

## Trabajo Práctico N° 3

### Chambucheria



Fecha presentación	06/06/2023
Fecha entrega	15/06/2023

## 1. Introducción

El local de Reuben y Stitch no para de crecer, y ahora tiene muchas mesas para que los clientes puedan comer ahí. Como les cuesta organizarse solos, nos contrataron para que hagamos un sistema para llevar un control de las reservas.

## 2. Objetivo

El presente trabajo práctico tiene como objetivo que el alumno:

- Manejo correcto de archivos.
- Se familiarice con y utilice correctamente el lenguaje de programación Python.
- Desarrolle una comunicación amigable y entendible con el usuario.

Por supuesto, se requiere que el trabajo cumpla con las buenas prácticas de programación profesadas por la cátedra.

Se considerarán críticos la modularización, reutilización y claridad del código.

## 3. Enunciado

El sistema se basará en una serie de comandos que el usuario puede utilizar para guardar, modificar, eliminar y mostrar las reservas del restaurante, las cuales estarán guardadas en un archivo de tipo csv.

Una reserva está compuesta por:

- El id para identificar el pedido (se genera cuando se crea el pedido, en modo ascendente)
- El nombre a quien se hace la reserva
- La cantidad de personas en la mesa
- El horario en el que van a asistir (en formato hh:mm)
- Si quieren adentro o afuera (identificado con F si quieren afuera y D si quieren adentro)

En el archivo las reservas se guardan con el formato

```
1 id;nombre;cant_personas;HH:MM;ubicacion
2
3 Por ejemplo
4 2;Sol;6;21:00;F
```

Cabe aclarar que solamente se deben guardar en el archivo datos correctos: horarios entre las 00 y las 23:59, la cantidad debe ser positiva y la ubicación sólo puede ser F o D.

### 3.1. Agregar

Para usarlo, el usuario deberá ingresar la palabra **agregar** seguida de los datos de la reserva separados por un espacio. Es decir, si se quiere agregar una reserva en nombre de Carla para 5 personas a las 21:30 en una mesa afuera, se deberá ingresar el comando

```
1 agregar Carla 5 21:30 F
```

### 3.2. Modificar

Para usarlo, el usuario deberá ingresar la palabra **modificar** seguida del id de la reserva a cambiar (si la reserva con ese id no existe, se le debe indicar con un mensaje). Una vez elegida la reserva, le aparecerá un mensaje preguntando qué quiere cambiar, ya sea el nombre, la cantidad de personas, el horario o adentro o afuera. El usuario tiene que ingresar el dato a modificar seguido del nuevo valor, es decir:

Para modificar la primer reserva:

```
1 modificar 1
2
3 # acá iría algún mensaje para el usuario
4
5 nombre Romina
```

Si se puede modificar correctamente, se debe avisar por pantalla.

La clave para cada dato a modificar son **nombre** para el nombre, **cant** para la cantidad de personas, **hora** es para el horario y **ubicacion** para indicar el cambio de adentro y afuera.

### 3.3. Eliminar

Para usarlo, el usuario deberá ingresar la palabra **eliminar** seguida del id de la reserva a cambiar. En caso de que no exista, se le debe indicar que no existe.

```
1 eliminar 1
```

Si se puede eliminar correctamente, se debe avisar por pantalla.

### 3.4. Listar

Este comando muestra las reservas por pantalla con un formato amigable para el usuario.

Para usarlo, se deberá ingresar la palabra **listar** seguida del rango de ids que se quieren listar, o si no ingresa ningún parámetro, se listan todas las reservas.

Si se ingresa un rango incorrecto, se debe indicar con un mensaje de error.

```
1 # muestra todas las reservas
2 listar
```

```
1 # muestra las reservas desde la 3 inclusive hasta la 5 inclusive
2 listar 3 5
```

## 4. Resultado esperado

Se espera que se creen las funciones y procedimientos para que el sistema se desarrolle con fluidez. Se debe tener una buena comunicación con el usuario, avisándole de sus errores e indicando qué cambios se van haciendo en el archivo y sus errores. Así mismo, se espera que se respeten las buenas prácticas de programación.

**IMPORTANTE!** Se deberá validar que los comandos reciban los parámetros necesarios, y en caso de no recibirlos, se deberá avisar al usuario y no realizar ninguna acción.

Muchas de las funcionalidades quedan a criterio del alumno, solo se pide que se respeten las estructuras y especificaciones brindadas.

## 5. Entrega

El trabajo práctico debe ser realizado en un archivo llamado chambucheria.py.

Por último debe ser entregado en la plataforma de corrección de trabajos prácticos **AlgoTrón** (patente pendiente), en la cual deberá tener la etiqueta **iExito!** significando que ha pasado las pruebas a las que la cátedra someterá al trabajo.

**IMPORTANTE!** Esto no implica necesariamente haber aprobado el trabajo ya que además será corregido por un colaborador que verificará que se cumplan las buenas prácticas de programación.

Para la entrega en **AlgoTrón** (patente pendiente), recuerde que deberá subir un archivo **zip** conteniendo únicamente los archivos antes mencionados, sin carpetas internas ni otros archivos. De lo contrario, la entrega no será validada por la plataforma.