

Trabajo Práctico N°2

Algoritmos I - Lanzillotta

Sistema automatizado de tickets para sala de cine

Una importante cadena de cines está renovando todo el sistema de ticket y venta de snacks para sus salas del país.

Nos contrató para la creación de una aplicación que deberá correr en los tótems interactivos que poseen.

La contratación consiste en crear dos aplicaciones, la primera estará disponible en los tótems, será una aplicación gráfica GUI utilizando la librería Tkinter, mientras que la segunda será una aplicación tipo consola/GUI (a elección) que permita leer un QR.

A continuación se explica el funcionamiento de cada aplicación.

Aplicación GUI para totems.

1. Se deberá crear una aplicación gráfica GUI mediante la librería **Tkinter**
2. Requerimientos
 - a. Pantalla principal:
 - i. Mostrar Ubicación del totem (Ejemplo Palermo, Puerto. Madero, Recoleta, etc)
 - ii. Mostrar cartelera. Se deberá mostrar en 2 columnas las portadas de las películas en cartelera. La información de las películas proviene de los endpoints ofrecidos por la empresa (*ver apartado de endpoints*)
 - iii. Input box que permita buscar una película ingresada por el usuario.
 - iv. Al realizar click en una de las portadas, se deberá pasar a la pantalla secundaria
 - b. Pantalla secundaria:
 - i. Mostrar sala a proyectar la película.
 - ii. Mostrar sinopsis, duración, actores y género de la película seleccionada en la pantalla principal.
 - iii. Debe existir en la parte superior un botón "Volver a pantalla principal"
 - iv. Debe existir un botón reservar, ***si hay disponibilidad de asientos.***
 - c. Pantalla reserva:
 - i. Al realizar click en el botón de reserva, se deberá pedir al usuario la cantidad de entradas a comprar, el valor unitario de cada entrada y un botón agregar al carrito.
 - ii. En la pantalla de reserva se debe colocar un botón de "Añadir Snack" que liste los snack existentes (*ver API reference*) y el usuario pueda agregar al carrito todo los que desee (debe estar la opción de agregar más de uno).

d. Checkout/Finalizar compra.

- i. Al ingresar al carrito o checkout se deberá listar todo lo comprado, detalle y precio unitario, finalizando con el precio total.
- ii. Una vez que el usuario haga click en el botón pagar se deberá generar un código QR que deberá ser guardado en un archivo .pdf, en la carpeta /QR (*ver apartado código QR*)

Aplicación lectura de códigos:

1. Deberá permitir el ingreso del Id del código QR por teclado
2. Deberá permitir tomar una foto del QR con la webcam
3. Una vez ingresado el código o leído el QR mediante la webcam, se deberá guardar en un archivo de texto "Ingresos.txt" la siguiente información separada por comas.
timestamp, Id_QR, nombre_película, cant_entradas, total_consumido

Endpoints disponibles:

Acceso a la API

Host: <http://vps-3701198-x.dattaweb.com:4000/>

- GET /movies
- GET /movies/{movie_id}
- GET /posters/{poster_id}
- GET /snacks
- GET /movies/{movie_id}/cinemas
- GET /cinemas
- GET /cinemas/{cinema_id}/movies

(Para saber cómo utilizar la API, leer documento "API Reference")

Formación del QR:

El QR se puede generar utilizando la librería **qrcode**.

Para generar el QR necesitarán un string. El string estará compuesto por los datos del pedido más el *timestamp* (fecha y hora) de la compra. Un ejemplo podría ser:

"ID_QR + película + ubicación_totem + cantidad_entradas + timestamp_compra"

Aclaraciones:

- Sólo puede existir una sala por película en una ubicación (Palermo, Puerto. Madero, Recoleta, etc)
- Los snacks listados en el endpoint poseen stock ilimitado.
- Las ubicaciones son sin numerar. Se tiene que tener en cuenta la capacidad de la sala. La capacidad la debe controlar la propia aplicación, el endpoint sólo indica la cantidad total.

- Se deberá tener todo el código en un repositorio GitHub, y todos los miembros del equipo deberán commitear código en dicho repositorio.

Librerías permitidas:

Cualquier librería que quieran usar fuera de estas, deberán consultarlas con los profesores.

- qrcode
- CV2 open-cv
- tkinter
- OS
- Requests

Documentación Requerida

Para la presentación del Trabajo Práctico, se pide al alumno realizar la documentación utilizando la aplicación online **overleaf**.

La documentación tendrá que tener un formato parecido a un paper. Deberá adecuarse a la información del “**Anexo Documentación**”.

La información volcada en el documento será utilizada por el equipo docente previo a corregir el trabajo por lo cual el alumno puede expresar allí lo que crea conveniente para ayudar en el proceso de corrección.

Defensa Oral

La entrega del Trabajo Práctico estará acompañada de una defensa oral tanto grupal como individual.

Se espera que tanto el alumno como el grupo en su totalidad pueda responder de forma completa las preguntas que llegara a plantear el docente en lo que respecta a la funcionalidad del trabajo, las herramientas provistas por el lenguaje de programación y el diseño del sistema en general.

Condiciones de Aprobación

NO se debe...

- usar variables globales
- usar ciclos infinitos
- escribir código repetido
- permitir al usuario ingresar valores/datos erróneos
- entregar un programa que no funcione
- entregar un programa con errores graves
- incumplir con la fecha de entrega

Sí se debe...

- validar el ingreso de datos, de forma tal que el programa no se rompa ante el ingreso de valores/datos erróneos
- modularizar el código
- usar buenas prácticas de programación
- utilizar un repositorio en github
- entregar un programa que cumpla con la funcionalidad completa especificada

Fecha de Entrega

El Trabajo Práctico cuenta con 2 fechas de entrega: 30/11/2023 y el 07/12/2023. Ambas tienen hora límite 18:00 hs. **Mientras que la primera fecha tiene posibilidad de recuperación, la segunda no.**

En esta línea, se espera que el grupo entregue el trabajo completo para la primera fecha, y -en caso de recibir correcciones por parte del equipo docente- tendrán posibilidad de realizar una segunda entrega en la segunda fecha. Esta primera instancia de entrega no restará puntos en caso de re-entregar en la segunda fecha. La segunda fecha definirá la nota del trabajo según corresponda.

Quienes decidan entregar únicamente en la segunda fecha (sin pasar por la primera) no tendrán posibilidad de correcciones intermedias y tendrán una nota final del trabajo según corresponda. Tanto la primera entrega como la segunda deben contener la totalidad de la funcionalidad solicitada, no se aceptarán entregas parciales o trabajos incompletos en ninguna de las fechas.

Importante

La aprobación del Trabajo Práctico N°2 es condición necesaria (junto con la aprobación del parcial) para regularizar la materia.