Proyecto #1 - StraviaTec

Instituto Tecnológico de Costa Rica
Área Académica Ingeniería en Computadores
Bases de Datos (CE3101)
II Semestre 2020
Valor 20%

Objetivo general

- ➔ Desarrollar una aplicación que permita manejar la descripción del caso expuesto en el punto 4.

Objetivos específicos

- ➔ Aplicar los 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, CSS, 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 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 y elites) para que lleven un registro de todas sus actividades que les permita desde compartir sus sesiones hasta mejorar sus tiempos y entrenamientos.

Vista Deportista: esta es la plataforma que permitirá a todos los deportistas visualizar de una forma agradable/amigable las sesiones realizadas.

Vista Organizador: Ésta es la plataforma que permite a los organizadores de eventos agregar eventos, incluir/aprobar tablas de posiciones, aceptar inscripciones/solicitudes de afiliación.

Requerimientos del Software

- ➔ Vista Deportista.
  - ◆ **Crear Cuenta:** El sistema permitirá la creación, modificación y eliminación de cuentas. Cuando se crea una cuenta se desea saber, nombre y apellidos, fecha de nacimiento, edad actual considerando la fecha de nacimiento, nacionalidad, foto, usuario y password.
  - ◆ **Log In:** Con una cuenta creada el deportista podrá ingresar a la aplicación una vez autenticadas sus credenciales.

- **Página de inicio:** Esta página mostrará todas las actividades registradas por orden ascendente de todos los amigos del deportista. En esta vista se podrá ver a manera de resumen: que tipo de actividad realizo, mapa de donde realizo la actividad y los kilómetros recorridos.

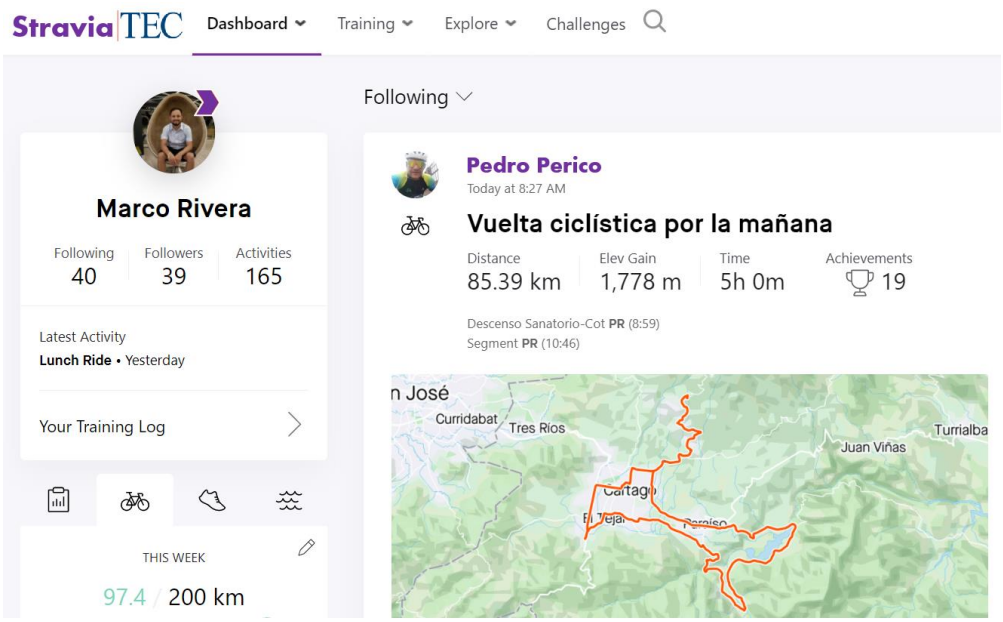

Ilustración 1. Página de inicio.

- **Búsqueda y seguimiento de atletas:** El sistema deberá facilitar la búsqueda de otros atletas a los cuales se desea seguir y facilitar agregarlo a la lista de amigos que sigue.

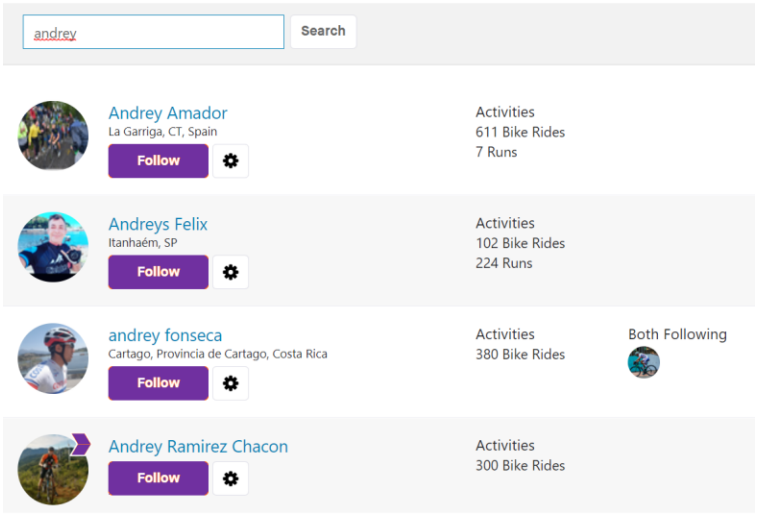

Ilustración 2. Búsqueda de deportistas.

- **Registrar Actividad**: el sistema debe permitir cargar una actividad de ejercicio realizada por un atleta de la cual se conoce: fecha y hora de la actividad, duración tipo de actividad, kilometraje, el recorrido que será un archivo **.gpx** y se debe indicar si es la completitud de un reto o carrera previamente inscrito.
- **Inscribirse en una Carrera:** El sistema debe permitir a los deportistas inscribirse a una carrera, para poder realizarlo debe anexar el recibo de pago del costo de la carrera.

- **Inscribirse en un Reto:** El sistema debe permitir a los deportistas inscribirse a un reto.
- **Asociarse a Grupo**: El sistema debe poder permitir a los deportistas asociarse a un grupo de deportistas.
- **Ver retos y carreras inscritos:** La aplicación deberá mostrar un listado de las carreras y retos inscritos.
  - En el caso de ser una carrera se debe poder ver la tabla de posiciones ordenado de menor a mayor por tiempo.
  - En el caso de ser un reto se debe ver el objetivo del reto y el avance que tiene el deportista así mismo debe informar cuantos días faltan para que se termine el reto.

➔ Vista Organizador.

- **Gestión de Carrera**: Las carreras son eventos de pago y se deben completar con una sola actividad.
  El sistema debe facilitar la posibilidad de crear, modificar y eliminar carreras. Para el registro de carreras se debe ingresar: nombre de la carrera, fecha de la carrera, recorrido de la carrera (**archivo .gpx**), tipo de actividad, si es privado o no (puede ser visto por todos o por ciertos grupos en este último caso se deben listar los grupos), costo de la carrera, cuentas bancarias, categorías disponibles (Junior, Sub-23, Elite, Master A, Master B, Master C) y la lista de patrocinadores.
- **Aceptar Inscripción:** El organizador validará el recibo de la inscripción y que efectivamente los fondos se encuentran en su cuenta de ser así procederá a aceptar la inscripción caso contrario denegará la inscripción.
- **Gestión de Retos**: Los retos son eventos gratuitos y se pueden completar con varias actividades, acumulará la cantidad de kilómetros recorridos o metros ascendidos según sea el reto.
  El sistema debe permitir crear, modificar y eliminar retos. Para el registro de retos se debe ingresar: nombre del reto, periodo en que esta disponible el reto, tipo de actividad, si es un reto de fondo o altitud, si es privado o no (puede ser visto por todos o por ciertos grupos en este último caso se debe listar los grupos) y la lista de patrocinadores.
  Los retos se pueden completar con una o más actividades.
- **Gestión de Grupos**: El sistema debe permitir crear, modificar y eliminar grupos, de un grupo se conoce: el nombre del grupo, un administrador.
- **Reporte de Participantes por Carrera:** Este reporte debe mostrar la lista de participantes a una carrera. Se desea visualizar nombre, apellidos, edad, categoría en la que participará y debe estar agrupada/ordenada por este atributo
- **Reporte de Posiciones de Carrera:** este reporte permitirá visualizar el nombre, apellidos, edad, tiempo registrado de completitud de la actividad y categoría participante. El reporte debe estar ordenado por el tiempo registrado de menor a mayor y debe estar agrupado por categoría.

➔ Valores por defecto.

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

        - Correr.
        - Nadar.
        - Ciclismo.
        - Senderismo.
        - Kayak
        - Caminata.
    - 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.
    - Debe existir un listado de categorías por defecto se conoce nombre y descripción. Las categorías que se tiene son:
        - Junior: menos de 15 años
        - Sub-23: de 15 a 23.
        - 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.

➔ **Aplicación Móvil**. La aplicación móvil permitirá realizar el registro de actividades a los deportistas.

- Esta mostrará el tiempo que lleva desde que se inicio la actividad, los kilómetros recorridos.
- Debe almacenar la ruta gps por donde transita el deportista.
- Los datos de esta aplicación móvil deben almacenarse en SQL lite para posteriormente ser sincronizados con la app web.

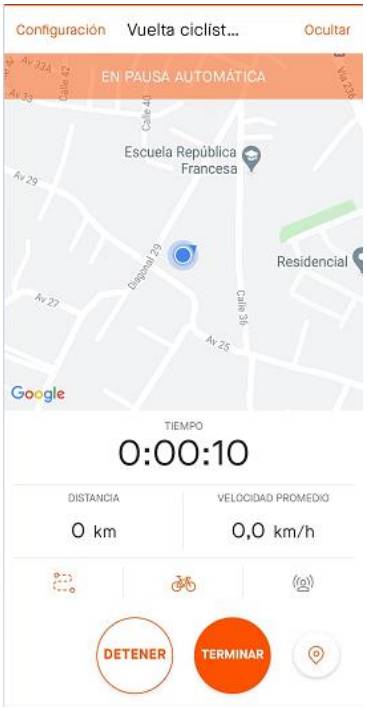

Ilustración 3. Aplicación Móvil.

Arquitectura del Software

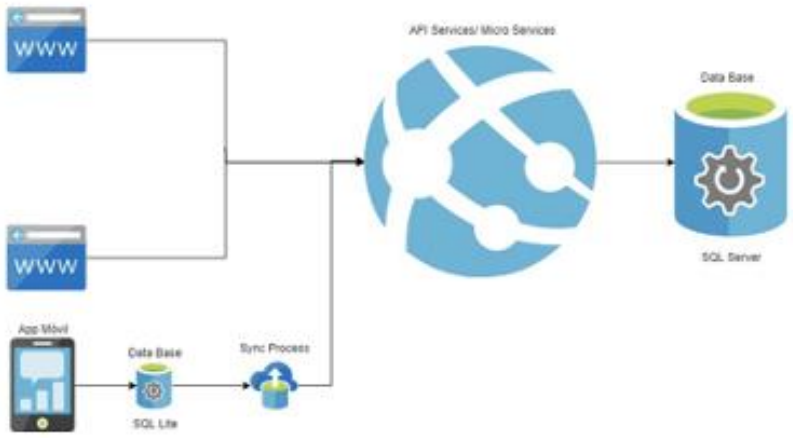

Ilustración 4. Arquitectura deseada

Requerimientos no funcionales del sistema

- ➔ El Sistema debe ser una aplicación web (utilizando Angular, Bootstrap, HTML5, CSS).
- ➔ La App Móvil debe ser desarrollada utilizando 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 Datos deben implementarse en la capa de datos.**
- ➔ La capa de servicios debe estar desarrollada en C# y debe ser desplegada en el IIS.
- ➔ El equipo de trabajo debe seleccionar a uno de sus miembros como único punto de contacto. Todas las comunicaciones y solicitudes deben ser a través de dicho punto de contacto.

Entregables

- ➔ Manual de Usuario.
- ➔ Documentación Técnica y del proyecto (como se solicita abajo).
- ➔ Documento de instalación.
- ➔ Plan de Proyecto.
- ➔ Script de Base de Datos.
- ➔ Script de población de Base de Datos.
- ➔ Aplicación WEB.
- ➔ Aplicación Móvil.
- ➔ Web API.
- ➔ Minutas

Documentación requerida

- ➔ Se deberá documentar el código fuente.
- ➔ Se deberá entregar un documento que contenga:
  - ◆ Modelo conceptual utilizando la notación de Chen.

- Modelo relacional.
- Descripción de las estructuras de datos desarrolladas (Tablas).
- Descripción detallada de la arquitectura desarrollada.
- Problemas conocidos: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.
- Problemas encontrados: descripción detallada, intentos de solución sin éxito, soluciones encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.

➔ Documentación de evidencia del trabajo en equipo.
- Descripción de roles.
- Actividades planeadas, su responsable, estimación de completitud de la tarea (menores a 8hs) y fecha de entrega. (Plan de trabajo)
- Minutas de sesiones de trabajo. (Seguimiento al plan de trabajo)
- Actividades realizadas por cada estudiante. (Bitácora en digital, donde se describen las actividades realizadas, desde reuniones con el compañero de trabajo, investigaciones, consultas, etc. Se debe describir todo por más insignificante que sea, esto demostrará si ustedes están trabajando en realidad. Este es su diario de trabajo, llevan seguimiento de todo en el tiempo, imaginen que, si un compañero los releva en su trabajo, le bastaría con leer sus bitácoras para seguir el trabajo.
- Evidencia de uso de un manejador de código (se recomienda Gitgub).

➔ Conclusiones del proyecto.

➔ Recomendaciones del proyecto.

➔ Bibliografía consultada en todo el proyecto

Aspectos operativos y evaluación:

1. **Fecha de entrega:** De acuerdo con el cronograma del curso y lo establecido en el TEC Digital. Se establece el siguiente plan de entregas parciales:
    a. Plan de proyecto: 28/Oct/2020
    b. Resumen Ejecutivo Avance 1: 04/Nov/2020
    c. Resumen Ejecutivo Avance 2: 11/Nov/2020
    d. Funcionalidad completa: 20/Nov/2020
2. El proyecto tiene un valor de 20% de la nota del curso.
3. El trabajo es **en grupos de 4 personas**.
4. La implementación tendrá un valor de un 70% de la nota final, debe estar funcional. La defensa vale un 10% y la documentación externa un 20%.
5. Cumplir con los requerimientos especificados en la documentación no significa que se tienen todos los puntos, se evaluará que la documentación sea coherente, acorde al tamaño del proyecto y el trabajo realizado, no escatimen en documentación.
6. Cada grupo recibirá una nota en cada uno de los siguientes apartados Código y Documentación.
7. El profesor no sólo evaluará la funcionalidad del proyecto, esto quiere decir que aunque el proyecto este 100% funcional esto no implica una nota de un 100, ya que se evaluarán aspectos de calidad de código, aplicación del paradigma OOP, uso de herramientas solicitadas, calidad de documentación interna y externa, trabajo en equipo.
8. No se revisarán funcionalidades parciales, ni funcionalidades no integradas.

9. Es responsabilidad de cada miembro del grupo conocer su código, el profesor puede preguntar a cualquier miembro del grupo que le explique alguna funcionalidad/porción de código.
10. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
11. Aún cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
    a. Si no se entrega documentación, automáticamente se obtiene una nota de cero en el proyecto.
    b. Si no se utiliza un manejador de código se obtiene una nota de cero en el proyecto.
    c. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de cero en el proyecto.
    d. Si el código no compila se obtendrá una nota de cero en el proyecto, por lo cual se recomienda realizar la defensa con un código funcional.
    e. Si el grupo no cuenta con los equipos necesarios para realizar la revisión y no avisó al profesor de esta situación obtendrá una nota de cero en el proyecto.
    f. El código debe ser desarrollado en C#, en caso contrario se obtendrá una nota de cero en el proyecto.
    g. Si el grupo no se presenta a la revisión se obtiene nota de cero en el proyecto. Si un estudiante no se presenta a la defensa y no cuenta con una justificación válida se le asignará al estudiante una nota de cero en el proyecto.
12. Cada grupo tendrá como máximo 60 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
13. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
14. Cada grupo es responsable de llevar los equipos requeridos para la revisión.
15. Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.
16. Las revisiones se realizan con los estudiantes matriculados en el curso, cualquier persona fuera de estos y los mencionados en el punto 15, no pueden participar en la revisión.
17. Si no se realiza la defensa del proyecto se asignará una nota de 0 en el proyecto.

Referencias

AngularJS (2018-10-04). Recuperado de**:** **https://angularjs.io**

Bootstrap Themes & Templates (2018-10-04). Recuperado de**:** **https://wrapbootstrap.com/**

How to Write Doc Comments for the Javadoc Tool. (2018-10-04). Recuperado de: http://www.oracle.com/technetwork/articles/java/index-137868.html

C# Coding Conventions (C# Programming Guide). (2018-10-04). Recuperado de: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions