

ANÁLISIS DEL PROYECTO

SPORT TRACKER

GESTOR DE EVENTOS DEPORTIVOS



JAVIER RODRÍGUEZ TORRES
2ºDAM IES FERNANDO AGUILAR QUIGNON

ÍNDICE

1. INTRODUCCIÓN	3
1.1. PROPÓSITO	3
1.2. OBJETIVOS	3
1.3. VIABILIDAD TÉCNICA Y ECONÓMICA DEL PROYECTO	3
1.4. TEMPORALIZACIÓN	4
2. ANÁLISIS	4
2.1. DOCUMENTACIÓN RELEVANTE	4
2.2. DEFINICIONES	5
2.3. REQUISITOS FUNCIONALES Y NO FUNCIONALES	5
2.4. DIAGRAMAS UML DE CASOS DE USO	7
2.5. MODELADO E/R. DIAGRAMAS E/R.	7
3. DISEÑO	8
3.1. ARQUITECTURA HARDWARE Y SOFTWARE DE LA SOLUCIÓN	8
3.2. MODELADO FUNCIONAL DE LA SOLUCIÓN. DIAGRAMAS DE CLASE	8
3.3. MODELO RELACIONAL.	8
4. IMPLEMENTACIÓN	11
4.1. REQUISITOS DE INSTALACIÓN Y EJECUCIÓN	11
4.2. IMPLEMENTACIÓN FUNCIONAL. CLASES	11
4.3. IMPLEMENTACIÓN DEL MODELO DE DATOS. TABLAS	12
5. PRUEBAS	13
5.1. PRUEBAS DE MÓDULOS	13
5.2. PRUEBAS DE INTEGRACIÓN Y SISTEMA	13
5.3. PRUEBAS DE INSTALACIÓN	14
6. CONCLUSIONES	15
6.1. GRADO DE CONSECUCIÓN DE LOS OBJETIVOS INICIALMENTE PLANTEADOS	15
6.2. DIFICULTADES ENCONTRADAS	16
6.3. PROPUESTAS DE MEJORA Y POSIBLES AMPLIACIONES	16
7. BIBLIOGRAFÍA Y RECURSOS ON-LINE	16

1. INTRODUCCIÓN

1.1. PROPÓSITO

Sport Tracker tiene como objetivo principal promover la actividad física, el deporte y la socialización a través de la organización de eventos deportivos, ya sea para competir en ligas, torneos o simplemente para jugar partidos informales.

La aplicación está diseñada para fomentar el deporte y la actividad física, reducir el sedentarismo, mejorar la salud y el bienestar, promover la socialización y el trabajo en equipo, desarrollar habilidades deportivas, así como promover la inclusión y la diversidad.

1.2 OBJETIVOS

- **Fomentar el deporte y la actividad física:** proporcionar una plataforma accesible donde los usuarios puedan encontrar y participar en una variedad de eventos deportivos, motivando a los usuarios a participar en actividades físicas regulares al ofrecer opciones diversas y atractivas de eventos deportivos.
- **Reducir el sedentarismo:** ofrecer alternativas atractivas a las actividades sedentarias mediante la promoción de eventos deportivos que sean divertidos y accesibles para personas de diferentes niveles de condición física.
- **Mejorar la salud y el bienestar:** facilitar el acceso a actividades deportivas que contribuyan a mejorar la salud cardiovascular, fortalecer los músculos y mejorar la salud mental y emocional.
- **Promover la socialización y el trabajo en equipo:** facilitar la formación de equipos y la interacción entre los miembros de la comunidad interesados en el deporte y fomenta la colaboración y el compañerismo a través de eventos deportivos grupales y competiciones en equipo.
- **Desarrollar habilidades deportivas:** Proporcionar oportunidades para que los usuarios mejoren sus habilidades deportivas participando en eventos competitivos.
- **Promover la inclusión y la diversidad:** Crear un entorno inclusivo donde todas las personas, independientemente de su género, edad, habilidad física o antecedentes culturales, se sientan bienvenidas y apoyadas para participar en eventos deportivos.

1.3 VIABILIDAD TÉCNICA Y ECONÓMICA DEL PROYECTO

La elección de usar Java y JavaFX para la versión de ordenador y Android Studio para la versión de teléfono se debe a la compatibilidad entre ellas, está desarrollado en java y da la posibilidades de compartir recursos como por ejemplo librerías y en caso de algún problema, permite encontrar fácilmente ayuda y recursos en línea.

Se usa una codificación de datos importantes con SHA-256, para mantener información importante del usuarios segura como por ejemplo la contraseña de su cuenta.

1.4 TEMPORALIZACIÓN

En un principio, se planteó desarrollar el proyecto durante los tres meses de Formación en Centros de Trabajo, pero debido al gran tiempo de desplazamiento por la distancia a la empresa, se redujo el tiempo de desarrollo y se pospuso la finalización del proyecto hasta la actual convocatoria. Finalizado ya el proyecto, se buscará mejorar su calidad e implementar nuevas funcionalidades y mejoras como desarrollo personal.

2. ANÁLISIS

2.1 DOCUMENTACIÓN RELEVANTE

El sistema permite a las personas registrarse y realizar diversas acciones adquiriendo varios roles:

- **Crear** un equipo (adquiriendo el rol de **capitán**)
- **Unirse** a un equipo (adquiriendo el rol de **jugador**)
- No pertenecer a ningún equipo ni ser gestor (**usuario**).

El jugador que cree un nuevo equipo será el **capitán** de este y **gestionará los partidos** que jugará este equipo. El capitán podrá **crear un partido amistoso** donde buscará que **otro capitán acepte jugar** contra este equipo. También podrá **crear una competición** o elegir **participar en alguna competición** que esté disponible, es decir que algún capitán la haya creado y necesiten equipos para completar la competición. El capitán también podrá **eliminar el equipo** que creó, con lo que conlleva, sacar al resto de jugadores pertenecientes a este.

El resto de **jugadores** de los equipos no tendrán las funciones del capitán, pero sí tendrán otras muchas funciones las cuales el capitán si podrá hacer ya que un **capitán también es un jugador de un equipo**. Los jugadores podrán **ver el calendario del equipo**, pudiendo ver los partidos futuros y pasados. También podrán **ver los datos de la competición** si está participando en ella. Si quisiese, un jugador podría **abandonar al equipo al que pertenece**.

En general, todos los usuarios pertenecientes a un deporte podrán hacer varias cosas en común:

- **Darse de baja**
- **Ver noticias**
- **Comentar noticias**
- **Unirse a un partido informal** sin necesidad de estar en un equipo.
- **Solicitar la creación de un nuevo deporte** (Rellenará un formulario con los datos del deporte y el administrador acepta o rechaza la creación del deporte).

2.2 DEFINICIONES

- **SportTracker:** SportTracker es el nombre de la aplicación gestora de eventos deportivos.
- **Usuario:** Toda persona registrada en la aplicación es un usuario de esta, adquiriendo un rol 'Usuario'.
- **Equipo:** Conjunto de usuarios que juegan juntos en diversos partidos.
- **Miembro:** Usuario perteneciente a un equipo.
- **Capitán:** Creador del equipo al que pertenece.
- **Informal:** Partido entre usuarios sin necesidad de pertenecer a ningún equipo.
- **Amistoso:** Partido entre dos equipos del mismo deporte.
- **Competición:** Evento donde se enfrentan varios equipos, y según el tipo de esta, liga o torneo, hay una forma de seleccionar al ganador de esta competición.
- **Jornada:** Partido generado por una competición donde se enfrentan dos equipos pertenecientes a una misma competición.
- **Evento:** Evento deportivo de la aplicación, pudiendo ser cualquier tipo de partido o una competición.
- **Noticia:** Información del final de una competición anunciando a su ganador.

2.3 REQUISITOS FUNCIONALES Y NO FUNCIONALES

Requisitos funcionales:

→ Aplicación del Móvil (Usuario):

- ◆ La aplicación debe permitir a los usuarios registrarse e iniciar sesión utilizando un nombre de usuario y contraseña.
- ◆ Los usuarios deben poder ver y editar su perfil.
- ◆ La aplicación debe permitir a los usuarios buscar y filtrar contenido de acuerdo con diferentes criterios.
- ◆ La aplicación debe cumplir con las regulaciones de privacidad y protección de datos.
- ◆ La aplicación debe permitir a los usuarios registrar equipos deportivos.
- ◆ Los usuarios deben poder crear y gestionar eventos deportivos.
- ◆ La aplicación debe permitir a los usuarios programar partidos entre equipos, estableciendo fechas y horarios.
- ◆ La aplicación debe permitir solicitar la creación de nuevos deportes.
- ◆ La aplicación permitirá interactuar con las noticias de cualquier evento.
- ◆ La aplicación permitirá la gestión del equipo si es el capitán, pudiendo buscar partidos entre equipos, crear y unirse a competiciones, gestionar a los jugadores del equipo, etc.

→ **Aplicación de escritorio (Administrador):**

- ◆ La aplicación debe permitir a los administradores iniciar sesión utilizando un nombre de usuario y contraseña.
- ◆ El administrador debe poder crear, modificar y eliminar cuentas de usuario, así como asignar diferentes niveles de permisos según el rol y las responsabilidades de cada usuario.
- ◆ El administrador debe poder eliminar eventos deportivos.
- ◆ El administrador debe poder eliminar comentarios de las noticias.
- ◆ El administrador debe poder aceptar o rechazar solicitudes de deportes.

Requisitos no funcionales:

- El sistema debe ser seguro, protegiendo la información personal de los usuarios utilizando medidas como el cifrado de datos y evitando accesos no autorizados.
- El sistema debe responder de manera rápida y eficiente, mostrando en vivo las modificaciones que se puedan haber hecho en el sistema.
- La interfaz de usuario debe ser intuitiva y fácil de usar para que los usuarios puedan acceder y realizar acciones sin dificultad.
- El sistema debe ser fácil de mantener y actualizar para incluir nuevas características y correcciones de errores.
- La aplicación debe ser compatible con una variedad de dispositivos móviles, asegurando una experiencia consistente para todos los usuarios, independientemente del dispositivo que utilicen

3. DISEÑO

3.1 ARQUITECTURA HARDWARE Y SOFTWARE DE LA SOLUCIÓN

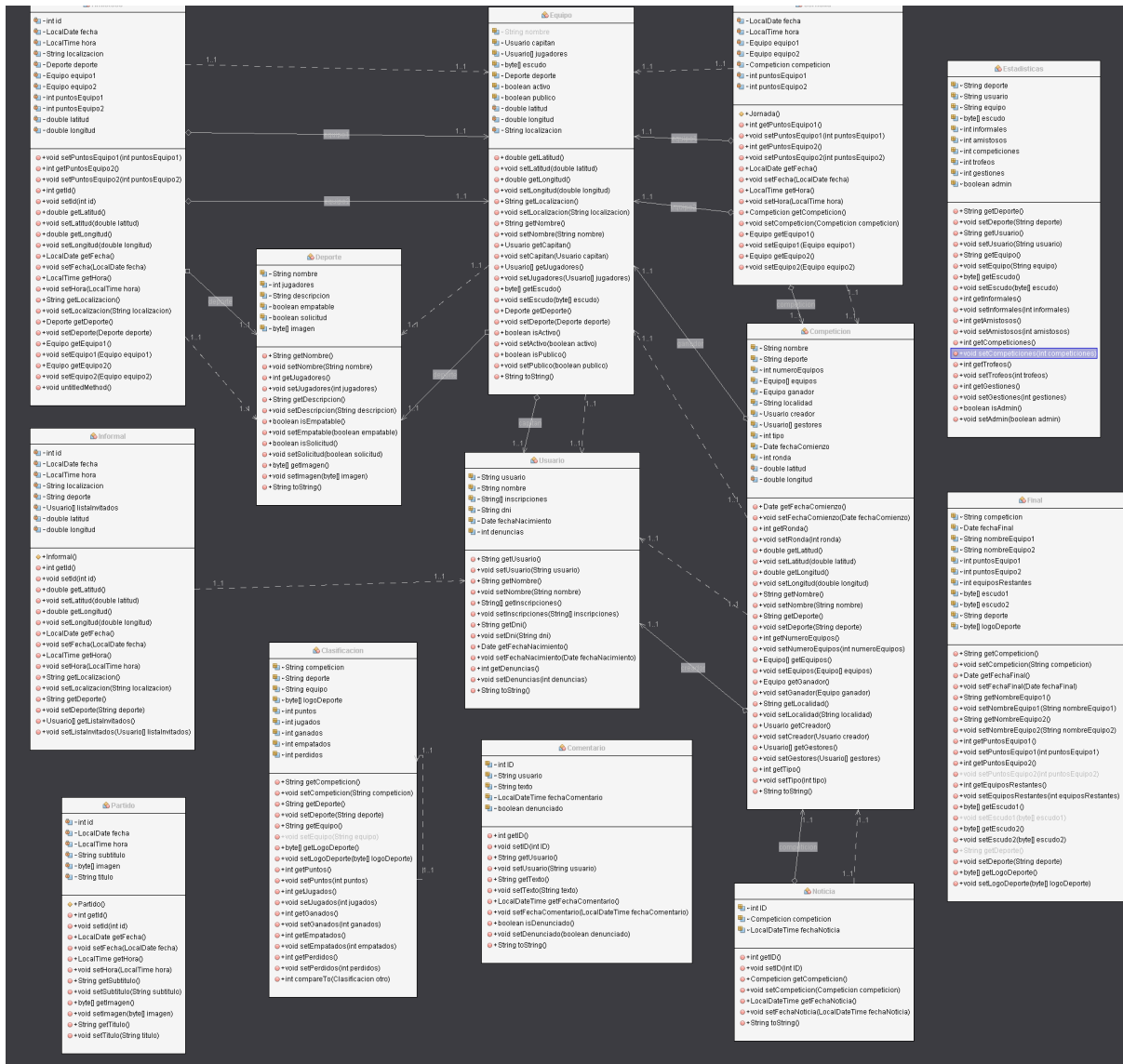
La infraestructura de hardware necesaria implica contar con un servidor robusto, capaz de manejar eficientemente las solicitudes gracias a su potencia de CPU y capacidad de RAM. Para asegurar una conexión estable entre las aplicaciones clientes y el servidor, se requiere una IP estática, aunque los clientes tienen la flexibilidad de ajustar las preferencias de red según sea necesario. Si bien es preferible alojar el servidor en un host remoto, también se puede optar por utilizar un equipo propio, considerando los costos asociados en términos de energía y espacio. Por otra parte, las aplicaciones clientes no demandan equipos de alto rendimiento, ya sean de escritorio o dispositivos móviles.

En lo que respecta a la arquitectura de software, el servidor depende del servicio de MySQL junto con la base de datos correspondiente para gestionar las solicitudes entrantes. Es necesario tener instalado Java (JRE versión 8 o superior) para ejecutar la aplicación de escritorio. Cada usuario utilizará la aplicación que se ajuste a su rol específico, aunque aquellos con varios roles específicos tendrán acceso a ambas opciones. Se implementa el modelo cliente-servidor para facilitar la comunicación entre el software, y las notificaciones entre clientes se canalizan a través del servidor mediante el patrón de observadores.

Para interactuar con la base de datos, se hace uso de la librería JDBC. La obtención de localizaciones reales se lleva a cabo mediante la API de Google Places, que proporciona información detallada como coordenadas y direcciones de lugares específicos. Asimismo, para simular transacciones de pago desde la aplicación móvil, se recurre a la API de PayPal, que facilita la creación de una pasarela de pago con el monto correspondiente.

Se integra un sistema de encriptación de contraseñas con SHA-256 para fortalecer la seguridad al transmitirlos por el protocolo. Este sistema aplica de manera uniforme en todas las aplicaciones del proyecto.

3.2 MODELADO FUNCIONAL DE LA SOLUCIÓN. DIAGRAMAS DE CLASE



3.3 MODELO RELACIONAL.

DEPORTE:

- ID (INTEGER, PRIMARY KEY)
- NOMBRE (VARCHAR(25), UNIQUE)
- NUMEROJUGADORES (INTEGER, NOT NULL)
- DESCRIPCION (VARCHAR(600), NOT NULL)
- EMPATABLE (BOOL, DEFAULT TRUE)
- SOLICITUD (BOOL, DEFAULT FALSE)
- IMAGEN (LONGBLOB)

USUARIO:

- ID (INTEGER, PRIMARY KEY)
- USUARIO (VARCHAR(25), UNIQUE)

- CONTRASEÑA (VARBINARY(100), NOT NULL)
- NOMBRE (VARCHAR(50), NOT NULL)
- DNI (VARCHAR(9), NOT NULL UNIQUE)
- FECHANACIMIENTO (DATE, NOT NULL)
- DENUNCIAS (INT, DEFAULT 0, NOT NULL)

ROL:

- ID (INTEGER, PRIMARY KEY)
- NOMBRE (VARCHAR(25), UNIQUE)

INSCRIPCION:

- ID (INTEGER, PRIMARY KEY)
- ID_USUARIO (INTEGER, NOT NULL, FOREIGN KEY REFERENCES USUARIO(ID))
- ID_DEPORTE (INTEGER, FOREIGN KEY REFERENCES DEPORTE(ID))
- ID_ROL (INTEGER, NOT NULL, FOREIGN KEY REFERENCES ROL(ID))
- FECHAINSCRIPCION (DATE, NOT NULL)

EQUIPO:

- ID (INTEGER, PRIMARY KEY)
- NOMBRE (VARCHAR(30), UNIQUE)
- ID_DEPORTE (INTEGER, NOT NULL, FOREIGN KEY REFERENCES DEPORTE(ID))
- ID_CAPITAN (INTEGER, FOREIGN KEY REFERENCES USUARIO(ID))
- IMAGEN (LONGBLOB, NOT NULL)
- ACTIVO (BOOL, NOT NULL, DEFAULT TRUE)
- DISPONIBLE (BOOL, NOT NULL, DEFAULT TRUE)
- LOCALIZACION (VARCHAR(100), NOT NULL)
- LATITUD (DECIMAL(9,6))
- LONGITUD (DECIMAL(9,6))

MIEMBROSEQUIPO:

- ID (INTEGER, PRIMARY KEY)
- ID_MIEMBRO (INTEGER, NOT NULL, FOREIGN KEY REFERENCES USUARIO(ID))
- ID_EQUIPO (INTEGER, NOT NULL, FOREIGN KEY REFERENCES EQUIPO(ID))

COMPETICION:

- ID (INTEGER, PRIMARY KEY)
- NOMBRE (VARCHAR(50), UNIQUE)
- ID_DEPORTE (INTEGER, NOT NULL, FOREIGN KEY REFERENCES DEPORTE(ID))
- NEQUIPOS (INTEGER, NOT NULL)
- ID_EQUIPO_GANADOR (INTEGER, FOREIGN KEY REFERENCES EQUIPO(ID))
- LOCALIDAD (VARCHAR(100), NOT NULL)
- ID_CREADOR (INTEGER, FOREIGN KEY REFERENCES USUARIO(ID))
- FECHA_COMIENZO (DATE, NOT NULL)
- TIPO (INTEGER, NOT NULL)
- RONDA (INTEGER, DEFAULT -1)
- DISPONIBLE (BOOLEAN, DEFAULT TRUE)
- LATITUD (DECIMAL(9,6))
- LONGITUD (DECIMAL(9,6))

COMPETIDORES:

- ID (INTEGER, PRIMARY KEY)
- ID_COMPETICION (INTEGER, NOT NULL, FOREIGN KEY REFERENCES COMPETICION(ID))
- ID_EQUIPO (INTEGER, NOT NULL, FOREIGN KEY REFERENCES EQUIPO(ID))

MIEMBROSCOMPETICION:

- ID (INTEGER, PRIMARY KEY)
- ID_COMPETIDORES (INTEGER, NOT NULL, FOREIGN KEY REFERENCES COMPETIDORES(ID))
- ID_MIEMBROSEQUIPO (INTEGER, NOT NULL, FOREIGN KEY REFERENCES MIEMBROSEQUIPO(ID))

NOTICIA:

- ID (INTEGER, PRIMARY KEY)
- ID_COMPETICION (INTEGER, NOT NULL, FOREIGN KEY REFERENCES COMPETICION(ID))
- FECHA (DATE, NOT NULL)
- HORA (TIME, NOT NULL)

COMENTARIO:

- ID (INTEGER, PRIMARY KEY)
- ID_NOTICIA (INTEGER, NOT NULL, FOREIGN KEY REFERENCES NOTICIA(ID))
- TEXTO (VARCHAR(100), NOT NULL)
- ID_USUARIO (INTEGER, NOT NULL, FOREIGN KEY REFERENCES USUARIO(ID))
- FECHA (DATE, NOT NULL)
- HORA (TIME, NOT NULL)
- DENUNCIADO (BOOLEAN, NOT NULL, DEFAULT FALSE)

INFORMAL:

- ID (INTEGER, PRIMARY KEY)
- FECHA (DATE, NOT NULL)
- HORA (VARCHAR(5), NOT NULL)
- LOCALIZACION (VARCHAR(100), NOT NULL)
- ID_DEPORTE (INTEGER, NOT NULL, FOREIGN KEY REFERENCES DEPORTE(ID))
- LATITUD (DECIMAL(9,6))
- LONGITUD (DECIMAL(9,6))
- DISPONIBLE (BOOL, DEFAULT TRUE)

JUGADORESINFORMAL:

- ID (INTEGER, PRIMARY KEY)
- ID_INFORMAL (INTEGER, NOT NULL, FOREIGN KEY REFERENCES INFORMAL(ID))
- ID_USUARIO (INTEGER, NOT NULL, FOREIGN KEY REFERENCES USUARIO(ID))

AMISTOSO:

- ID (INTEGER, PRIMARY KEY)
- FECHA (DATE, NOT NULL)
- HORA (VARCHAR(5), NOT NULL)
- LOCALIZACION (VARCHAR(100), NOT NULL)
- ID_DEPORTE (INTEGER, NOT NULL, FOREIGN KEY REFERENCES DEPORTE(ID))
- ID_EQUIPO1 (INTEGER, FOREIGN KEY REFERENCES EQUIPO(ID))
- ID_EQUIPO2 (INTEGER, FOREIGN KEY REFERENCES EQUIPO(ID))
- PUNTOS_EQUIPO1 (INTEGER, NOT NULL, DEFAULT -1)
- PUNTOS_EQUIPO2 (INTEGER, NOT NULL, DEFAULT -1)
- LATITUD (DECIMAL(9,6))

- LONGITUD (DECIMAL(9,6))
- DISPONIBLE (BOOLEAN, DEFAULT TRUE)

JORNADA:

- ID (INTEGER, PRIMARY KEY)
- ID_COMPETICION (INTEGER, NOT NULL, FOREIGN KEY REFERENCES COMPETICION(ID))
- EQUIPOSRESTANTES (INTEGER, NOT NULL)
- FECHA (DATE, NOT NULL)
- HORA (VARCHAR(5), NOT NULL)
- LOCALIZACION (VARCHAR(100), NOT NULL)
- ID_EQUIPO1 (INTEGER, NOT NULL, FOREIGN KEY REFERENCES EQUIPO(ID))
- ID_EQUIPO2 (INTEGER, NOT NULL, FOREIGN KEY REFERENCES EQUIPO(ID))
- PUNTOS_EQUIPO1 (INTEGER, DEFAULT NULL)
- PUNTOS_EQUIPO2 (INTEGER, DEFAULT NULL)
- LATITUD (DECIMAL(9,6))
- LONGITUD (DECIMAL(9,6))

4. IMPLEMENTACIÓN

4.1 REQUISITOS DE INSTALACIÓN Y EJECUCIÓN

Para comenzar, es necesario configurar el entorno donde se instalarán las distintas aplicaciones. A continuación, se detalla los requisitos a considerar:

Para la aplicación del servidor:

- Se requiere un sistema operativo Windows.
- Es necesario instalar MySQL en su versión 8.0 para crear la base de datos.
- Se debe contar con Java JRE en su versión 8.0 para ejecutar la aplicación.
- Es indispensable disponer de una conexión de red para enviar y recibir las peticiones.

Para la aplicación de escritorio:

- Se necesita un sistema operativo Windows.
- Java JRE en su versión 8.0 es necesario para ejecutar la aplicación.
- Se debe tener acceso a una conexión de red para establecer comunicación con el servidor.

Para la aplicación móvil:

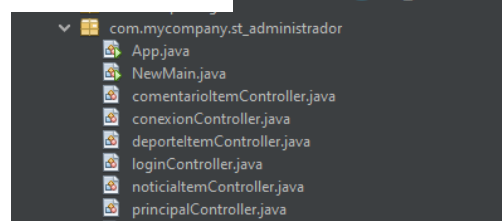
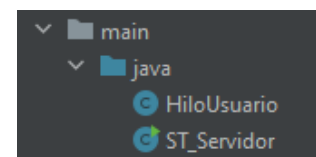
- Se requiere un dispositivo Android con una versión mínima de Android 9.
- Es necesario otorgar permisos de ubicación para cargar el mapa y determinar la ubicación del dispositivo.
- Se debe tener una conexión a internet para poder conectarse al servidor remoto.

4.2 IMPLEMENTACIÓN FUNCIONAL. CLASES

Aplicación del servidor:

Contiene dos clases, la clase HiloUsuario donde maneja un hilo que se conecta con el servidor para enviar y recibir información, y la clase principal ST_Servidor donde maneja todos los hilos conectados al servidor.

Aplicación de escritorio:



El paquete principal contiene los controladores de cada vista (almacenadas en el segundo paquete), que gestionan la comunicación de la interfaz que ve el usuario con su conexión al servidor y contiene la clase principal para ejecutar la aplicación.

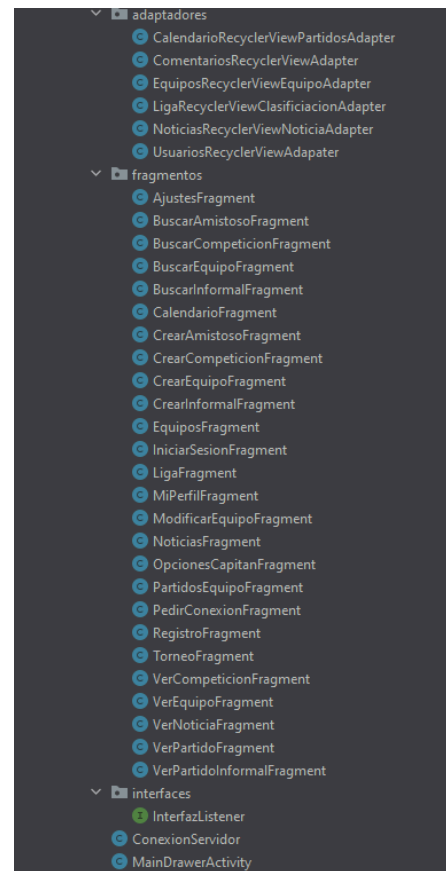
Aplicación de móvil:

Las dos clases principales son `MainDrawerActivity`, donde gestiona la comunicación de los fragmentos con el servidor y la clase `ConexionServidor`, la cual envía y recibe datos del servidor.

En el paquete `fragmento` se almacenan las clases que gestionan las vistas de la aplicación.

En el paquete `adaptadores`, se almacenan los distintos adaptadores usados para mostrar elementos de una lista en los fragmentos.

En el paquete `interfaces` tenemos una interfaz que comunica a la clase principal con los fragmentos y viceversa.



4.3 IMPLEMENTACIÓN DEL MODELO DE DATOS. TABLAS

- **DEPORTE:** Almacena información sobre los deportes disponibles, incluyendo su nombre, número de jugadores, descripción, entre otros detalles relacionados.
- **USUARIO:** Contiene detalles de los usuarios registrados, como su nombre de usuario, contraseña (en formato binario), nombre, DNI, fecha de nacimiento y número de denuncias.
- **ROL:** Guarda información sobre los roles disponibles en el sistema.
- **INSCRIPCION:** Registra la inscripción de usuarios en diferentes deportes y roles, con la fecha de inscripción correspondiente.

- **EQUIPO:** Almacena datos relacionados con los equipos participantes en competencias, incluyendo su nombre, deporte, capitán, ubicación, entre otros detalles.
- **MIEMBROSEQUIPO:** Asocia usuarios a equipos, indicando la membresía de cada usuario en un equipo específico.
- **COMPETICION:** Contiene información sobre las competencias, incluyendo el deporte, número de equipos, creador, fecha de inicio, entre otros detalles relevantes.
- **COMPETIDORES:** Asocia equipos a competencias específicas.
- **MIEMBROSCOMPETICION:** Asocia miembros de equipos a competencias en las que participan.
- **NOTICIA:** Almacena noticias relacionadas con competencias, incluyendo la fecha y hora de publicación.
- **COMENTARIO:** Registra comentarios realizados por usuarios en noticias, con detalles como el texto del comentario, fecha y hora.
- **INFORMAL:** Guarda información sobre eventos informales, como fecha, hora, ubicación y deporte.
- **JUGADORESINFORMAL:** Asocia usuarios a eventos informales en los que participan como jugadores.
- **AMISTOSO:** Contiene detalles sobre partidos amistosos, como fecha, hora, ubicación, equipos participantes y resultados.
- **JORNADA:** Registra jornadas de competencias, incluyendo la fecha, hora, ubicación, equipos participantes y resultados parciales.

5. PRUEBAS

5.1 PRUEBAS DE MÓDULOS

Aplicación escritorio:

- **Autenticación:** Se procede a iniciar sesión con el usuario y contraseña proporcionados. Si la información es correcta, se muestra un mensaje de confirmación y se carga la interfaz principal. De lo contrario, muestra un mensaje de error. En el primer acceso, se solicita crear al administrador del sistema con los detalles proporcionados.
- **Creación de deporte:** A través de un formulario, se rellenan una serie de campos que da lugar a la creación de un nuevo deporte con el que podrán interactuar los usuarios.
- **Filtrado:** Pruebas de filtrados en las distintas vistas de la aplicación, haciendo las búsqueda más específica.

- Eliminar elementos: Opciones de eliminar distintos elementos en la aplicación de forma exitosa mostrando el cambio en la vista.
- Cierre de sesión: Permite finalizar la sesión del usuario activo, retornando a la pantalla de inicio de sesión.
- Configuración de conexión: Facilita la modificación de los parámetros de conexión (dirección IP y puerto) para establecer conexión con el servidor remoto. Se muestra un mensaje indicativo según el estado de la conexión. Los datos ingresados se almacenan en un archivo de configuración para futuras referencias.

Aplicación móvil:

- Autenticación: Se procede a iniciar sesión con el usuario y contraseña proporcionados. Si la información es correcta, se muestra un mensaje de confirmación y se carga la interfaz principal. De lo contrario, muestra un mensaje de error. En el primer acceso, se solicita crear al administrador del sistema con los detalles proporcionados.
- Creación de elementos: A través de un formulario, se rellenan una serie de campos que da lugar a la creación de nuevos elementos como lo son Equipos, partidos, competiciones o comentarios. En caso de algún error mostrará un mensaje por pantalla.
- Filtrado: Pruebas de filtrados en las distintas vistas de la aplicación, haciendo las búsqueda más específica.
- Eliminar elementos: Opciones de eliminar distintos elementos en la aplicación de forma exitosa mostrando el cambio en la vista, como por ejemplo equipos o partidos. En caso de algún error mostrará un mensaje por pantalla.
- Cierre de sesión: Permite finalizar la sesión del usuario activo, retornando a la pantalla de inicio de sesión.
- Configuración de conexión: Facilita la modificación de los parámetros de conexión (dirección IP y puerto) para establecer conexión con el servidor remoto. Se muestra un mensaje indicativo según el estado de la conexión. Los datos ingresados se almacenan en un archivo de configuración para futuras referencias.

5.2 PRUEBAS DE INTEGRACIÓN Y SISTEMA

- **Prueba de Conectividad:** Se efectuará una exhaustiva prueba de conectividad para garantizar que las interacciones entre los clientes y el servidor se establezcan de manera fluida. Se revisarán minuciosamente la dirección IP y el puerto del servidor para asegurarse de que estén correctamente configurados y accesibles desde la red. Una vez confirmada la accesibilidad del servidor, se procederá a que los clientes se conecten sin contratiempos utilizando la dirección IP correspondiente.
- **Prueba de Usabilidad:** se llevarán a cabo pruebas con usuarios reales para evaluar la usabilidad de las aplicaciones en condiciones de uso real. Se analizará la experiencia del usuario, prestando especial atención a la simplicidad e intuición de la interfaz de las aplicaciones.

- **Prueba de Rendimiento:** se realizarán pruebas de estrés exhaustivas para evaluar la capacidad de respuesta del servidor ante múltiples solicitudes simultáneas de clientes, tanto de dispositivos de escritorio como móviles. Se verificará que el servidor pueda manejar estas peticiones de manera eficiente y oportuna.
- **Prueba de Seguridad:** se realizará una exhaustiva evaluación de la seguridad del entorno, centrándose en la encriptación de datos sensibles antes de ser transmitidos a través de la red. Se llevarán a cabo pruebas rigurosas de inicio de sesión para garantizar la seguridad y protección de los procesos de autenticación y registro de usuarios.
- **Prueba de Manejo de Errores:** se asegurará que todas las operaciones realizadas con la base de datos estén protegidas mediante transacciones para evitar la corrupción de datos en caso de fallos o problemas con el gestor de la base de datos. Esta medida garantizará la integridad y consistencia de los datos almacenados en el sistema.

5.3 PRUEBAS DE INSTALACIÓN

Para probar los instaladores, se cuenta con dos sistemas Windows: una máquina física y una máquina virtual. La instalación del servidor se realizará en la máquina virtual, la cual estará conectada a la red de la máquina física. Por otro lado, el cliente de escritorio se instalará en la máquina física. Ambos sistemas estarán conectados entre sí utilizando el modo puente en la configuración de red de la máquina virtual.

Consideraciones para la instalación del servidor:

- La máquina virtual debe tener instalado MySQL Server, y se debe contar con un usuario "root" con la contraseña "admin". Este usuario será utilizado por el instalador para crear la base de datos y cargar las tablas necesarias.
- Se debe verificar que la conexión de red esté activa en la máquina virtual, para permitir que los clientes puedan conectarse al servidor.
- Es crucial asegurarse de que el puerto introducido esté disponible y no sea utilizado por ninguna otra aplicación, ya que será utilizado por el protocolo de comunicación del servidor.

Consideraciones para la instalación del cliente del escritorio:

- Se requiere una conexión de red activa en la máquina física para que el cliente pueda conectarse al servidor. Se debe verificar que el antivirus o el cortafuegos no interfieran con la conexión.
- Si se realiza la instalación en una ubicación con privilegios de administrador, el usuario que ejecute el cliente debe contar con los permisos necesarios.

En ambos casos, tanto para el servidor como para el cliente de escritorio, es fundamental que las máquinas tengan instalado el entorno de ejecución de Java (JRE), con una versión mínima de la 8, para poder ejecutar el software correctamente.

Consideraciones para la instalación del cliente del móvil:

- Se requiere como mínimo la versión 9 de Android para poder ejecutar las aplicaciones.
- Se necesita una conexión a internet activa en el dispositivo para que el cliente pueda conectarse al servidor. Además, se debe asegurar que el servidor esté en la misma red.

- Para hacer uso del mapa integrado en los clientes Android, el dispositivo debe tener instalados los Servicios de Google Play y tener la ubicación activada.

6. CONCLUSIONES

6.1 GRADO DE CONSECUCIÓN DE LOS OBJETIVOS INICIALMENTE PLANTEADOS

Para evaluar el grado de consecución de los objetivos inicialmente planteados para la aplicación "Sport Tracker", podemos analizar cómo cada objetivo se ha abordado y si se han logrado alcanzar. Basándonos en los propósitos y objetivos proporcionados, podemos realizar la siguiente evaluación:

Fomentar el deporte y la actividad física:

- Se proporciona una plataforma accesible para encontrar y participar en eventos deportivos.
- Se ofrecen opciones diversas y atractivas de eventos deportivos para motivar la participación.
- *Grado de Consecución: Alto*

Reducir el sedentarismo:

- Se ofrecen alternativas atractivas a las actividades sedentarias.
- *Grado de Consecución: Alto*

Mejorar la salud y el bienestar:

- Se facilita el acceso a actividades deportivas que contribuyen a mejorar la salud cardiovascular, fortalecer los músculos y mejorar la salud mental y emocional.
- *Grado de Consecución: Alto*

Promover la socialización y el trabajo en equipo:

- Se facilita la formación de equipos y la interacción entre los miembros de la comunidad interesados en el deporte.
- Se fomenta la colaboración y el compañerismo a través de eventos deportivos grupales y competiciones en equipo.
- *Grado de Consecución: Alto*

Desarrollar habilidades deportivas:

- Se proporcionan oportunidades para mejorar las habilidades deportivas participando en eventos competitivos.
- *Grado de Consecución: Medio*

Promover la inclusión y la diversidad:

- Se crea un entorno inclusivo donde todas las personas se sienten bienvenidas y apoyadas para participar en eventos deportivos, independientemente de su género, edad, habilidad física o antecedentes culturales.
- *Grado de Consecución: Alto*

En general, la aplicación "Sport Tracker" ha logrado un alto grado de consecución en la mayoría de sus objetivos iniciales, especialmente en aquellos relacionados con la promoción del deporte, la actividad física, la salud y el bienestar, así como la socialización y la inclusión. Sin embargo, puede haber áreas donde el desarrollo de habilidades deportivas podría ser mejorado para alcanzar un grado de consecución más alto en todos los objetivos planteados.

6.2 DIFICULTADES ENCONTRADAS

El proyecto ha causado diversas dificultades como establecer un protocolo de comunicación efectivo entre el servidor y los clientes. Esto implicaba considerar todas las posibles solicitudes de los clientes y las respuestas correspondientes del servidor.

Otra dificultad fue gestionar las acciones de los distintos eventos (partidos y comienzos de competiciones) y para ello, tuve que implementar un hilo en el servidor que duerma hasta un evento próximo, y llegado el momento, se despierte y gestiona tal evento. También tuve que controlar que al añadir o eliminar un evento, se despierte el hilo y si ha cambiado el evento más próximo, cambiar el tiempo que dormirá el hilo.

6.3 PROPUESTAS DE MEJORA Y POSIBLES AMPLIACIONES

Algunas propuestas de mejoras de la aplicación serían las siguientes:

- **Implementar un rol gestor:** una persona se encargará de las gestiones de las distintas competiciones, pudiendo existir más de un gestor por competición.
- **Mejorar el filtrado de los eventos:** implementar una mejor navegación por el mapa y filtrados de los distintos eventos deportivos.
- **Crear un sistema de comunicación:** implementar un chat donde se puedan comunicar los miembros de un mismo equipo.

7. BIBLIOGRAFÍA Y RECURSOS ON-LINE

- MySQL Server: <https://www.mysql.com/>
- NetBeans IDE: <https://netbeans.apache.org/>
- Drive JDBC: <https://dev.mysql.com/downloads/connector/j/>
- Java FX: <https://openjfx.io/>
- Android Studio IDE: <https://developer.android.com/>
- Diseño de aplicación móvil: <https://openwebinars.net/academia/>
- API de Google Maps: <https://developers.google.com/maps?hl=es-419>
- API de PayPal: <https://www.youtube.com/watch?v=lcbDdeZVPnk>
- Inno Setup: <https://jrsoftware.org/isinfo.php>
- Launch4J: <https://launch4j.sourceforge.net/>