

**Instituto Tecnológico de Costa Rica**

**Proyecto 2: Gestión de Espacios de Trabajo**

Curso:

IC4700 – Lenguajes de programación

Profesor:

Allan Rodríguez Dávila

Estudiantes:

**Mary Paz Álvarez Navarrete**

**Keingell Moodie Jaentschke**

Carnés:

2023138604

2021495925

II Semestre 2024

## Manual de Usuario

### Compilación y Ejecución

1. Abra la terminal y entre al subdirectorío del programa del proyecto con el comando: **cd programa**.
2. Inicie el intérprete de Haskell ejecutando el siguiente comando: **ghci**
3. Una vez que GHCi esté cargado, cargue el archivo del programa con el siguiente comando: **:l programa.hs**
4. Finalmente, ejecute el programa escribiendo: **main**

### Uso

#### Instrucciones Generales:

1. Para seleccionar cualquier opción digite el numero de la opción que se muestra y presione enter.
2. Para volver a un menú o submenú seleccione la opción **Volver**.
3. Para salir del programa seleccione la opción **Salir**. Si está en otro menú solo devuélvase hasta el menú principal.

#### Instrucciones del Menú principal:

1. **Ingresar opción:** Al ejecutar el programa, el usuario verá un menú principal con las siguientes opciones:

```
--Menú Principal--  
1. Opciones Operativas  
2. Opciones Generales  
3. Salir  
Ingrese la opcion deseada:  
|
```

- Si desea importar mobiliarios, crear nuevas salas, consultar la información de una sala, ver todas las reservas realizadas y generar un informe estadístico, seleccione la opción **1 (Opciones Operativas)**. Esta opción está disponible únicamente para usuarios previamente guardados.
- Si desea crear, cancelar o modificar reservas, consultar los detalles de una reserva o verificar la disponibilidad de salas, seleccione la opción **2 (Opciones Generales)**.

## Instrucciones de Opciones Operativas

1. **Cédula del usuario:** Una vez ingrese se le pedirá la cedula del usuario esta debe ser sin espacios, guiones o ceros para indicar un espacio. Como se menciona anteriormente para poder ingresar debe ser un usuario previamente guardado.

Una vez ingresado el usuario se mostrará el siguiente menú:

```
--Opciones Operativas--
1. Crear y mostrar mobiliario
2. Cargar y mostrar sala de reunión
3. Informe de reservas
4. Volver
Ingrese la opcion deseada:
|
```

- Si desea cargar mobiliarios guardados en un archivo seleccione la opción **1 (Crear y mostrar mobiliario)**.
- Si desea crear salas o ver la información de una sala seleccione la opción **2 (Cargar y mostrar sala)**.
- Si desea ver toda la información de todas las reservas, el usuario con más reservas, la sala con más reservas y la fecha más reservada seleccione la opción **3 (Informe reservas)**.

## Instrucciones de crear y mostrar mobiliario:

1. **Mobiliarios previamente guardados:** Una ingresado a esta opción se le mostraran los mobiliarios que fueron cargados anteriormente.
2. **Ingrese la ruta del archivo:** La ruta del archivo debe tener el siguiente formato /ruta/archivo.txt
3. **Mobiliarios guardados:** Cuando ya se ingreso la ruta del archivo se le mostrara todos los mobiliarios guardados incluidos los que se acaban de cargar.

### Instrucciones de cargar y mostrar sala:

1. **Al ingresar esta opción:** Una vez ingrese a esta opción se le mostrara el siguiente menú:

```
--Salas--  
1. Cargar Sala  
2. Mostrar salas  
3. Volver  
Ingrese la opcion deseada:  
█
```

- Si desea crear una sala seleccione la opción **1 (Cargar sala)**.
- Si desea ver toda la información de una sala seleccione **2 (Mostrar sala)**.

### Instrucciones cargar sala:

1. **Ingresar nombre de la sala:** Ingrese el nombre de la sala esta puede contener letras y números.
2. **Ingresar nombre del edificio:** Ingrese el edificio en el que se encuentra la sala. Este puede contener letras y números.
3. **Ingrese el piso:** Ingrese el número de piso en el que se encuentra la sala. Este debe ser un entero mayor a 0.
4. **Ingrese la ubicación:** Ingrese la ubicación de la sala. Esta puede ser una oración y puede incluir tanto letras como números.
5. **Ingrese la capacidad:** Ingrese la capacidad de la sala, esta debe ser un número entero mayor a 0.
6. **Agregue el mobiliario:** Se le mostrara una lista de la información de los mobiliarios guardados. Para que así vea el id de estos. Ingrese el id del mobiliario para agregarlo.
7. **Continuar agregando:** Si dese agregar otro mobiliario cuando se le haga la pregunta seleccione "Y" para si y "N" para no.

### Instrucciones para mostrar sala:

1. **Ingresar id:** Al ingresar a esta opción se le pedirá el id de la sala que desea ver. El id o número de sala debe ser un número entero mayor a 0.

### Instrucciones de informe de reservas:

Esta opción es automática por lo que solo se mostrara la información general de la reserva, la sala relacionada a esta y el mobiliario relacionado a la sala. Después se le

mostrara los datos estadísticos como el usuario con más reservas, la sala con más reservas y la fecha más reservada.

### Instrucciones Opciones Generales:

1. **Al ingresar esta opción:** Una vez ingrese a esta opción se le mostrara el siguiente menú:

```
--Opciones Generales--
1. Gestión de reserva
2. Consultar reserva
3. Cancelar reserva
4. Modificar reserva
5. Consulta de disponibilidad de sala
6. Volver
Ingrese la opcion deseada:
```

- Si desea crear una reserva seleccione la opción **1 (Gestión de reserva)**.
- Si desea consultar la información de una reserva seleccione la opción **2 (Consultar reserva)**.
- Si desea cancelar una reserva hecha anteriormente seleccione la opción **3 (Cancelar reserva)**.
- Si desea modificar una reserva hecha anteriormente seleccione la opción **4 (Modificar reserva)**.
- Si desea consultar la disponibilidad de una sala seleccione la opción **5 (Consulta de disponibilidad de sala)**.

### Instrucciones de gestión de reserva:

1. **Ingresar cédula del usuario:** Una vez ingrese se le pedirá la cedula del usuario esta debe ser sin espacios, guiones o ceros para indicar un espacio. Como se menciona anteriormente para poder ingresar debe ser un usuario previamente guardado.
2. **Ingresar fecha:** Ingrese la fecha de cuando requiere la sala. Esta debe cumplir con el formato AAAA-MM-DD.
3. **Ingrese la sala:** Ingrese el id de la sala que se va a usar, este debe existir y ser un número entero mayor a 0.
4. **Ingresar cantidad:** Ingrese la cantidad de personas que van a estar en la sala el día de la reserva. Este debe ser un número entero y mayor a 0.

### Instrucciones de consultar reserva:

1. **Ingresar id:** Cuando se ingrese a esta opción se le pedirá al usuario la id o número de reserva. Este debe ser un número entero mayor a 0. Si la reserva existe se le mostrara solo la información general de la reserva.

#### Instrucciones de cancelar reserva:

1. **Ingresar id:** Cuando se ingrese a esta opción se le pedirá al usuario la id o número de reserva. Este debe ser un número entero mayor a 0. Si la reserva existe se cancelará.

#### Instrucciones de modificar reserva:

1. **Ingresar id:** Cuando se ingrese a esta opción se le pedirá al usuario la id o número de reserva. Este debe ser un número entero mayor a 0. Si la reserva existe pasara al siguiente paso.
2. **Modificar sala:** Se le preguntara si desea modificar la sala de la reserva; si lo desea seleccione "S" para si y "N" para no. Si selecciona "S" ingrese el nuevo id de la sala que desea.
3. **Modificar fecha.** Se le preguntara si desea modificar la fecha de la reserva; si lo desea seleccione "S" para sí y "N" para no. Si selecciona "S" ingrese la nueva fecha con formato AAAA-MM-DD.
4. **Modificar cantidad:** Se le preguntara si desea modificar la cantidad de personas de la reserva; si lo desea seleccione "S" para sí y "N" para no. Si selecciona "S" ingrese la nueva cantidad, esta debe ser un número entero mayor a 0.

#### Instrucciones de consulta de disponibilidad de sala:

1. **Al ingresar esta opción:** Cuando entra a esta opción se le mostrara un menú con las siguientes opciones:

```
--Consultas--
1. Por fecha
2. Por Rango de fechas
3. Volver
Ingrese la opcion deseada:
```

- Si desea ver la disponibilidad en una sala en una fecha específica seleccione la opción **1 (Por fecha)**
- Si dese ver la disponibilidad de una sala en un rango de fechas seleccione la opción **2 (Por rango de fechas).**

#### Instrucciones de Por fecha:

1. **Ingresar fecha:** Al seleccionar esta opción se le pedirá ingresar una fecha. Esta debe tener el formato AAAA-MM-DD.

### Instrucciones de Por fecha:

1. **Ingresar fecha inicial:** Al seleccionar esta opción se le pedirá ingresar una fecha inicial. Esta debe tener el formato AAAA-MM-DD.
2. **Ingresar fecha final:** Ingrese una fecha final. Esta debe tener el formato AAAA-MM-DD. Además, debe ser mayor que la fecha inicial.

### Pruebas de funcionalidad

#### Inicio del programa:

Al ejecutar el programa, se muestra en consola un menú con 3 diferentes opciones:

1. Opciones Operativas
2. Opciones Generales
3. Salir

```
--Menú Principal--  
1. Opciones Operativas  
2. Opciones Generales  
3. Salir  
Ingrese la opcion deseada:  
█
```

#### 1. Opciones operativas:

Al ingresar en este apartado, el programa le pide al usuario su id, y una vez ingresado el id, se abre otro submenú, con 4 diferentes opciones:

1. Crear y mostrar mobiliario
2. Cargar y mostrar sala de reunión
3. Informe de reservas
4. Volver

```
--Opciones Operativas--  
1. Crear y mostrar mobiliario  
2. Cargar y mostrar sala de reunión  
3. Informe de reservas  
4. Volver  
Ingrese la opcion deseada:  
█
```

### 1.1 Crear y mostrar mobiliario:

Al elegir esta opción se muestra una lista de todos los mobiliarios almacenados en el sistema y se le pide al usuario ingresar la dirección de un archivo para almacenar más mobiliarios en el sistema, una vez cargados los mobiliarios, los imprime.

```
Ingrese la opcion deseada:
1
Mobiliario previamente cargado:
Mobiliario {codigoM = 1, nombreMobiliario = "Silla de oficina", descripcion = "Una silla ergonomica", tipo = "Menaje"}
Mobiliario {codigoM = 2, nombreMobiliario = "Lampara de pie", descripcion = "Lampara moderna", tipo = "Electronico"}
Mobiliario {codigoM = 3, nombreMobiliario = "Mesa de comedor", descripcion = "Mesa de madera", tipo = "Menaje"}
Mobiliario {codigoM = 0, nombreMobiliario = "Mesa de Billar", descripcion = "Mesa de madera", tipo = "Menaje"}
Mobiliario {codigoM = 4, nombreMobiliario = "Silla Gamer", descripcion = "Silla de cuero", tipo = "Menaje"}
Ingresa la ruta del archivo para cargar nuevos ítems:
archivos/mobiliario.txt
Mobiliario cargado:
Mobiliario {codigoM = 4, nombreMobiliario = "Silla Gamer", descripcion = "Silla de cuero", tipo = "Menaje"}
Mobiliario {codigoM = 0, nombreMobiliario = "Mesa de Billar", descripcion = "Mesa de madera", tipo = "Menaje"}
Mobiliario {codigoM = 3, nombreMobiliario = "Mesa de comedor", descripcion = "Mesa de madera", tipo = "Menaje"}
Mobiliario {codigoM = 2, nombreMobiliario = "Lampara de pie", descripcion = "Lampara moderna", tipo = "Electronico"}
Mobiliario {codigoM = 1, nombreMobiliario = "Silla de oficina", descripcion = "Una silla ergonomica", tipo = "Menaje"}
```

### 1.2 Cargar y mostrar sala de reunión:

Se abre un submenú con 3 diferentes opciones:

1. Cargar sala
2. Mostrar salas
3. Volver

```
--Salas--
1. Cargar Sala
2. Mostrar salas
3. Volver
Ingresa la opcion deseada:
█
```



### 1.2.1 Cargar sala:

Se piden datos al usuario para almacenar una sala nueva dentro de un archivo.

```
Ingrese el nombre de la sala:  
Sala 6  
Ingrese el edificio en el que se encuentra la sala:  
Edificio 6  
Ingrese el piso en el que se encuentra la sala:  
2  
Ingrese la ubicación en el que se encuentra la sala:  
Frente al cuarto de videojuegos  
Ingrese la capacidad de la sala:  
100  
Sala añadida exitosamente
```

Además también se le pide al usuario que ingrese los mobiliarios que tendria la sala.

```
Código: 1  
Nombre: Silla de oficina  
Descripción: Una silla ergonomica  
Tipo: Menaje  
  
Elija los mobiliarios a agregar  
Ingrese el codigo del mobiliario:  
1  
Desea agregar otro mobiliario (Y/N):  
N  
Mobiliario cargado exitosamente.
```

### 1.2.2 Mostrar salas:

Dado el código de la sala, el programa devuelve la información de la sala y los mobiliarios relacionados con ella.

```
Código: 1
Nombre: Sala 1
Edificio: Edificio 1
Piso: 4
Ubicación: Frente oficina
Capacidad: 20

Mobiliario de la sala:
Código: 4
Nombre: Silla Gamer
Descripción: Silla de cuero
Tipo: Menaje
```

### 1.2.3 Volver:

El programa vuelve al menú de opciones operativas.

```
--Salas--
1. Cargar Sala
2. Mostrar salas
3. Volver
Ingrese la opcion deseada:
3
Volviendo al Opciones Operativas...

--Opciones Operativas--
1. Crear y mostrar mobiliario
2. Cargar y mostrar sala de reunión
3. Informe de reservas
4. Volver
Ingrese la opcion deseada:

```

### 1.3 Informe de reservas:

El programa muestra una serie de informes estadísticos almacenados en el sistema, como las reservas hechas por cada usuario, sala más utilizada, usuario con mayor reserva, día con mayor reserva.

```
---Reserva---
Código: 3
Reserva hecha por el usuario: Mary Paz Alvarez Navarrete
Cédula del usuario: 123456789
Código de la sala reservada: 4
Fecha: 2023-09-12
Capacidad reservada: 5 personas
---Sala---
Código: 4
Nombre: Sala 4
Edificio: Edificio 4
Piso: 4
Ubicación: Frente a la oficina B
Capacidad: 20

Mobiliario de la sala:
Código: 2
Nombre: Lampara de pie
Descripción: Lampara moderna
Tipo: Electronico
```

```
-----Usuario con mayor reservas-----
Cédula: 123456789
Nombre: Mary Paz Alvarez Navarrete
Puesto: Desarrollador

-----Día con mayor cantidad de reservas-----

--Opciones Operativas--
1. Crear y mostrar mobiliario
2. Cargar y mostrar sala de reunión
3. Informe de reservas
4. Volver
Ingrese la opcion deseada:
█
```

#### 1.4 Volver:

El programa vuelve al Menú principal.

```
--Opciones Operativas--
1. Crear y mostrar mobiliario
2. Cargar y mostrar sala de reunión
3. Informe de reservas
4. Volver
Ingrese la opcion deseada:
4
Volviendo al Menú Principal...

--Menú Principal--
1. Opciones Operativas
2. Opciones Generales
3. Salir
Ingrese la opcion deseada:
█
```

## 2. Opciones Generales:

Al ingresar en este apartado, el programa abre un submenú, con 4 diferentes opciones:

1. Gestión de reserva
2. Consultar reserva
3. Cancelar reserva
4. Modificar reserva
5. Consultar disponibilidad de sala
6. Volver

### 2.1 Gestión de reserva:

El programa le pide al usuario su id, y acto seguido los datos necesarios para almacenar una reserva en el sistema.

```
Ingrese la opcion deseada:
1

Ingrese el id de usuario:
234567891

Ingrese la fecha de la reserva (YYYY-MM-DD):
2025-09-29

Ingrese el id de la sala:
6

Ingrese el número de personas para la reserva:
90

Código: 3
Reserva hecha por el usuario: Keingell Moodie
Cédula del usuario: 234567891
Código de la sala reservada: 6
Fecha: 2025-09-29
Capacidad reservada: 90 personas

La reserva fue creada con exito...
```

## 2.2 Consultar reserva:

El programa le pide al usuario el id de alguna reserva almacenada en el sistema y muestra su información.

```
Ingrese la opcion deseada:
2
Ingrese el id de la reserva:
3

Código: 3
Reserva hecha por el usuario: Mary Paz Alvarez Navarrete
Cédula del usuario: 123456789
Código de la sala reservada: 4
Fecha: 2023-09-12
Capacidad reservada: 5 personas
```

## 2.3 Cancelar reserva:

El programa le pide al usuario el id de alguna reserva almacenada en el sistema y la elimina.

```
Ingrese la opcion deseada:
3
Ingrese el id de la reserva:
3
La reserva fue cancelada exitosamente...
```

## 2.4 Modificar reserva:

El programa le pide al usuario el id de alguna reserva almacenada en el sistema para modificarla. Para la modificación se le da la opción al usuario de modificar 3 diferentes datos de una reserva en específico.

```
Ingrese la opcion deseada:
4
Ingrese el id de la reserva:
5
El codigo ingresado no es válido
Ingrese el id de la reserva:
2
¿Desea modificar la sala? (S/N)
S
Ingrese el nuevo identificador de sala:
6
¿Desea modificar la fecha? (S/N)
N
¿Desea modificar la cantidad de personas? (S/N)
N

Código: 2
Reserva hecha por el usuario: Jorge Mendoza Valverde
Cédula del usuario: 184927427
Código de la sala reservada: 6
Fecha: 2024-02-12
Capacidad reservada: 20 personas

La reserva fue modificada con éxito...
```

## 2.5 Consulta de disponibilidad de sala:

Se abre un submenú con 3 diferentes opciones:

1. Por fecha
2. Por Rango de fechas
3. Volver

### 2.5.1 Por fecha:

El programa le pide al usuario que ingrese una fecha y devuelve un listado de todas las salas disponibles en dicha fecha.

```
Ingrese la fecha (YYYY-MM-DD):
2024-07-01
Salas disponibles en 2024-07-01:
Sala 1 (Capacidad: 20)
Sala 3 (Capacidad: 10)
Sala 4 (Capacidad: 20)
Sala 5 (Capacidad: 10)
Sala 6 (Capacidad: 100)

--Consultas--
1. Por fecha
2. Por Rango de fechas
3. Volver
Ingrese la opcion deseada:
```

### 2.5.2 Por rango de fechas:

El programa le pide al usuario que ingrese un rango de fechas y devuelve un listado de todas las salas disponibles en cada fecha de ese rango.

```
Fecha Inicial:

Ingrese la fecha (YYYY-MM-DD):
2024-07-01

Fecha Final:

Ingrese la fecha (YYYY-MM-DD):
2024-07-03

Fecha: 2024-07-01
Salas disponibles:
Sala 1 (Capacidad: 20)
Sala 3 (Capacidad: 10)
Sala 4 (Capacidad: 20)
Sala 5 (Capacidad: 10)
Sala 6 (Capacidad: 100)

Fecha: 2024-07-02
Salas disponibles:
Sala 1 (Capacidad: 20)
Sala 2 (Capacidad: 15)
Sala 3 (Capacidad: 10)
Sala 4 (Capacidad: 20)
Sala 5 (Capacidad: 10)
Sala 6 (Capacidad: 100)
```

### 2.5.3 Volver:

El programa vuelve al menú de opciones generales.

```
--Consultas--
1. Por fecha
2. Por Rango de fechas
3. Volver
Ingrese la opcion deseada:
3
Volviendo al Menu de Opciones Generales...
```

### 2.6 Volver:

El programa vuelve al menú principal.

```
Ingresa la opcion deseada:
6
Volviendo al Menú Principal...

--Menú Principal--
1. Opciones Operativas
2. Opciones Generales
3. Salir
Ingresa la opcion deseada:
█
```

### 3. Salir:

Se sale del menú principal y finaliza el programa.

```
--Menú Principal--
1. Opciones Operativas
2. Opciones Generales
3. Salir
Ingresa la opcion deseada:
3
Saliendo del programa...
```



## **Descripción del Problema**

El proyecto consiste en desarrollar un sistema de gestión de reservas para salas de reuniones que permita, de manera eficiente, importar el mobiliario, crear y consultar salas, gestionar las reservas, y generar informes estadísticos. El sistema estará basado en un menú interactivo accesible desde la consola, a través del cual el usuario podrá cargar mobiliarios, crear nuevas salas, consultar salas existentes y gestionar reservas (crearlas, modificarlas, consultarlas o eliminarlas). Además, se generarán informes para facilitar el análisis de las reservas. El programa garantizará la persistencia de los datos mediante el almacenamiento en archivos, y hará uso de estructuras de datos avanzadas (data, type), respetando los principios del paradigma funcional, como la inmutabilidad, funciones puras, y el uso de funciones de orden superior.

## **Diseño del programa**

### **Decisiones del diseño**

Para este proyecto, se ha optado por utilizar la consola de comandos como interfaz de usuario. Se han implementado mensajes claros y concisos para evitar saturar al usuario con demasiada información. Dado que el programa incluye un menú principal y submenús, se ha decidido ofrecer siempre la opción de volver y mostrar el menú o submenú actual al finalizar cada tarea.

El programa gestiona los datos en memoria para realizar las operaciones, pero para garantizar la persistencia de la información, los datos agregados, modificados o eliminados se guardan inmediatamente tanto en memoria como en archivos de texto correspondientes. Esto evita la pérdida de información en caso de una interrupción inesperada durante la ejecución del programa.

Se han elegido archivos de texto (TXT) como medio de almacenamiento para mantener la simplicidad del sistema. En caso de errores durante el manejo de estos archivos, el programa generará mensajes de error claros y específicos, para informarle al usuario sobre el problema y facilitar su resolución.

Con el fin de mantener el código sencillo y legible, se ha optado por dividir la lógica en funciones auxiliares, evitando la concentración de toda la funcionalidad en una única función. Este enfoque modular facilita la mantenibilidad y mejora la claridad del código.

## Algoritmos usados

### **manejarErrorArchivo :: IOException -> IO [[String]]**

Gestiona errores al abrir o manipular un archivo. Si el archivo no existe, muestra un mensaje y retorna una lista vacía. En caso de otro error, lo relanza.

### **manejarErrorEscribir :: IOError -> IO Bool**

Gestiona errores al escribir en un archivo. Muestra un mensaje con la descripción del error y retorna False como indicador del fallo.

### **leerArchivo :: FilePath -> IO [[String]]**

Lee el contenido de un archivo especificado por la ruta proporcionada y devuelve una lista de listas de String. Si ocurre un error, como que el archivo no exista, se gestiona con la función manejarErrorArchivo.

### **leerLineas :: Handle -> IO [[String]]**

Lee línea por línea un archivo usando un Handle, separando cada línea en campos por medio de comas (,) y almacenando el contenido en una lista de listas de String. Si ocurre un error de fin de archivo, retorna una lista vacía.

### **escribirAppendArchivo :: FilePath -> String -> IO Bool**

Escribe una cadena de texto al final de un archivo en la ruta especificada. Si la escritura es exitosa, devuelve True. En caso de error, gestiona el error y retorna False.

### **esEntero :: String -> Bool**

Verifica si una cadena de texto puede ser interpretada como un número entero. Devuelve True si el valor puede ser convertido a entero, y False en caso contrario.

### **validarFecha :: String -> IO Bool**

Verifica si una cadena de texto cumple con el formato de fecha YYYY-MM-DD. Retorna True si el formato es válido y False en caso contrario.

### **verificarReserva :: Reservas -> Int -> IO Bool**

Verifica si un identificador de reserva existe en la lista de reservas. Retorna True si la reserva con el ID proporcionado se encuentra en la lista y False si no.

### **verificarSala :: Salas -> Int -> IO Bool**

Verifica si un identificador de sala existe en la lista de salas. Retorna True si la sala con el ID proporcionado se encuentra en la lista y False si no.

### **verificarMobiliario :: Mobiliarios -> Int -> IO Bool**

Verifica si un identificador de mobiliario existe en la lista de mobiliarios. Retorna True si el mobiliario con el ID proporcionado se encuentra en la lista y False si no.

#### **verificarUsuario :: String -> IO Bool**

Verifica si un identificador de usuario existe en el archivo usuarios.txt. Retorna True si el usuario está presente en el contenido del archivo y False si no.

#### **getNombreUsuario :: String -> IO String**

Devuelve el nombre de un usuario dado su identificador. Busca en el archivo usuarios.txt y retorna el nombre correspondiente al primer usuario que coincide con el ID proporcionado. Si no se encuentra el ID, puede generar un error.

#### **getPuestoUsuario :: String -> IO String**

Devuelve el puesto de un usuario dado su identificador. Busca en el archivo usuarios.txt y retorna el puesto correspondiente al primer usuario que coincide con el ID proporcionado. Si no se encuentra el ID, puede generar un error.

#### **getMobiliarioSala :: [String] -> MobiliarioSala**

Convierte una lista de cadenas en una estructura MobiliarioSala. Se espera que la lista contenga el código de la sala y el código del mobiliario, que se transforman en tipos numéricos y se asignan a los campos correspondientes de la estructura.

#### **getSala :: [String] -> Sala**

Convierte una lista de cadenas en una estructura Sala. Se espera que la lista contenga el código, nombre, edificio, piso, ubicación y capacidad de la sala. Los campos numéricos se convierten de cadena a tipo numérico.

#### **getReserva :: [String] -> Reserva**

Convierte una lista de cadenas en una estructura Reserva. Se espera que la lista contenga el código de la reserva, código del usuario, código de la sala, fecha y cantidad. Los campos numéricos se convierten de cadena a tipo numérico.

#### **reservaToString :: Reserva -> String**

Convierte una estructura Reserva en una cadena de texto. La cadena resultante contiene los campos de la reserva separados por comas.

#### **guardarReservas :: FilePath -> Reservas -> IO ()**

Reescribe el archivo de reservas en la ruta especificada. Convierte cada Reserva en su representación en cadena y escribe todas las reservas en el archivo, separando cada una por una nueva línea.

**cancelarReserva :: Reservas -> IO Reservas**

Elimina una reserva del archivo y de la memoria utilizando su ID. Solicita al usuario el ID de la reserva y verifica si existe. Si es así, filtra la lista de reservas para excluir la cancelada, actualiza el archivo de reservas y retorna la nueva lista. En caso de que el ID no sea válido, solicita al usuario que ingrese uno nuevo.

**mostrarInfoReserva :: Reserva -> IO ()**

Imprime en pantalla la información de una reserva. Muestra el código de la reserva, el nombre del usuario que la hizo, su cédula, el código de la sala reservada, la fecha de la reserva y la capacidad reservada en personas.

**consultarReserva :: Reservas -> IO ()**

Muestra la información de una reserva dada su ID. Solicita al usuario el ID de la reserva, verifica su existencia y, si se encuentra, llama a mostrarInfoReserva para imprimir la información correspondiente. Si no es válida, pide al usuario que intente nuevamente.

**preguntarXUsuario :: IO String**

Solicita al usuario el ID de un usuario y verifica su existencia. Si el ID es válido, lo devuelve; si no, informa al usuario y vuelve a solicitar el ID.

**preguntarXFecha :: IO String**

Solicita al usuario una fecha para la reserva y valida su formato. Si la fecha es válida, la devuelve; de lo contrario, informa al usuario y vuelve a solicitar una fecha.

**preguntarXSala :: Salas -> IO Int**

Solicita al usuario el ID de una sala y verifica su existencia y formato. Si el ID es válido y la sala existe, lo devuelve; si no, informa al usuario y vuelve a solicitar el ID.

**preguntarXCantidad :: IO Int**

Solicita al usuario el número de personas para la reserva y verifica que sea un número entero positivo. Si la cantidad es válida, la devuelve; de lo contrario, informa al usuario y vuelve a solicitar la cantidad.

**guardarReservas :: FilePath -> Reservas -> IO ()**

Reescribe el archivo de reservas en la ruta especificada. Toma como argumento la ruta del archivo y una lista de reservas, y escribe las reservas en el archivo utilizando el formato de texto definido por reservaToString.

**verificarDatos :: Reservas -> Salas -> String -> Int -> Int -> IO Bool**

Verifica si la sala está disponible en la fecha dada y si la cantidad de personas es válida en relación con la capacidad de la sala.

**gestionDeReservas :: Reservas -> Salas -> IO Reservas**

Solicita al usuario la información necesaria para crear una reserva, validando los datos antes de proceder a guardar la reserva.

**preguntarFecha :: IO String**

Pide al usuario que ingrese una fecha en formato 'YYYY-MM-DD', validando el formato de la fecha ingresada.

**consultarDispo :: Reservas -> Salas -> IO Reservas**

Permite al usuario consultar la disponibilidad de salas, ya sea por fecha específica o por un rango de fechas.

**mostrarSalaConsulta :: Sala -> IO ()**

Imprime el nombre y la capacidad de una sala en consulta.

**salasDisponiblesEnFecha :: Reservas -> Salas -> String -> [Sala]**

Devuelve una lista de salas disponibles en una fecha específica, excluyendo aquellas que ya están reservadas.

**generarFechasRango :: String -> String -> [String]**

Genera un arreglo de fechas en un rango específico.

**consultarPorRango :: Reservas -> Salas -> String -> String -> IO ()**

Genera un arreglo de salas disponibles en un rango de fechas.

**modificarReserva :: Reservas -> Salas -> IO Reservas**

Modifica una reserva almacenada en el sistema, permitiendo cambiar sala, fecha y cantidad de personas.

**opcionesGenerales :: Reservas -> Salas -> IO()**

Muestra un submenú con opciones generales para gestionar reservas y consultar disponibilidad.

**mostrarMobiliarios :: Mobiliarios -> IO ()**

Muestra los mobiliarios en memoria.

**mobiliarioXCodigo :: Int -> Mobiliarios -> IO ()**

Imprime en pantalla el mobiliario correspondiente al código dado.

**preguntaConfirmacion :: IO Bool**

Pregunta si el usuario desea continuar con la acción actual.

**preguntarXMobiliario :: Int -> MobiliariosSala -> Mobiliarios -> IO MobiliariosSala**

Pregunta por el mobiliario que se va a añadir a la sala y guarda la información correspondiente.

**cargarMobiliarioSala :: Salas -> MobiliariosSala -> Mobiliarios -> IO MobiliariosSala**

Le pide al usuario por el mobiliario a agregar a la nueva sala. Luego guarda en memoria y en un archivo.

**preguntarXPiso :: IO String**

Le pregunta al usuario por el piso en el que se encuentra la sala y verifica si el dato ingresado por el usuario es correcto.

**preguntarXCapacidad :: IO String**

Le pregunta al usuario por la cantidad de la sala y verifica si el dato ingresado por el usuario es correcto.

**cargarSalas :: Salas -> IO Salas**

Le pide al usuario la información para crear una sala. Luego se guarda en memoria y en un archivo y, por último, le muestra el código y su información.

**mostrarinfoSala :: Int -> Salas -> MobiliariosSala -> Mobiliarios -> IO ()**

Imprime la información de la sala.

**mostrarSala :: Salas -> MobiliariosSala -> Mobiliarios -> IO ()**

Muestra la sala dependiendo del código dado.

**cargarMostrarSalas :: Salas -> MobiliariosSala -> Mobiliarios -> IO ()**

Submenú para preguntarle si quiere cargar una sala o ver todas las salas.

**cargarYMostrarMobiliario :: Mobiliarios -> IO Mobiliarios**

Carga nuevos ítems de mobiliario desde un archivo y los muestra, guardando el mobiliario actualizado en un archivo.

**cargarMobiliario :: [[String]] -> Mobiliarios -> Mobiliarios**

Carga nuevos ítems desde una lista de filas, evitando duplicados basados en el código.

**mobiliarioToString :: Mobiliario -> String**

Convierte un Mobiliario a su representación en String, separando atributos con comas.

**guardarMobiliario :: FilePath -> Mobiliarios -> IO ()**

Guarda la lista de mobiliario en un archivo, usando la representación en String de cada ítem.

**cargarMobiliarioDesdeArchivo :: FilePath -> IO Mobiliarios**

Carga mobiliario desde un archivo y retorna una lista de Mobiliarios, evitando duplicados.

**parseMobiliario :: [String] -> Mobiliario**

Convierte una lista de strings en un Mobiliario; genera error si el formato es incorrecto.

**mostrarReservas :: Reserva -> Salas -> MobiliariosSala -> Mobiliarios -> IO ()**

Muestra información sobre una reserva específica, incluyendo detalles de la sala.

**salaMasUsada :: Reservas -> Salas -> MobiliariosSala -> Mobiliarios -> IO ()**

Identifica y muestra la sala más reservada, analizando una lista de reservas.

**usuarioMasReservas :: Reservas -> IO ()**

Identifica al usuario con más reservas realizadas y muestra su información.

**informeReservas :: Reservas -> Mobiliarios -> IO ()**

Genera un informe de reservas, mostrando detalles de cada reserva y estadísticas sobre salas y usuarios.

**opcionesOperativas :: Mobiliarios -> IO()**

Muestra un submenú para las opciones operativas relacionadas con el mobiliario y reservas.

**menuPrincipal :: IO ()**

Muestra el menú principal de la aplicación y permite seleccionar opciones para navegar entre las funcionalidades.

**main :: IO ()**

Punto de entrada de la aplicación, inicia el menú principal.

## Librerías Usadas

### Data.List.Split (splitOn)

Usado para separar caracteres según el carácter que se le sea indicado. Para instalarla seguir estas instrucciones:

1. Se debe estar en la carpeta del programa, ejecutar: **cd programas.**
2. Abrir la terminal y ejecutar: **cabal install --lib Split.**
3. Ejecutar **ghci.**
4. Ejecutar el siguiente comando: **:m + Data.List.Split.**

### Data.Time (parseTimeM, defaultTimeLocale, Day, formatTime, parseTimeOnError)

Usado para la manipulación de fechas y horas. Para instalarla seguir estas instrucciones:

1. Se debe estar en la carpeta del programa, ejecutar: **cd programas.**
2. Abrir la terminal y ejecutar: **cabal install --lib time.**
3. Ejecutar **ghci.**
4. Ejecutar el siguiente comando: **:set -package time.**
  - **parseTimeM:** Usado para analizar una cadena de texto en tipo Day.
  - **defaultTimeLocale:** Usada para presentar la fecha en un formato correcto.
  - **Day:** Usado para presentar la fecha sin mostrar la hora.
  - **formatTime:** Usado para pasar de un tipo Day a una cadena de texto.
  - **parseTimeOnError:** Usado para manipular errores relacionados con fechas.

### System.Exit (exitSuccess)

Usado para terminar la ejecución de un programa de forma exitosa, sin ningún problema.

### Control.Exception (catch, bracket, IOException)

Usada para la manipulación de errores o excepciones en el programa.

- **catch:** Se utiliza para manejar errores en caso de que ocurran durante la ejecución de una acción de IO.
- **bracket:** Se usa para liberar recursos adecuadamente, garantizando que se liberen incluso si se produce un error o excepción.
- **IOException:** Se utiliza junto con catch para gestionar errores o excepciones específicos relacionados con operaciones de IO.



### **Data.List (group, sort)**

Usado para manipular las listas.

- **group:** Se emplea para agrupar elementos idénticos en sublistas dentro de una lista.
- **Sort:** Se utiliza para ordenar los elementos de una lista en un orden específico.

### **System.IO (Handle, hClose, hIsEOF, hGetsLine, hPutStrLn, openFile, IOMode (AppendMode, ReadMode))**

Usado para manejar lectura de archivos y para manejar entradas y salidas.

- **Handle:** Usado para manejar archivos como lo es la escritura en ellos o la lectura.
- **hClose:** Usado en conjunto con handle, es usado para cerrar un archivo después de ser usado.
- **hIsEOF:** Usado para verificar si llego al final del archivo.
- **hGetsLine:** Usado para leer la línea completa de un archivo.
- **hPutStrLn:** Usado para escribir una cadena de texto en un archivo incluyendo el salto de línea.
- **openFile:** Abre un archivo y devuelve un handle.
- **IOMode (AppendMode, ReadMode)**

Define como se va a abrir un archivo.

- **AppendMode:** En este modo el archivo se abre de manera que se pueda agregar líneas al archivo sin borrar todo su contenido.
- **ReadMode:** En este modo se abre el archivo para solo leer o extraer su contenido.

### **System.IO.Error (isEOFError, isDoesNotExistError)**

Se usa para manejar y verificar errores.

- **isEOFError:** Usado para verificar si existe un error tipo "En Of File".
- **isDoesNotExistError:** Usado para verificar los errores a la hora de abrir un archivo.

### **Text.Read ( readMaybe )**

Es usado para convertir una cadena de texto en un tipo de dato. Por ejemplo, es usado para la verificar si un dato dado por el usuario es entero.

### Análisis de resultados

Objetivo	Alcanzado	No alcanzado
Cargar y mostrar mobiliario	✓	
Cargar y mostrar sala de reuniones	✓	
Informe de reservas	✓	
Gestión de reserva	✓	
Consulta de reserva	✓	
Cancelación de reserva	✓	
Modificación de reserva	✓	
Consulta disponibilidad sala	✓	
Salir	✓	

### Enlace al Repositorio de GitHub

<https://github.com/Mary-Paz-AN/PY2-LenguajesProgramacion.git>