

Proyecto Programado #2

Alquiler de Bicicletas

Introducción

Los sistemas de registro de información permiten llevar control de las acciones diarias y mejorar el reporte de información, por medio de la automatización y digitalización de procesos involucrados en la gestión de este.

Adicionalmente, estos sistemas son usados para manejar todas las tareas administrativas de los administradores. Permite gestionar la base de datos de personas, visualizar información detallada de ellas, insumos, inventario, utilización, entre otros.

Actualmente, ante la saturación de nuestras calles por todo tipo de vehículo automotor, el hecho de viajar unos pocos kilómetros en el centro de una ciudad puede convertirse en un caos. Muchos gobiernos locales están impulsando el uso de la bicicleta, pero con el inconveniente que no todos cuentan con una, por ellos, cada uno de estos municipios tienen a disposición del público bicicletas para su alquiler gracias al apoyo de empresas.

El proyecto busca someter al estudiante a entornos y realidades lo más cercano a la realidad, con el objetivo de generar un aprendizaje efectivo.

Proyecto a desarrollar

Su trabajo consiste en implementar una aplicación de escritorio para administrar la gestión de alquiler de bicicletas utilizando el lenguaje Haskell y la Programación Funcional.

El programa deberá desplegar un menú al usuario con dos submenús o grupo de funciones (el menú debe mostrarse recurrentemente después de cada opción realizada -a excepción del salir- y tener la opción de subir al menú principal desde los menús secundarios) y salir:

Opciones Operativas

La solución debe disponer de una sección de opciones operativas en donde se accederá a todas estas funcionalidades indicando un usuario (debe estar indicado en la documentación).

Al indicar el usuario se habilitará un menú donde se habilitarán las siguientes funcionalidades:

- Información comercial (no va en menú)
- Cargar y Mostrar parqueos
- Mostrar y asignar bicicletas
- Cargar usuarios
- Estadísticas
- volver

Información comercial

El sistema debe tener registro de la información de la empresa gestora del sistema, que será el **nombre de la empresa, sitio web, contacto, tarifa en colones por kilómetro pedal (TR), tarifa por kilómetro eléctrico (AE) y tarifa por kilómetro gasolina (AG).**

Carga y mostrar parqueos

El sistema debe permitir cargar un archivo con la base de datos de los parqueos de bicicletas, por cada parqueo se deberá indicar **id, nombre, dirección, provincia** (LI, PU, GU, SJ, AL, CA o HE), **ubicación x y ubicación y** (hace un par (x,y)). La información se cargará desde un archivo, que el usuario indicará la ruta, donde cada línea tendrá información de un parqueo y los datos estarán separados por coma. Al finalizar se mostrarán todos los parqueos del sistema.

Ejemplo línea: id,nombre,dirección,provincia,x,y

Mostrar y asignar bicicletas

El sistema debe poseer al menos 15 bicicletas precargadas en el sistema. Por cada bicicleta se debe registrar **identificador** (código ej: B001), **tipo** (Tradicional **TR**, con asistencia Eléctrica **AE** o con asistencia Gasolina **AG**).

Campos: código, tipo

El sistema permitirá cargar un archivo con el que actualiza la ubicación de las bicicletas, el archivo tendrá, por cada línea y separado por comas, el identificador de bicicleta y el identificador de parqueo (actualización parcial del lote):

idbicicleta,idparqueo

Adicionalmente, se podrá consultar por nombre de parqueo y mostrará las bicicletas en dicho parqueo. **No se muestran las bicicletas en tránsito.** Al indicar nombre “#”, muestra todas las bicicletas del sistema, al indicar “transito”, mostrará las bicicletas en tránsito (activas en alquiler).

Carga usuarios

El sistema debe permitir cargar un archivo con la base de datos de usuarios, por cada persona se deberá indicar la **cédula** (cadena numérica de largo 9) y **nombre completo**. La información se cargará desde un archivo, que el usuario indicará la ruta, donde cada línea tendrá información de una persona y los datos estarán separados por coma.

Ejemplo línea: cédula,nombre completo

Estadísticas

El sistema debe permitirle al usuario, por medio de un submenú, acceder a las siguientes estadísticas:

- a) Top 5 de bicicletas con más viajes, indicar bicicleta y cantidad de viajes.
- b) Top 5 de parqueos con más viajes (salida + destino) indicar parqueo y cantidad de viajes.
- c) Top 3 de usuarios con más kilómetros recorridos (según fórmula de distancia). Indicar usuario y cantidad.
- d) Resumen: total de viajes, total de kilómetros y total facturado (facturas generadas).

Opciones Generales

Para acceder a estas funcionalidades el usuario deberá ingresar por medio del Menú Principal y se deben habilitar las siguientes funcionalidades (Menú General):

- Consultar bicicletas
- Alquilar
- Facturar
- Volver

Consultar bicicletas

El usuario tendrá que indicar una posición (x,y) y el sistema le mostrará el parqueo más cercano, junto con sus bicicletas (debe utilizar una fórmula para encontrarlo, no puede ser “alambrado”). Investigar fórmula o proponer una, en ambos casos debe documentar el origen.

Alquilar

El sistema le solicitará al usuario que indique: **usuario (cedula)**, **un parqueo de salida** y **el parqueo de llegada**. Posteriormente le mostrará las **bicicletas disponibles** (identificador y tipo) en el parqueo de salida. **El usuario indicará el identificador** de la bicicleta y se le **generará** un identificador de alquiler por parte del sistema. La bicicleta cambia de **ubicación** a “**tránsito**”, se debe generar una estructura de datos con la información del alquiler (al menos salida, destino, código de la bicicleta, código del alquiler), este quedará “**activo**”.

Facturar

El sistema le solicitará al usuario que indique el **identificador de alquiler**, el sistema verificará que el **alquiler está activo** y procederá a facturarle. Se le muestra al usuario una factura en pantalla con la siguiente información: **identificador** (autogenerado), **información del negocio** (información comercial), **usuario**, **salida**, **destino**, **bicicleta** (identificador, tipo), **cantidad de kilómetros** (calculado con la misma fórmula de la consulta de bicicletas), **tarifa por kilómetro** aplicada (depende del tipo de bicicleta) y **total en colones**. Debe mostrar un formato **amigable al usuario**.

Al facturar, el alquiler cambia a estado “**facturado**” y la bicicleta se le coloca de ubicación el parqueo destino.

Salir

Los datos del sistema no son persistentes, toda la información indicada anteriormente se libera, se hace un manejo de estructuras en memoria.

Puntos Extra

Se darán **2.5 puntos adicionales** al entregar a más tardar el miércoles 19 de septiembre a las 11:55:55 PM el Documento de Requerimientos, ver plantilla suministrada en el Tec Digital. Debe subirse en la documentación llamada “Proyecto Programado II (archivos adicionales)” debajo de la carpeta de “Proyectos”.

Se darán **7.5 puntos adicionales** si se maneja persistencia de datos, es decir que al salir del Menú Principal y volver a entrar al Menú Principal se mantengan los datos. La persistencia debe realizarse por medio de un medio físico (archivo o base de datos). Con la carga de datos, si se carga por segunda vez una información, se anexará lo nuevo y se mantendrá lo anterior, sin permitir repetidos.

Aspectos técnicos

El proyecto deberá estar escrito en el lenguaje de programación Haskell. En caso de requerir librerías adicionales para compilar y ejecutar el programa, deberán especificarlo en la documentación, ya que de lo contrario se descontarán puntos en la evaluación.

Deberán utilizar el sistema de control de versiones GitHub, el repositorio deberá ser público o incluir al profesor en el control de acceso del mismo.

Se valorará el aporte generado por cada estudiante, considerando, entre otras cosas, los commit generados por cada uno. Por lo que el puntaje obtenido por cada uno de los estudiantes puede ser diferenciado.

Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de los mismos.

Para la documentación interna, deberán incluir comentarios descriptivos para cada función, con sus entradas, salidas, restricciones y **objetivo**.

La documentación externa deberá incluir:

1. Portada.
2. Manual de usuario: **instrucciones de compilación, ejecución y uso**.
3. Pruebas de funcionalidad: incluir *screenshots*.
4. Descripción del problema.
5. Diseño del programa: decisiones de diseño, algoritmos usados.
6. Librerías usadas: manejo entradas-salidas, archivos, etc.
7. Análisis de resultados: objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).
8. Bitácora (autogenerada en git, commit por usuario incluyendo comentario)

Forma de trabajo

El trabajo se debe realizar en parejas.

Evaluación

La evaluación se va a centrar en dos elementos: programación y documentación.

El proyecto programado tiene un valor de **10%** de la nota final, en el rubro de Proyectos.

Desglose de la evaluación del proyecto programado:

1. Documentación interna 2 ptos.
2. Documentación externa 8 ptos.
3. Funcionalidad 80 ptos (ver detalle en Proyecto a Desarrollar)
4. Revisión del proyecto (según completitud del proyecto y gestión del tiempo) 5 ptos.
5. Hora de Entrega 5 ptos.

Aspectos administrativos

Debe crear un archivo **.zip** ("PP2_Integrante1_Integrante2.zip") que contenga únicamente un archivo **info.txt** y 2 carpetas llamadas **documentacion** y **programa**, en la primera deberá incluir el documento de *word* o pdf solicitado y en la segunda los archivos y carpetas necesarias para la implementación de este proyecto programado, y/o link en git del repositorio. El archivo **info.txt** debe contener la siguiente información (cualidades):

- a. Nombre del curso
- b. Número de semestre y año lectivo
- c. Nombre de los Estudiantes
- d. Número de carnet de los estudiantes
- e. Número de proyecto programado
- f. Fecha de entrega
- g. Estatus de la entrega (debe ser **CONGRUENTE** con la solución entregada):
[Deplorable | Regular | Buena | MuyBuena | Excelente | Superior]

Entrega

Deberá subir el archivo antes mencionado al TEC Digital en el curso de LENGUAJES DE PROGRAMACIÓN GR 60, en la asignación llamada "P2" debajo del rubro de "Proyectos". En la evaluación del Proyecto el rubro de "Hora de Entrega" valdrá por 5 puntos de la nota total del proyecto, según la siguiente escala:

- a. Si se entrega antes de las 11:55:55 **PM** del miércoles 04 de octubre de 2023, 5 puntos.
- b. Si se entrega antes de las 11:55:55 **AM** del jueves 05 de octubre de 2023, 2.5 puntos.
- c. Si se entrega antes de las 11:55:55 **PM** del jueves 05 de octubre de 2023, 0 puntos.

NO SE ACEPTARÁN trabajos que contengan "commits" posterior a esta fecha.

Todo el contenido de cada proyecto debe ser 100% original y en caso de conducta fraudulenta se procederá según el artículo 75 del RREA.

Todos los miembros del grupo deberán participar de la revisión, ya que de lo contrario no se les asignará el puntaje correspondiente.