



ISIS-1221

# INTRODUCCIÓN A LA PROGRAMACIÓN

## Nivel 2 - Proyecto CupiEmpanadas

### Objetivo general

El objetivo general de este proyecto es que usted practique los conceptos estudiados en el nivel 2 del curso.

Recuerde que este proyecto debe realizarse de forma **completamente individual**.

### Objetivos específicos

1. Llamar funciones con parámetros y componer funciones.
2. Usar la técnica Dividir y Conquistar
3. Crear y usar un módulo.
4. Utilizar condicionales.
5. Utilizar diccionarios.
6. Construir interfaces de usuario basadas en consola.

### Contexto

Fundado en el lejano año de 2023, ✨Cupiempanadas✨ se ha vuelto un negocio triunfante que domina completamente el mercado de empanadas de todo el barrio Las Aguas. Tras haber superado su fase inicial, el negocio ya cuenta con un catálogo nutrido de empanadas, salsas, toppings y demás. Ahora, está lista para lanzar un agresivo modelo de negocio de empanadas *supergourmet* que finalmente le permita extender su influencia de mercado más allá de la calle 26. Es nuestra tarea implementar un sistema que permita a Cupiempanadas su éxito indiscutible.

Por el momento, el programa se implementará con 4 empanadas para probar las funcionalidades de la aplicación. Cada empanada será representada por un diccionario con las siguientes llaves:

- Nombre
- Precio

- Vegana
- Ingredientes
- Toppings
- Rating
- Unidades Vendidas
- Próxima Fecha de Entrega

En primera instancia se desea construir una aplicación capaz de:

- Buscar una empanada por su nombre
- Buscar la empanada que ha generado más ganancias
- Buscar las empanadas que contengan un ingrediente específico
- Busca la empanada con mejor rating que contenga un topping específico
- Calcula el promedio de rating de las empanadas.
- Comparar el rating promedio de las empanadas veganas y no veganas
- Modificar el rating y los toppings de una empanada
- Calcular el tiempo entre una fecha y la fecha próxima de entrega de una empanada

## Actividad 1 | Preparación del ambiente de trabajo

1. Cree una carpeta para trabajar, poniéndole su nombre o login.
2. Descargue de Bloque Neón el archivo con el “esqueleto” del proyecto (**n2-esqueleto.zip**) y descomprímalo en su carpeta de trabajo. El esqueleto consiste en un conjunto de archivos que usted va a usar o a modificar.
3. Descomprima el esqueleto en la carpeta que usted acaba de crear.
4. Abra Spyder y cambie la carpeta de trabajo para que sea la carpeta con el esqueleto.

## Actividad 2 | Construir un módulo de empanadas

5. Abra el archivo llamado “**empanadas.py**”. En este archivo usted debe completar las funciones necesarias para cumplir con los requerimientos base que le han sido encargados para la aplicación. Note que las funciones de este módulo están debidamente documentadas y tienen una etiqueta **TODO**, indicando que usted debe completarlas.

**ATENCIÓN:** este archivo debe permanecer dentro de la carpeta que usted acaba de descomprimir.

6. Complete las funciones marcadas con **TODO** en el archivo de acuerdo con la documentación.

Cada empanada se va a representar utilizando un diccionario que debe tener las siguientes llaves, que serán cadenas de caracteres (tenga cuidado a las combinaciones de mayúsculas y minúsculas y utilice las llaves exactamente como aparecen a continuación):

Llave	Tipo	Descripción de la llave
nombre	str	Nombre de la empanada. Por ejemplo: “Empanada Criolla de la Casa”
precio	int	Precio de la empanada en pesos colombianos como 2500, 3000
vegana	bool	Indica si la empanada es vegana
ingredientes	str	Ingredientes de la empanada separados por comas
toppings	str	Toppings de la empanada separados por comas
rating	float	Calificación de la empanada en un rango de [0, 5]
unidades_vendidas	int	Unidades vendidas de la empanada
proxima_fecha_entrega	int	Proxima fecha en que el proveedor traerá más empanadas de este tipo. Estará en formato YYYYMMDD (Por ejemplo: 20240325)

**NOTA IMPORTANTE:** Repetir código que podría reutilizar es considerado una muy mala práctica de programación. Si usted no utiliza una función existente en un lugar donde podría hacerlo, se considerará un error así su programa genere los resultados esperados.

### Actividad 3 | Construir interfaces de usuario basadas en consola

- En esta actividad usted debe construir la interfaz basada en consola para que el usuario interactúe con la aplicación. Para construir esta interfaz usted debe completar el archivo **consola\_empanadas.py**, el cual ya tiene una parte implementada que le facilitará su trabajo. La parte que ya está implementada incluye algunos elementos de Python que se estudiarán en el siguiente nivel así que usted sólo tiene que modificar los elementos marcados con la etiqueta **TODO**.

El programa debe ofrecer las siguientes funcionalidades:

- Buscar empanada por su nombre:** El programa debe solicitar el nombre de una empanada y buscarla entre todas las empanadas ofrecidas, y mostrar

el nombre, el precio y los ingredientes de la empanada en caso de encontrarla con un mensaje de la forma "*X cuesta Y y tiene Z*", donde X es el nombre de la empanada, Y es el precio de esta y Z son los ingredientes. En caso de no encontrar ninguna empanada, debe mostrar el mensaje "*No se encontró la empanada*".

- b. **Buscar la empanada que ha generado más ganancias:** El programa debe buscar y mostrar el nombre de la empanada que haya generado más ganancias de entre todas las empanadas con el siguiente mensaje: "*La empanada con más ganancias es X con ganancias de Y*", donde X es el nombre de la empanada y Y son las ganancias que generó la empanada. Las ganancias se entienden como el producto entre el precio de la empanada y el número de unidades vendidas de la misma. En caso de empate, retorne la empanada cuyo nombre sea menor alfabéticamente.
- c. **Buscar empanadas que contengan un ingrediente específico:** El programa debe solicitar el nombre de un ingrediente para luego buscar qué empanadas contienen ese ingrediente en específico. El programa debe mostrar un mensaje de la forma "*Las empanadas que contienen X son Y*", donde X es el ingrediente y Y son los nombres de las empanadas que contienen el ingrediente. Los nombres de las empanadas deben estar separados por comas. Si no se encuentra ninguna empanada debe mostrar: "*No se encontró ninguna empanada con X*", donde X es el ingrediente.
- d. **Buscar la empanada con mejor rating que contenga un topping específico:** El programa debe solicitar un topping y debe informar al usuario la empanada con mayor rating que tiene este topping recomendado. El programa debe mostrar un mensaje de la forma "*La empanada con mejor rating con X es Y con Z de rating*", donde X es el topping, Y es el nombre de la empanada y Z es el rating de la empanada. En caso de que no se encuentre ninguna empanada con ese topping recomendado se debe mostrar: "*No se encontró ninguna empanada con X*", donde X es el topping recomendado. En caso de empate, retorne la primera empanada encontrada.
- e. **Calcular el promedio de rating de empanadas veganas o no veganas:** el programa debe solicitar si se debe calcular el rating de las empanadas veganas o no veganas y posteriormente debe informar cual es el promedio de rating de las empanadas de ese tipo con el siguiente formato: "*El promedio de rating de las empanadas X es Y*", donde X es el tipo de empanada (vegana o no vegana) y Y es el promedio de rating de las empanadas de ese tipo redondeado a 2 decimales.
- f. **Comparar el promedio de rating de empanadas veganas y no veganas:** el programa debe comparar el rating promedio de las empanadas veganas vs el rating promedio de empanadas no veganas y mostrar alguno de los mensajes: "El promedio de rating de las empanadas veganas es mayor que el de las no veganas", "El promedio de rating de las empanadas no veganas es mayor que el de las veganas" o "El promedio de rating de las empanadas veganas y no veganas es el mismo" según sea el caso.

- g. **Modificar el rating y los toppings de una empanada:** el programa debe solicitar el nombre de una empanada, para posteriormente modificar el rating y los toppings de esta. Debe verificar que el nuevo rating sea un número flotante entre 0 y 5. Tenga en cuenta que al cambiar las características de una empanada las respuestas algunos de los otros requerimientos variarán y que, si el nombre de la empanada ingresado por el usuario no existe dentro del menú de Cupiempanadas, debe mostrar un mensaje que diga: "No se encontró la empanada".
- h. **Calcular el tiempo entre una fecha y la fecha próxima de entrega de una empanada:** El programa debe solicitar un nombre de empanada y una fecha en formato YYYYMMDD y debe mostrar un mensaje de la forma "Faltan X años, Y meses y Z días entre la fecha A y la próxima fecha de entrega de B", donde X, Y y Z son los años, meses y días entre la fecha ingresada por el usuario y la fecha de entrega de la empanada. A es la fecha ingresada por el usuario y B es la próxima fecha de entrega de la empanada. Si la fecha ingresada por el usuario es anterior a la de entrega de la empanada. En el caso contrario, debe mostrar un mensaje que diga: "La entrega de X ya pasó", donde X es el nombre de la empanada. Para este requerimiento asuma que los meses tienen 30 días y un año tiene 360 días

Además, si el nombre de la empanada ingresado por el usuario no existe dentro del menú de Cupiempanadas, debe mostrar un mensaje que diga: "No se encontró la empanada".

8. **Ejecute el programa y pruebe cada una de las funciones para asegurar que esté funcionando.**

## **Entrega**

9. Comprima la carpeta con su proyecto resuelto. El archivo debería llamarse **N2-PROY-login.zip**, donde login es su nombre de usuario de Uniandes.
10. Entregue el archivo comprimido a través de Bloque Neón en la tarea designada como **Proyecto del Nivel 2**.