



Fredd Badilla Víquez

Randall Corella Castillo

Allan Rodríguez Dávila

Proyecto Programado #2

Semestre II, 2023

## Manual de usuario

Primero hay que ejecutar el código main el cual es llamado “main.hs” con el comando “ghc -o main main.hs”, luego de esto se llama al ejecutable “main”, se desplegará el menú principal con 3 opciones “Opciones específicas”, “Opciones generales” y “Estadísticas”, en opciones específicas nos encontraremos las opciones “cargar y mostrar parqueos”, donde se nos dará la opción de ingresar un path de un archivo de texto, donde si se sigue el formato correcto se agregaran los parqueos deseados, “mostrar y asignar bicicletas”, en esta opción se nos desplegará un submenú con 3 opciones, “actualizar bicicletas”, “cargar bicicletas” y “mostrar bicicletas”, donde al igual que con los parqueos se pedirá un path donde si se siguen los parámetros conocidos se realizarán dichas acciones sin problemas, “cargar usuarios”, sigue la misma dinámica.

Luego “Opciones generales”, donde se nos dará la opción de “consultar”, se nos pedirá coordenadas con las cuales encontrará el parqueo más cercano y se nos mostrarán las bicis asociadas a ese parqueo, luego en “alquilar”, se nos pedirá la cedula de un usuario registrado previamente, con esta indicaremos el ID de un parqueo de inicio y de un parqueo de fin, nos mostrará las bicis en el parqueo de inicio, seleccionamos una según ID y se nos dará el alquiler.

Luego en facturar se nos pedirá un ID de alquiler válido, si es válido, se nos hará la factura con todos los datos del alquiler, finalmente en estadísticas se nos mostrarán datos relevantes de la aplicación.

Pruebas de funcionalidad

### **Menu principal.**

```
Bienvenido al Menú
Selecciona una opción:
1. Opciones especificas
2. Opciones generales
3. Estadísticas
4. Salir
|
```

### **Opciones operativas.**

```
MENU OPERATIVO
1. Cargar y mostrar parqueos
2. Mostrar y asignar bicicletas
3. Cargar usuarios
4. Estadísticas
5. Volver
```

```
Ingresa su opcion deseada:
```

```
|
```

## Cargar parqueos.

1. Cargar y mostrar parqueos
2. Mostrar y asignar bicicletas
3. Cargar usuarios
4. Estadísticas
5. Volver

Ingrese su opcion deseada:

1

Ingresar la ruta de tu archivo

test1.txt

AVISO -> Informacion agregada con exito

ID: 1| Nombre: parqueoSj| Ubicacion: Desamparados centro| Provincia: SJ| Coordenadas: X->154 Y->70  
ID: 2| Nombre: santaParks| Ubicacion: Al frente del parquesito| Provincia: LI| Coordenadas: X->20 Y->45  
ID: 3| Nombre: puertoParks| Ubicacion: En el paseo de los turistas| Provincia: PU| Coordenadas: X->207 Y->110  
ID: 4| Nombre: tamarindoParks| Ubicacion: Bagaces centro| Provincia: GU| Coordenadas: X->512 Y->458  
ID: 5| Nombre: ligaParks| Ubicacion: al frente del Morera| Provincia: AL| Coordenadas: X->215 Y->256  
ID: 6| Nombre: cristianParks| Ubicacion: a la par del TEC| Provincia: CA| Coordenadas: X->195 Y->115  
ID: 7| Nombre: oxigenParks| Ubicacion: al frente de Oxígeno| Provincia: HE| Coordenadas: X->89 Y->96  
ID: 8| Nombre: wyntaParks| Ubicacion: entrando a Pueblo| Provincia: LI| Coordenadas: X->30 Y->70  
ID: 1813| Nombre: parqueoSj| Ubicacion: Desamparados centro| Provincia: SJ| Coordenadas: X->154 Y->70  
ID: 1833| Nombre: tamarindoParks| Ubicacion: Bagaces centro| Provincia: GU| Coordenadas: X->512 Y->458  
ID: 1853| Nombre: ligaParks| Ubicacion: al frente del Morera| Provincia: AL| Coordenadas: X->215 Y->256  
ID: 1873| Nombre: cristianParks| Ubicacion: a la par del TEC| Provincia: CA| Coordenadas: X->195 Y->115  
ID: 1883| Nombre: oxigenParks| Ubicacion: al frente de Oxígeno| Provincia: HE| Coordenadas: X->89 Y->96

MENU OPERATIVO

1. Cargar y mostrar parqueos
2. Mostrar y asignar bicicletas
3. Cargar usuarios
4. Estadísticas
5. Volver

Ingrese su opcion deseada:

|

## Cargar bicicletas.

### MENU OPERATIVO

1. Cargar y mostrar parqueos
2. Mostrar y asignar bicicletas
3. Cargar usuarios
4. Estadísticas
5. Volver

Ingrese su opcion deseada:

2

### MENU DE BICIS

1. Cargar bicis
2. Actualizar bicicletas
3. Mostrar bicicletas
4. Volver

Ingrese su opcion deseada:

1

Ingrese la ruta de su archivo con las bicicletas:

test2.txt

AVISO -> Informacion agregada con exito

Bicis añadidas con exito!!

### MENU DE BICIS

1. Cargar bicis
2. Actualizar bicicletas
3. Mostrar bicicletas
4. Volver

Ingrese su opcion deseada:

|

## Mostrar bicicletas.

```
Ingresa su opcion deseada:
3

Ingresa un ID de parqueo, # o la palabra ->transito<-, segun lo que quieras ver
#

Estas son las bicis que no estan en transito de manera global

ID: B004| Tipo: TR| Ubicacion: 3
ID: B005| Tipo: AE| Ubicacion: 3
ID: B007| Tipo: TR| Ubicacion: 3
ID: B009| Tipo: TR| Ubicacion: 3
ID: B011| Tipo: TR| Ubicacion: 4
ID: B014| Tipo: TR| Ubicacion: 8
ID: B015| Tipo: TR| Ubicacion: 8
ID: B044| Tipo: TR| Ubicacion: 1
ID: B045| Tipo: TR| Ubicacion: 1
ID: B487| Tipo: TR| Ubicacion: 1
ID: B497| Tipo: TR| Ubicacion: 1
ID: B407| Tipo: AE| Ubicacion: 1

MENU DE BICIS
1. Cargar bicis
2. Actualizar bicicletas
3. Mostrar bicicletas
4. Volver

Ingresa su opcion deseada:
|
```

## Actualizar bicicletas.

```
MENU DE BICIS
1. Cargar bicis
2. Actualizar bicicletas
3. Mostrar bicicletas
4. Volver

Ingrese su opcion deseada:
2

Ingrese la ruta de su archivo con los nuevos datos:
test2.txt

AVISO -> Informacion sobreescrita con exito

Bicis actualizadas con exito

AVISO -> Informacion sobreescrita con exito

MENU DE BICIS
1. Cargar bicis
2. Actualizar bicicletas
3. Mostrar bicicletas
4. Volver

Ingrese su opcion deseada:
|
```

## Cargar usuarios.

```
MENU OPERATIVO
1. Cargar y mostrar parqueos
2. Mostrar y asignar bicicletas
3. Cargar usuarios
4. Estadísticas
5. Volver

Ingrese su opcion deseada:
3

Ingrese su archivo de usuarios:
test3.txt

AVISO -> Informacion agregada con exito

Usuarios añadidos con exito!!

MENU OPERATIVO
1. Cargar y mostrar parqueos
2. Mostrar y asignar bicicletas
3. Cargar usuarios
4. Estadísticas
5. Volver

Ingrese su opcion deseada:
```



## Opciones generales.

```
Bienvenido al Menú
Selecciona una opción:
1. Opciones especificas
2. Opciones generales
3. Estadisticas
4. Salir
2
```

```
Opciones operativas:
Selecciona una opción:
1. Consultar
2. alquilar
3. facturar
4. Volver
|
```

## Consultar parqueo.

```
Opciones operativas:
Selecciona una opción:
1. Consultar
2. alquilar
3. facturar
4. Volver
1
Ingresa la cordenada x:
205
Ingresa la cordenada y:
105

El parqueo mas cercano es: puertoParks

=====bicicletas asociadas=====
Codigo de la bicicleta: B009 Tipo: TR Id del parqueo actual 3

=====bicicletas asociadas=====

Opciones operativas:
Selecciona una opción:
1. Consultar
2. alquilar
3. facturar
4. Volver
|
```

## Alquilar.

```
Opciones operativas:
Selecciona una opción:
1. Consultar
2. alquilar
3. facturar
4. Volver
2
Ingrese la cedula:
7-295-394

Ingrese el parqueo de inicio:
3

Ingrese el parqueo de llegada:
1

=====bicicletas asociadas=====
===== En caso de no existir ingrese '-' =====

Codigo de la bicicleta: B009 Tipo: TR Id del parqueo actual 3

=====bicicletas asociadas=====
Ingrese el codigo de la bicicleta:
B009

AVISO -> Informacion agregada con exito

AVISO -> Informacion sobreescrita con exito
Transaccion completada, Id del alquiler: 626

Opciones operativas:
Selecciona una opción:
1. Consultar
2. alquilar
3. facturar
4. Volver
|
```

## Facturar.

```
Opciones operativas:
Selecciona una opción:
1. Consultar
2. alquilar
3. facturar
4. Volver
3
Ingrese el ID del alquiler:
626

Nombre de la empresa: Corella Bikes  Pagina: bikesCorella.net  Correo: bikesCorr@yahoo.com
Nombre del usuario: Randall Corella
Parqueo de salida: puertoParks
Parqueo de salida: parqueoSj
kilometros: 66.40030120413611
Total en colones: 2324.010542144764

AVISO -> Informacion sobreescrita con exito

AVISO -> Informacion sobreescrita con exito

AVISO -> Informacion agregada con exito

Opciones operativas:
Selecciona una opción:
1. Consultar
2. alquilar
3. facturar
4. Volver
|
```

## Estadísticas.

```
Menu General
Bienvenido al Menú
Selecciona una opción:
1. Opciones específicas
2. Opciones generales
3. Estadísticas
4. Salir
3

Estadísticas:
Selecciona una opción:
1. Top 5 de bicicletas con más viajes, indicar bicicleta y cantidad de viajes.
2. Top 5 de parqueos con más viajes (salida + destino) indicar parqueo y cantidad de viajes
3. Top 3 de usuarios con más kilómetros recorridos (según fórmula de distancia). Indicar usuario y cantidad.
4. Resumen: total de viajes, total de kilómetros y total facturado (facturas generadas)
5. Volver
|
```

motor, el hecho de viajar unos pocos kilómetros en | Con algunos archivos, una alternativa que tambie

## Top 5 bicicletas.

Estadísticas:

Selecciona una opción:

1. Top 5 de bicicletas con más viajes, indicar bicicleta y cantidad de viajes.
2. Top 5 de parqueos con más viajes (salida + destino) indicar parqueo y cantidad de viajes
3. Top 3 de usuarios con más kilómetros recorridos (según fórmula de distancia). Indicar usuario y cantidad.
4. Resumen: total de viajes, total de kilómetros y total facturado (facturas generadas)
5. Volver

1

El código de la bicicleta es: B004 La cantidad de viajes: 6  
El código de la bicicleta es: B005 La cantidad de viajes: 3  
El código de la bicicleta es: B003 La cantidad de viajes: 3  
El código de la bicicleta es: B009 La cantidad de viajes: 2  
El código de la bicicleta es: B014 La cantidad de viajes: 1

## Top 5 parqueos.

Estadísticas:

Selecciona una opción:

1. Top 5 de bicicletas con más viajes, indicar bicicleta y cantidad de viajes.
2. Top 5 de parqueos con más viajes (salida + destino) indicar parqueo y cantidad de viajes
3. Top 3 de usuarios con más kilómetros recorridos (según fórmula de distancia). Indicar usuario y cantidad.
4. Resumen: total de viajes, total de kilómetros y total facturado (facturas generadas)
5. Volver

2

El id del parqueo es: 3 La cantidad de viajes: 16  
El id del parqueo es: 4 La cantidad de viajes: 7  
El id del parqueo es: 5 La cantidad de viajes: 3  
El id del parqueo es: 2 La cantidad de viajes: 3  
El id del parqueo es: 7 La cantidad de viajes: 2

## Top 3 usuarios.

Estadísticas:

Selecciona una opción:

1. Top 5 de bicicletas con más viajes, indicar bicicleta y cantidad de viajes.
2. Top 5 de parqueos con más viajes (salida + destino) indicar parqueo y cantidad de viajes
3. Top 3 de usuarios con más kilómetros recorridos (según fórmula de distancia). Indicar usuario y cantidad.
4. Resumen: total de viajes, total de kilómetros y total facturado (facturas generadas)
5. Volver

3

Cédula del usuario: 7-295-394 Kilómetros recorridos: 4476.047  
Cédula del usuario: 7-275-394 Kilómetros recorridos: 0.0  
Cédula del usuario: 1-575-956 Kilómetros recorridos: 0.0

## Descripción del problema

-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.

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

## Diseño del programa

Para este programa se utilizó sobre todo recursión para manejar las listas, creamos funciones como Split, esEntero, leerArchivo, agregarAArchivo o sobrescribirArchivo, las cuales fueron vitales para manejar todo el tema de archivos, para validar entradas de archivos, dividíamos las posiciones de las de filas por medios de las “;”, y así se nos hacía más fácil acceder a ciertos datos, en algunos casos se tuvieron que usar Brackets para el manejo de hilos con algunos archivos, una alternativa que también usamos fueron los archivos auxiliares que ayudaran a manejar dichos hilos sin necesidad de Brackets, también en algunos puntos se utilizaron handles para manejar los archivos, para casi todo el programa se uso la programación funcional, haciendo uso del tipado, recursión y estructuras condicionales como if, else if, else y cases, para los menús, decidimos implementar un menú principal que condujera a los dos submenús “Opciones operativas y Opciones Generales”, en opciones generales se usó un submenú para “asignar y mostrar bicicletas”, ya que esta tenía muchas opciones, entonces pensamos que lo más amigable para el usuario es implementarlo de esa manera.

Además, que nunca se pensó en el proyecto como un programa sin persistencia, por ende se realizó con esto por default, logrando así la sección de puntos extra.

## Librerías usadas

`System.IO`: Es usada para entrada y salida, además de otras funciones como `putStrLn` o `putStr`, que son otra alternativa al `print`.

`System.Directory (doesPathExist)`: Es un módulo de `System.Directory` el cual verifica si una ruta existe en el administrador de archivos del sistema.

`Control.Exception`: Es una librería encargada del manejo de excepción, ejemplos de estas pueden ser `try`, `catch` o `Brackets`.

`System.Directory`: Módulo encargado del manejo de archivos del sistema de manera más amplia.

`Control.Concurrent`: Librería para manejar programación paralela como hilos o cosas por el estilo.

`Text.Read (readMaybe)`: Es un módulo que permite analizar si es posible hacer el parseo de un tipo a otro.

`Data.IORef`: Es un módulo en Haskell que proporciona una forma de crear y manipular referencias a datos mutables dentro del sistema de tipos puros de Haskell. Estas referencias se conocen como "variables de referencia" y se utilizan para mantener datos mutables en un contexto de código puro.

`Data.Char (toUpper)`: Librería que manipula los caracteres, en este caso se utiliza la función `toUpper`, la cual realiza o pasa los caracteres a su símbolo mayúscula si es posible/existe.

`import Control.Concurrent`: Este módulo proporciona facilidades para la programación concurrente en Haskell. Permite crear hilos de ejecución y sincronizar su ejecución.

`import System.IO.Unsafe (unsafePerformIO)`: El módulo `System.IO.Unsafe` se utiliza para realizar operaciones de entrada/salida (IO) en código puro de Haskell utilizando la función `unsafePerformIO`.

`import Data.List (sortOn)`: `Data.List` proporciona varias funciones para trabajar con listas en Haskell. La función `sortOn` se utiliza para ordenar una lista.

`import Data.Function (on)`: `Data.Function` proporciona varias funciones de orden superior para trabajar con funciones en Haskell. La función `on` se utiliza

comúnmente para especificar una función de proyección en combinación con funciones de ordenación, como `sortBy` o `sortOn`.

`import Control.DeepSeq`: Este módulo se utiliza para realizar evaluación en profundidad (deep evaluation) de estructuras de datos.

`import Data.List (sortBy)`: `Data.List` proporciona funciones para trabajar con listas en Haskell. La función `sortBy` se utiliza para ordenar una lista utilizando una función de comparación personalizada que determina el orden de los elementos en la lista.



Análisis de resultados

Información comercial

LOGRADO

Carga y mostrar parqueos

LOGRADO

Mostrar y asignar bicicletas

LOGRADO

Carga usuarios

LOGRADO

Estadísticas

LOGRADO EXEPTUANDO LA GENERAL.

Consultar bicicletas

LOGRADO

Alquilar

LOGRADO

Facturar

LOGRADO

Puntos extra

Persistencia de datos

LOGRADO

Bitácora

Repositorio de GitHub

<https://github.com/FR3DD221/Lenguajes-Proye2.git>