

# Instituto Tecnológico de Costa Rica

## Escuela de Ingeniería en Computación

Centro Académico de Limón



### Proyecto Programado 02 - Alquiler de bicicletas

#### Lenguajes de programación

Hansol Antay & Alex Sánchez

II Semestre, Año 2022

Fecha de entrega: 27 de setiembre, 2022

# Tabla de contenidos

<b>1</b>	<b>Introducción</b>	<b>4</b>
1.1	Breve descripción del proyecto . . . . .	4
1.2	Descripción técnica del proyecto . . . . .	4
1.3	Descripción del problema . . . . .	4
<b>2</b>	<b>Diseño del programa</b>	<b>4</b>
2.1	Modelos (data's) . . . . .	4
2.1.1	Bicicleta . . . . .	5
2.1.2	Comercio . . . . .	5
2.1.3	Factura . . . . .	5
2.1.4	Parqueo . . . . .	6
2.1.5	Recibo . . . . .	6
2.2	Cálculo de distancia de bicicletas . . . . .	7
2.3	Algoritmos utilizados . . . . .	7
2.3.1	Mostrar top n de cierto criterio . . . . .	7
2.3.2	Generar identificador para una nueva factura . . . . .	8
2.3.3	Obtener parqueo más cercano . . . . .	9
<b>3</b>	<b>Manual de usuario</b>	<b>9</b>
3.1	Requerimientos no funcionales . . . . .	9
3.2	Compilación del programa . . . . .	10
3.3	Menú principal . . . . .	10
3.4	Menú de opciones operativas . . . . .	11
3.4.1	Gestión del comercio (información comercial) . . . . .	12
3.4.2	Gestión de parqueos . . . . .	13
3.4.3	Gestión de bicicletas . . . . .	16
3.4.4	Mostrar usuarios registrados . . . . .	17
3.4.5	Menú de estadísticas . . . . .	18
<b>4</b>	<b>Menú de opciones generales</b>	<b>20</b>
<b>5</b>	<b>Dependencias externas</b>	<b>24</b>
<b>6</b>	<b>Análisis de Resultados</b>	<b>24</b>
<b>7</b>	<b>Fuentes digitales</b>	<b>25</b>
<b>8</b>	<b>Librerías utilizadas</b>	<b>25</b>

## 9 Conclusiones

## Tabla de figuras

1	Teorema de pitágoras . . . . .	7
2	Algoritmo para mostrar top 3 bicicletas con más recorrido . . . . .	8
3	Algoritmo para generar id's de facturas . . . . .	8
4	Algoritmo para obtener el parqueo más cercano . . . . .	9
5	Compilación correcta y ambiente listo . . . . .	10
6	Menú principal del programa . . . . .	11
7	Menú de opciones operativas . . . . .	11
8	Menú de gestión del comercio . . . . .	12
9	Mostrando la información del comercio en consola . . . . .	12
10	Mostrando la información del comercio en consola . . . . .	13
11	Menú de gestión de parqueos . . . . .	13
12	Mostrando todos los parqueos en consola . . . . .	14
13	Mostrando todos los parqueos de una provincia . . . . .	14
14	Mostrando todas las bicicletas en consola . . . . .	15
15	Mostrando bicicletas en tránsito . . . . .	15
16	Menú de gestión de bicicletas . . . . .	16
17	Mostrando todas las bicicletas (B) . . . . .	16
18	Mostrando bicis por tipo . . . . .	17
19	Mostrando los usuarios registrados en consola . . . . .	17
20	Menú de estadísticas . . . . .	18
21	Top 5 usuarios con más viajes . . . . .	18
22	Top 5 parqueos con más viajes . . . . .	19
23	Top 3 bicis con más kilómetros recorridos . . . . .	19
24	Resumen del comercio . . . . .	20
25	Menú de opciones generales . . . . .	20
26	Mostrando parqueo más cercano (A) . . . . .	20
27	Mostrando parqueo más cercano (B) . . . . .	21
28	Alquilando una bicicleta (A) . . . . .	21
29	Alquilando una bicicleta (B) . . . . .	22
30	Alquilando una bicicleta (C) . . . . .	22
31	Facturando alquiler (A) . . . . .	23
32	Facturando alquiler (B) . . . . .	23
33	Lista de cotejo (según requerimientos) . . . . .	24

# 1 Introducción

## 1.1 Breve descripción del proyecto

Sistema de software basado en interacción directa con la consola de comandos mediante menús de opciones funcionales que permitirá la gestión de un sistema de alquiler de bicicletas. Este sistema permitirá gestionar las actividades de gestión e inventario de bicicletas, clientes (usuarios), facturación y parqueos.

## 1.2 Descripción técnica del proyecto

Este sistema se desarrolló en **Haskell** mediante el uso de una interfaz de consola e implementado para hacer uso de **data-types (data)**. Diseñado para ejecutarse en cualquier sistema operativo que soporte **ghci - stack**, ya que se desarrolló el proyecto utilizando esta herramienta. Adicionalmente, se utilizaron archivos **.json** para la persistencia de datos de forma local.

## 1.3 Descripción del problema

Se requiere de un sistema capaz de gestionar labores sencillas de un local de alquiler de bicicletas con distintos parqueos previamente registrados en el sistema. Entre los requerimientos solicitados se encuentran la carga e impresión (muestra) de los parqueos, bicicletas, usuarios, estadísticas, facturación y alquiler de las mismas (estos divididos en *opciones operativas* y *opciones generales*).

# 2 Diseño del programa

El sistema se diseñó para que se utilizara a través de la **consola de comandos** y la persistencia de datos se dio a través de archivos **.json**. Los modelos se implementaron a través de estructuras **data**, y estos se detallan a continuación:

## 2.1 Modelos (data's)

Se definirán los modelos utilizados en la aplicación que facilitaron el manejo y almacenamiento de datos. Estos modelos se encuentran implementados en el directorio (**/programa/src/models/**).

### 2.1.1 Bicicleta

Modelo correspondiente a las bicicletas que son propiedades del local comercial, son las podrán ser alquiladas por el usuario y serán el eje económico de la empresa.

1. **codigo (String)**: El código de la bicicleta, será el identificador propio del objeto.
2. **tipo (String)**: El tipo de bicicleta, este podrá ser (**TR** - Tradicional y **AE** - Asistencia eléctrica).
3. **parqueo (String)**: El nombre del parqueo en el que se encuentra la bicicleta, este también puede tener el valor de "**En tránsito**".

### 2.1.2 Comercio

Modelo que servirá para representar de forma persistente la información del comercio (local comercial). Se diseñó el programa de forma que solo pueda existir un solo comercio, por lo que debe considerarse en una futura extensión del programa.

1. **nombre (String)**: El nombre del local comercial.
2. **web (String)**: Dirección web del local comercial.
3. **contacto (String)**: Correo electrónico o número de teléfono (Según sea requerido).
4. **tarifaColonesKmPedal (Double)**: La tarifa cobrada para las bicicletas de tipo (TR - tradicional).
5. **tarifaColonesKmElectrico (Double)**: La tarifa cobrada para las bicicletas de tipo (AE - asistencia eléctrica).

### 2.1.3 Factura

**Nota importante:** debido a negligencia estudiantil se interpretó mal este apartado, el nombre correcto de este modelo sería "**Alquiler**", es por eso que luego se define un modelo "**Recibo**".

1. **idFactura (Int)**: número pseudo-aleatorio generado por el programa, será el identificador único del objeto.
2. **cedulaCliente (String)**: cédula del cliente asociado al alquiler (factura).

3. **parqueoSalida (String)**: nombre del parqueo de salida (punto de partida de la bicicleta).
4. **parqueoLlegada (String)**: nombre del parqueo de llegada (punto de llegada de la bicicleta).
5. **codigoBici (String)**: código de la bicicleta alquilada.
6. **estado (String)**: estado en el que se encuentra el alquiler (activo/facturado).

#### 2.1.4 Parqueo

Modelo correspondiente a los parqueos registrados por el local comercial, estos servirán para definir las localizaciones de las bicicletas, además de servir como referencia de los puntos de llegada y salida de los alquileres que permitirán definir el monto cobrado en cada factura.

1. **nombre (String)**: nombre del parqueo.
2. **dirección (String)**: dirección del parqueo (prosa).
3. **provincia (String)**: provincia en donde se encuentra el parqueo.
4. **x (Double)**: coordenada en x del parqueo.
5. **y (Double)**: coordenada en y del parqueo.

#### 2.1.5 Recibo

Modelo correspondiente a las verdaderas facturas, que serán el detalle de todos los alquileres en calidad de "**cancelado**".

1. **idFactura (Int)**: identificador del alquiler (Factura).
2. **cedulaCliente (String)**: cédula del cliente asociado a la factura.
3. **parqueoSalida (String)**: nombre del parqueo de salida de la bicicleta.
4. **parqueoLlegada (String)**: nombre del parqueo de llegada de la bicicleta.
5. **codigoBici (String)**: identificador de la bicicleta.
6. **tipoBici (String)**: el tipo de la bicicleta.

7. **tarifaPorKilometro**: tarifa por kilómetro aplicada en la facturada.
8. **distanciaRecorridaEnKm**: total recorrido en kilómetros por la bicicleta.
9. **costoTotal**: costo total cobrado al cliente ( $\text{tarifaPorKilometro} * \text{distanciaRecorridaEnKm}$ )

## 2.2 Cálculo de distancia de bicicletas

Debido que se trabajar con coordenadas (x,y) se planteó y se decidió utilizar el teorema de pitágoras. La imagen [1] se sacó de la página web [sciencenotes.org](http://sciencenotes.org).

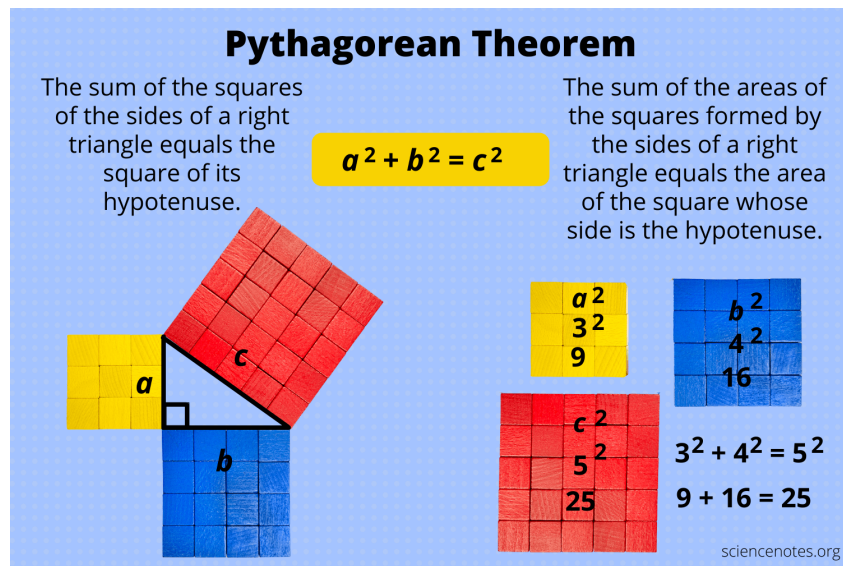


Figure 1: Teorema de pitágoras

## 2.3 Algoritmos utilizados

Algunos de los algoritmos utilizados en el programa se detallarán mediante gráficos y/o pseudocódigo:

### 2.3.1 Mostrar top n de cierto criterio

Para sacar los tops (sección de estadística) se seguirá el siguiente esquema, que es el propio para sacar top 3 bicicletas con más distancia recorrida. El algoritmo es el mismo para los demás variando en la cantidad de posiciones en el top.



Algoritmo: mostrar top 3 bicis con más distancia recorrida.

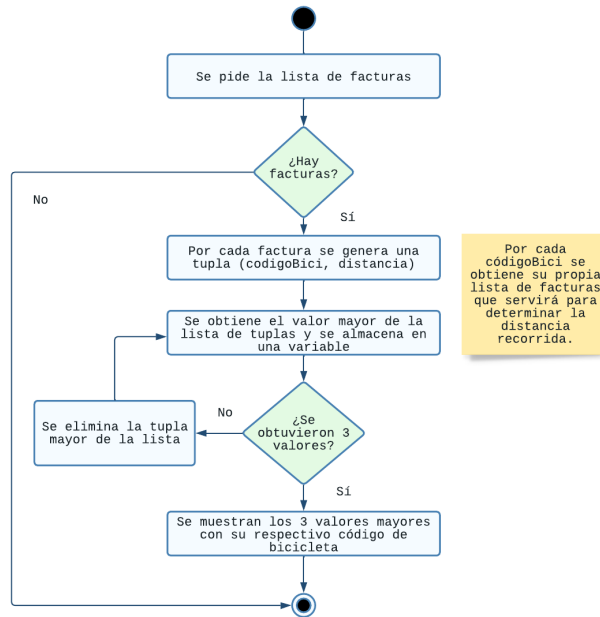


Figure 2: Algoritmo para mostrar top 3 bicicletas con más recorrido

### 2.3.2 Generar identificador para una nueva factura

Algoritmo: generar id de factura.

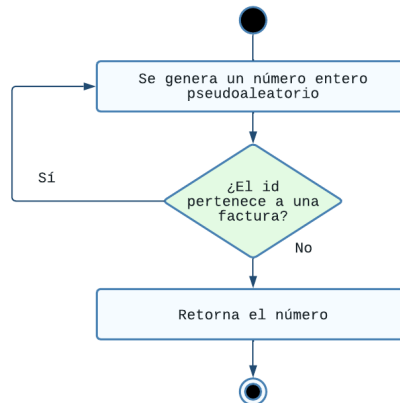


Figure 3: Algoritmo para generar id's de facturas

### 2.3.3 Obtener parqueo más cercano

Algoritmo: retornar el parqueo más cercano.

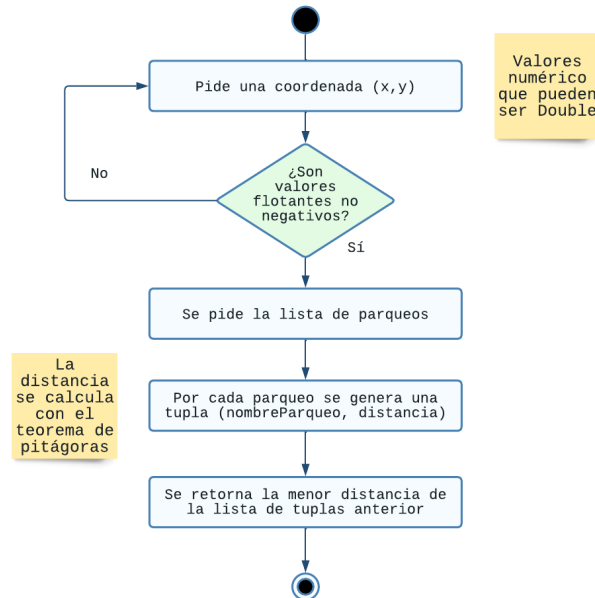


Figure 4: Algoritmo para obtener el parqueo más cercano

## 3 Manual de usuario

### 3.1 Requerimientos no funcionales

A continuación se enumera el software y librerías que son necesarios para la ejecución del programa.

1. **Sistema operativo:** Windows 10 o Windows 11. No se han realizado pruebas de funcionalidad en versiones anteriores o posteriores a estas.
2. **Glasgow Haskell Compiler:** se puede seguir la guía oficial en la [página de Haskell.org](https://www.haskell.org/) o seguir [este vídeo](#) paso a paso para instalar usando ghcup.
3. **The Haskell Tool Stack:** esta herramienta es obligatoria para la ejecución del programa, debido a las librerías que se utilizaron para el manejo de datos y persistencia de archivos. Puede seguir [esta página](#) para su instalación. Una

vez instalado, basta con ejecutar el comando (**stack build**) en la consola de comandos (en la ubicación **/programa/**).

### 3.2 Compilación del programa

Una vez con el ambiente listo, siguiendo la última instrucción, se ejecuta el comando (**stack repl**) dentro de la misma carpeta (**/programa/**). En caso de que todo haya sido preparado correctamente, deberá aparecer listo el ambiente del ghci para empezar a ejecutar la aplicación (como se muestra a continuación):

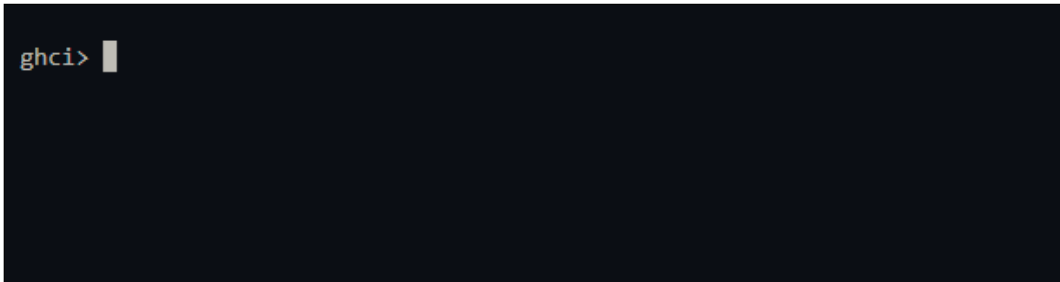


Figure 5: Compilación correcta y ambiente listo

De ser así, basta con llamar a la función **main**, que será el punto inicial del programa. Que mostrará el menú principal de la aplicación.

### 3.3 Menú principal

Este menú ofrecer las opciones de **opciones operativas**, **opciones generales** y **salir** del programa.

```
===== [MENU PRINCIPAL] =====  
1. Opciones operativas  
2. Opciones generales  
0. Sair  
Digite la opcion deseada:  
> 
```

Figure 6: Menú principal del programa

### 3.4 Menú de opciones operativas

Las opciones operativas se muestran y detallan a continuación:

```
===== [MENU OPERATIVO] =====  
1. Gestión del comercio (info. comercial)  
2. Gestión de los parqueos  
3. Gestión de las bicicletas  
4. Mostrar usuarios registrados  
5. Estadísticas  
0. Volver al menú principal  
Digite la opcion deseada:  
> 
```

Figure 7: Menú de opciones operativas

### 3.4.1 Gestión del comercio (información comercial)

```
===== [GESTION DEL COMERCIO] =====  
1. Mostrar información del comercio.  
2. Editar información del comercio.  
0. Volver al menú operativo.  
Digite la opción deseada:  
> 
```

Figure 8: Menú de gestión del comercio

Las opciones del menú se detallan a continuación:

1. **Mostrar información del comercio:** muestra la información básica del comercio.

```
===== [INFORMACION DEL COMERCIO] =====  
Nombre: CleTicas  
Web: https://cleticas.cr  
Contacto: hola@cleticas.cr  
Tarifa por kilómetro de pedal: 400.0  
Tarifa por kilómetro de eléctrico: 705.5  
=====  
[!] Presione una tecla para continuar... [!]  
 
```

Figure 9: Mostrando la información del comercio en consola

2. **Editar información del comercio:** permite editar la información básica del comercio, mostrando los datos actuales y solicitando los nuevos desde consola.

```
===== [INFORMACION ACTUAL DEL COMERCIO] =====  
Nombre: CleTicas  
Web: https://cleticas.cr  
Contacto: hola@cleticas.cr  
Tarifa por kilómetro de pedal: 400.0  
Tarifa por kilómetro de eléctrico: 705.5  
=====  
NOTA: Si desea cancelar, presione (enter) en cualquier momento.  
  
Ingrese el nuevo nombre del comercio:  
Nuevo nombre  
Ingrese la nueva dirección web:  
web.nueva.cr  
Ingrese la nueva info. de contacto:  
contacto@mail.com
```

Figure 10: Mostrando la información del comercio en consola

El usuario puede cancelar la operación presionando la tecla **enter** (dejando la entrada "vacía") en cualquier momento.

3. **Volver:** simplemente regresa al **menú de opciones operativas**.

### 3.4.2 Gestión de parqueos

```
===== [GESTION DE LOS PARQUEOS] =====  
1. Mostrar todos los parqueos.  
2. Mostrar parqueos por provincia.  
3. Mostrar bicicletas de parqueo. (OP.3)  
0. Volver al menú operativo.  
Dígame la opción deseada:  
>
```

Figure 11: Menú de gestión de parqueos

1. **Mostrar todos los parqueos:** imprime la información de cada uno de los parqueos en consola. Primero pide una ruta en donde se encuentren dichos

parqueos, el usuario puede presionar simplemente (**enter**) para usar la ruta por defecto del sistema.

```
===== [INFORMACION DE LOS PARQUEOS] =====
Ingrese la ubicacion del archivo (enter para el valor por defecto):
Nombre del Parqueo: PARQ01
Direccion del Parqueo: Av. 1, en el parque
Provincia del Parqueo: LI
Coordenadas del Parqueo(x,y): (111.8,76.3)
-----
Nombre del Parqueo: PARQ02
Direccion del Parqueo: Por el palo de mango
Provincia del Parqueo: SJ
Coordenadas del Parqueo(x,y): (50.5,3.4)
-----
Nombre del Parqueo: PARQ03
Direccion del Parqueo: Por ahí en el centro
Provincia del Parqueo: GU
Coordenadas del Parqueo(x,y): (175.4,201.2)
-----

[!] Presione una tecla para continuar... [!]
```

Figure 12: Mostrando todos los parqueos en consola

2. **Mostrar parqueos por provincia:** pide una ruta para leer el archivo con la información de todos los parqueos (el usuario puede digitar **enter** para seleccionar la ruta por defecto), pide la provincia (LI, PU, GU, AL, SJ, CA, HE) y muestra todos los parqueos que se encuentren en dicha provincia.

```
===== [PARQUEOS POR PROVINCIA] =====
Ingrese la ubicacion del archivo (enter para el valor por defecto):
Ingrese la provincia: LI
=====
Nombre del Parqueo: PARQ01
Direccion del Parqueo: Av. 1, en el parque
Provincia del Parqueo: LI
Coordenadas del Parqueo(x,y): (111.8,76.3)
-----

[!] Presione una tecla para continuar... [!]
```

Figure 13: Mostrando todos los parqueos de una provincia

3. **Mostrar bicicletas de un parqueo en específico:** pide la ruta del archivo (el usuario puede digitar **enter** para seleccionar la ruta por defecto), puede digitar '#' para mostrar todas las bicicletas o también '**transito**' para mostrar todas las bicicletas en tránsito.

```
===== [BICICLETAS DE UN PARQUEO] =====
Ingrese la ubicacion del archivo (enter para el valor por defecto):
('#' para todos los parqueos/ 'transito' para bicicletas en transito)
Ingrese el nombre del parqueo: #
=====
Código: B001
Tipo: TR
Parqueo: PARQ01
-----
Código: B002
Tipo: AE
Parqueo: PARQ01
-----
Código: B003
Tipo: AE
Parqueo: PARQ01
-----
```

Figure 14: Mostrando todas las bicicletas en consola

```
===== [GESTION DE LOS PARQUEOS] =====
1. Mostrar todos los parqueos.
2. Mostrar parqueos por provincia.
3. Mostrar bicicletas de parqueo. (OP.3)
0. Volver al menú operativo.
Digite la opción deseada:
>3
===== [BICICLETAS DE UN PARQUEO] =====
Ingrese la ubicacion del archivo (enter para el valor por defecto):
('#' para todos los parqueos/ 'transito' para bicicletas en transito)
Ingrese el nombre del parqueo: transito
===== [ BICICLETAS EN TRANSITO ] =====
=====
[!] Presione una tecla para continuar... [!]
```

Figure 15: Mostrando bicicletas en tránsito



4. **Volver:** simplemente regresa al **menú de opciones operativas**.

### 3.4.3 Gestión de bicicletas

```
===== [GESTION DE BICICLETAS] =====  
1. Mostrar todas las bicicletas.  
2. Mostrar bicicletas por tipo.  
0. Volver al menú operativo.  
Digite la opción deseada:  
>
```

Figure 16: Menú de gestión de bicicletas

1. **Mostrar todas las bicicletas:** permite visualizar la información de todas las bicicletas registradas. Esta opción también permite al usuario ingresar un path con los objetos almacenados de forma persistente.

```
===== [TODAS LAS BICICLETAS] =====  
Ingrese la ubicacion del archivo (enter para el valor por defecto):  
=====  
Código: B001  
Tipo: TR  
Parqueo: PARQ01  
-----  
Código: B002  
Tipo: AE  
Parqueo: PARQ01  
-----  
Código: B003  
Tipo: AE  
Parqueo: PARQ01  
-----
```

Figure 17: Mostrando todas las bicicletas (B)

2. **Mostrando bicicletas por tipo:** permite consultar las bicicletas de un tipo específico (TR/AE). También solicitando inicialmente una ruta de archivo al usuario.

```
===== [BICICLETAS POR TIPO] =====
Ingrese la ubicacion del archivo (enter para el valor por defecto):
Ingrese el tipo de bicicleta (TR/AE): TR
=====
Código: B001
Tipo: TR
Parqueo: PARQ01
-----
Código: B004
Tipo: TR
Parqueo: PARQ02
-----
Código: B006
Tipo: TR
Parqueo: PARQ02
-----
```

Figure 18: Mostrando bicis por tipo

3. **Volver:** simplemente regresa al **menú de opciones operativas**.

#### 3.4.4 Mostrar usuarios registrados

Pide una ruta con el archivo con los usuarios registrados e imprimir dicha información en pantalla.

```
===== [USUARIOS REGISTRADOS] =====
Ingrese la ubicacion del archivo (enter para el valor por defecto):
=====
Cédula: 123456789
Nombre: Hansol Antay Rostran
-----
Cédula: 987654321
Nombre: Alexander Sanchez Cespedes
-----
Cédula: 123456781
Nombre: Juan Perez Marques
-----

=====
[!] Presione una tecla para continuar... [!]
```

Figure 19: Mostrando los usuarios registrados en consola

### 3.4.5 Menú de estadísticas

```
===== [ESTADISTICAS] =====  
1. Top 5 usuarios con más viajes.  
2. Top 5 parqueos con más viajes.  
3. Top 3 bicicletas con más kilómetros recorridos.  
4. Resumen  
0. Volver al menú principal.  
Digite la opción deseada:  
>
```

Figure 20: Menú de estadísticas

1. **Top 5 usuarios con más viajes:** muestra a los cinco usuarios con más viajes realizados.

```
Top 5 de usuarios con más viajes:  
1. 123456789 - 3 viajes  
2. 987654321 - 2 viajes  
3. - 0 viajes  
4. - 0 viajes  
5. - 0 viajes  
  
[!] Presione una tecla para continuar... [!]  
|
```

Figure 21: Top 5 usuarios con más viajes

2. **Top 5 parqueos con más viajes:** muestra los parqueos con más viajes registrados (ya sea como llegada o como salida).

```
Top 5 de parqueos con más viajes:
1. PARQ01 - 5 viajes
2. PARQ03 - 3 viajes
3. PARQ02 - 2 viajes
4. - 0 viajes
5. - 0 viajes

[!] Presione una tecla para continuar... [!]
```

Figure 22: Top 5 parqueos con más viajes

3. **Top 3 bicicletas con más kilómetros recorridos:** muestra la bicicletas con más kilómetros recorridos.

```
Top 3 de bicicletas con más distancia recorrida:
1. B011 - 280.32103024924834 km
2. B004 - 190.49514429507119 km
3. B015 - 140.16051512462417 km

[!] Presione una tecla para continuar... [!]
```

Figure 23: Top 3 bicis con más kilómetros recorridos

4. **Resumen:** muestra el total de viajes realizados, el total de kilómetros recorridos y el total facturado (cobrado a los clientes).

```
Total de viajes: 5
Total de distancia recorrida por todas las bicicletas: 610.9766896689437 km
Total facturado por todas las bicicletas: 372847.7879792955 $.
[!] Presione una tecla para continuar... [!]
```

Figure 24: Resumen del comercio

## 4 Menú de opciones generales

```
===== [MENU GENERAL] =====
1. Consultar parqueo más cercano (por coordenadas).
2. Alquilar bicicleta
3. Facturar
0. Volver al menú principal
Digite la opcion deseada:
>
```

Figure 25: Menú de opciones generales

1. **Consultar parqueo más cercano:** pide una ruta del archivo de los parques (puede seleccionar la ruta por defecto presionando **enter**), indica las coordenadas (x,y).

```
===== [PARQUEO MAS CERCANO] =====
Ingrese la ubicacion del archivo (enter para el valor por defecto):
Ingrese la coordenada en x: 0
Ingrese la coordenada en y: 0
```

Figure 26: Mostrando parqueo más cercano (A)

Al indicar las coordenadas, se calcula el parqueo más cercano y se muestra en pantalla, junto con la distancia al respectivo punto indicado por el usuario.

```
~~~~~  
[!!] Resultado [!!]  
El parqueo mas cercano es: PARQ02 a 50.61432603522445 metros.  
~~~~~  
[!] Presione una tecla para continuar... [!]  
█
```

Figure 27: Mostrando parqueo más cercano (B)

2. **Alquilar bicicleta:** primero se pide la ruta del archivo de bicicletas (puede seleccionar la ruta por defecto presionando **enter**), pide la cédula (existente) del usuario, los nombres de los parqueos de llegada y salida (no pueden ser el mismo parqueo), y posteriormente, se indica que la bicicleta fue alquilada.

```
===== [ALQUILER DE BICICLETA] =====  
Ingrese la ubicacion del archivo de las bicicletas (enter para el valor por defecto):  
Ingrese la cedula del usuario: 123456789  
Ingrese el parqueo de salida: PARQ01  
Ingrese el parqueo de llegada: PARQ02█
```

Figure 28: Alquilando una bicicleta (A)

```
Código: B001
Tipo: TR
-----
Código: B002
Tipo: AE
-----
Código: B003
Tipo: AE
-----
Código: B011
Tipo: AE
-----
Código: B015
Tipo: AE
-----

Ingrese el codigo de la bicicleta: 
```

Figure 29: Alquilando una bicicleta (B)

Una vez elegida una bicicleta, simplemente se muestra un mensaje y el código generado automáticamente por el sistema.

```
~~~~~
[!!] Resultado [!!]
La bicicleta ha sido alquilada. Factura: 235335
~~~~~
[!] Presione una tecla para continuar... [!]

```

Figure 30: Alquilando una bicicleta (C)

3. **Facturar:** esta opción solicita el identificador existente de una factura y posteriormente realiza a facturarla solamente si el alquiler se encuentra activo. Este incluirá todos los detalles de la factura que son mostrados al usuario.

```
===== [REALIZAR FACTURACION] =====  
Ingrese el id de la factura: 235335
```

Figure 31: Facturando alquiler (A)

```
===== [DETALLE DE FACTURA] =====  
Número de factura: 235335  
-----  
Código: B011  
Tipo: AE  
Parqueo: En tránsito  
-----  
-- > Parqueo de salida  
Nombre del Parqueo: PARQ01  
Direccion del Parqueo: Av. 1, en el parque  
Provincia del Parqueo: LI  
Coordenadas del Parqueo(x,y): (111.8,76.3)  
  
-- > Parqueo de llegada  
Nombre del Parqueo: PARQ02  
Direccion del Parqueo: Por el palo de mango  
Provincia del Parqueo: SJ  
Coordenadas del Parqueo(x,y): (50.5,3.4)  
  
Distancia recorrida: 95.24757214753559 km  
Tarifa aplicada: 705.5$ por km  
Costo total: 67197.16215008635$  
=====  
[!] Presione una tecla para continuar... [!]  
█
```

Figure 32: Facturando alquiler (B)



## 5 Dependencias externas

1. **Aeson:** librería principal para la gestión de archivos JSON.
2. **Aeson-pretty:** librería compatible con Aeson que permite el formato de archivos JSON en una forma más legible.
3. **bytestring:** librería para el acceso a ficheros externos del programa (archivos JSON).
4. **process:** librería para ejecutar comandos en consola desde código fuente.
5. **directory:** para la validación de existencia de directorios (validar que existan archivos JSON indicados por rutas del usuario).
6. **tuple:** para el acceso a tuplas por índice.
7. **random:** para la generación de números pseudoaleatorios.

## 6 Análisis de Resultados

A continuación se presentará una breve tabla con los lineamientos solicitados y el estado final en la entrega del producto. **Nota:** la persistencia de datos se realizó con todos los datos manejados en el programa al 100%.

<b>C = Completo, I = Incompleto</b>	
<b>Requerimiento</b>	<b>Resultado</b>
Persistencia de datos en todos los módulos del proyecto (facturas, alquileres, bicicletas y demás).	C
Opciones operativas (carga-muestra de parqueos, mostrar bicicletas, cargar de usuarios y todas las estadísticas).	C
Opciones generales (consulta de bicicletas, alquiler y facturación).	C

Figure 33: Lista de cotejo (según requerimientos)

## 7 Fuentes digitales

A continuación se en listarán fuente digitales que ayudaron directa o indirectamente en el desarrollo de este programa.

1. **www.schoolofhaskell.com** [está página](#) sirvió para la consulta de módulos externos que se podían utilizar, como además, sirvió para un primer acercamiento de manipulación de archivos JSON en Haskell.
2. **hackage.haskell.org** [está página](#) permitió detectar el sistema operativo para poder ejecutar comandos desde la aplicación de forma consistente.
3. **docs.haskellstack.org** [está página](#) permitió complementar el entorno stack con el módulo Aeson.
4. **stackoverflow** página de consulta principal, más directamente ayudo a resolver el problema de dependencias que generaba el entorno stack.
5. **youtube.com (ThijnandMike)**: [este vídeo](#) permitió ofrecer un acercamiento introductorio sobre como utilizar Aeson para manipular directamente los datos del archivo JSON a los **data** dentro del programa y viceversa.

## 8 Librerías utilizadas

1. **Data.Maybe**: manipulación de encapsulamiento de posibles valores de retorno.
2. **Data.Aeson**: gestor de archivos JSON desde código fuente.
3. **Data.Aeson.Encode.Pretty**: codificador en formato más legible de los JSON.
4. **Data.Tuple.Select**: permite acceder a tuplas mediante un índice.
5. **Data.ByteString.Lazy**: acceso a ficheros externos del programa (los JSON's).
6. **System.IO()**: Entrada y salida de consola.
7. **System.Info**: permite acceder al nombre del sistema operativo usado por el usuario.
8. **System.Process**: permite ejecutar líneas de comando de consola dentro del sistema.

9. **System.Directory**: usado primordialmente para validación de existencia de ficheros (rutas indicadas por el usuario).
10. **Text.Read**: lectura de valores no-numéricos como tipos de datos específicos.
11. **GHC.Generics**: para la definición de datos **genéricos** en Haskell. Permitiendo que la labor de parsing a archivos JSON sea más sencillo.

## 9 Conclusiones

Se lograron completar todos los requerimientos solicitados, adicionalmente se logró investigar e implementar exitosamente una forma de manejar archivos JSON utilizando el lenguaje de programación Haskell para una aplicación enfocada en la interacción por consola de comandos a través de menús interactivos con opciones funcionales que cumplen con persistencia de datos en todo el programa.