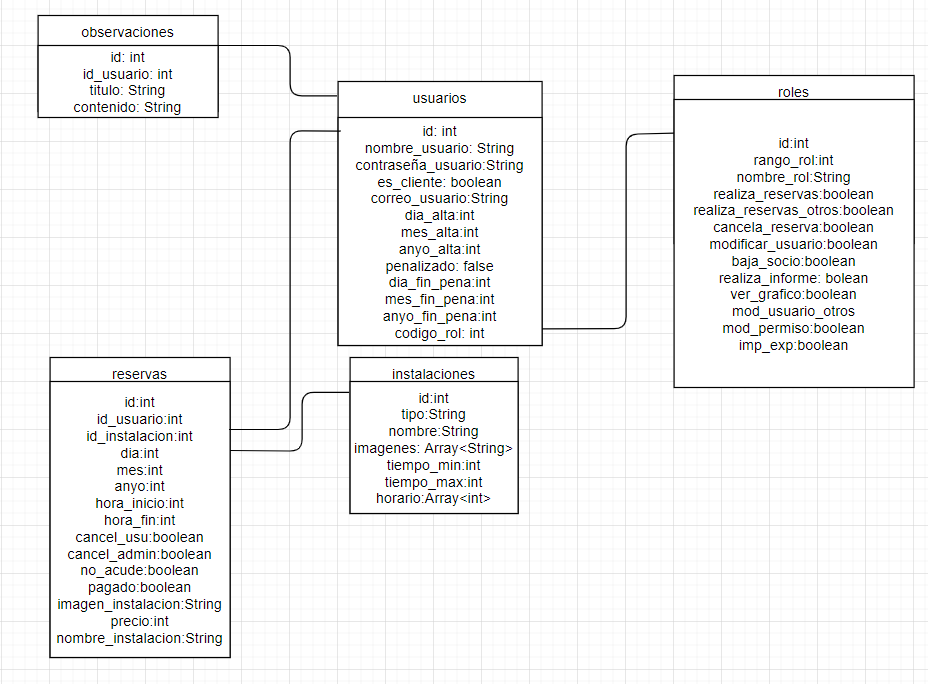
Para la solución del problema se ha optado por el desarrollo de dos aplicaciones Android, las cuales se comunicarán con el servidor.

## Herramientas utilizadas

Las herramientas empleadas para la realización de dicha tarea, han sido:

1. Librería Picasso: En concreto la versión “2.71828” es una librería empleada para cargar imágenes, si bien es cierto que hay otras opciones, a la hora de descargar varias imágenes esta librería es más rápida que otras como el uso de Inputstream, además de facilitar la descarga de las mismas.
2. Librería Retrofit 2. Para este proyecto se ha empleado la versión “2.7.0”. La finalidad del uso de la misma es la de obtener datos desde un api json, permitiendo dicha librería customizar los enlaces, con los cuales se realizan las interacciones con los datos, de manera sencilla.
3. Json-server: para todo el desarrollo, esta herramienta permite crear un servidor falso en local. La finalidad de esta herramienta es el testeo de la aplicación.

## Base de datos



La base de datos se compone de 5 listas:

1. Observaciones: Representa las observaciones realizadas por los empleados.

Está compuesta por:

* id\_usuario: representa al usuario al cual hace referencia.
* titulo: almacena un pequeño texto representativo de la observación.
* conenido: forma el cuerpo de la observación.

Esta tabla se podría comparar con el artículo de un periódico.

1. Usuarios: Contiene la lista de personas con acceso a la aplicación, ya sean clientes o empleados:

Se compone por:

* nombre\_usuario: representa el nombre, el cual el usuario o un trabajador con permisos ha elegido para dicho usuario.
* contraseña\_usuario: almacena la contraseña del usuario
* es\_cliente: determina si dicho usuario es un cliente o un empleado. Esto determinará a que aplicación tiene acceso.
* Correo\_usuario: almacena el correo del usuario.
* Dia, mes, anyo\_alta: contiene la fecha en la que se creó el usuario.
* Penalizado: si el usuario está penalizado
* Dia, mes, anyo\_fin\_pena: establece el fin de la penalización, si el usuario está penalizado.
* Código\_rol: representa al rol del usuario

1. Roles: guarda los distintos grupos de permisos, a los cuales pueden acceder los usuarios.

Se compone por:

* Rango\_rol: determina el nivel de restricción del rango, un empleado no puede asignar roles de un rango superior al suyo.
* Nombre\_rol: representa el nombre del rol.
* Los demás parámetros, representan los diferentes permisos que se le asignan a los usuarios a los cuales se les asigna dicho rol.

1. Reservas: almacena las reservas que realizan los usuarios.

Se compone por:

* Id\_usuario: indica el usuario al cual se ha asignado la reserva.
* Id\_instalación: guarda la instalación seleccionada para dicha reserva.
* Dia, mes, anyo, hora\_inicio, hora\_fin: indica la fecha de la reserva.
* Cancel\_usu, cancel\_admin: representa si ha sido cancelado por el usuario el cual ha realizado la reserva o por un empleado.
* no\_acude: si la fecha de la reserva es anterior los empleados pueden marcar si el usuario no ha asistido a la reserva.
* Pagado: indica si el usuario ha pagado la reserva.
* Imagen\_instalación: para favorecer la previsualización las reservas almacenan la primera imagen de la instalación.
* Precio: representa el precio en monedos aportados a la hora de realizar la reserva.
* Nombre\_instalacion: contiene el nombre de la instalación asignada a la reserva.

1. Instalaciones:

Se compone por:

* Tipo: contiene el tipo de instalación (baloncesto, tenis…)
* Nombre: almacena el nombre de la instalación.
* Imágenes: es una lista de enlaces donde están las imágenes relacionadas con la instalación con el fin de ayudar a los clientes a visualizar el aspecto de la instalación la cual van a reservar.

## Estructura del proyecto

Para favorecer la integridad del proyecto he dividido el proyecto en dos aplicaciones.

La primera está dirigida al usuario general donde los clientes pueden gestionar su cuenta, realizar reservas y administrar las mismas.

La segunda va dirigida para el uso profesional, incluye más funciones.

## Organización de la aplicación

## Explicación de las clases

## Interfaz gráfica

## Utilización del programa

## Aplicación móvil

# **CONCLUSIONES**

## Dificultades encontradas

Configuración de la herramienta json-server.

Diseño de la aplicación.

## Ampliaciones

Pagos en línea.

* Implementación de pagos en línea mediante paypal. Han surgido ciertas dificultades derivados del api de Paypal.

Servidor en la nube.

* Se ha estudiado varias opciones para el almacenamiento del servidor. A continuación, enumero una serie de propuestas a tener en cuenta:
  + Hostinger
  + Amazon Web Services
  + Google Cloud Platform
  + Microsoft Azure

Importar y exportar datos: Para realizar este módulo, lo más recomendable sería desarrollar una aplicación de escritorio, lo cual facilitaría la ubicación de los archivos.

## Conclusiones personales