

# Trabajo Práctico N°2

GRUPO 5

Segundo Cuatrimestre 2023

## 1. Integrantes

- 111335 - Cettour Ignacio Ariel - icettour@fi.uba.ar
- 111560 - Pecuch Agustina - apecuch@fi.uba.ar
- 111248 - Farias Erika Aldana - eafarias@fi.uba.ar

## 2. Introducción

Este documento presenta los conceptos básicos que se realizaron en la creación de la interfaz gráfica que permite la gestión de sistemas comerciales de cines. No solo establecemos el funcionamiento del proyecto en sí, sino que también describimos los problemas que encontramos durante la realización del programa, los recursos que utilizamos, las hipótesis que hicimos y las soluciones a los mayores problemas durante la ejecución.

## 3. Palabras clave/Keywords

- Python
- Interfaz gráfica
- Tkinter
- Graphic User Interface
- GitHub
- Código QR
- Pantalla
- API

## 4. Guía de uso

Este trabajo se divide en dos aplicaciones. Por un lado, tenemos la principal en la cual se simula una compra en el cine, generando un código QR junto con un id al final. En la segunda, permite validar el id/QR generados anteriormente. En el apartado 4.2 se detallan las partes que componen cada aplicación. En este apartado se detallan los pasos a seguir para poder utilizar correctamente las aplicaciones.

### 4.1. Librerías necesarias

Utilizamos algunas librerías externas cuya instalación es necesaria para el funcionamiento del programa. En la aplicación principal, usamos *Requests* para hacer consultas a una API externa, *Pillow* para trabajar con imágenes y *qrcode* para la creación del código QR de la compra. Para la segunda aplicación, utilizamos *OpenCV* para la lectura del código mediante la cámara web y la lectura del mismo. A continuación se listan los comandos para la instalación:

- **requests:** *pip install requests*
- **Pillow:** *pip install Pillow*
- **qrcode:** *pip install qrcode*
- **OpenCV:** *pip install opencv-python*

## 4.2. Detalle aplicaciones

Para la aplicación del cine, se ejecuta el archivo *main.py*. Se abrirá una pantalla que muestra la cartelera con las películas disponibles para el totem seleccionado, con la posibilidad de modificar el mismo. Asimismo, se puede mostrar las películas que coincidan con una búsqueda. Al seleccionar una película se mostrará en pantalla su detalle, pudiendo proceder con la reserva o volver al menú principal. En la reserva, se seleccionan la cantidad de entradas y los snacks elegidos. Esta información viaja a la pantalla de pago, donde se ingresan los datos de la tarjeta y, si son válidos, al aceptar se genera el QR con los datos de la compra. Para la segunda aplicación, se ejecuta el archivo *aplicacion leer qr.py*. El mismo tiene dos opciones: ingresar el ID de la compra o escanear un QR. En el primero, se ingresa el código de la compra que se informa al finalizar la misma. El segundo, permite escanear el código QR generado mediante la cámara web de nuestra computadora.

## 5. Solución propuesta

La estructura del código define de manera precisa el flujo del programa. Inicia con la ejecución de la llamada a *main()*, que alberga el diccionario principal y la invocación de la función de la primera pantalla. El diccionario sirve como el contenedor principal de información, facilitando el intercambio de datos entre las ventanas a medida que se obtiene información del usuario. La llamada a la función marca la apertura del programa y, por ende, de las ventanas subsiguientes. Optamos por emplear únicamente funciones para la creación de cada ventana. Dentro de estas funciones, se definían los widgets proporcionados por la librería *tkinter*. El elemento crucial que permitió la conexión entre las ventanas fue el uso de *.Button()*, al cual se le proporcionaron como parámetros las ventanas anteriores y otras variables esenciales.

## 6. Pruebas y/o validación

Llevamos a cabo pruebas unitarias que consistieron principalmente en un proceso de prueba y error, donde probábamos manualmente el código entre los miembros del equipo. Este proceso se facilitó en gran medida gracias al repositorio en *GitHub* que teníamos, permitiéndonos visualizar y actualizar el código de manera simultánea mientras lo testeábamos.

## 7. Plan de actividades

Después de la formación del grupo, nos contactamos rápidamente para organizar y asignar las tareas que cada uno llevaría a cabo. Optamos por dividir el trabajo práctico en segmentos, asegurándonos de que cada miembro se encargara de un problema específico. Esta medida principal tenía como objetivo evitar errores al integrar todo el trabajo en una única rama. Es importante destacar que, a pesar de esta

división de tareas, no estábamos aislados ni dejábamos de colaborar. Por el contrario, manteníamos una comunicación constante a través de Slack. Además de consultarnos sobre problemas, organizábamos reuniones en Meet para discutirlos en profundidad.

## **8. Hipótesis y Supuestos**

### **8.1. Aplicación Cine**

En la primer pantalla, la del menú principal, se decidió que se listen únicamente las películas por sucursal elegida. Nos pareció correcto que esta pudiera ser modificada por el usuario para visualizar el catálogo correspondiente a otros cines. El filtro que se realiza mediante la barra de búsqueda también tiene en cuenta la localidad seleccionada. Para no cargar la pantalla de información innecesaria, la información que se muestra de cada película son su nombre y el poster correspondiente, para acceder al detalle de cada una se debe seleccionar la deseada.

La pantalla secundaria de esta aplicación de cine despliega información detallada sobre la película seleccionada en la pantalla anterior, presentando datos relevantes como el título, sinopsis, duración, género y la cantidad exacta de asientos disponibles en la sala. Esta interfaz proporciona a los usuarios la posibilidad de realizar reserva de asientos, siempre y cuando haya al menos un asiento disponible. En caso de que la sala esté completamente ocupada, se muestra un aviso indicando la falta de disponibilidad. Además, se incluye un botón en la parte superior para permitir a los usuarios regresar a la pantalla inicial en caso de reconsiderar su elección o realizar una nueva selección.

En la tercera pantalla, o pantalla de reserva, se plantearon hipótesis clave con el objetivo de simplificar el código sin sacrificar ninguna funcionalidad requerida. Una de estas hipótesis, la más destacada, consistió en presentar en una lista en pantalla tanto los snacks como sus precios unitarios. Esta estrategia fue diseñada para facilitar el proceso de codificación, haciéndolo más accesible y visible para la interacción del usuario, sin comprometer o fallar en los requisitos establecidos. Otra acción implementada fue la visualización permanente del carrito, asegurando que el usuario siempre tenga conocimiento de los elementos que está comprando y brindándole la posibilidad de cancelar cualquier pedido en cualquier momento.

La pantalla de checkout es donde los usuarios revisan su selección de entradas y snacks antes de pagar. Se muestra en detalle lo que está en el carrito: tanto las entradas como los snacks, indicando cantidades y precios. Al final de la pantalla se indica el costo total y se presenta un botón que lleva a la ventana de pago.

Finalmente, la pantalla de pago. Aquí, se asumió como hipótesis que el usuario realizaría el pago mediante una tarjeta bancaria de su elección, validando en cada caso que los datos cumplan con ese formato. Una vez que decide hacer clic en "Pagar", el programa no solo genera el ID de la compra, un código QR en formato .png y lo guarda posteriormente en un PDF, sino que también registra el momento exacto en que se procesa el pago. Toda la información posterior y los detalles del pago se almacenan tanto en el código QR como en un archivo .json, donde las llaves son los ID de cada compra. Cabe destacar que con cada compra, se agregarán nuevos ID sin perder los anteriores.

## 8.2. Aplicación Lectura de QR

Como se mencionó anteriormente, al finalizar la compra se genera un ID y un código QR con los cuales podemos conocer la información del pedido: ID de compra, nombre de la película seleccionada, localidad del cine, cantidad de entradas, fecha y hora de la compra y el precio final (incluye entradas y snacks) Por un lado, se puede ingresar y validar el ID de compra. Este se verifica comparando con un archivo json en el cual se listan todos los pedidos realizados con sus datos. Por otro lado, se puede leer el código QR. Para este apartado, decidimos que el programa se encargaría de abrir la cámara web y tomar una foto, luego la misma sería analizada para detectar el QR. En ambos casos, se informa mediante un mensaje la validación exitosa o el error, según corresponda. Luego, se agrega la información del pedido al archivo *Ingresos.txt*

## 9. Resumen/Abstract

### 9.1. Resumen

Podemos entender el programa que desarrollamos como una coartada para lo que en realidad fue un camino de aprendizaje continuo. Implementamos los conocimientos adquiridos después del parcial, los cuales fueron clave para la complejización del trabajo y su realización. El uso de nuevas librerías, como lo fue Tkinter, nos permitió acercarnos a la creación de interfaces gráficas y, a su vez, al formato que proporciona los estilos, los cuales, en líneas generales, se asemejan a la creación de páginas web.

### 9.2. Abstract

We can understand the program we developed as a scaffold for what was, in reality, a continuous learning journey. We implemented the knowledge acquired after the midterm, which was crucial for the complexity of the work and its execution. The use of new libraries that we were not familiar with, such as Tkinter, allowed us to approach the creation of graphical interfaces and, at the same time, the formatting that provides styles, which, in broad terms, resemble the creation of web pages.

## 10. Referencias

- Graphical user interfaces with TK. (s. f.). Recuperado de <https://docs.python.org/3/library/tk.html>
- Interfaz gráfica con Tkinter — Documentación de Guia Tkinter - 0.1.1. (s. f.). Recuperado de <https://guia-tkinter.readthedocs.io/es/develop/>
- Requests. (2023, 22 mayo). Recuperado de <https://pypi.org/project/requests/>
- Requests: HTTP for humans™ — Requests 2.31.0 documentation. (s. f.). Recuperado de <https://requests.readthedocs.io/>
- Pybase64. (2023, 30 septiembre). Recuperado de <https://pypi.org/project/pybase64/>
- Base64 — Base16, Base32, Base64, Base85 Data encodings. (s. f.). Recuperado de <https://docs.python.org/3/library/base64.html>
- Pillow. (2023, 15 octubre). Recuperado de <https://pypi.org/project/Pillow/>

- IO — core tools for working with streams. (s. f.). Recuperado de <https://docs.python.org/3/library/io.html>
- QRCode. (2023, 5 febrero). Recuperado de <https://pypi.org/project/qrcode/>