

Presentación de Avance II “ArchData”

Autores: Angel Alarcón
Mathiu Orellana

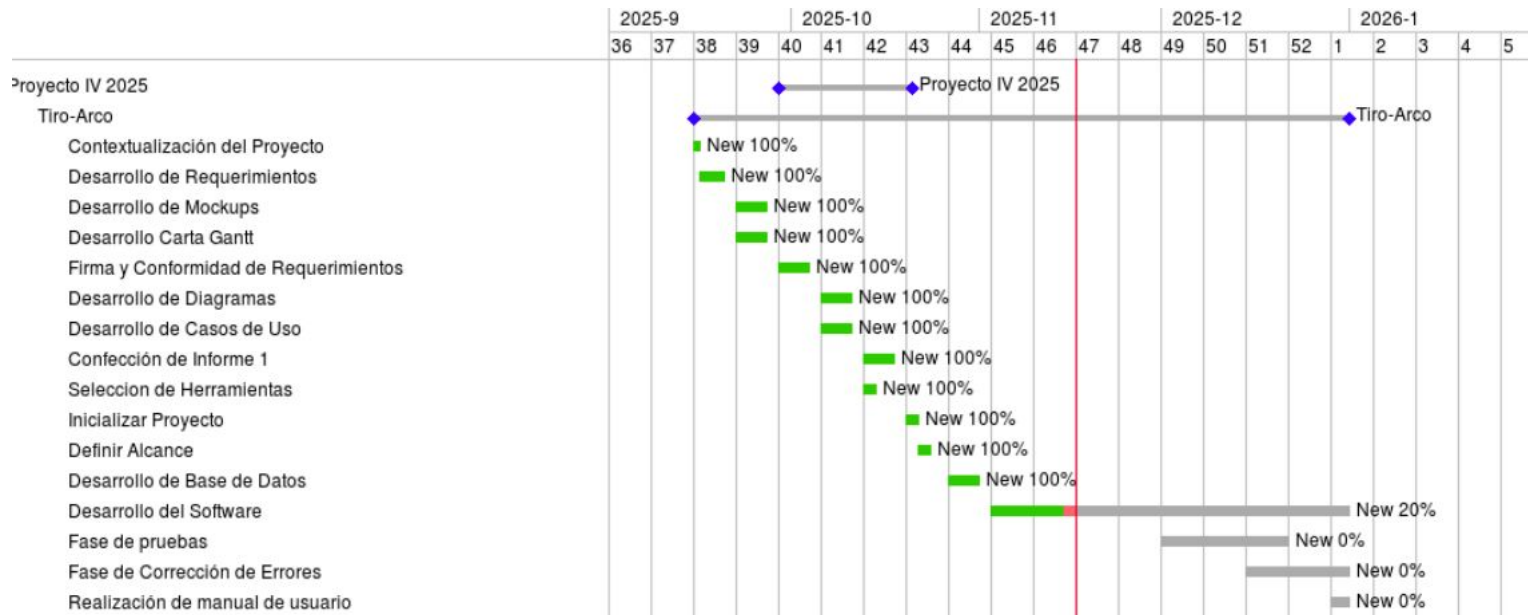
Curso: Proyecto IV

Profesor: Diego Aracena





Planificación Cronológica



La planificación del proyecto se determinó a través de una carta gantt, en la cual se tiene agendado como fecha de inicio del proyecto la primera semana de Septiembre, y finalizará la última semana de Diciembre.



Herramientas a utilizar

- **Desarrollo de la aplicación:** Para el desarrollo de la aplicación se decidió utilizar **Flutter** el cual es un framework elaborado por Google que permite construir aplicaciones nativas de alto rendimiento y además es de código abierto. Consta de diversas características destacables como el desarrollo rápido gracias a la función Hot Reload, la cual permite reflejar cambios de manera instantánea, también contiene bibliotecas gratuitas para implementar la funcionalidad TTS en el sistema.

- **Backend y Base de Datos:** Para la base de datos junto al backend se utilizará **Supabase**, este software permite escuchar a la base de datos en tiempo real, lo que permite el reflejo de los tiros de manera instantánea, funciona a través de base de datos PostgreSQL, y además es gratuita.

- **Desarrollo y Almacenamiento de Documentación:** Se utilizará **Google Drive** el cual es una plataforma de almacenamiento en la nube que permite guardar, compartir y colaborar en archivos de manera eficiente. Es ideal para trabajar en equipo, ya que facilita el acceso a los archivos desde cualquier dispositivo con conexión a internet y tiene opciones de control de permisos y versiones de los documentos.

- **Versionado y Almacenamiento de Código:** Para el versionado y almacenamiento de código se utilizará **GitHub**, el cual es una plataforma de desarrollo colaborativo que permite a los programadores alojar, gestionar y controlar versiones de proyectos de software. Se basa en Git, un sistema de control de versiones distribuido que facilita el seguimiento de cambios en el código fuente a lo largo del tiempo, permitiendo que varios desarrolladores trabajen en paralelo sin perder el historial de modificaciones.



Herramientas a utilizar

- **Organización del proyecto:** Para la organización del proyecto se utilizará **Redmine**, el cual es una herramienta de gestión de proyectos y seguimiento de incidencias de código abierto. Ofrece funcionalidades como gestión de tareas, control de versiones, seguimiento de tiempo y gestión de recursos, lo que facilita la planificación y ejecución de proyectos. Además, permite la colaboración entre equipos, soporta múltiples proyectos y cuenta con un sistema de notificaciones y foros para discusiones. Es altamente personalizable e integrable con otras herramientas a través de plugins.
- **Gestión de proyecto:** Para la gestión del proyecto se utilizará **YouTrack** debido a que la metodología que se aplicará en este proyecto es Kanban, lo que significa que es necesario un tablero digital, es por esto que Trello es una opción ideal ya que permite aplicar la metodología kanban de una manera intuitiva, gestionar y asignar tareas de forma ágil y flexible, y además es gratuita.
- **Diseño y prototipado:** Para la fase de prototipado se utilizará **Figma**, esta permite crear prototipos visuales y mockups, la elección de este software fue determinada debido a que se puede utilizar de manera colaborativa con el equipo y además tiene un servicio gratuito.



Requerimientos Funcionales

Nº	Descripción	Prioridad
RF1	El sistema debe recibir la información proveniente de un programa recolector de datos.	Alta
RF2	El sistema debe ser capaz de mostrar la información recibida en tiempo real por medio de dispositivos móviles.	Alta
RF3	El sistema debe guardar la información histórica de la sesión de tiro.	Media
RF4	El sistema debe ser capaz de seleccionar al deportista que está llevando a cabo el entrenamiento.	Media
RF5	El sistema debe implementar un TTS para mencionar los resultados del tiro más reciente.	Alta
RF6	El sistema debe mostrar los resultados del tiro más reciente en una diana virtual.	Alta
RF7	Optimizar o Refactorizar el programa recolector de datos.	Baja



Requerimientos no Funcionales

Nº	Descripción
RnF1	El sistema debe presentar una interfaz amigable para el usuario.
RnF2	Los colores de la diana virtual deben representar los de una diana real
RnF3	El lugar donde cayo la flecha debe ser representado por un punto de color claramente visible.

Los requerimientos funcionales y no funcionales del proyecto fueron definidos acorde a los requerimientos de alto nivel los cuales fueron definidos con anterioridad, estos requerimientos fueron definidos junto con el cliente del proyecto con el fin de que sea acorde a las necesidades propuestas.

Importante: Se debe priorizar la fase de desarrollo de la aplicación para dispositivos móviles y tablets.



Características Principales

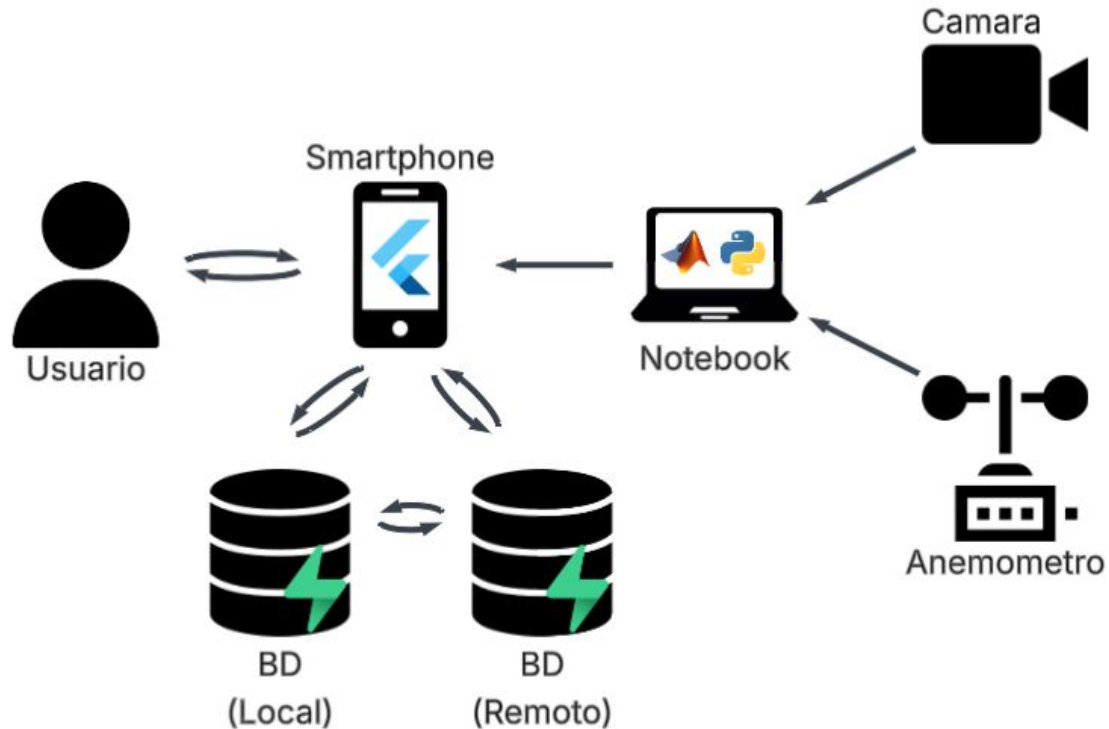
La definición de los requerimientos de alto nivel, requerimientos funcionales y no funcionales permitió estructurar al proyecto de la siguiente manera:

C1: Módulo de retroalimentación en tiempo real: Componente central de la aplicación orientado al atleta y al entrenador, tiene como objetivo principal presentar y anunciar el resultado de cada disparo de manera inmediata.

C2: Módulo de gestión de deportistas y sesiones: Componente que permite administrar la gestión del atleta en la sesión activa y almacenar su progreso histórico.

C3: Módulo de Interoperabilidad y Captura de datos: Componente que permite establecer una conexión entre el software existente y permite gestionar la recepción de datos.

Arquitectura del sistema



Modelo de Contexto

Los siguientes diagramas de contexto y nivel 1 describen los límites y subprocessos del sistema principal.

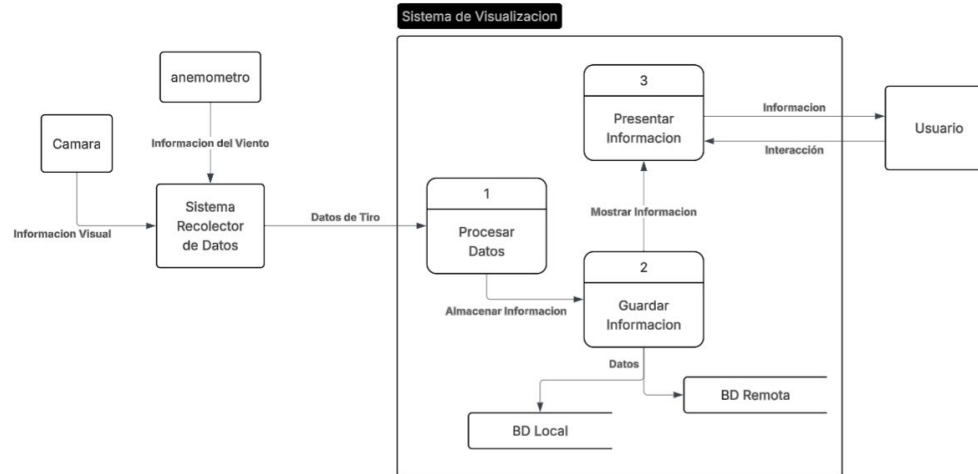


Diagrama de casos de uso

Con la información definida en los procesos anteriores, se pudo construir un diagrama de casos de uso en el cual determinaron 3 actores diferentes, sistema recolector de datos, deportista y entrenador.

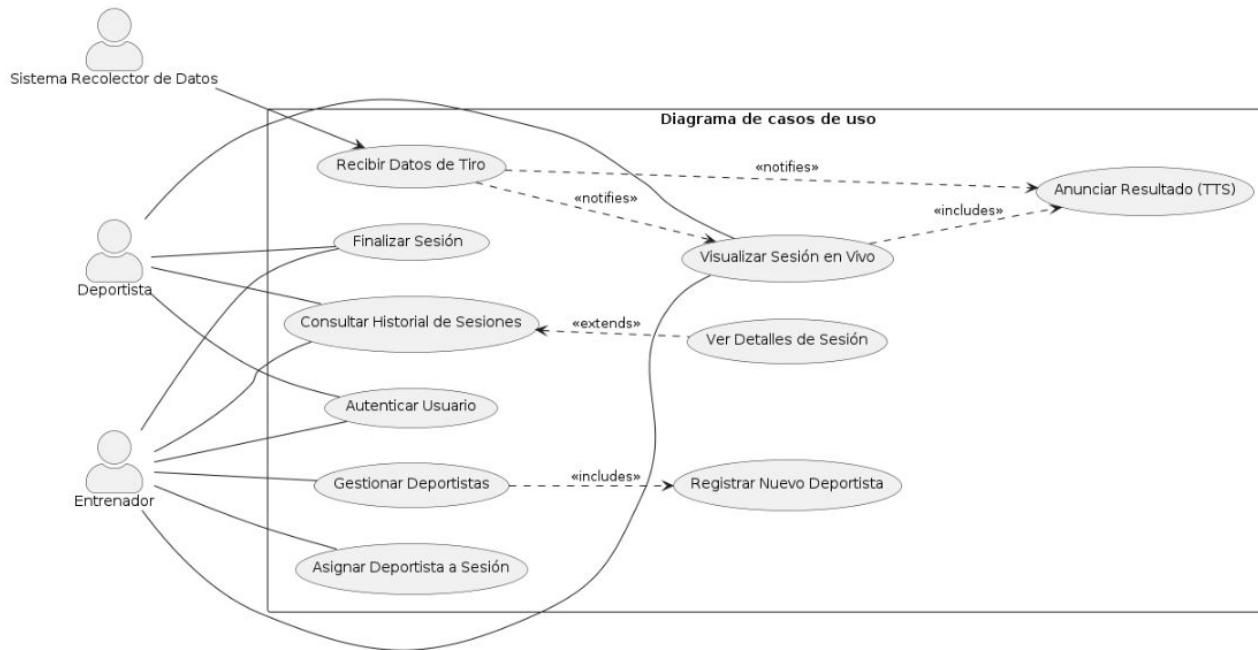
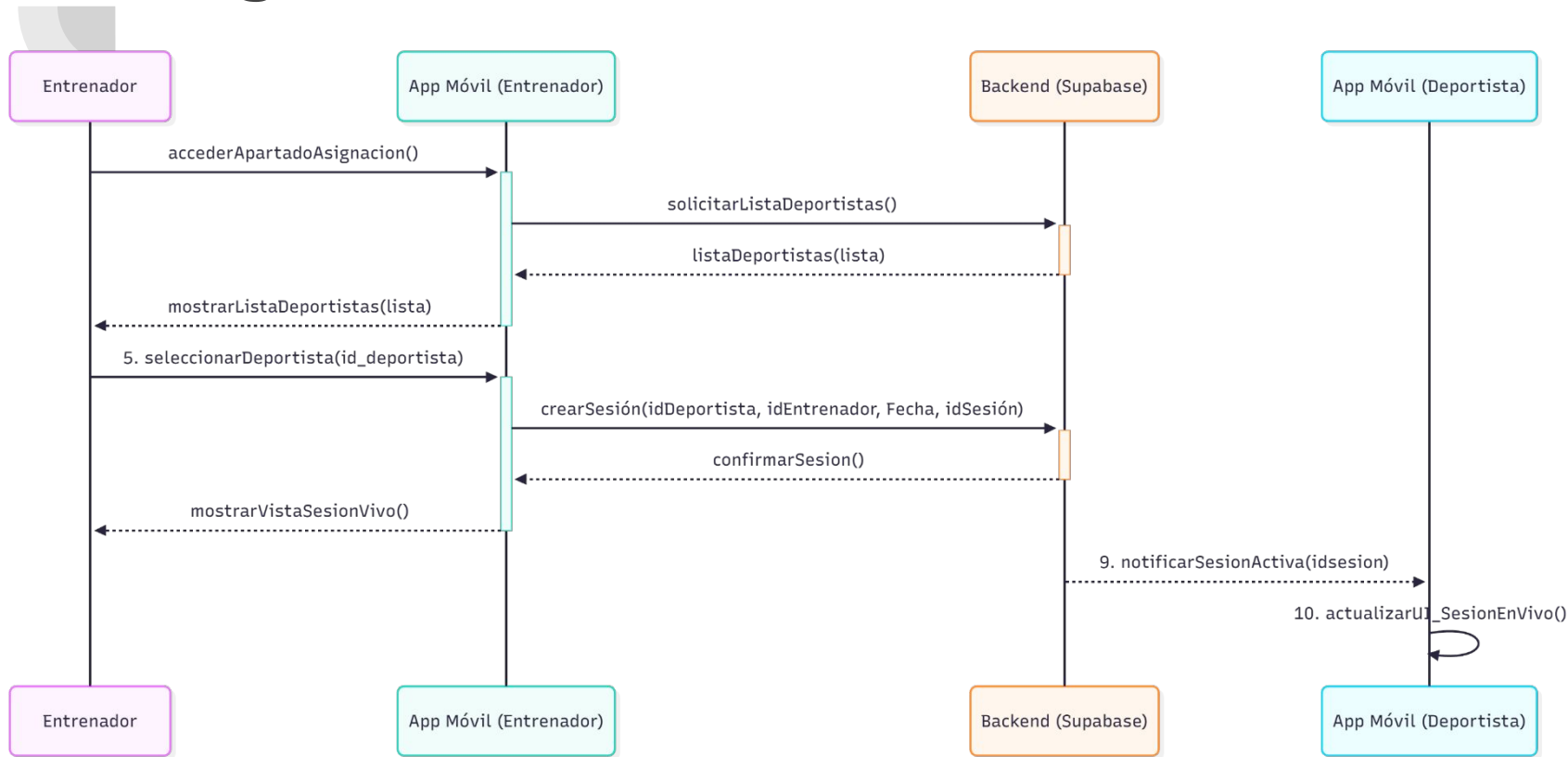


Diagrama de casos de uso

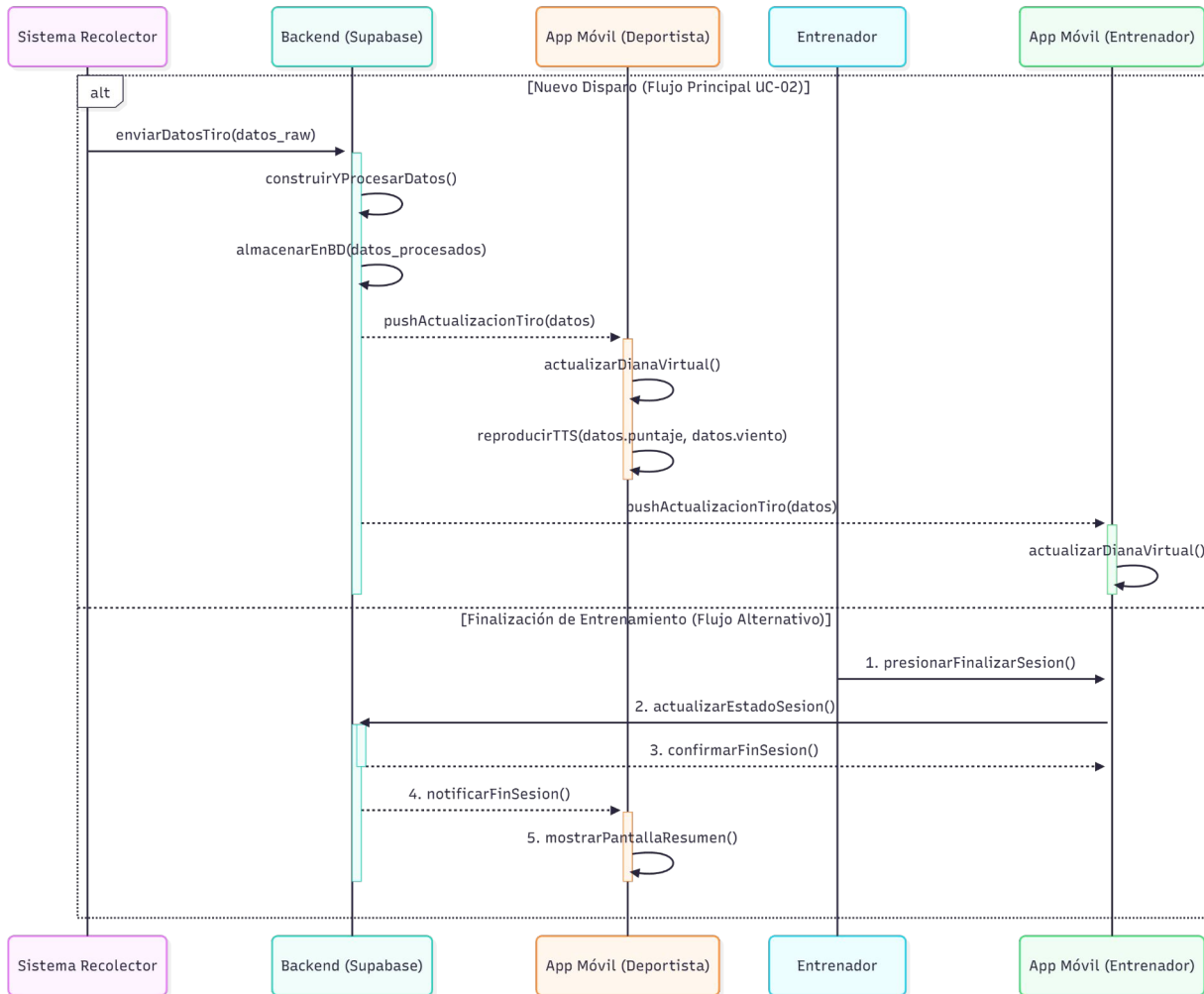
Campo	Descripción
ID	UC-01
Título	Asignar deportista a sesión
Actor(es)	Entrenador
Descripción	Permite al entrenador seleccionar un deportista registrado y asignarlo a una diana para iniciar un entrenamiento.
Disparador	El entrenador inicia sesión y accede al apartado de "Asignación".
Precondiciones	El entrenador debe estar autenticado en "Modo entrenador". Deben existir deportistas registrados en el sistema.
Flujo Principal (Éxito)	1. El Entrenador accede al apartado de asignación. 2. El Sistema muestra la lista de deportistas disponibles. 3. El Entrenador selecciona un deportista. 4. El Sistema actualiza el estado del deportista (a "activo" o "en sesión"). 5. El Sistema inicia el proceso de registro de tiros para ese deportista.
Poscondiciones	El deportista seleccionado está activo y su dispositivo móvil se actualiza para mostrar la sesión en vivo.
Flujos Alternativos	

Diagrama de interacción



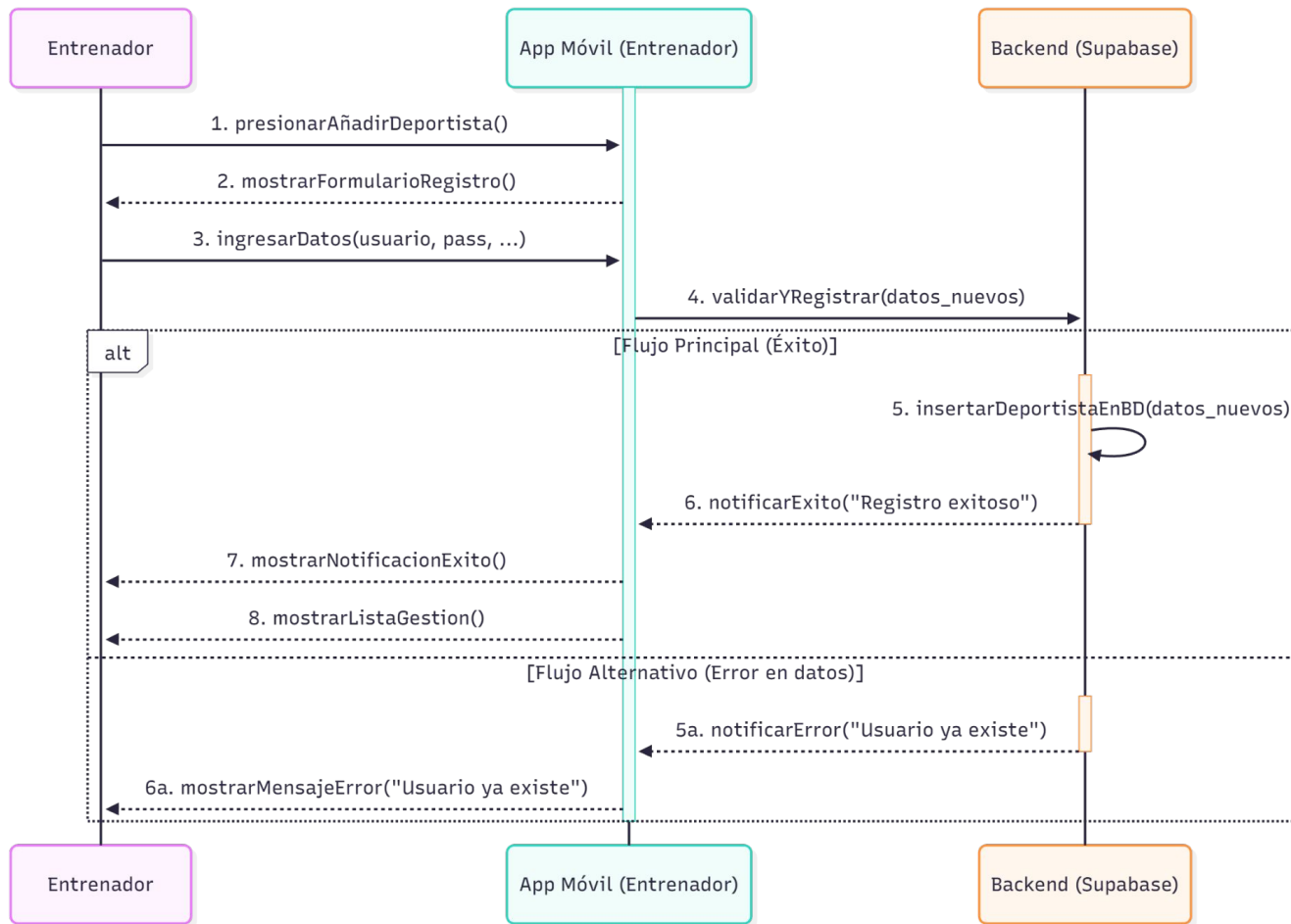


Campo	Descripción
ID	UC-02
Título	Visualizar sesión en vivo
Actor(es)	Deportista (inicia), Sistema Recolector de Datos (provee), Sistema (procesa).
Descripción	Captura, procesa, almacena y presenta la información de un tiro en tiempo real.
Disparador	El deportista dispara una flecha y esta impacta en la diana.
Precondiciones	Un deportista debe estar asignado a la sesión (Ver UC-01).
Flujo Principal	1. El deportista dispara la flecha. 2. El Sistema (Recolector) captura la información del tiro (posición, datos de sensores). 3. El Sistema recibe la información. 4. El Sistema construye la información. 5. El Sistema almacena la nueva información en la base de datos. 6. El Sistema refleja la información en el dispositivo móvil (actualiza diana virtual). 7. El Sistema reproduce la información del tiro (puntaje, viento, etc.) en el locutor TTS. 8. El sistema espera el siguiente disparo, a menos que el entrenador finalice el entrenamiento.
Poscondiciones	El tiro está almacenado y el deportista/entrenador han recibido feedback visual y auditivo.
Flujos Alternativos	Finalización de entrenamiento: 1. El Entrenador decide finalizar el entrenamiento. 2. El Entrenador actualiza el estado del deportista en el sistema. 3. El proceso de registro de tiros termina.

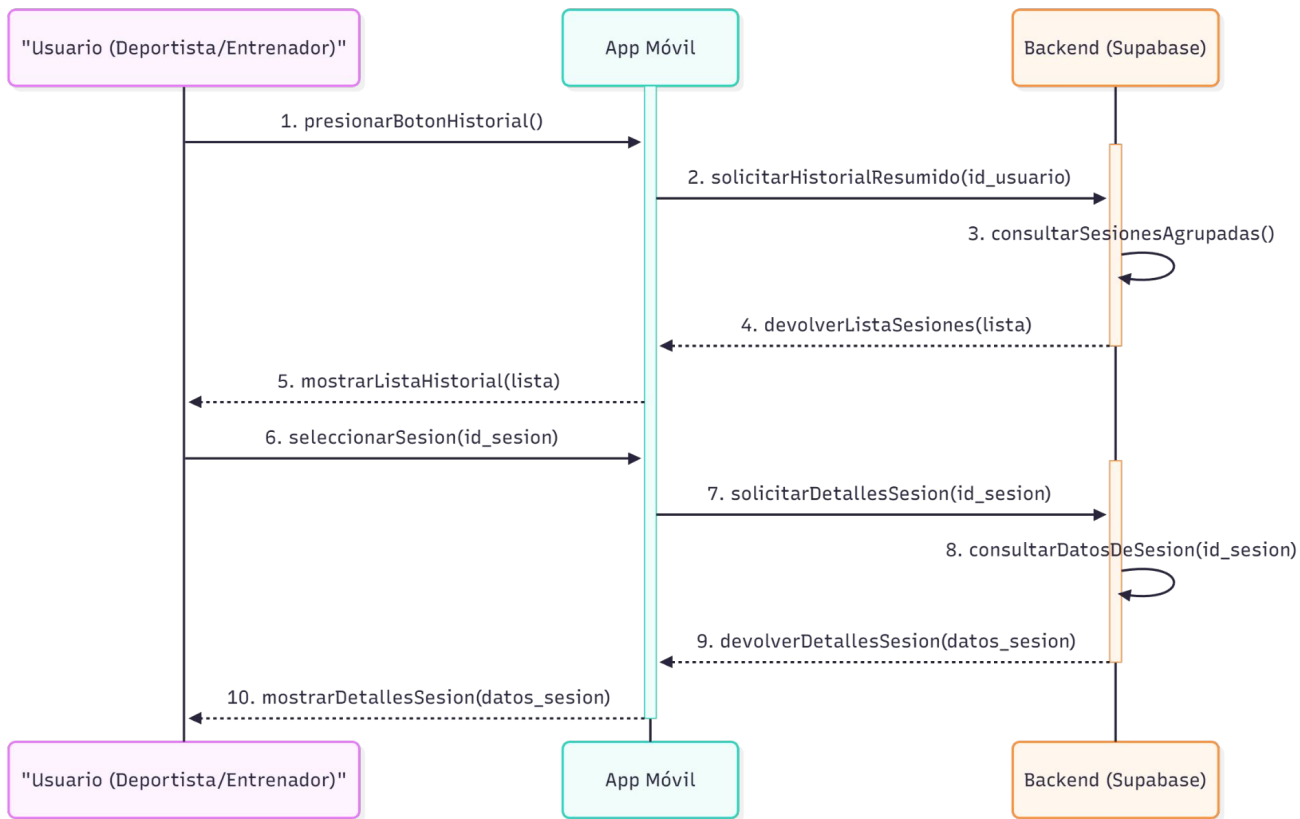




Campo	Descripción
ID	UC-03
Título	Registrar Nuevo Deportista
Actor(es)	Entrenador
Descripción	Permite al entrenador dar de alta a un nuevo deportista en el sistema.
Disparador	El entrenador accede al apartado de "Gestión" y selecciona "Añadir Deportista".
Precondiciones	El entrenador debe estar autenticado en "Modo entrenador".
Flujo Principal (Éxito)	1. El Entrenador accede al apartado de gestión. 2. El Sistema muestra un formulario para rellenar. 3. El Entrenador ingresa los datos del nuevo deportista (Usuario, Contraseña, etc.) y confirma. 4. El Sistema comprueba los datos ingresados. 5. El Sistema ingresa los datos del deportista en la base de datos. 6. El Sistema notifica al entrenador que el registro fue exitoso.
Poscondiciones	El nuevo deportista está registrado y puede ser asignado a sesiones.
Flujos Alternativos	Error en datos: 1. En el paso 4, el Sistema detecta un error en los datos (ej. usuario ya existe, campos vacíos). 2. El Sistema notifica el error al Entrenador. 3. El Entrenador recepciona el error y puede corregir los datos (vuelve al paso 3).

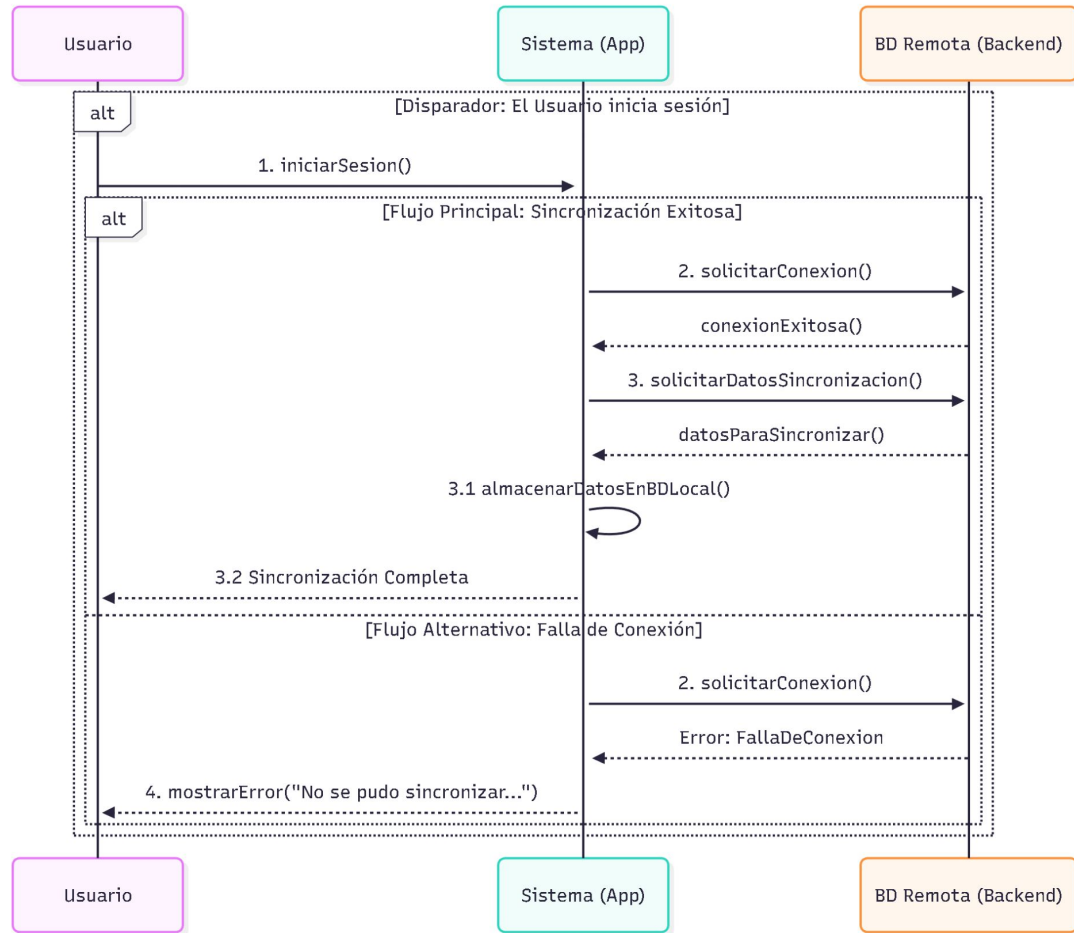


Campo	Descripción
ID	UC-04
Título	Consultar Historial de Sesiones.
Actor(es)	Deportista (principal), Entrenador (secundario).
Descripción	Permite al usuario (Deportista o Entrenador) revisar las sesiones de entrenamiento pasadas y ver los detalles de una sesión específica.
Disparador	El Deportista (o Entrenador) ingresa a la aplicación y selecciona el apartado de "Historial".
Precondiciones	El usuario debe estar autenticado. Deben existir sesiones históricas para ese deportista.
Flujo Principal (Éxito)	1. El Deportista ingresa al apartado de historial. 2. El Sistema consulta los datos del deportista en la BD y muestra un listado de sesiones resumidas (agrupadas por fecha). 3. El Deportista selecciona una sesión específica. 4. El Sistema solicita y consulta los datos de esa sesión en la BD. 5. El Sistema muestra los datos de la sesión específica (diana con todos los tiros, puntaje total y lista de flechas).
Poscondiciones	El usuario ha consultado su rendimiento en una sesión pasada.
Flujos Alternativos	





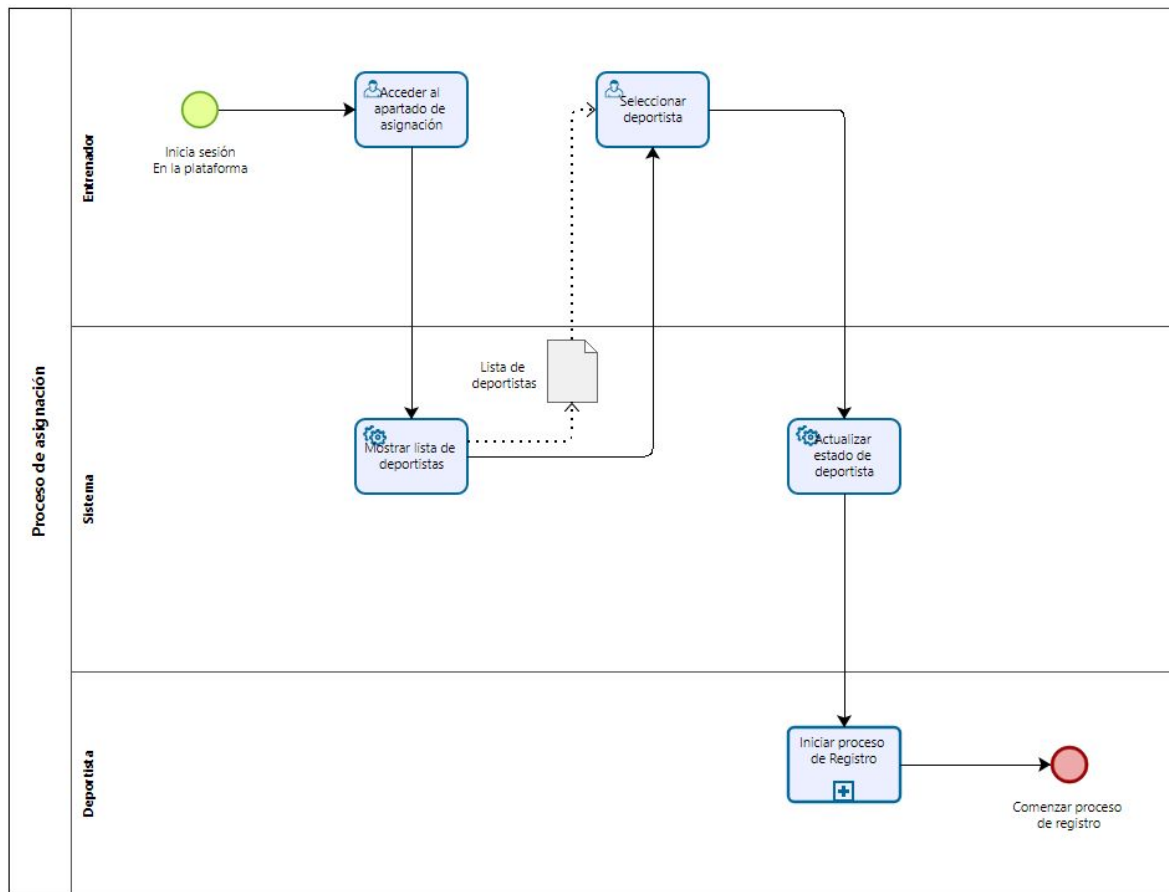
Campo	Descripción
ID	UC-05
Título	Sincronizar Bases de Datos
Actor(es)	Usuario (Entrenador/Deportista)
Descripción	Permite sincronizar la base de datos local con la base de datos remota.
Disparador	El Usuario inicia sesión. El Entrenador termina una sesión.
Precondiciones	El Usuario debe estar autenticado.
Flujo Principal (Éxito)	1. El Usuario accede al sistema. 2. El Sistema conecta con la base de datos Remota 3. El Sistema realiza la sincronización con la base de datos remota.
Poscondiciones	La información en la base de datos local y remota debe coincidir.
Flujos Alternativos	3. El Sistema detecta la imposibilidad de conectar a la base de datos remota. 4. El Sistema informa al usuario: “No se pudo sincronizar la base de datos”.

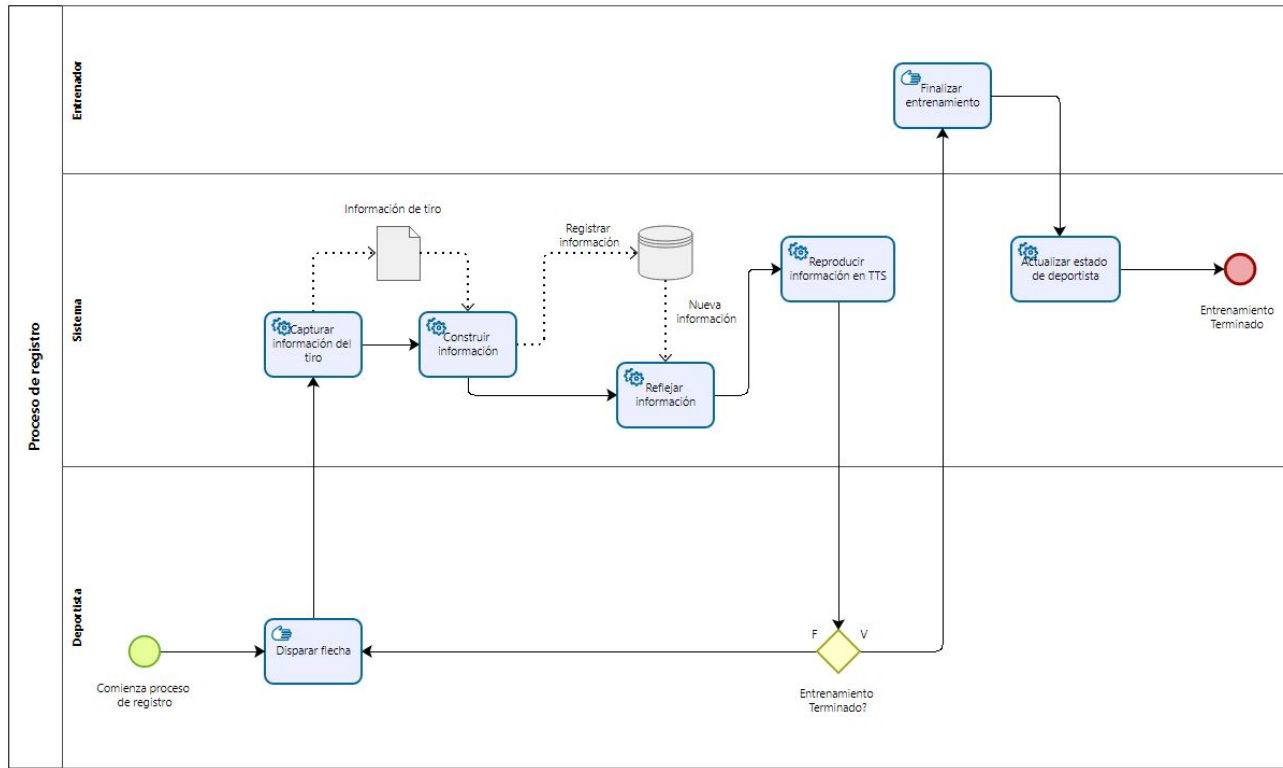




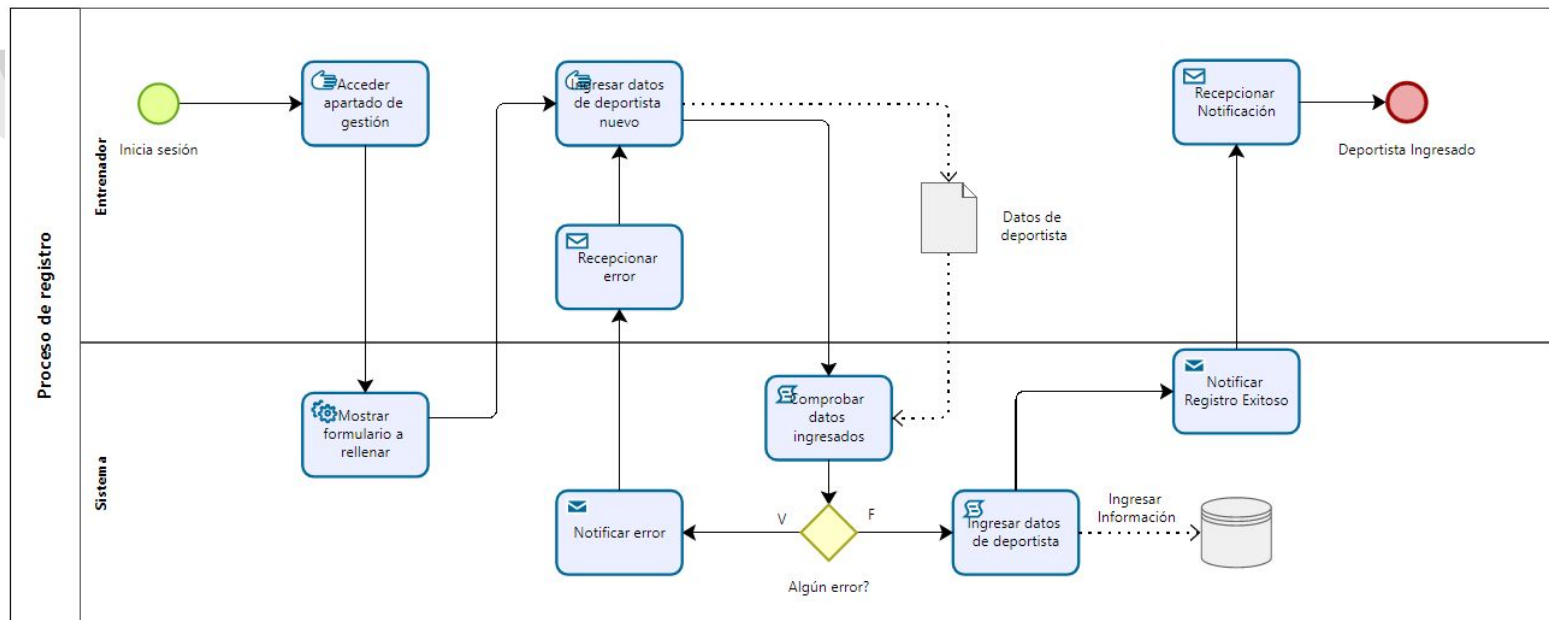
Modelos BPMN

En el primer proceso llamado “Proceso de Asignación”, el entrenador solicita al sistema una lista de deportistas registrados anteriormente y selecciona a uno para poder asignarle la diana que registrará las flechas, esto permitirá iniciar el subproceso llamado “Proceso de Registro”.

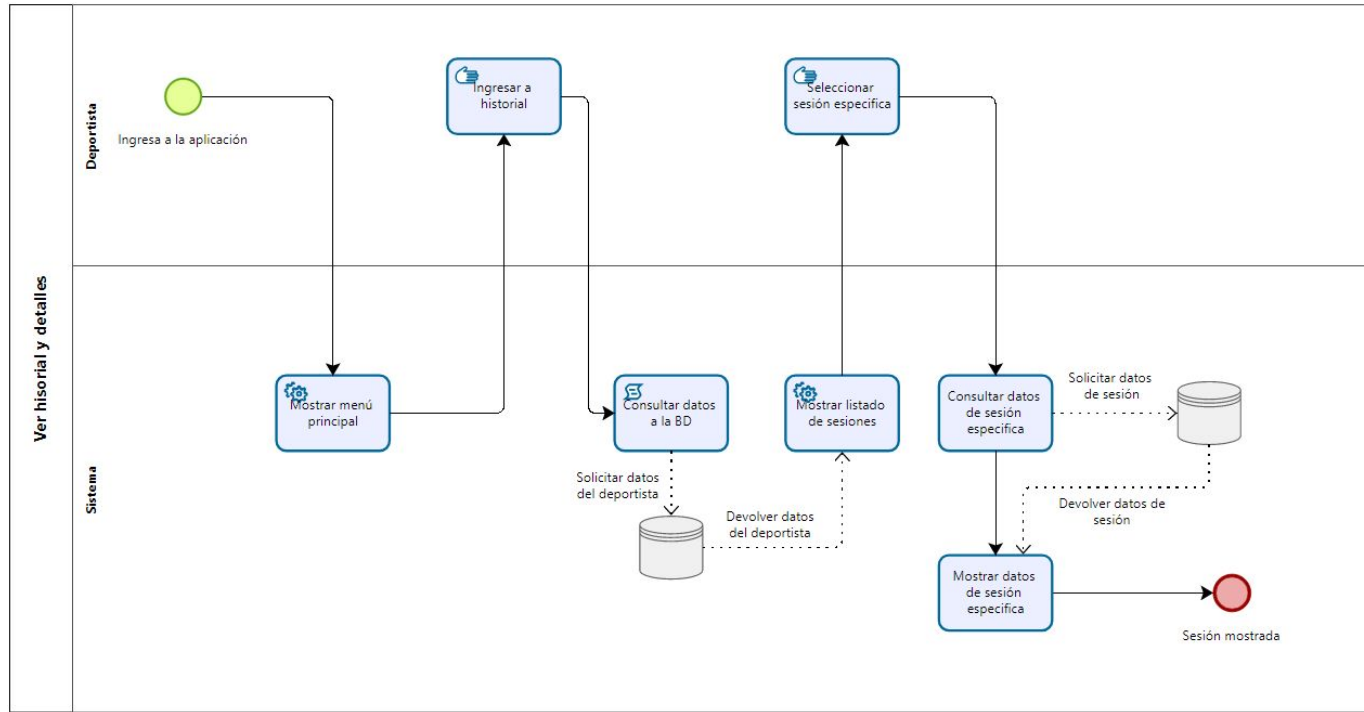




El proceso de registro permite registrar las flechas impactadas en la diana, para esto el deportista dispara la flecha y al impactar el sistema captura la información del tiro, esta información pasa por un proceso de construcción el cual permite generar un archivo legible para la base de datos, luego de construir esa información, es almacenada en la base de datos y se refleja automáticamente en el dispositivo móvil, reproduciendo la información recibida a través del altavoz. Para poder terminar el entrenamiento, el entrenador debe actualizar el estado en el sistema.



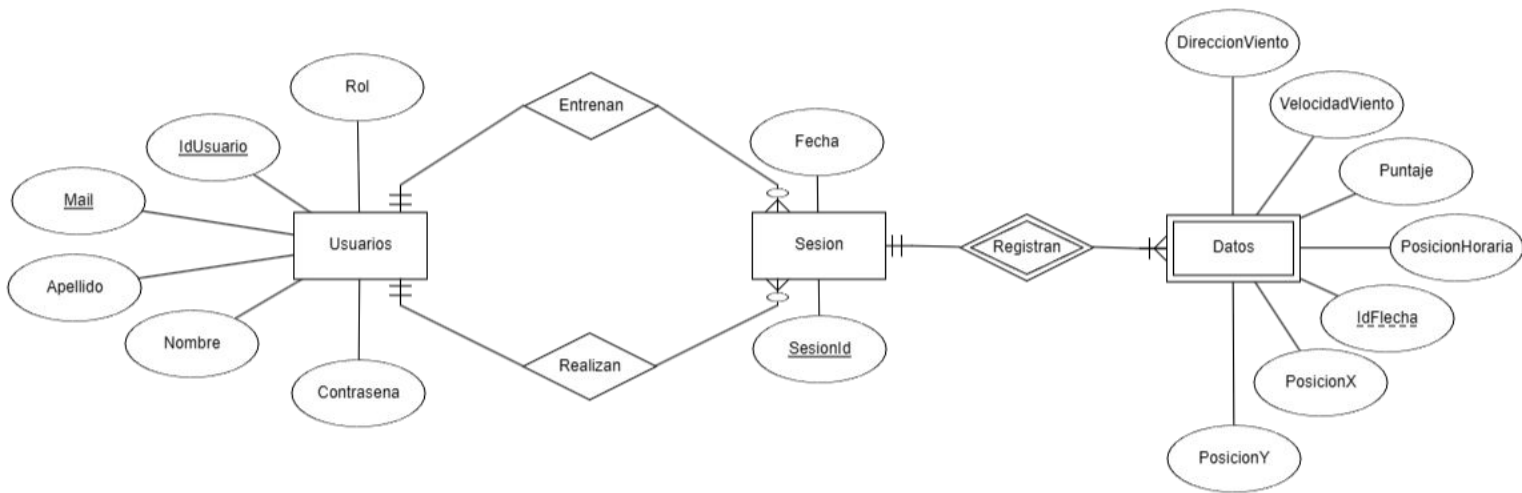
Permite que un entrenador pueda registrar un nuevo deportista, para esto el entrenador deberá acceder al apartado de gestión brindado por el sistema y se mostrará un formulario que deberá rellenar. Una vez rellenado, el sistema comprobará los datos ingresados y si es que hay algún error será notificado, de no haber ningún problema, el sistema ingresará los datos a la base de datos y notificará al entrenador que se ha registrado un nuevo deportista exitosamente.



El deportista podrá revisar su historial de sesiones realizadas junto a sus detalles, para esto se realizó el proceso de “Ver historial y detalles”, en donde el deportista ingresará a la aplicación y se dirigirá al apartado de historial, el sistema buscará toda la información del deportista en la base de datos y mostrará todas las sesiones resumidas. Para poder ver a detalles se deberá elegir una sesión y el sistema buscará toda la información de la sesión específica.

Modelado de datos

Se desarrolló una base de datos relacional para el proyecto, la cual consta de 3 entidades llamadas Usuarios, Sesión y Datos.





Modelado de datos

La entidad “Usuarios” es la que contiene todos los datos personales de los deportistas y los entrenadores, sus atributos son los siguientes:

Atributo	Descripción	Tipo de dato
IdUsuario	Id único de cada usuario.	INT
Mail	Email del usuario.	Varchar(30)
Nombre	Nombre del usuario.	Varchar(30)
Apellido	Apellido del usuario.	Varchar(30)
Contraseña	Contraseña del usuario.	Varchar(60)
Rol	Atributo discriminante que permite reconocer si un usuario es capaz de ingresar al modo entrenador.	INT



Modelado de datos

La entidad “Sesión” es la que contiene el dato principal de cada sesión:

Atributo	Descripción	Tipo de dato
Fecha	Fecha de cuando se creó la sesión	DATE
SesionId	Id de cada sesión.	INT
IdDeportista	Clave foránea de la tabla Usuarios para referenciar al deportista asignado a la sesión.	INT
IdEntrenador	Clave foránea de la tabla Usuarios que permite referenciar al entrenador encargado de la sesión.	INT

Modelado de datos

La entidad “Datos” contiene la información detallada de cada sesión, sus atributos son los siguientes:

Atributo	Descripción	Tipo de dato
IdFlecha	Id de cada flecha registrada.	INT
IdSesion	Clave foránea de la tabla de “sesión” que permite referenciar a qué sesión se registra cada flecha.	
DireccionViento	Fecha de cuando se creó la sesión.	DATE
VelocidadViento	Id de cada sesión.	INT
Puntaje	Atributo foráneo de la tabla Usuarios para referenciar al deportista asignado a la sesión.	INT
PosicionHoraria	Atributo foráneo de la tabla Usuarios que permite referenciar al entrenador encargado de la sesión.	INT
PosicionX	Posicion X de la flecha en el plano de la diana para graficar en imagen.	FLOAT
PosicionY	Posicion Y de la flecha en el plano de la diana	FLOAT



Conclusión

Esta primera fase estableció con éxito las bases fundamentales para el desarrollo del proyecto, se identificó el problema que es la interrupción del rendimiento que sufren los deportistas al tener que verificar manualmente sus puntajes y se acordó una solución de software y almacenamiento de datos, además, se logró proponer las funcionalidades principales del proyecto, así como mostrar el funcionamiento de los casos de uso y como estos cumplen con la principal problemática propuesta por el cliente.

Se hace necesario el proceso de investigación de Flutter y supabase para de esta forma dar comienzo al desarrollo de la solución de software a la brevedad. Esto con el objetivo de optimizar el tiempo y alcanzar la refactorización del programa recolector de datos el cual a de conectar el sistema a implementar.

Presentación de Avance “ArchData”

Autores: Angel Alarcón
Mathiu Orellana

Curso: Proyecto IV

Profesor: Diego Aracena

