

PIZZERIA

1) Organización de los archivos:

Para tener los archivos lo más organizados posibles, los he dividido en varias carpetas. Por una parte, tengo la carpeta "Pizzeria_Builder" donde está todo el código Python relacionado con los patrones de diseño y estructurales, y otras clases necesarias para la web.

Por otra parte, en "static" guardo los archivos relacionados con el diseño de la página, como el css y las imágenes utilizadas para hacer la web más bonita.

Por último, en la carpeta "templates", tengo todas las páginas html, separadas cada una en su archivo correspondiente.

Fuera de estas carpetas, tengo el interfazuser.py, con las funciones flask y otras, relacionadas con la implementación de los patrones en la página, y el run.py, para ejecutar el código.

Todo esto, dentro de la carpeta del proyecto "pizzería".

2) Diseño de la web:

Para el diseño de la web, me he decantado por algo sencillo: una caja arriba con los tipos de productos que se pueden comprar y un inicio de sesión para el usuario. Al hacer click en cada uno de los enlaces, le llevará a una página de selección de productos según a dónde haya accedido.

Por ejemplo, en el enlace Pizza personalizada, el usuario tendrá que escoger uno a uno los productos que quiere para su pizza, así como el postre y la bebida. De igual manera para los menús personalizados. El único enlace que cambia un poco es el de los Combos Ana la Rana, en el que el usuario solo podrá decidir uno de los menús que ofrece la pizzería. Una vez haya decidido el, o los productos, tendrá que enviar el pedido. Una vez enviado, le saldrá un mensaje de confirmación y el precio total del pedido.

Si el usuario no tiene una sesión y quiere registrarse, puede darle al enlace "regístrate", y le llevará a una página en la que tiene que meter sus datos. Una vez enviados, se iniciará sesión automáticamente y devolverá a la página principal home.

3) Patrones de diseño:

Tanto para los menús como para las pizzas personalizadas, he utilizado el patrón de diseño Builder. Con este patrón, puedo crear cada elemento del pedido de manera eficiente y ordenada, y guardar los datos en un csv. Accedo a los datos del formulario, los guardo en variables y luego se los paso al builder, para que me construya paso a paso la pizza o el menú.

Para los menús personalizados, además, he utilizado el patrón estructural composite, que me da los precios de cada producto recogiendo de un csv (que sería como la carta), y si está en el pedido, lo añade como hijo. Luego, sumo todos los hijos para obtener el precio final del menú.