TEC | Tecnológico de Costa Rica

Escuela de Ingeniería en Computadores
CE-3101 - Bases de Datos

# Documentación Técnica

**Profesor:**

Marco Rivera Meneses

**Estudiantes:**


- David Alberto Robles Vargas
- Carlos Andrés Mata Calderon
- Luis Felipe Vargas Jiménez
- Jose Ignacio Calderón Diaz
- Jose Carlos Umaña Rivera


II Semestre, 2023

# Repositorio de Github

https://github.com/Bases-Crew/StraviaTEC

# Plan de Trabajo en Github

**Para más detalles visite la pestaña Board y Roadmap en el enlace a continuación:**

https://github.com/orgs/Bases-Crew/projects/2

# Estructuras de datos desarrolladas

*Descripción General y API StraviaTEC*

Esta aplicación está dirigida a deportistas de diversas disciplinas, se apoya en una robusta arquitectura de datos y servicios para proporcionar una experiencia de usuario integral y eficiente. La implementación de dos bases de datos, SQL Server y MongoDB, se alinea con la diversidad de datos manejados, desde perfiles de usuario hasta detalladas estadísticas de actividades deportivas. La API, desarrollada en C#, juega un papel crucial al centralizar la funcionalidad, facilitando la comunicación entre la aplicación móvil y las bases de datos, y garantizando la coherencia y seguridad de los datos. Esta API, desplegada en la nube específicamente en azure, no solo agiliza el procesamiento de las solicitudes de los usuarios sino que también asegura una escalabilidad óptima del sistema, aprovechando las funcionalidades que tiene Microsoft para los estudiantes.

*Estructura de la Base de Datos y Funcionalidades*

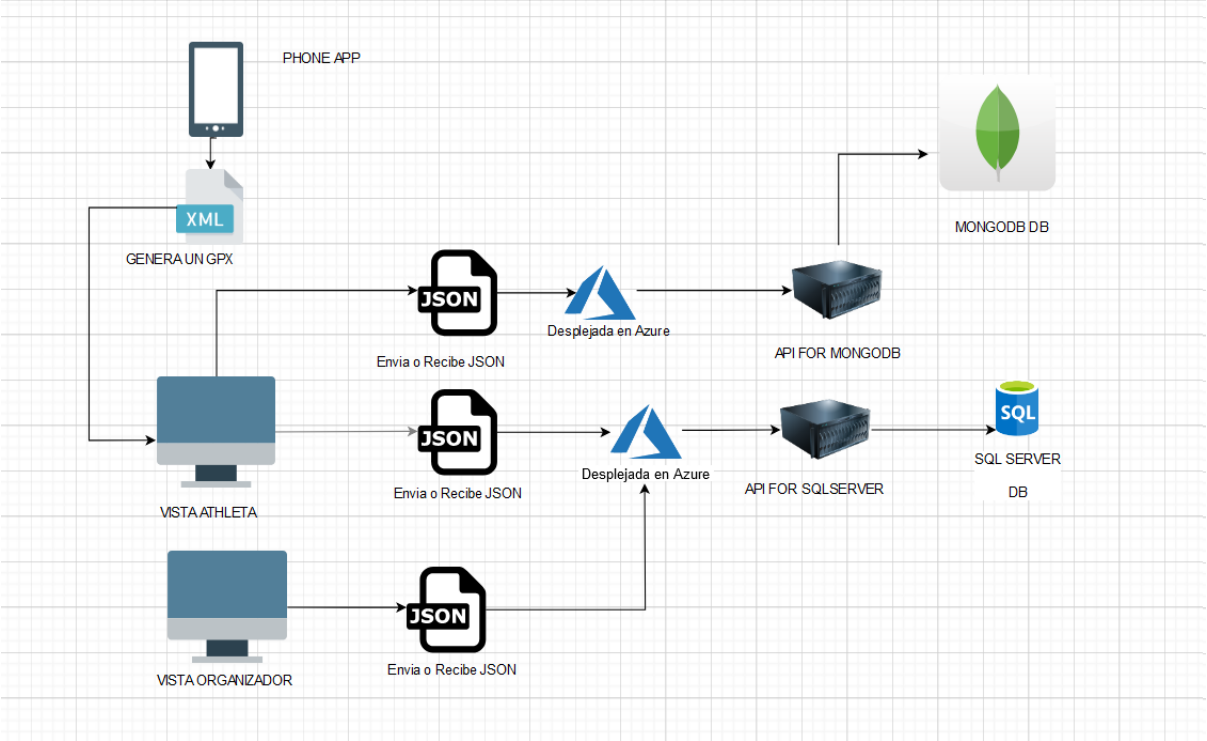
La base de datos en SQL Server maneja la estructura principal de los datos, incluyendo tablas para usuarios, actividades deportivas, eventos, inscripciones y grupos. Cada tabla está diseñada para capturar y relacionar información específica basándonos en nuestro mapa tanto conceptual como relacion y de la misma forma con el mapeo de las entidades, como los detalles personales de los usuarios en la tabla Users, o las métricas de las actividades deportivas en Activities. Además, se utiliza MongoDB para manejar aspectos más dinámicos y menos estructurados, como los comentarios en las actividades, aprovechando su naturaleza orientada a documentos. Procedimientos almacenados, vistas y triggers se implementan para manejar la lógica de negocios, asegurando así la integridad y el rendimiento óptimo de la base de datos.

*Aplicación Móvil y Web*

La aplicación móvil, desarrollada para proporcionar una interfaz amigable y funcional a los deportistas, permite el registro y seguimiento de actividades, así como la inscripción en eventos y retos. Integrada estrechamente con la API, la app móvil garantiza una experiencia de usuario fluida y actualizada. Por otro lado, la aplicación web, diseñada para los organizadores de eventos, facilita la gestión de eventos, la aprobación de inscripciones y la generación de reportes. Ambas aplicaciones, desarrolladas utilizando tecnologías modernas como Angular o React y Bootstrap, destacan por su diseño responsivo y su capacidad para adaptarse a diferentes dispositivos y tamaños de pantalla, asegurando así una accesibilidad y usabilidad óptimas.

## Descripción de la arquitectura

El almacenamiento de los datos se da con el uso de SQL Server y MongoDB, posteriormente se desarrolla un API usando las tecnologías de ASP.NET en C# que permiten el acceso y la modificación de los datos almacenados en la base de datos en la nube mediante stored procedures escritas utilizando el lenguaje SQL. A su vez, el API permite definir rutas web que las vistas web pueden utilizar para acceder a la información necesaria. Las vistas web hacen uso de Angular, mientras que la aplicación móvil hace uso de Flutter y Dart.




## Problemas Conocidos

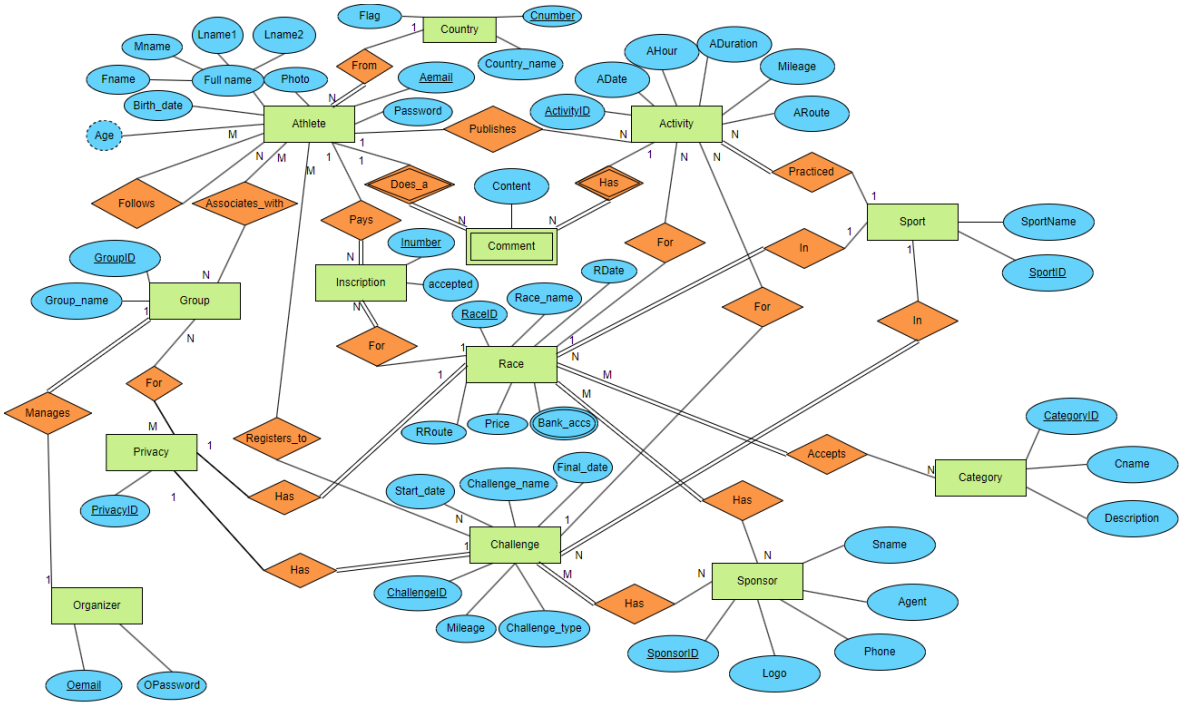
Durante el desarrollo se encontró un problema con el despliegue de más de un mapa al mismo tiempo. Es imposible desplegar más de 1 mapas al mismo tiempo. En StraviaTEC web se utilizó la librería leaflet y openbox con el fin de mostrar los archivos gpx. Sin embargo, cuando se intentó mostrar más de dos mapas este se sobreponía al anterior. Por lo tanto, utilizando estas librerías mencionadas es imposible desplegar 2 mapas al mismo tiempo.

## Problemas Encontrados

El primer problema encontrado es la superposición de registros con el login. Las componentes no se están desplegando correctamente por lo tanto se optó por

almacenarlas en un contenedor div con el fin de mostrar correctamente las componentes sin que se sobrepongan entre ellas.
Por otra parte, el segundo problema hay situaciones donde el login no pone la imagen de la persona. Muestra un espacio vacío en donde se debería ver la imagen del usuario. Se soluciona optando por utilizar otro tipo de objeto de tipo image en angular.

# Diagrama Conceptual




# Mapeo Relacional

## Entidades Fuertes:

| **Atleta** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Email** | **Contrase ña** | **Foto** | **Nombre 1** | **Nombre 2** | **Apellido 1** | **Apellido 2** | **Fecha de nacimie nto** |

| **Pais** | | |
|---|---|---|
| **Cnumber** | **Bandera** | **Nombre del pais** |

| Actividad | | | | | |
|---|---|---|---|---|---|
| ActividadID | fecha | hora | duración | distancia | ruta |

| Deporte | |
|---|---|
| id_Deporte | nombre del deporte |

| Grupo | |
|---|---|
| id_grupo | nombre del grupo |

| Inscripción | |
|---|---|
| No_inscripcion | aceptado |

| Carrera | | | | |
|---|---|---|---|---|
| id_carrera | Nombre | fecha | ruta | precio |

| Reto | | | | | |
|---|---|---|---|---|---|
| id_reto | fecha inicio | fecha fin | nombre | tipo de reto | distancia |

| Organizador | |
|---|---|
| Oemail | Contraseña |

| Patrocinador | | | | |
|---|---|---|---|---|
| id_patrocinador | Logo | numero telefono | agente | Sname |

| Categoría | | |
|---|---|---|
| id_categoria | Nombre | Descripción |

| Privacidad |
| --- |
| **id_privacidad** |

**Entidades Débiles:**

| Comentario |
| --- |
| **Contenido** |

**Asociaciones Binarias [1:1]:**

- **Grupo - Organizador**

Se añade como Llave foránea, la llave primaria de organizador en grupo.

| Grupo | | |
| --- | --- | --- |
| **id_grupo** | **nombre del grupo** | **Ouser** |

- **Reto - Privacidad**

Igualmente se añade como llave foránea, la llave primaria de privacidad en reto.

| Reto | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **id_reto** | **fecha inicio** | **fecha fin** | **nombre** | **tipo de reto** | **distancia** | **Pid** |

- **Carrera - Privacidad**

Se añade la llave primaria de privacidad en carrera.

| Carrera | | | | | |
| --- | --- | --- | --- | --- | --- |
| **id_carrera** | **Nombre** | **fecha** | **ruta** | **precio** | **Pid** |

**Asociaciones Binarias [1:N]:**

- **Atleta- inscripción**

Se añade la llave primaria de atleta en inscripción.

| **Inscripción** | | |
|---|---|---|
| **No_inscripcion** | **aceptado** | **Auser** |

- **Atleta- comentario**

Se añade la llave primaria de atleta en comentario.

| **Comentario** | |
|---|---|
| **Contenido** | **Auser** |

- **Atleta- actividad**

Se añade la llave primaria de atleta en actividad.

| **Actividad** | | | | | | |
|---|---|---|---|---|---|---|
| **ActividadID** | **fecha** | **hora** | **duración** | **distancia** | **ruta** | **Auser** |

- **Pais- Atleta**

Se añade la llave primaria de país en atleta.

| **Atleta** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Email** | **Contraseña** | **Foto** | **Nombre1** | **Nombre2** | **Apellido1** | **Apellido2** | **Fecha de nacimiento** | **Cno** |

- **Reto - Actividad**

Se añade la llave primaria de reto en actividad.

| **Reto** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **id_reto** | **fecha inicio** | **fecha fin** | **nombre** | **tipo de reto** | **distancia** | **Pid** | **Rid** |

- **Deporte - Reto**

Se añade la llave primaria de deporte en reto.

| **Reto** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **id_reto** | **fecha inicio** | **fecha fin** | **nombre** | **tipo de reto** | **distancia** | **Pid** | **Rid** | **Spid** |

- **Deporte- Carrera**

Se añade la llave primaria de deporte en carrera.

| **Vuelo** | | | | | | |
|---|---|---|---|---|---|---|
| **No_vuelo** | **Estado** | **Precio** | **Fecha** | **Pid** | **VOrigen** | **VDestino** |

- **Deporte- Actividad**

Se añade la llave primaria de deporte en actividad.

| **Actividad** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **ActividadID** | **fecha** | **hora** | **duración** | **distancia** | **ruta** | **Auser** | **Dno** | **Spid** |

- **Carrera - Actividad**

Se añade la llave primaria de carrera en actividad.

| **Actividad** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **ActividadID** | **fecha** | **hora** | **duración** | **distancia** | **ruta** | **Auser** | **Dno** | **Spid** | **Rid** |

- **Actividad - Comentario**

Se añade la llave primaria de actividad en comentario.

| **Comentario** | | |
|---|---|---|
| **Contenido** | **Auser** | **Actid** |

**Asociaciones Binarias [N:M]:**

- **Atleta- Atleta**

Se agregó una tabla de referencia cruzada con las llaves primarias de Atleta y Atleta para el caso de esta relación.

| **Follows** | |
|---|---|
| **Afollower** | **Afollows** |

- **Atleta- Grupo**

Se agregó una tabla de referencia cruzada con las llaves primarias de Atleta y grupo para el caso de esta relación.

| **Athlete_Gropu** | |
|---|---|
| **Auser** | **Gid** |

- **Atleta- Reto**

Se agregó una tabla de referencia cruzada con las llaves primarias de Atleta y reto para el caso de esta relación.

| **Athlete_challenge** | |
|---|---|
| **Auser** | **Challid** |

- **Grupo- Privacidad**

Se agregó una tabla de referencia cruzada con las llaves primarias de grupo y privacidad para el caso de esta relación.

| **Group_privacy** | |
|---|---|
| **Gid** | **Pid** |

- **Reto - patrocinador**

Se agregó una tabla de referencia cruzada con las llaves primarias de reto y patrocinador para el caso de esta relación.

| **Challenge_sponsor** | |
|---|---|
| **Challid** | **Spnid** |

- **Carrera - patrocinador**

Se agregó una tabla de referencia cruzada con las llaves primarias de carrera y patrocinador para el caso de esta relación.

| **Race_sponsor** | |
|---|---|
| **Rid** | **Spnid** |

- **Carrera - categoria**

Se agregó una tabla de referencia cruzada con las llaves primarias de carrera y categoría para el caso de esta relación.

| **Race_category** | |
|---|---|
| **Rid** | **Catid** |

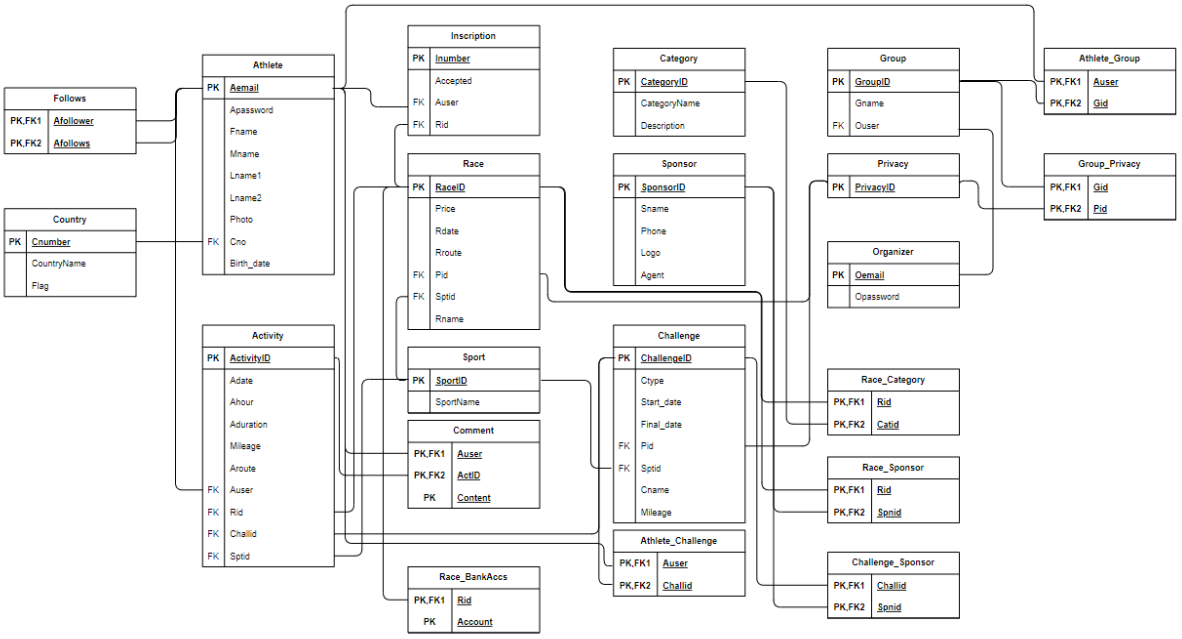
**Atributos Multivaluados:**

- **Cuentas bancarias**

Para tratar el caso de atributo multivaluado de las cuentas bancarias se creó la nueva tabla que asocia la llave primaria de carrera con el atributo cuenta.

| **Race_BankAccs** | |
|---|---|
| **Rid** | **Account** |

# Diagrama Relacional




# Conclusiones

1. La mayoría de los sistemas de reporte necesitan de NuGet packages disponibles en las bibliotecas de .NET de Visual Studio.
2. Es posible desarrollar aplicaciones Web bastante robustas con Angular ya que posee una gran cantidad de funcionalidades.
3. El manejador de bases de datos de Microsoft SQL Server es un excelente sistema en la que se pueden tener bases de datos extensas, y de una manera sencilla de entender
4. El entorno de desarrollo de Android para desarrollar una aplicación móvil permite crear aplicaciones móviles de manera sencilla y eficiente.
5. Es muy importante el entender el problema que se propone y así obtener un modelo conceptual y relacional de la base de datos, y de esta manera plasmar todas las necesidades antes de montar la base de datos.
6. El uso de la nube facilita la escalabilidad, seguridad y despliegue de aplicaciones web, API’s y bases de datos. Permitiendo a la aplicación crecer según sea necesario sin inversiones muy grandes.

## Recomendaciones

Como se refleja en la sección de problemas encontrados, el no definir y seguir una nomenclatura a la hora de nombrar o definir variables asociadas a cada columna de las tablas en la base de datos puede llevar al caos, por lo que se recomienda definir un orden y compartirlo con todos los desarrolladores para que la conexión entre las diferentes partes de la aplicación se dé con facilidad.

Se recomienda altamente el tomar en cuenta la mayor cantidad de factores posibles a la hora del mapeo de la base de datos, tanto del modelo conceptual como del relacional, para así evitar lo máximo posible la necesidad de hacer cambios a la hora de que la base de datos este creada, pues en caso de necesitar de añadir una nueva relación, o agregar o eliminar una columna de una tabla, puede causar muchos conflictos y retrasar el proyecto.

Es recomendable estimar el flujo de datos que va a estar presente en la aplicación y base de datos, ya que esto permite calcular precisamente cuánto será el costo por alojar la base de datos, API y aplicación web en la nube. Una mala estimación podría llevar a su empresa a pagar más dinero del necesario.

# Plan de Trabajo

**Metas:**

● Se espera un conjunto de vistas web desarrolladas en angular que permitan visualizar la información almacenada en la base de datos, así como su respectiva modificación o actualización.
● Permitir el acceso a algunas de las funciones de la aplicación a dispositivos móviles mediante una aplicación móvil.
● Manejar y crear archivos .gpx para la visualización de las rutas del deporte.
● Permitir el acceso y la modificación de la información de la base datos a través de un API desarrollado en C#.
● Almacenar los datos en Microsoft SQL Server y MongoDB en la nube. Mediante el servicio de Azure

**Roles:**

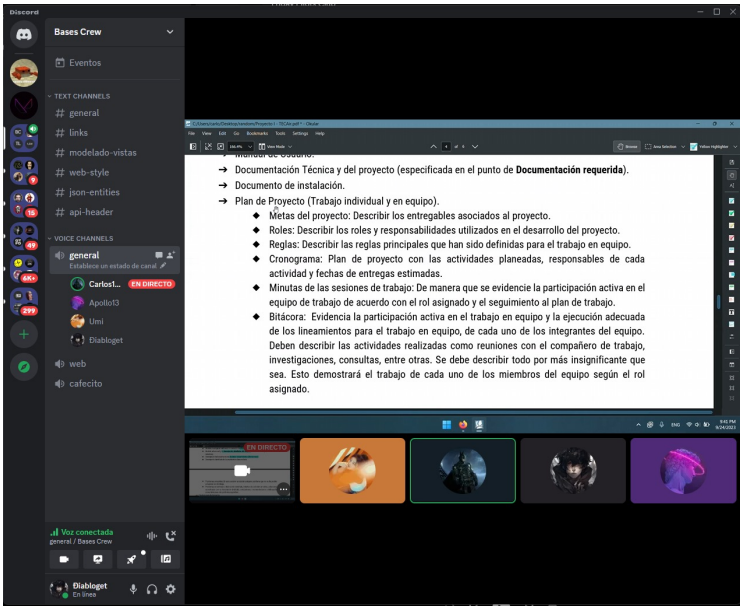
**Front End Manager:** Se encarga de llevar un control de las tareas a realizar y posibles adiciones que se pueden necesitar a la hora del uso de la aplicación y la web. Además, coordina con los BackEnd Developer y el Mobile Developer acerca de la conexión.
**Front End Designer:** Crea mediante código la interfaz e implementa las funcionalidades necesarias para una sencilla utilización de la página.
**Back End API Developer:** Responsable de crear y mantener la interfaz de programación de la API, permitiendo una buena comunicación entre las aplicaciones.
**Back End DataBase Designer:** Se encarga de diseñar, implementar y mantener la estructura de la base de datos donde se almacenan y gestionan los datos del sistema.
**Mobile Developer:** Se encarga de diseñar y crear la interfaz gráfica para mostrar al cliente. Además, implementa toda funcionalidad necesaria con el fin de que el app sea amigable con el usuario.
**Reglas:**

● Las reuniones sincrónicas se harán en su mayoría mediante discord, las cuales se emplearán para definir detalles específicos de las tareas a realizar. En el caso de ser posible o necesario, se harán reuniones presenciales.

● La comunicación restante se dará mediante un grupo de WhatsApp, en su mayoría para presentar actualizaciones en el desarrollo, y notificar de cualquier problema encontrado al resto del equipo.
● El control de versiones del código se hará mediante GitHub, con el fin de que todos los miembros del grupo tengan acceso al código actualizado por otros miembros.
● El código se realizará y comentará en el idioma inglés con el fin de mantener la coherencia del desarrollo.
● Los commits de GitHub se realizan usando la estandarización de Conventional Commits, por lo que también se comenta en inglés.
● Cada característica a realizar puede abrir un branch respectivo, el cual se debe de eliminar una vez la característica hace merge con el código fuente.
● Nunca comentar en el master en el GitHub.
● Comentar al estilo javadocs o utilizando XML.
● Nombrar funciones y variables lo más significativo y corto posible. (Al estilo clean code).

**Minutas:**

31 - 10 (Todos)



4 - 11
Explicación de GPX


6 - 11
Diseño de Apariencia de Página web

**Bitácora:**

**Carlos:**

- 19/11/2023 : Merge branch 'master' into web_athlete_home_user
- 19/11/2023 : feat: displays
- 18/11/2023 : Merge branch 'web_athlete_home_user'
- 18/11/2023 : Merge branch 'web_athlete_home_user' into sqlserver_api
- 18/11/2023 : build: asdf
- 18/11/2023 : Merge branch 'web_friends' into web_athlete_home_user
- 18/11/2023 : feat: add activities
- 18/11/2023 : feat: add routing complex
- 18/11/2023 : feat: change fotter
- 17/11/2023 : build: add packages
- 17/11/2023 : Merge branch 'web_athlete_home_user' into web_friends
- 17/11/2023 : feat: route example
- 17/11/2023 : fix: change size activity
- 17/11/2023 : feat: show actibity component
- 17/11/2023 : feat: add challenge activity renames
- 17/11/2023 : fix: repair init
- 17/11/2023 : fix: change Kayak
- 16/11/2023 : Merge branch 'sqlserver_api' into web_start_init
- 16/11/2023 : Merge branch 'sqlserver_api' into web_athlete_home_user
- 16/11/2023 : feat:
- 16/11/2023 : fix: register
- 16/11/2023 : chore: rename
- 16/11/2023 : fix: add again the footer
- 16/11/2023 : Merge branch 'master' of https://github.com/Bases-Crew/StraviaTEC
- 16/11/2023 : fix: add again the footer
- 16/11/2023 : feat: delete unnecessary footer
- 16/11/2023 : Merge branch 'web_athlete_home_user' into web_friends
- 15/11/2023 : Merge branch 'web_start_init' into sqlserver_api
- 15/11/2023 : Merge branch 'web_athlete_home_user' into web_start_init
- 15/11/2023 : feat: login
- 15/11/2023 : feat: add follow charactericts
- 15/11/2023 : Merge branch 'web_athlete_home_user' into sqlserver_api
- 15/11/2023 : fix: sp

- 15/11/2023 : feat: login
- 15/11/2023 : feat: add attributes user
- 15/11/2023 : chore: rename attributes user
- 15/11/2023 : feat: change link router header works
- 15/11/2023 : feat: add user var save data
- 15/11/2023 : fix: change header
- 13/11/2023 : chore: rename
- 13/11/2023 : chore: change icon
- 13/11/2023 : perf: delete display unnecessary
- 13/11/2023 : feat: settings
- 13/11/2023 : Merge branch 'web_start_init' into web_athlete_home_user
- 13/11/2023 : feat: footer component
- 13/11/2023 : feat: header inital
- 03/11/2023 : chore: format code
- 02/11/2023 : feat: example for the project
- 02/11/2023 : feat: boostrap and normalize
- 02/11/2023 : feat: create athlete view
- 28/10/2023 : feat: struture project
- 26/10/2023 : Initial commit

**Felipe:**

- 17/11/2023 : update: create the pdf participants
- 17/11/2023 : update code
- 17/11/2023 : update: age to the web
- 17/11/2023 : update
- 17/11/2023 : update logic secuence
- 17/11/2023 : update code
- 17/11/2023 : update
- 17/11/2023 : feat:update the component with the age
- 16/11/2023 : feat:update
- 14/11/2023 : feat:group
- 14/11/2023 : feat: trial creado
- 14/11/2023 : feat:update the data base of the angular
- 10/11/2023 : feat: create the profile menu
- 04/11/2023 : feat: Create and update the componente challenges
- 04/11/2023 : feat: create the component challenges

- 03/11/2023 : feat: create the component friends

**Jose:**

- 18/11/2023 : feat: categories procedure
- 18/11/2023 : feat: getcategories and categories in population script
- 18/11/2023 : feat: added new sponsors in population script
- 18/11/2023 : feat: newchallenge, newrace, newgroup, getgroups, getsponsors
- 17/11/2023 : feat: some challenge gets and posts, athlete delete
- 16/11/2023 : feat: get sports, post activities and follow/unfollow athletes
- 15/11/2023 : feat: get athlete by aemail
- 15/11/2023 : feat: get athelte by email and athlete login
- 15/11/2023 : fix: now images can be sent to the db
- 14/11/2023 : feat: DB scripts
- 14/11/2023 : feat: sql server first commit

**David**

- 16/11/2023 : feat: register fix and connection with api
- 14/11/2023 : feat: register form
- 07/11/2023 : feat: login page
- 05/11/2023 : fix: updated the icon image and some colors
- 05/11/2023 : feat: added some components to simulate an initial web page

**Ignacio**

- 31-10: Reunión para planificación de plan de trabajo.
- 1-11: Investigación de GPS en aplicaciones móviles.
- 5-11: Interfaz gráfica de la aplicación móvil.
- 6-11: Se agrega el mapa de google maps y se obtiene la localización.
- 10-11: investigación de librerías de Flutter para archivo .gpx.
- 13-11: Se agrega el cálculo de distancia y cálculo de velocidad.
- 14-11: Creación de archivos .gpx completa.
- 15-11: Corrección de la ruta de almacenamiento de la ruta .gpx.
- 18-11: documentación interna del código.
- 18/11/2023 : Merge branch 'gpsMobile'
- 18/11/2023 : documentation
- 15/11/2023 : fix external file management
- 15/11/2023 : change file permisions
- 14/11/2023 : generate gpx
- 13/11/2023 : route line
- 13/11/2023 : distance and velocity
- 06/11/2023 : mapa funciona, timer no

- 05/11/2023 : primera version gps

## Bibliografía (Enlaces)

**GPS Visualizer**
**xml | Dart Package (pub.dev)**
**dart - Flutter Countdown Timer - Stack Overflow**