



Documentación Técnica v1.0

StraviaTEC, Documentación Técnica.

Instituto Tecnológico de Costa Rica,

Área Académica de Ingeniería en Computadores.

Esquivel Jonathan, Ortiz Angelo, Solís Iván, Venegas Agustín.

Versión 1.0, 27 páginas.

Hoja de Control

Organismo	Instituto Tecnológico de Costa Rica	
Proyecto	StraviaTEC - Proyecto #1, Curso de Bases de Datos	
Entregable	Documentación Técnica	
Autor	ce-itcr	
Aprobado por	ce-itcr	Fecha de Aprobación: 21/11/2020
		N° Total de Páginas: 26

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
001	Versión Inicial	Angelo Ortiz Vega	21/11/2020

EQUIPO DE DESARROLLO

Nombre	Correo Electrónico	Rol
Esquivel Sanchez Jonathan	jesquivelsa@gmail.com	Desarrollador / Backend / DB
Ortiz Vega Angelo	angelo2898@gmail.com	Desarrollador / Frontend /DB
Solís Ávila Iván	isolis@solavcr.com	Desarrollador / Backend / DB
Venegas Vega Agustín	joseagustinvneveg@gmail.com	Desarrollador / Frontend / DB

Tabla de Contenidos

Hoja de Control	2
Tabla de Contenidos	3
Presentación	5
Objetivos	5
Objetivo General	5
Objetivos Específicos	5
Descripción del Problema	5
Tabla de Definiciones y Acrónimos	6
Tipos de Usuario	6
Historias de Usuario	7
Vista Deportista	7
Aplicación Móvil	7
Vista Organizador	8
Requerimientos Funcionales de Software	8
Módulo Login/Logout y Registro de Usuarios : DEP,ORG,WEB	8
Módulo Dashboard : DEP,WEB	9
Módulo Búsqueda y Seguimiento de Atletas : DEP,WEB	9
Módulo Registro de Actividades y Carreras : DEP,WEB	10
Módulo Registro de Actividades : DEP, MOV	10
Módulo Gestión de Carreras : ORG, WEB	10
Módulo Reportes de Aplicación : ORG, WEB	11
Módulo Administración de Usuarios : ORG, WEB	11
Diagrama de Funcionalidades	12
Requerimientos de Bases de Datos	13
Valores por defecto	13
Modelo Conceptual	15
Modelo Relacional	16
Requerimientos no Funcionales	17
Facilidad de Uso	17
Desarrollo	17
Descripción de las estructuras de datos desarrolladas	18
Descripción detallada de la arquitectura desarrollada	19

Problemas conocidos	20
Problemas encontrados	20
Minutas de Sesiones de Trabajo	21
Bitácora de Trabajo	22
Conclusiones	25
Recomendaciones	25
Bibliografía	26

Presentación

Objetivos

Objetivo General

- Desarrollar una aplicación que permita manejar la descripción del caso expuesto.

Objetivos Específicos

- Aplicar conceptos del modelo conceptual y relacional.
- Crear una Base de Datos en Postgresql para que permita el almacenamiento de los datos.
- Crear un servicio API para que centralice la funcionalidad.
- Crear una página web para que exponga la funcionalidad al usuario.
- Crear una aplicación móvil utilizando SQL Lite como base de datos empotrada.
- Usar herramientas como Angular, Bootstrap, HTML5, CSS3, Entity Framework y Reporting Services o Cristal Reports.
- Crear un documento de instalación que permita el despliegue correcto de la aplicación.
- Crear un plan de proyecto que evidencie la participación en el equipo de trabajo asignado de acuerdo a su rol y la ejecución de lineamientos para el trabajo en equipos.

Descripción del Problema

Para nadie es un secreto que durante la pandemia y las medidas sanitarias impuestas por los gobiernos para salvaguardar la vida, han hecho que los deportes individuales (atletismo, ciclismo, senderismo, etc..) tomen mucho auge.

Por tal motivo se expone StraviaTEC como una opción a todos los deportistas (principiantes, intermedios, elites) para que lleven un registro de todas sus actividades de tal forma que se les permita compartir sus sesiones hasta mejorar sus tiempos y entrenamientos. StraviaTEC se compone principalmente de dos grandes vistas:

1. **Vista Deportista:** Le permite a los deportistas visualizar de forma agradable las sesiones realizados por él y personas a quién sigue,
2. **Vista Organizador:** Le permite a los organizadores de eventos agregar eventos, incluir/aprobar tablas de posiciones, aceptar inscripciones/solicitudes de afiliación.

Tabla de Definiciones y Acrónimos

Acrónimo	Significado
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
RBD	Requerimiento de Base de Datos
DEP	Deportista
ORG	Organizador
WEB	Tipo de Aplicación Web
MOV	Tipo de Aplicación Móvil
ACT	Actividad

Tipos de Usuario

Tipo de Usuario	Descripción
Usuario Deportista	El usuario deportista tiene acceso únicamente a funciones de deportista, en esta plataforma permitirá a todos los deportistas visualizar de una forma agradable/amigable las sesiones realizadas.
Usuario Organizador	El usuario organizador tiene acceso únicamente a funciones de organizador, en esta plataforma permitirá a todos los organizadores de eventos, incluir/aprobar tablas de posiciones, aceptar inscripciones/solicitudes de afiliación.

Historias de Usuario

Vista Deportista

1. Como Deportista quiero que se me permita la creación de una cuenta donde incluya los siguientes datos: nombre y apellidos, fecha de nacimiento, edad actual considerando fecha de nacimiento, nacionalidad, foto, nombre de usuario y contraseña.
2. Como Deportista quiero que se me permita la modificación y eliminación de mi cuenta.
3. Como Deportista quiero que se me permita el ingreso a mi cuenta con mi nombre de usuario y contraseña.
4. Como Deportista quiero que al ingresar a mi cuenta visualice todas las actividades registradas de todos mis amigos en orden ascendente.
5. Como Deportista quiero que se me facilite la búsqueda de otros atletas a los cuales deseo seguir.
6. Como Deportista quiero que se me permita agregar otros deportistas a lista de amigos.
7. Como Deportista quiero que se me permita cargar actividades de ejercicio donde se conozca: fecha, hora de la actividad, duración, tipo de actividad, kilometraje, recorrido en un archivo .gpx, completitud de un reto o carrera.
8. Como Deportista quiero que se me permita inscribirme a una carrera, donde deba anexar el recibo de pago del costo de la carrera.
9. Como Deportista quiero que se me permita inscribirme a un reto,
10. Como Deportista quiero que se me permita asociarme a grupos de deportistas.
11. Como Deportista quiero que se me permita ver carreras y retos inscritos.
12. Como Deportista quiero ver tablas de posiciones de una carrera ordenada de menor a mayor.
13. Como Deportista quiero ver objetivos de retos y avances que tengo así mismo informar cuántos días hacen falta para terminar el reto.

Aplicación Móvil

1. Como Deportista quiero realizar el registro de actividades dentro de una aplicación móvil.
2. Como Deportista quiero que se muestre el tiempo que se lleva desde que se inicia una actividad junto con los kilómetros recorridos.
3. Como Deportista deseo que la ruta gps se almacene.

Vista Organizador

1. Como Organizador quiero que se me permita gestionar carreras.
2. Como Organizador quiero que se permita aceptar inscripciones.
3. Como Organizador quiero que se me permita gestionar retos.
4. Como Organizador quiero que se me permita gestionar grupos.
5. Como Organizador quiero observar el reporte de participantes por carrera y posiciones.

Requerimientos Funcionales de Software

Módulo Login/Logout y Registro de Usuarios : DEP,ORG,WEB

RF001 - Iniciar Sesión

- **Descripción:** Con una cuenta creada el deportista y/o organizador podrá ingresar a la aplicación una vez autenticadas sus credenciales.
- **Restricciones:** El nombre de usuario y contraseña deben coincidir.

RF002 - Cerrar Sesión

- **Descripción:** El usuario acepta cerrar sesión, se guardan datos y la aplicación se redirige a Inicio de Sesión.
- **Restricciones:** El usuario debe estar dentro de la aplicación.

RF003 - Registrarse como Deportista

- **Descripción:** El usuario de tipo deportista crea su cuenta para poder hacer inicio de sesión y acceder a sus funcionalidades.
- **Restricciones:**
 - El usuario debe proveer los siguientes datos: nombre y apellidos, fecha de nacimiento, edad actual, nacionalidad, foto, nombre de usuario y contraseña.
 - Deben ser texto: nombre, apellidos, fecha de nacimiento, nacionalidad, nombre de usuario, y contraseña.
 - Debe ser número entero, edad actual.

RF004 - Registrarse como Organizador

- **Descripción:** El usuario de tipo organizador crea su cuenta para poder hacer inicio de sesión y acceder a sus funcionalidades.
- **Restricciones:**
 - El usuario debe proveer los siguientes datos: nombre y apellidos, fecha de nacimiento, edad actual, nacionalidad, foto, nombre de usuario y contraseña.
 - Deben ser texto: nombre, apellidos, fecha de nacimiento, nacionalidad, nombre de usuario, y contraseña.
 - Debe ser número entero, edad actual.

Módulo Dashboard : DEP,WEB

RF005 - Mostrar datos de usuario

- **Descripción:** Se muestra en la pantalla inicial el nombre de usuario, con sus respectivos seguidores, seguidos y número de actividades.
- **Restricciones:** N/A

RF006 - Mostrar listado de mis actividades en orden ascendente

- **Descripción:** Debajo de la sección “*Mostrar datos de usuario*”, se muestra las actividades realizadas por el usuario en orden ascendente.
- **Restricciones:** N/A

RF007 - Filtrar mis actividades por deporte

- **Descripción:** Filtra las actividades que realiza cada deportista por deporte en orden ascendente.
- **Restricciones:** Para aplicar los filtros a las actividades la lista no debe estar vacía.

RF008 - Mostrar actividades de deportistas a quienes sigo por orden ascendente

- **Descripción:** En esta sección se muestran las actividades que han realizado deportistas a quienes sigo, se muestra a manera de resumen: tipo de actividad, mapa de donde realizó la actividad y los kilómetros recorridos.
- **Restricciones:** Antes de mostrarse en pantalla, se debe verificar que ninguno de los espacios antes mencionados estén vacíos.

RF009 - Seleccionar puntos específicos dentro del mapa

- **Descripción:** Se le permite al usuario presionar sobre un punto específico dentro del mapa para ver características específicas como velocidad y altitud.
- **Restricciones:** Antes de mostrarse en pantalla, se debe verificar que ninguno de los espacios antes mencionados estén vacíos.

Módulo Búsqueda y Seguimiento de Atletas : DEP,WEB

RF010 - Buscar atletas

- **Descripción:** El sistema debe permitir al usuario buscar usuarios mediante un filtro.
- **Restricciones:** N/A

RF011 - Seguir atletas

- **Descripción:** El sistema debe permitir al usuario, una vez buscado un atleta, seguirlo.
- **Restricciones:** N/A

RF012 - Asociarse a un grupo

- **Descripción:** El sistema debe permitirle al usuario ingresar a un grupo seleccionado.

- **Restricciones:** El usuario debe ser aceptado en el grupo para ingresar.

Módulo Registro de Actividades y Carreras : DEP, WEB

RF013 - Registrar actividad

- **Descripción:** El sistema debe permitir al usuario cargar la información de una actividad de ejercicio
- **Restricciones:** Debe tener fecha y hora de la actividad, duración, tipo de actividad, kilometraje, el recorrido (en archivo .gpx) y debe indicarse si es la completitud de un reto o carrera previamente inscrito.

RF014 - Inscribirse a una carrera

- **Descripción:** El sistema debe permitirle a un usuario inscribirse en una carrera
- **Restricciones:** Se debe asegurar de anexar el recibo de pago del costo de la carrera

RF015 - Inscribirse en un reto

- **Descripción:** El sistema debe permitirle al usuario inscribirse a los retos disponibles.
- **Restricciones:** Para quedar inscrito debe ser aceptado.

RF016 - Ver retos y carreras inscritos

- **Descripción:** La aplicación debe brindarle al usuario la posibilidad de ver una tabla con los retos y carreras inscritos.
- **Restricciones:** N/A

Módulo Registro de Actividades : DEP, MOV

RF017 - Iniciar actividad

- **Descripción:** El sistema debe permitir el inicio de una actividad
- **Restricciones:** Sólo las actividades listadas (Correr, Nadar, Ciclismo, Senderismo, Kayak, Caminata)

RF018 - Mostrar mapa de actividad en curso

- **Descripción:** El sistema deberá mostrar el mapa con el recorrido de la actividad en curso
- **Restricciones:** N/A

Módulo Gestión de Carreras : ORG, WEB

RF019 - Crear, modificar y eliminar carreras

- **Descripción:** En esta vista el usuario debe tener la posibilidad de crear, modificar y eliminar carreras.
- **Restricciones:** N/A

RF020 - Aceptar Inscripciones

- **Descripción:** El sistema debe permitir a un organizador validar el recibo de una inscripción y validar si la cuenta cuenta con los fondos requeridos, de ser así debe aceptar la inscripción y de no ser así deberá denegar la inscripción
- **Restricciones:** Validar que tenga los fondos requeridos en la cuenta

RF021 - Crear, modificar y eliminar retos

- **Descripción:** En esta vista el usuario debe tener la posibilidad de crear, modificar y eliminar retos.
- **Restricciones:** N/A

Módulo Reportes de Aplicación : ORG, WEB**RF022 - Reporte de participantes por carrera**

- **Descripción:** La aplicación debe permitir al usuario ver todos los participantes inscritos en una carrera en esta vista.
- **Restricciones:** Debe estar ordenada en base a los siguientes atributos: Nombre, apellidos, edad, categoría.

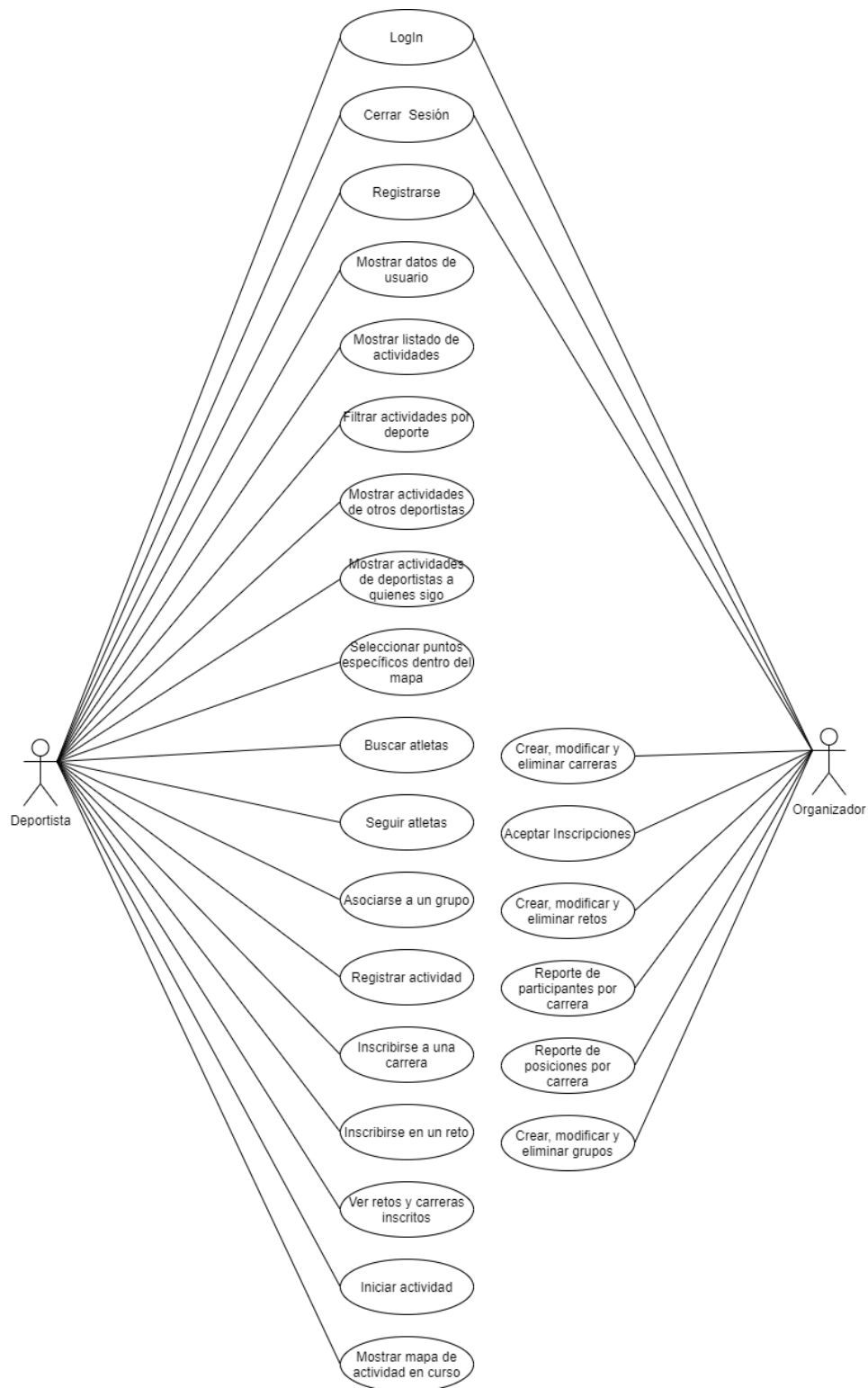
RF023 - Reporte de posiciones por carrera

- **Descripción:** En caso de haber una carrera, se debe poder ver la tabla de posiciones ordenado
- **Restricciones:** La tabla debe ser ordenada basada en las posiciones de menor a mayor

Módulo Administración de Usuarios : ORG, WEB**RF024 - Crear, modificar y eliminar grupos**

- **Descripción:** El sistema debe permitir crear, modificar y eliminar grupos, del grupo se conoce: nombre del grupo, y un administrador que por defecto es quién crea el grupo.
- **Restricciones:** Un grupo siempre debe poseer un administrador asociado.

Diagrama de Funcionalidades



Requerimientos de Bases de Datos

Valores por defecto

Debe existir un listado de actividades por defecto, se conocen las siguientes:

Nombre de Actividad	Id de Actividad
Atletismo	ACT001
Natación	ACT002
Ciclismo	ACT003
Senderismo	ACT004
Kayak	ACT005
Caminata	ACT006

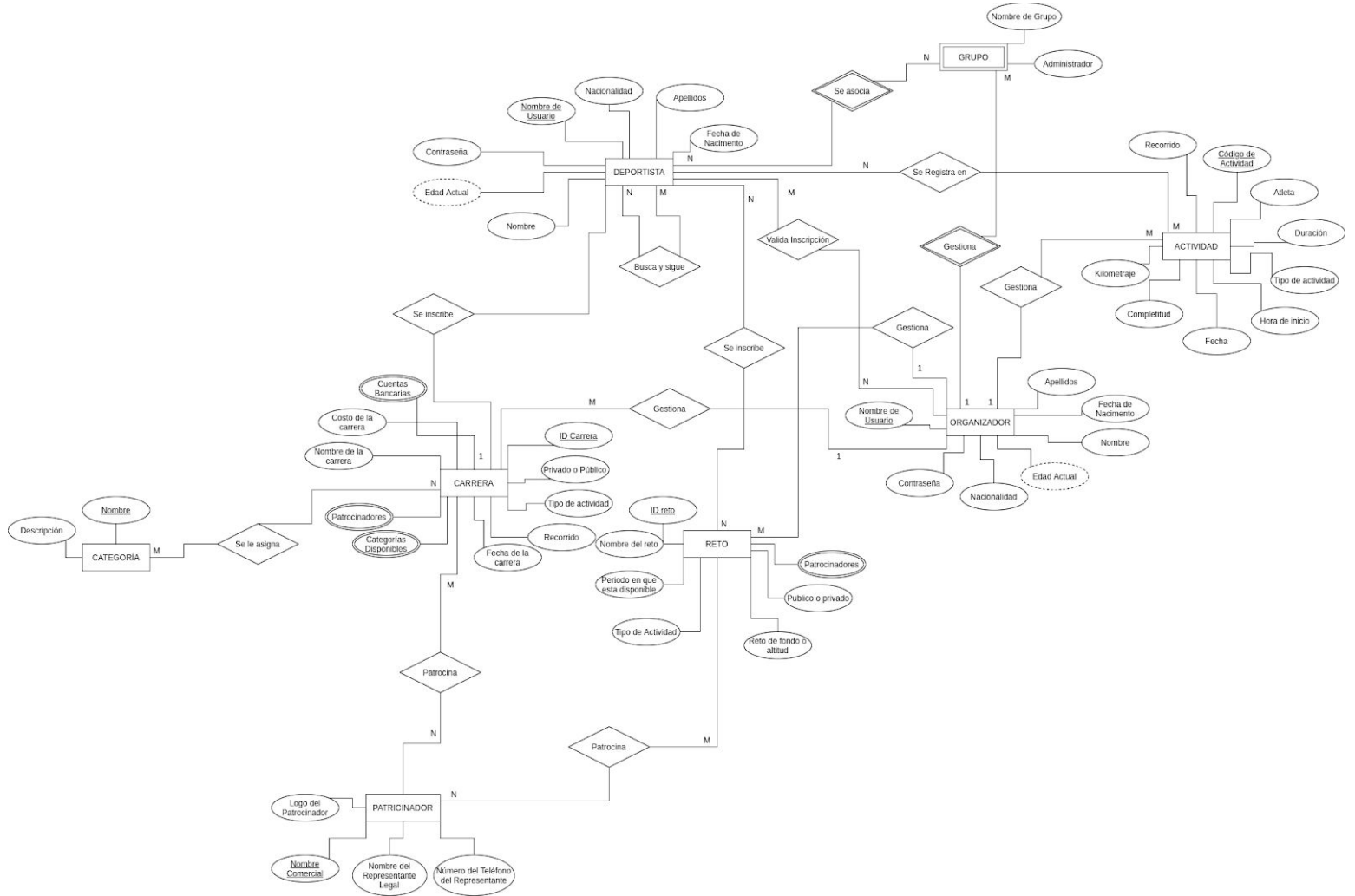
Debe existir un listado de categorías por defecto se conoce nombre y descripción. Las categorías que se tienen:

Nombre de Categoría	Descripción
Junior	Menos de 15 años
Sub-23	De 15 a 23 años
Open	De 24 a 30 años
Elite	Cualquiera que quiera inscribirse
Master A	De 30 a 40 años
Master B	De 41 a 50 años
Master C	Más de 51 años

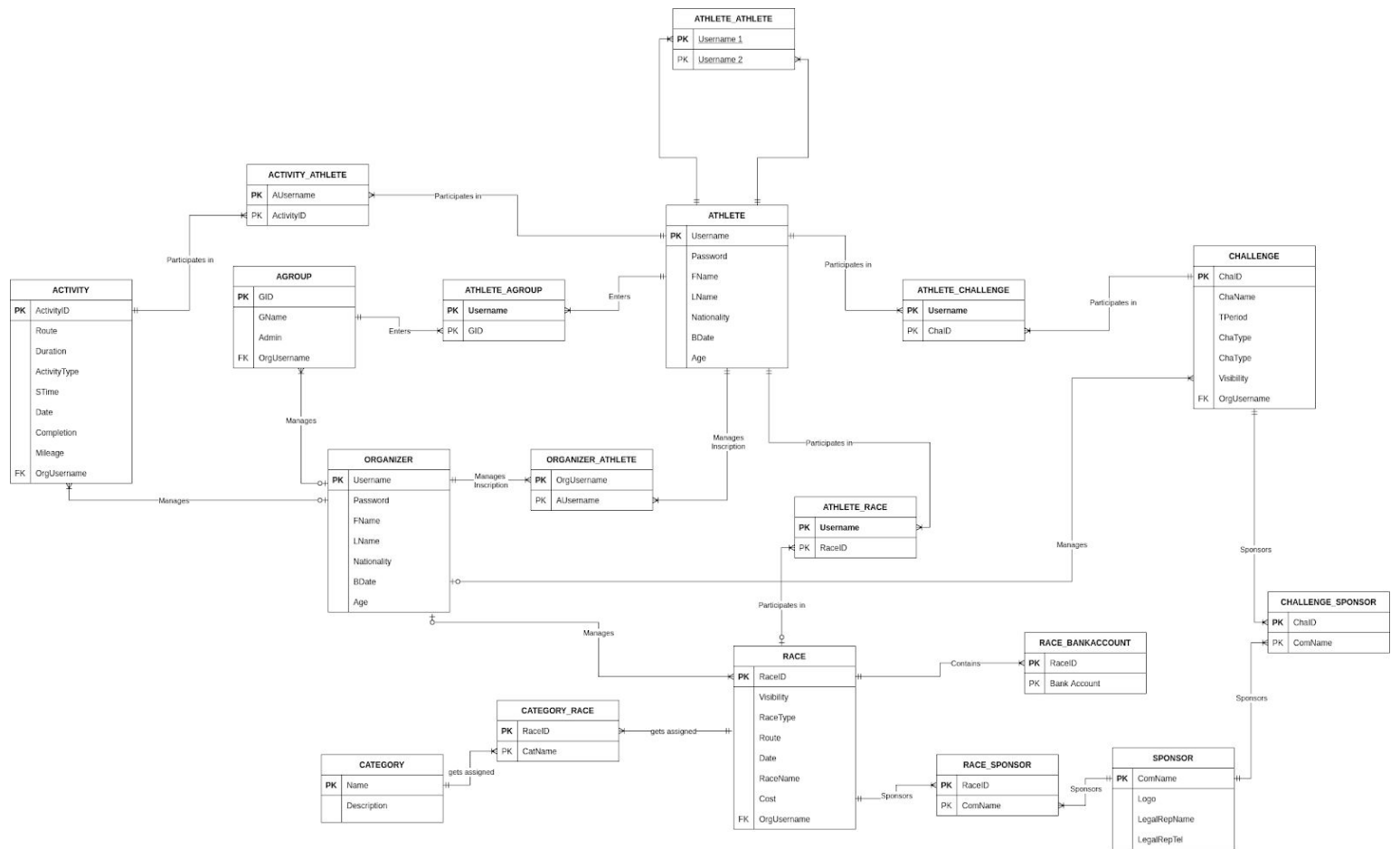
Debe existir un listado de patrocinadores por defecto de los cuales se conoce un nombre comercial, el nombre del representante legal, número de teléfono del representante y el logo del patrocinador. Se tienen los siguientes:

Nombre Comercial	Nombre Representante	Número de Teléfono	Logo
StraviaTEC	Angelo Ortiz Vega	+506 88998899	STRAVIA 
The North Face	Jonathan Esquivel Sanchez	+506 88778877	THE NORTH FACE 
Asics	Agustín Venegas Vega	+506 88448844	
Red Bull	Iván Solís Ávila	+506 88338833	
Huawei Mobile	John Doe	+506 88222288	
Strava	Jane Doe	+506 8877993322	STRAVA TM

Modelo Conceptual



Modelo Relacional



Requerimientos no Funcionales

Facilidad de Uso

- La aplicación tendrá como idioma base el español.
- La aplicación al entregarse debe contener con un manual de usuario, documento de instalación, documentación técnica y plan de proyecto.

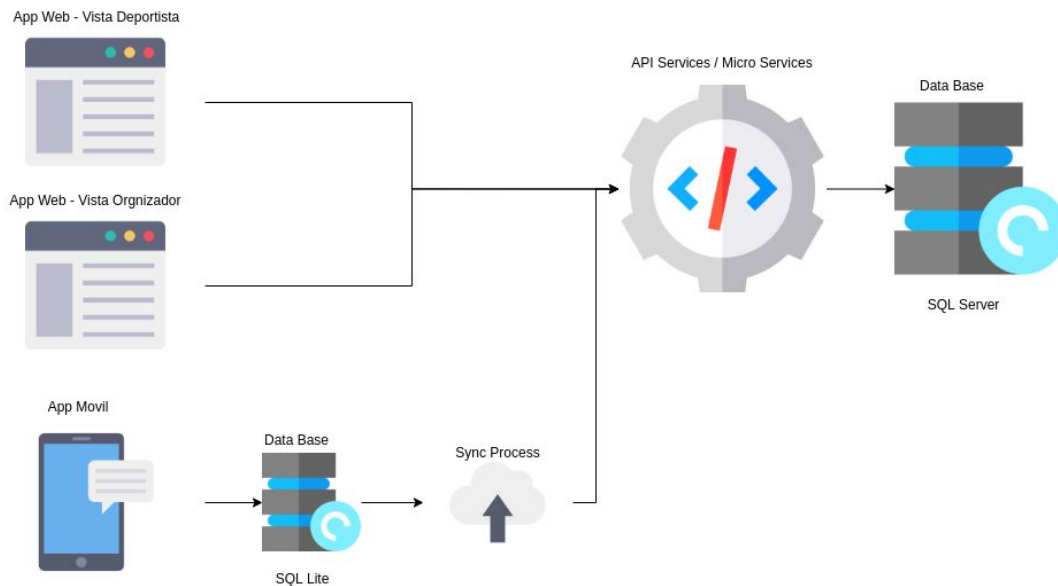
Desarrollo

- El sistema debe ser una aplicación web (utilizando Angular, Bootstrap, HTML5, CSS)
- La App Móvil debe ser desarrollada utilizando Android Studio y SQL Lite como base de datos.
- La base de datos debe estar en Postgresql.
- No se permite el uso de Store Procedures, Vistas o Triggers. Los scripts de Base de Datos deben implementarse en la capa de datos.
- La capa de servicios debe estar desarrollada en C# y debe ser desplegada en IIS.
- La aplicación móvil debe ser desarrollada con Ionic utilizando Angular como base.

Descripción de las estructuras de datos desarrolladas

- En la tabla de athlete se crea f_name y l_name para guardar el nombre y apellido del usuario, estos no pueden ser valores nulos ya que toda persona debe tenerlos para identificarse. Nationality, b_date y age se agregan como atributos simples de la tabla estos sí pueden ser nulos y age es un atributo derivado del valor b_date, estos son principalmente para funciones de perfil de la persona. Por último, se le agrega username como llave primaria de athlete y u_password para poder hacer login a este perfil y prof_img para personalizar el usuario. Para organizer se hace exactamente lo mismo.
- Para la tabla activity, se agrega un atributo activity_id que va funcionar como llave primaria de la tabla. También contiene los atributos s_time, activity_date, activity_type y milage para que se pueda ver en el muro de la persona que hizo, cuando lo hizo y cuánto duró haciéndolo. También se agrega una representación visual de la ruta realizada por medio de route, en esta se va guardar un link en donde se pueda acceder al recorrido. Por último, se agrega un atributo completion para asignar esta actividad como parte de un reto o simplemente como una actividad aparte.
- Para la tabla race se agrega un race_id como llave primaria de la tabla. Se agregan los atributos race_name, race_type, race_cost, race_date y route para darle información a los usuarios acerca de la carrera. Visibility se utiliza para decir si una carrera es pública o exclusiva para un grupo. Por último, se agrega org_username como llave foranea de la relación organizer y race.
- La tabla race_bankaccount se utiliza como atributo multivaluado de la tabla race, para colocar una o más valores de cuentas bancarias para el pago de la carrera. Este contiene únicamente race_id y bank_account como llaves primarias y race_id es la llave foranea que relación.
- La tabla category tiene la llave primaria cat_name como llave primaria de la relación y se agrega un atributo description para agregarle una descripción a la categoría
- La tabla challenge con un atributo cha_id que funciona como llave primaria de la tabla. Se le agrega cha_name, cha_type y t_perido para darle información al usuario acerca del reto y al igual que carrera se le agrega un visibility para hacer el reto público o exclusivo para un grupo. Por último se agrega org_username como llave foranea de la realción entre reto y organizer.
- La última tabla es sponsor, esta tiene comp_name como llave primaria, esto se decidió ya que no pueden haber dos compañías con el mismo nombre, después se le agregó logo solo para información de la compañía y por último legal_repname y legal_reptel para la información del representante legal de la compañía.

Descripción detallada de la arquitectura desarrollada



StraviaTec se compone de una aplicación web con dos vistas principales: Deportista y Organizador, las cuales acceden a un API el cual es desarrollado en .NET Framework y accede a capa de datos.

En la abstracción se concluye que se encuentran tres grandes Módulos a los cuales acceden los usuarios, cada módulo corresponde a un componente en el Backend.

1. **Módulo Login/Registro de Usuarios:** Gestiona el inicio de sesión de usuarios de tipo organizador y deportista, además del registro de usuarios.
2. **Módulo Vista Organizador:** Gestiona las funcionalidades y acceso de datos al tipo de usuario organizador, dentro de las principales se encuentran: gestionar carreras, retos y grupos; gestión de afiliaciones y reportes.
3. **Módulo Vista Deportista:** Gestiona las funcionalidades y acceso de datos al tipo de usuario deportista, dentro de las principales se encuentran: visualización de actividades realizadas por mis amigos, registrar actividades, inscribirse en carreras, retos y grupos.

La funcionalidad principal que posee la aplicación móvil es cargar actividades desde el dispositivo, esta actividad se registra primero en una Base de Datos de tipo SQL y después de cierto tiempo realiza un proceso de sincronización con el API, para guardar esta actividad en SQL Server.

Problemas conocidos

- Dependiendo del dispositivo en el que se quiera correr el servidor, se debe cambiar el path respectivo de la carpeta de reportes.
- La actualización automática de reportes

Problemas encontrados

- Los reportes no se actualizaban automáticamente cuando se actualizaba la Base de Datos, este mismo se solucionó implementando un refresh.
- Las tablas no se actualizaban inmediatamente después de un cambio, esto se solucionó ajustando el “ngOnInit” para llamarlo siempre después de un cambio visual.
- Por un problema de dependencias durante un periodo se trabajó por separado la vista del organizador y la vista del deportista en el control de versiones de GitHub.

Minutas de Sesiones de Trabajo

No	Tipo	Asistentes	Día	Descripción
1	Reunión	Todos	10/26/20	Realizamos distribución de tareas y completamos el archivo del plan de proyecto.
2	Reunión	Todos	10/29/20	Luego de la Primera Reunión con el punto de contacto, denotamos puntos importantes en cuanto a la mejora del trabajo en equipo
3	Reunión	Todos	10/30/20	Se redistribuyen las tareas, se decide trabajar de la siguiente forma: <ol style="list-style-type: none"> 1. FrontEnd: Angelo Ortiz - Agustín Venegas. 2. Backend: Jonathan Esquivel - Iván Solís.
4	Reunión	Todos	11/5/20	Reunión de avance, se encuentran a nivel inicial la visualización del perfil de usuario y las actividades que ha realizado. Se distribuyen las tareas de Backend donde implica desarrollo de API.
5	Reunión	Angelo Ortiz, Agustín Venegas	11/8/20	Se distribuyen las tareas de Vista Deportista y Vista Organizador para asegurar un avance significativo.
6	Reunión	Todos	11/12/20	Luego de la reunión con el punto de contacto, se crea distribución para las siguientes semanas.
7	Reunión	Todos	11/15/20	Se anotan tareas que hacen falta a la fecha del 11/15/20, y se planean fechas finales para cada tarea.
8	Reunión	Todos	11/17/20	Se distribuyen tareas para trabajos entregables donde incluye Documentación Técnica, Documento de Instalación y Manual de Usuario.
9	Reunión	Todos	11/19/20	Se realizan pruebas sobre el proyecto en diferentes computadoras, verificando el correcto funcionamiento de todos los componentes.

Bitácora de Trabajo

No	Tipo	Participantes	Día	Descripción
1	Documentación	Angelo Ortiz	27/10	Creación de Documentos: Administración de Proyecto y Plan de Proyecto.
2	Documentación	Angelo Ortiz	28/10	Desarrollo de Plan de Proyecto y Requisitos Iniciales de Software.
3	Desarrollo	Jonathan Esquivel	30/10	Desarrollo de Estructura de aplicación frontend.
4	Desarrollo	Angelo Ortiz	30/10	Desarrollo de Estructura de aplicación backend.
5	Desarrollo - Documentación	Angelo Ortiz, Jonathan Esquivel	30/10	Se inicia el modelo entidad relación.
6	Desarrollo	Angelo Ortiz	31/10	Validaciones y desarrollo de inicio de sesión y registro de usuarios a nivel de frontend.
7	Desarrollo	Agustín Venegas	1/11	Estructura inicial de la vista de deportista; desarrollo de RF005.
8	Desarrollo	Agustín Venegas	2/11	Desarrollo de RF002, RF006, RF007.
9	Desarrollo	Agustín Venegas, Jonathan Esquivel	2/11	Desarrollo de cartas dinámicas para colocar datos de usuarios seguidos en vista deportista.
10	Desarrollo	Angelo Ortiz	3/11	Desarrollo de tablas para vista de organizador.
11	Desarrollo	Agustín Venegas, Jonathan Esquivel	4/11	Implantación de Mapas (.gpx) a las cartas de usuarios seguidos, continuación a requerimiento RF008.
12	Documentación	Jonathan Esquivel, Iván Solís	4/11	Finalización del modelo conceptual y relacional.
13	Desarrollo	Jonathan Esquivel, Iván Solís	5/11	Creación del script de base de datos.

14	Desarrollo	Angelo Ortiz	5/11	Impletación de Paper Dashboard Angular
15	Desarrollo	Angelo Ortiz	6/11	Actualización de las cartas de amigos dentro de dashboard
16	Desarrollo	Agustín Venegas	8/11	Creación de una branch aparte "frontend-rebuild" para desarrollar vista del deportista.
17	Desarrollo	Agustín Venegas	8/11	Se estableció comunicación con el servidor para iniciar sesión.
18	Desarrollo	Jonathan Esquivel, Iván Solís, Agustín Venegas	9/11	Se realizó comunicación entre API, Base de Datos y Frontend para la población de actividades de usuario.
19	Desarrollo	Jonathan Esquivel, Iván Solís, Agustín Venegas	10/11	Se realizó comunicación para la actualización de actividades de los usuarios seguidos, en orden ascendente.
20	Desarrollo	Angelo Ortiz	11/11	Se crea a nivel de interfaz la sección de actualización de perfil, y retos/actividades en los que estoy inscrito.
21	Desarrollo	Jonathan Esquivel, Iván Solís, Agustín Venegas	11/11	Se realiza comunicación para llenar los datos del usuario (perfil).
22	Desarrollo	Jonathan Esquivel, Iván Solís, Agustín Venegas	13/11	Se crea el componente para subir una actividad y se muestran las carreras y retos en los que está inscrito el usuario.
23	Desarrollo	Iván Solís	14/11	Generación Básica de Reportes
24	Desarrollo	Angelo Ortiz	16/11	Creación de Aplicación Móvil - Inicio de Sesión y registro de actividades listo.
25	Desarrollo	Jonathan Esquivel, Agustín Venegas	17/11	Se realizó la comunicación entre frontend y backend sobre la gestión de carreras y retos.
26	Desarrollo	Jonathan Esquivel, Agustín Venegas	18/11	Se realizó la comunicación entre frontend y backend sobre la gestión de grupos y afiliaciones.

27	Documentación	Angelo Ortiz	18/11	Finalización del Documento de Instalación, se añaden dependencias y requisitos de software.
28	Desarrollo	Jonathan Esquivel, Agustín Venegas	19/11	Actualización de perfil de usuario y display de reportes en vista de organizador
29	Desarrollo	Angelo Ortiz	20/11	Finalización de Aplicación Móvil, se pueden visualizar actividades realizadas por personas a quienes sigo, y las mías.
30	Desarrollo	Jonathan Esquivel, Agustín Venegas	21/11	Mejoras del proyecto, correcciones de errores.
31	Desarrollo	Todos	21/11	Creación de Script para creación y población de la base de datos.
32	Documentación	Todos	21/11	Finalización de Manual de Usuario y Documentación Técnica.

Conclusiones

- PostgreSQL es uno de los mejores sistemas de bases de datos, ya que posee muchas características que triunfan en cuanto a otros DBMS.
- PostgreSQL es un gestor de bases de datos poderoso, avanzado y muy completo que retorna los mejores resultados a la hora de montar bases de datos relacionales.
- SQLite realiza operaciones de manera más eficiente y es más rápido que MySQL y PostgreSQL.
- SQLite cuenta con diferentes interfaces del API, el cual le permite trabajar en cualquier entorno, se ejecuta en muchas plataformas y sus bases de datos pueden ser fácilmente portadas sin ninguna configuración o administración.

Recomendaciones

- Se recomienda usar el principio LIFT sobre la estructuración de archivos dentro de un proyecto, LIFT por sus siglas significa:
 - Location our code easy.
 - Identify code at a glance.
 - Flat structure as long as we can.
 - Try to stay dry
- Organizar el código es fundamental para proyectos escalables, lo más importante debe ir arriba, se recomienda usar Camelcase para nombramiento de métodos, propiedades y componentes.
- Una buena práctica es separar Servicios de Componentes, donde el servicio funcione para recolectar datos y realizar comunicación con el servidor.

Bibliografía

6 Best Practices & Pro Tips when using Angular CLI Recuperado de <https://medium.com/@tomastrajan/6-best-practices-pro-tips-for-angular-cli-better-developer-experience-7b328bc9db81>

Angular University. (2020). Recuperado de <https://angular-university.io/>

Angular JS (2018-10-04) Recuperado de <https://angularjs.io>

Bootstrap Themes & Templates (2018-10-04) Recuperado de <http://wrapbootstrap.com>

Ionic Framework Recuperado de <https://ionicframework.com/>

Jackson, L. (2020). *.NET Core 3.1 MVC REST API - Full Course* [Video]. Recuperado de <https://youtu.be/fmvcAzHpsk8>

Paper Dashboard Angular Recuperado de <https://www.creative-tim.com/product/paper-dashboard-angular>

SQLite Ionic Recuperado de <https://ionicframework.com/docs/native/sqlite>