

GUÍA #2 DISEÑO E IMPLEMENTACIÓN DE CLASES

JEISON AFRICANO MARTINEZ

JOSEPH IMANOL REYES CHAPARRO

UNIVERSIDAD MANUELA BELTRAN

INGENIERIA DE SOFTWARE

PROGRAMACION ORIENTADA A OBJETOS

INGENIERA: DIANA MARCELA TOQUICA RODRÍGUEZ

BOGOTA D.C, COLOMBIA

03 DE SEPTIEMBRE DE 2023

INTRODUCCION

A continuación, daremos una explicación detallada sobre el programa creado para explicar una plataforma basada en la organización de recetas de cocina, donde se implementará la creatividad y la precisión de los usuarios que pretendan retroalimentar su capacidad para tener una organización en los diferentes métodos instructivos para una eficaz organización culinaria.

De igual manera contextualizamos diferentes términos de habilidades de programación, donde ponemos en requerimientos diferentes clases y métodos de la lógica del programa, así mismo aprendemos un diseño de interfaz diferente acoplando iconos y elementos visuales para tener e identificar diversos tamaños de fuente, colores y espaciado creando una jerarquía visual. Elementos importantes, como los títulos de las recetas, deben destacar.

DESCRIPCIÓN DEL PROGRAMA

```
class Ingrediente:
```

```
    def _init_(self, nombre, cantidad, unidad):
```

```
        self.nombre = nombre
```

```
        self.cantidad = cantidad
```

```
        self.unidad = unidad
```

```
class PasoPreparacion:
```

```
    def _init_(self, descripcion):
```

```
        self.descripcion = descripcion
```

```
class Receta:
```

```
    def _init_(self, nombre, categoria):
```

```
        self.nombre = nombre
```

```
        self.categoria = categoria
```

```
        self.ingredientes = []
```

```
        self.pasos_preparacion = []
```

```
    def agregar_ingredientes(self, nombre, cantidad, unidad):
```

```
        ingrediente = Ingrediente(nombre, cantidad, unidad)
```

```
self.ingredientes.append(ingrediente)
```

```
def agregar_paso_preparacion(self, descripcion):
```

```
    paso = PasoPreparacion(descripcion)
```

```
    self.pasos_preparacion.append(paso)
```

```
def mostrar_receta(self):
```

```
    print(f"Receta: {self.nombre}")
```

```
    print(f"Categoría: {self.categoria}")
```

```
    print("\nIngredientes:")
```

```
    for ingrediente in self.ingredientes:
```

```
        print(f"- {ingrediente.cantidad} {ingrediente.unidad} de {ingrediente.nombre}")
```

```
    print("\nPasos de Preparación:")
```

```
    for i, paso in enumerate(self.pasos_preparacion, start=1):
```

```
        print(f"{i}. {paso.descripcion}")
```

```
mi_receta = Receta("Tarta de Manzana", "Postre")
```

```
mi_receta.agregar_ingrediente("Manzanas", 4, "unidades")
```

mi_receta.agregar_ingredient("Azúcar", 200, "gramos")

mi_receta.agregar_ingredient("Harina", 150, "gramos")

mi_receta.agregar_ingredient("Canela", 1, "cucharadita")

mi_receta.agregar_paso_preparacion("Pelar y cortar las manzanas en rodajas.")

mi_receta.agregar_paso_preparacion("Mezclar las manzanas con el azúcar y la canela.")

mi_receta.agregar_paso_preparacion("Forrar un molde con la masa de harina y verter la mezcla de manzanas.")

mi_receta.agregar_paso_preparacion("Hornear a 180°C durante 40 minutos.")

mi_receta.mostrar_receta()

EXPLICACIÓN DEL PROCESO

1. Proceso creación de recetas

Función: Este proceso permite a los usuarios crear nuevas recetas culinarias, agregando detalles como el nombre, la categoría, los ingredientes y los pasos de preparación.

Operaciones:

El usuario inicia el proceso de creación de receta, lo que podría ser a través de un botón "Crear Receta" en la interfaz.

El sistema muestra un formulario donde el usuario ingresa el nombre de la receta y selecciona una categoría (por ejemplo, postres, platos principales, etc.).

El usuario agrega ingredientes proporcionando el nombre, la cantidad y la unidad de cada uno. El sistema valida los datos ingresados para asegurarse de que sean correctos.

El usuario agrega pasos de preparación, proporcionando una descripción detallada para cada paso.

Una vez que el usuario completa todos los detalles, el sistema crea una instancia de la clase Receta y almacena la información ingresada, como nombre, categoría, ingredientes y pasos.

2. Proceso de Visualización de Recetas:

Función: Este proceso permite a los usuarios ver los detalles completos de una receta, incluidos los ingredientes y los pasos de preparación.

Operaciones:

El usuario navega a la lista de recetas o realiza una búsqueda para encontrar la receta deseada.

El sistema muestra un resumen de la receta, que generalmente incluirá el nombre y la categoría.

Si el usuario selecciona una receta, el sistema muestra los detalles completos de la receta, incluidos los ingredientes y los pasos de preparación.

Los ingredientes se muestran con su nombre, cantidad y unidad correspondiente.

Los pasos de preparación se muestran en orden, cada uno con su descripción detallada.

3. Proceso de Manipulación de Ingredientes y Pasos:

Función: Este proceso permite a los usuarios editar y gestionar los ingredientes y los pasos de preparación de una receta existente.

Operaciones:

- El usuario selecciona una receta para editar desde la lista de recetas o desde la vista detallada de la receta.
- El sistema muestra la información actual de la receta, incluidos los ingredientes y los pasos.
- El usuario tiene la opción de editar ingredientes y pasos existentes. Puede cambiar nombres, cantidades, unidades y descripciones.
- Además, el usuario puede agregar nuevos ingredientes y pasos si es necesario.
- Después de realizar los cambios, el sistema actualiza la instancia de la receta con la información editada y la almacena nuevamente.

Estos procesos permiten a los usuarios crear, visualizar y manipular recetas en la plataforma de manera coherente y efectiva, dando una experiencia organizada y funcional para los amantes de la cocina.

FLUJO DE DATOS

1. Se define una clase llamada `Ingrediente` para representar los ingredientes de una receta.

Tiene un constructor `__init__` que toma tres parámetros: `nombre`, `cantidad`, y `unidad`, y los asigna como atributos de la instancia.

2. Se define una clase llamada `PasoPreparacion` para representar los pasos de preparación de una receta. Tiene un constructor `__init__` que toma un parámetro `descripcion` y lo asigna como un atributo de la instancia.

3. Se define una clase llamada `Receta` para representar una receta en sí. Tiene un constructor `__init__` que toma dos parámetros: `nombre` y `categoria`, y crea una instancia

de receta con estos atributos, además de dos listas vacías: ``ingredientes`` y ``pasos_preparacion``.

4. La clase ``Receta`` tiene dos métodos para agregar ingredientes y pasos de preparación a la receta:

- ``agregar_ingredientes``: Toma tres parámetros (``nombre``, ``cantidad``, ``unidad``) y crea un objeto de la clase ``Ingrediente`` con estos datos. Luego, agrega ese objeto a la lista de ingredientes de la receta.

- ``agregar_paso_preparacion``: Toma un parámetro (``descripcion``) y crea un objeto de la clase ``PasoPreparacion`` con esta descripción. Luego, agrega ese objeto a la lista de pasos de preparación de la receta.

5. La clase ``Receta`` tiene un método llamado ``mostrar_receta`` que imprime la información de la receta, incluyendo su nombre, categoría, lista de ingredientes y lista de pasos de preparación.

6. Se crea una instancia de la clase ``Receta`` llamada ``mi_receta`` con el nombre "Tarta de Manzana" y la categoría "Postre".

7. Se agregan ingredientes a la receta ``mi_receta`` utilizando el método ``agregar_ingredientes`` con información sobre las manzanas, el azúcar, la harina y la canela.

8. Se agregan pasos de preparación a la receta ``mi_receta`` utilizando el método ``agregar_paso_preparacion`` con las descripciones de los pasos necesarios para preparar la tarta de manzana.

9. Finalmente, se llama al método ``mostrar_receta`` en la instancia ``mi_receta``, que muestra por pantalla la información de la receta, incluyendo su nombre, categoría, lista de ingredientes y lista de pasos de preparación.

En resumen, este código define una estructura de clases para representar una receta culinaria y luego crea una instancia de esa clase para una receta específica ("Tarta de Manzana").

Luego, se agregan ingredientes y pasos de preparación a la receta y se muestra la información de la receta por pantalla.

Flujo de datos del programa

Inicio

1. Se crea una instancia de la clase ``Receta`` llamada ``mi_receta`` con el nombre "Tarta de Manzana" y la categoría "Postre". En este punto, ``mi_receta`` es un objeto de la clase ``Receta`` con atributos iniciales.

2. Se utilizan los métodos ``agregar_ingredientes`` y ``agregar_paso_preparacion`` para agregar ingredientes y pasos de preparación a ``mi_receta``:

- Se llama a ``agregar_ingredientes("Manzanas", 4, "unidades")``, lo que crea un objeto de la clase ``Ingrediente`` con los datos proporcionados (nombre: "Manzanas", cantidad: 4, unidad: "unidades"). Este objeto se agrega a la lista de ingredientes de ``mi_receta``.

- Se llama a ``agregar_ingrediente("Azúcar", 200, "gramos")``, lo que crea otro objeto de la clase ``Ingrediente`` (nombre: "Azúcar", cantidad: 200, unidad: "gramos") y lo agrega a la lista de ingredientes.

- Se repiten estos pasos para los ingredientes "Harina" y "Canela".

- Luego, se llaman a ``agregar_paso_preparacion`` para agregar los pasos de preparación uno por uno. Cada llamada crea un objeto de la clase ``PasoPreparacion`` con la descripción del paso y lo agrega a la lista de pasos de preparación de ``mi_receta``.

3. Después de agregar todos los ingredientes y pasos de preparación, se llama al método ``mostrar_receta`` en ``mi_receta``.

4. ``mostrar_receta`` imprime la información de la receta, incluyendo el nombre, la categoría, la lista de ingredientes y la lista de pasos de preparación. Los datos se muestran en la consola.

Este es el flujo de datos a través del código, desde la creación de la instancia ``mi_receta`` hasta la visualización de la información de la receta en la consola.

CONCLUSIONES

Se logro desarrollar satisfactoriamente el programa de implementación de clases asociando el desarrollo aplicativo realizado en Python, de igual manera se logra establecer una estructura clara y precisa para determinar la capacidad de implementar un objeto lógico en diferente método de clase.

ANEXOS

```
class Ingrediente:
```

```
def __init__(self, nombre, cantidad, unidad):
```

```
    self.nombre = nombre
```

```
    self.cantidad = cantidad
```

```
    self.unidad = unidad
```

```
class PasoPreparacion:
```

```
def __init__(self, descripcion):
```

```
    self.descripcion = descripcion
```

```
class Receta:
```

```
def __init__(self, nombre, categoria):
```

```
self.nombre = nombre
```

```
self.categoria = categoria
```

```
self.ingredientes = []
```

```
self.pasos_preparacion = []
```

```
def agregar_ingredientes(self, nombre, cantidad, unidad):
```

```
    ingrediente = Ingrediente(nombre, cantidad, unidad)
```

```
    self.ingredientes.append(ingrediente)
```

```
def agregar_paso_preparacion(self, descripcion):
```

```
    paso = PasoPreparacion(descripcion)
```

```
    self.pasos_preparacion.append(paso)
```

```
def mostrar_receta(self):
```

```
    print(f"Receta: {self.nombre}")
```

```
    print(f"Categoría: {self.categoria}")
```

```
    print("\nIngredientes:")
```

```
    for ingrediente in self.ingredientes:
```

```
        print(f"- {ingrediente.cantidad} {ingrediente.unidad} de {ingrediente.nombre}")
```

```
    print("\nPasos de Preparación:")
```

```
for i, paso in enumerate(self.pasos_preparacion, start=1):
```

```
    print(f'{i}. {paso.descripcion}')
```

```
mi_receta = Receta("Tarta de Manzana", "Postre")
```

```
mi_receta.agregar_ingredient("Manzanas", 4, "unidades")
```

```
mi_receta.agregar_ingredient("Azúcar", 200, "gramos")
```

```
mi_receta.agregar_ingredient("Harina", 150, "gramos")
```

```
mi_receta.agregar_ingredient("Canela", 1, "cucharadita")
```

```
mi_receta.agregar_paso_preparacion("Pelar y cortar las manzanas en rodajas.")
```

```
mi_receta.agregar_paso_preparacion("Mezclar las manzanas con el azúcar y la canela.")
```

```
mi_receta.agregar_paso_preparacion("Forrar un molde con la masa de harina y verter la  
mezcla de manzanas.")
```

```
mi_receta.agregar_paso_preparacion("Hornear a 180°C durante 40 minutos.")
```

```
mi_receta.mostrar_receta()
```

