

PROYECTO FINAL ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN

SERGIO ALEJANDRO GONZÁLEZ SÁNCHEZ

FICHA: 2206205

SERVICIO NACIONAL DE APRENDIZAJE (SENA)

TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN

MOSQUERA

2022

APLICACIÓN WEB 'WEEKLYMENU'

SERGIO ALEJANDRO GONZÁLEZ SÁNCHEZ

Informe acerca del proyecto de creación de aplicación web para gestionar menús semanales (análisis de requerimientos, diseño, construcción y documentación)

**SERVICIO NACIONAL DE APRENDIZAJE (SENA)
TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN
MOSQUERA**

2021

CONTENIDO

| | |
|--|-----------|
| 1. Contextualización del proyecto | 4 |
| 2. Objetivos | 5 |
| 3. Requerimientos técnicos | 6 |
| 4. Definición de requerimientos | 8 |
| 5. Cronograma de actividades | 9 |
| 6. Diagramas de casos de uso | 12 |
| 7. Manual Técnico (Backend) | 20 |
| 8. Manual de Usuario (Frontend) | 26 |

1. Contextualización del proyecto

Este proyecto surge, por una parte, como la integración final de todos los conocimientos y habilidades adquiridas durante la etapa lectiva de mi formación como tecnólogo en análisis y desarrollo de sistemas de información en el SENA (Servicio Nacional de Aprendizaje) durante los últimos dieciocho meses. Buscando evidenciar tanto la calidad y pertinencia de los conocimientos adquiridos como la habilidad y destreza para ponerlos en práctica en un proyecto como el que presento a continuación.

Por otra parte, este proyecto también surge gracias a una necesidad personal y particular. En los próximos meses comenzaré a vivir solo y una de las facetas de la vida que más me inquieta es la comida, sobre todo porque actualmente cuando voy a cocinar algo tomo mucho tiempo decidiendo qué hacer, resultando en comidas hechas a la ligera que no resultan de mi total agrado, tal vez no sean del mejor contenido nutricional que me debería y siendo muy desordenado en general.

Por tales razones, me propuse desarrollar una aplicación web que me permitiese planificar con antelación las comidas que iba a tener en una semana. De manera que no gastase tiempo para cocinar o hacer otras cosas en esta decisión. Además, verificando que lo que vaya a comer sea de mi agrado y que sea de buen contenido nutricional, en la medida de lo posible, ya que no soy ningún experto, pero sí me interesa poder alimentarme de manera equilibrada. Por último, dicha planificación conllevaría un control más adecuado de los ingredientes que necesite comprar para realizar cada comida y así también aportar un mejor cálculo a los gastos económicos que supone esta materia.

En el presente texto se expondrán un análisis de los requerimientos de la aplicación, las necesidades técnicas para su desarrollo, las diferentes fases de diseño y los manuales de usuario y técnicos para el funcionamiento de la misma.

2. Objetivos del proyecto

- Objetivo general:

Desarrollar una solución informática que contribuya a una mejor gestión de recursos, alimentos, tiempo y dinero para mejorar la planeación de los alimentos a consumir a lo largo de una semana.

- Objetivos específicos

Analizar y definir los requerimientos funcionales del proyecto.

Diseñar la solución informática, particularmente la arquitectura tecnológica y el diseño de la base de datos (Modelo Entidad Relación).

Diseñar los prototipos visuales sobre los cuales se desarrollará el aspecto frontend de la solución.

Desarrollar la solución informática de acuerdo con los requerimientos y fase de diseño.

Realizar pruebas y solucionar fallos e inconvenientes.

3. Requerimientos técnicos

3.1. Requerimientos técnicos de Hardware

| | |
|---|---|
| Ficha técnica requisitos de Hardware | |
| Proyecto | WeeklyMenu |
| Objetivo | Desarrollar un sistema de información que permita apoyar la planeación de un menú semanal con la intención de mejorar la organización, ser más eficientes y asistir en el proceso de compra de ingredientes y/o materiales |
| Descripción | Aplicación web que ayuda la organización de un menú semanal detallando ingredientes, recetas y lista de compra |
| Requerimientos para el desarrollo en términos de Hardware | <ul style="list-style-type: none">- Procesador a 1.6 GHz o superior.- 1 GB (32 bits) o 2 GB (64 bits) de RAM - 4GB disponible de almacenamiento.- Tarjeta de vídeo compatible con DirectX 9 con resolución de pantalla de 1024 x 768 o más.- Periféricos teclado, mouse y monitor. |

3.2. Requerimientos técnicos de Software

| | |
|---|--|
| Ficha técnica requisitos de Software | |
| Proyecto | Weekly menu |
| Objetivo | Desarrollar un sistema de información que permita apoyar la planeación de un menú semanal con la intención de mejorar la organización, ser más eficientes y asistir en el proceso de compra de ingredientes y/o materiales |
| Descripción | Aplicación web que ayuda la organización de un menú semanal detallando ingredientes, recetas y lista de compra |
| Requerimientos para el desarrollo en términos de Software | <ul style="list-style-type: none">- Python 3.1+- Django- IDE Visual Studio Code- Xampp- Navegadores web con soporte HTML5, CSS3 y JS ES6- Windows 10 |

4. Definición de requerimientos funcionales del sistema

- Inicio de sesión y roles de usuario.
- Desarrollar un CRUD para gestionar las recetas de la aplicación (agregar, eliminar, modificar, leer).
- Obtener un menú de recetas para una semana distribuido por días y por momento del día (desayuno, almuerzo, cena).
- Obtener una lista de compras con los ingredientes necesarios para cocinar todas las comidas planeadas de la semana.

5. Cronograma de actividades

| CRONOGRAMA DE ACTIVIDADES | | | | | | | | |
|---|---------|---|---|---|---|---|---|---|
| Actividades | Semanas | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Objetivo 1: | | | | | | | | |
| Analizar los requerimientos de la solución de acuerdo con las necesidades del cliente | X | | | | | | | |
| Actividad 1: | | | | | | | | |
| Estudiar, observar y consultar cómo se manejan los procesos actualmente para definir la técnica de recolección de datos | X | | | | | | | |
| Actividad 2: | | | | | | | | |
| Definir los requerimientos del cliente basados en la técnica de recolección de información | X | | | | | | | |
| Objetivo 2: | | | | | | | | |
| Diseñar el prototipo del sistema de acuerdo a los requerimientos de la solución | X | | | | | | | |
| Actividad 1: | | | | | | | | |
| Diseñar el prototipo de acuerdo a los diagramas de caso de uso, secuencia y actividades | X | | | | | | | |
| Actividad 2: | | | | | | | | |
| Desarrollo del prototipo de sistema de información de acuerdo a los requerimientos del cliente | X | | | | | | | |
| Objetivo 3: | | | | | | | | |
| Construir el sistema de información de acuerdo a los requerimientos del cliente | | | X | | X | | | |
| Actividad 1: | | | | | | | | |
| Definir el lenguaje de programación para la construcción del sistema de acuerdo al hardware disponible | | | X | | X | | | |
| Actividad 2: | | | X | | X | | | |

| | | | | | |
|--|--|--|--|---|-----------------|
| Desarrollar y ejecutar el sistema de acuerdo con el proceso asignado por cada usuario | | | | | Sergio González |
| Objetivo 4: | | | | | |
| Implementar el sistema de información según requerimientos del cliente | | | | X | Sergio González |
| Actividad 1: | | | | | |
| Instalar el sistema de información de acuerdo al proceso automatizado en el servicio de hosting | | | | X | Sergio González |
| Actividad 2: | | | | | |
| Capacitar al usuario acerca del uso del sistema de información implementado. Creación de manuales de uso | | | | X | Sergio González |
| Objetivo 5 | | | | | |
| | | | | X | Sergio González |

Project Planning con fechas para cada fase del proyecto y costos proyectados para cada fase:

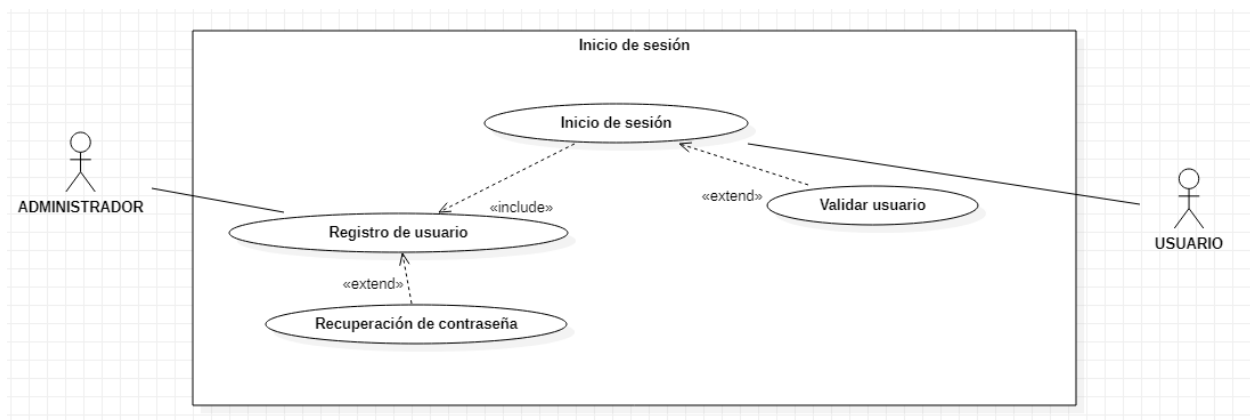
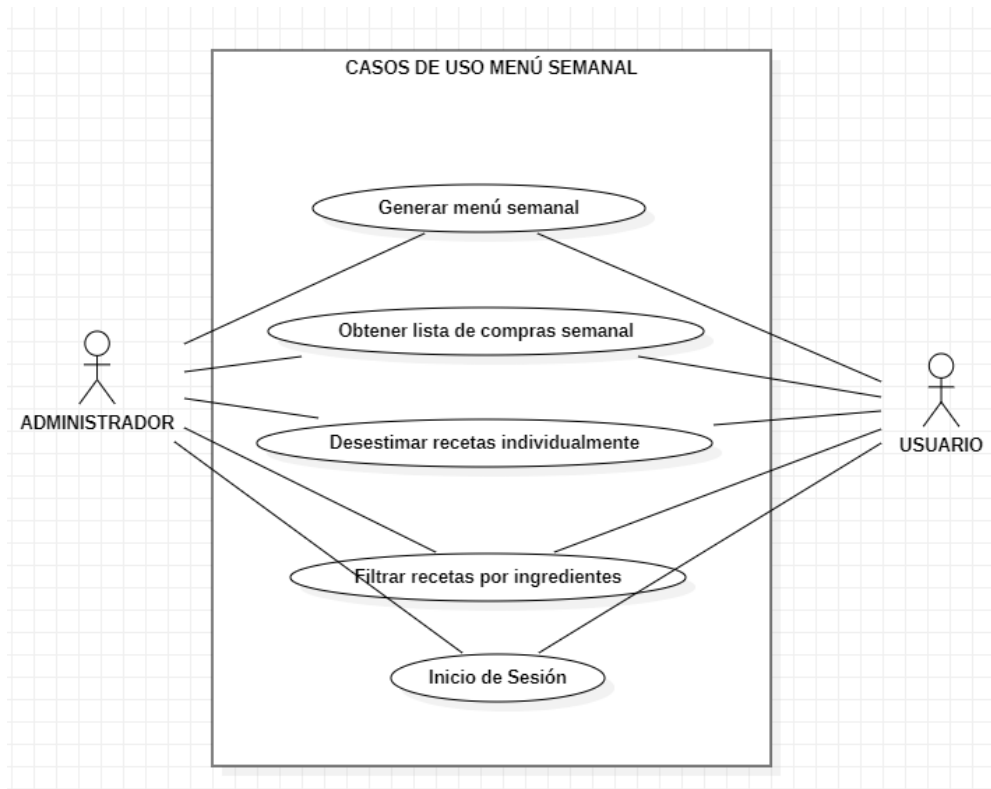
| | Task Name | Duration | Planned Start Date | Planned Finish... | Assigned | Percent Com... |
|----|---|----------------|--------------------|-------------------|---------------------|----------------|
| 1 | Analizar los requerimientos de la solución de acu... | 14 days | 1/02/2022 | 18/02/2022 | Sergio Go... | 64% |
| 2 | Estudiar, observar y consultar cómo se manejan los... | 4 days | 1/02/2022 | 4/02/2022 | | 100% |
| 3 | Definir los requerimientos del cliente basados en la... | 10 days | 7/02/2022 | 18/02/2022 | | 50% |
| 4 | Diseñar el prototipo del sistema de acuerdo a los... | 10 days | 21/02/2022 | 4/03/2022 | Sergio Go... | |
| 5 | Diseñar el prototipo de acuerdo a los diagramas de... | 5 days | 21/02/2022 | 25/02/2022 | | |
| 6 | Desarrollo del prototipo de sistema de información ... | 5 days | 28/02/2022 | 4/03/2022 | | |
| 7 | Construir el sistema de información de acuerdo a ... | 15 days | 7/03/2022 | 25/03/2022 | Sergio Go... | |
| 8 | Definir el lenguaje de programación para la constr... | 1 day | 7/03/2022 | 7/03/2022 | | |
| 9 | Desarrollar y ejecutar el sistema de acuerdo con e... | 14 days | 8/03/2022 | 25/03/2022 | | |
| 10 | Implementar el sistema de información según requ... | 15 days | 28/03/2022 | 15/04/2022 | Sergio Go... | |
| 11 | Instalar el sistema de información de acuerdo al pr... | 5 days | 28/03/2022 | 1/04/2022 | | |
| 12 | Capacitar al usuario acerca del uso del sistema d... | 10 days | 4/04/2022 | 15/04/2022 | | |
| 13 | Analizar resultados de la implementación y añadir ca... | 11 days | 18/04/2022 | 2/05/2022 | Sergio Gon... | |

| Pri... | A... | P... | Planned Cost | Planned Re... | Linked From | Linked To | Jan, 31 '22 | Feb, 07 '22 | Feb, 14 '22 | Feb, 21 '22 |
|--------|------|------------|--------------|---------------|-------------|-----------|---------------|---------------|---------------|---------------|
| None | 0... | \$ 933.324 | | | | | M T W T F S S | M T W T F S S | M T W T F S S | M T W T F S S |
| None | | \$ 266.664 | | | 3 | | | | | |
| None | | \$ 666.660 | | 2 | | | | | | |
| None | 0... | \$ 666.660 | | | | | | | | |
| None | | \$ 333.330 | | | 6 | | | | | |
| None | | \$ 333.330 | | 5 | | | | | | |
| None | 0... | \$ 999.990 | | | | | | | | |
| None | | \$ 66.666 | | | 9 | | | | | |
| None | | \$ 933.324 | | 8 | | | | | | |
| None | 0... | \$ 999.990 | | | | | | | | |
| None | | \$ 333.330 | | | 12 | | | | | |
| None | | \$ 666.660 | | 11 | 13 | | | | | |
| None | | \$ 733.326 | | 12 | | | | | | |

| Task Name | Feb, 28 '22 | Mar, 07 '22 | Mar, 14 '22 | Mar, 21 '22 | Mar, 28 '22 | Apr, 04 '22 | Apr, 11 '22 |
|---------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1 Analizar los... | S M T W T F S S | S M T W T F S S | S M T W T F S S | S M T W T F S S | S M T W T F S S | S M T W T F S S | S M T W T F S S |
| 2 Estudiar, o... | | | | | | | |
| 3 Definir los... | | | | | | | |
| 4 Diseñar el p... | | | | | | | |
| 5 Diseñar el ... | | | | | | | |
| 6 Desarrollo ... | | | | | | | |
| 7 Construir el... | | | | | | | |
| 8 Definir el ... | | | | | | | |
| 9 Desarrollar... | | | | | | | |
| 10 Implementar... | | | | | | | |
| 11 Instalar el... | | | | | | | |
| 12 Capacitar a... | | | | | | | |
| 13 Analizar resu... | | | | | | | |

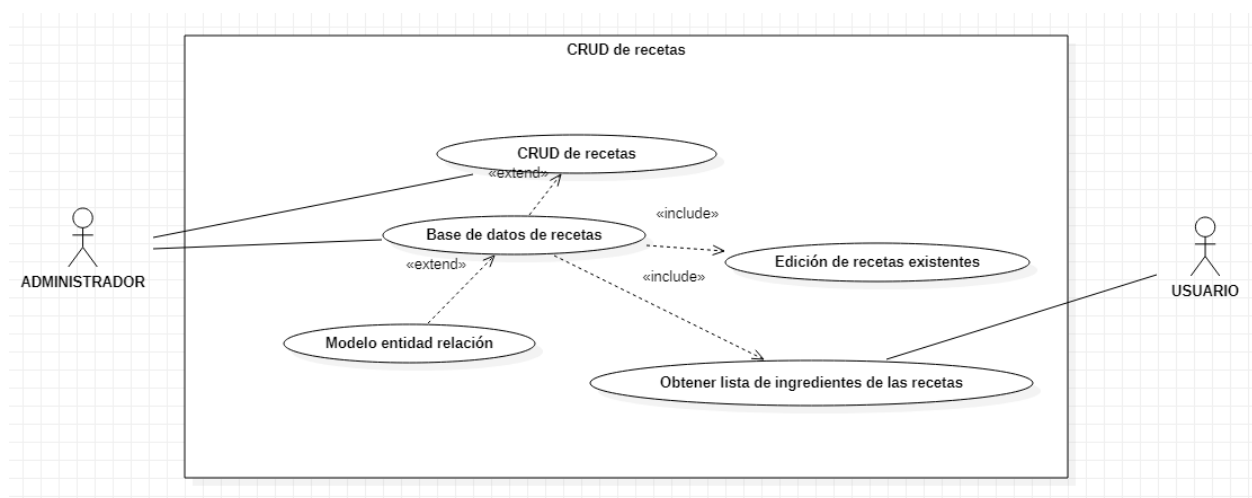
| Task Name | Apr, 18 '22 | Apr, 25 '22 | May, 02 '22 | May, 09 '22 | May, 16 '22 | May, 23 '22 | May, 30 '22 |
|---------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 1 Analizar los... | S S M T W T F S | S M T W T F S S | S M T W T F S S | S M T W T F S S | S M T W T F S S | S M T W T F S S | S M T W T F S S |
| 2 Estudiar, o... | | | | | | | |
| 3 Definir los... | | | | | | | |
| 4 Diseñar el p... | | | | | | | |
| 5 Diseñar el ... | | | | | | | |
| 6 Desarrollo ... | | | | | | | |
| 7 Construir el... | | | | | | | |
| 8 Definir el ... | | | | | | | |
| 9 Desarrollar... | | | | | | | |
| 10 Implementar... | | | | | | | |
| 11 Instalar el... | | | | | | | |
| 12 Capacitar a... | | | | | | | |
| 13 Analizar resu... | | | | | | | |

6. Diagramas de casos de uso



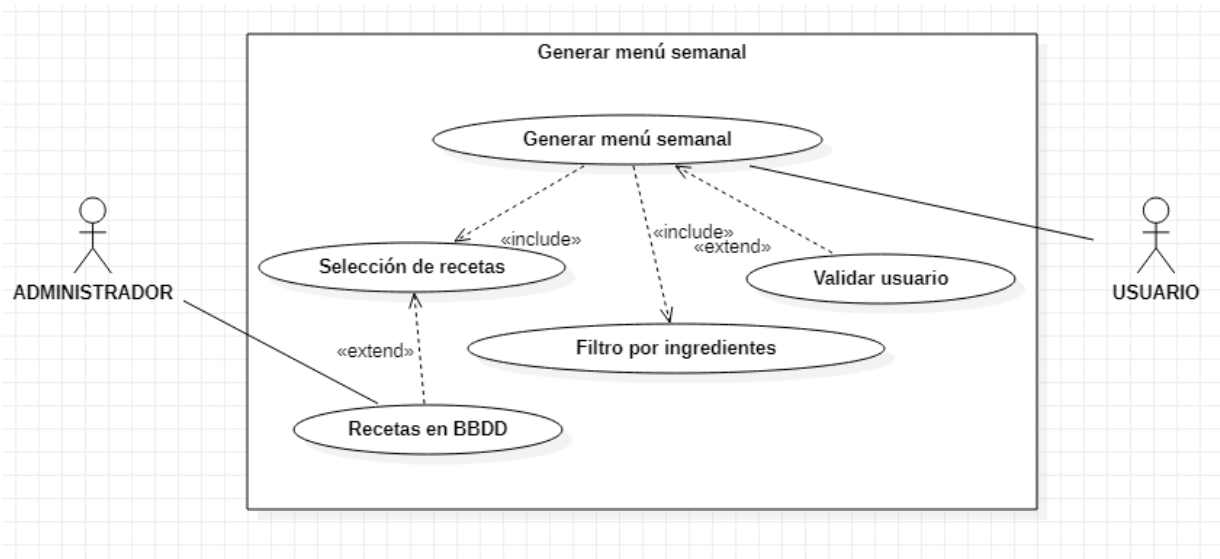
| Inicio de sesión | |
|----------------------------|---|
| Actores | Usuario |
| Descripción | Permite ingresar al usuario al sistema para hacer uso del aplicativo |
| Disparador | Entrar a la página web y digitar los datos necesarios para iniciar sesión |
| Precondiciones | <ol style="list-style-type: none"> 1. Tener conexión a internet e ingresar a la url de la web 2. Haber creado una cuenta previamente, de lo contrario crearla |
| Postcondiciones | <ol style="list-style-type: none"> 1. El usuario ya está dentro de la aplicación, puede empezar a crear su menú semanal o consultar el que haya creado previamente, si aplica |
| Flujo normal | <ol style="list-style-type: none"> 1. El usuario ingresa a la web 2. Digita los datos para ingresar a su cuenta <ol style="list-style-type: none"> a. Si no se ha registrado, debe crear una cuenta haciendo clic en 'Registrarse' |
| Flujos alternativos | |
| Excepciones | <p>E1. No se cargaron todos los datos requeridos.</p> <p>E1.1. El sistema le indica al usuario qué datos son requeridos</p> <p>E1.2. Vuelve al flujo principal</p> <p>E2. Ya existe una cuenta con ese correo electrónico</p> <p>E2.1. El sistema le indica al usuario que ya existe una cuenta usando ese correo electrónico</p> |

| | |
|--------------------------|---|
| | <p>E2.2. El sistema guía al usuario a la opción de recuperación de contraseña</p> <p>E2.3. El usuario puede digitar los datos para así recuperar sus credenciales</p> |
| Prioridad | Alta |
| Frecuencia de uso | Alta |
| Reglas de negocio | Se crea un nuevo usuario y sus registros en la base de datos |



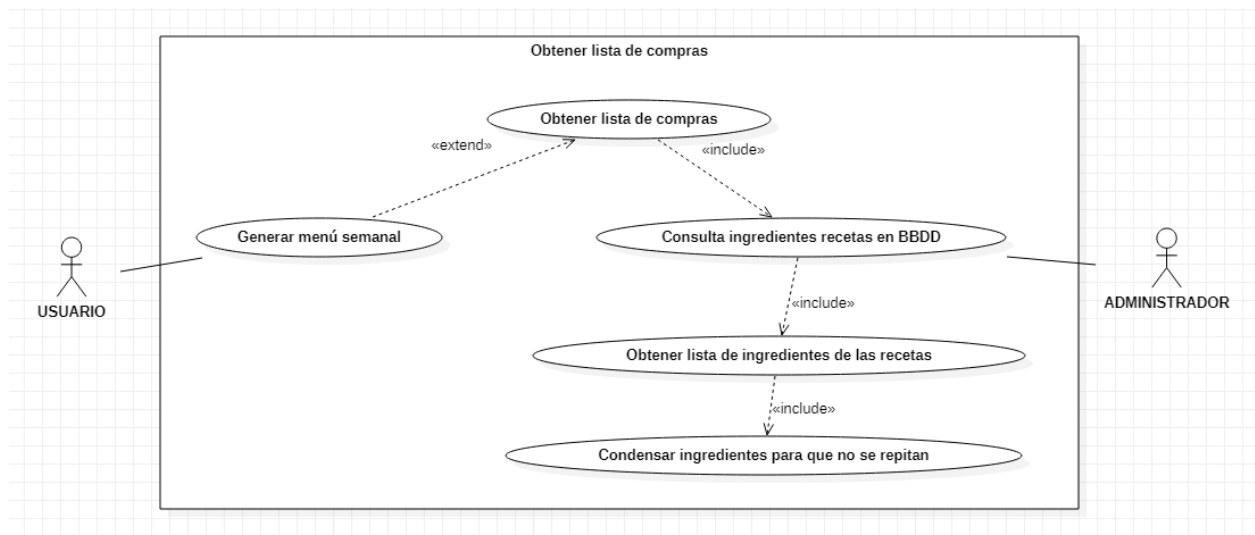
| Desarrollar CRUD para agregar recetas a la aplicación | |
|---|---|
| Actores | Administrador |
| Descripción | Permite al administrador ingresar, modificar o eliminar recetas dentro de la aplicación |
| Disparador | Entrar a la página web con permisos de administrador e ingresar a la opción para hacer uso del CRUD |
| Precondiciones | 1. Tener credenciales de administrador |

| | |
|----------------------------------|---|
| | 2. Ingresar los datos necesarios y actualizar la base de datos con los nuevos registros |
| Postcondiciones | 1. Las nuevas recetas estarán disponibles para agregarlas al menú semanal, así como las modificadas. 2. Las recetas eliminadas no se encontrarán disponibles |
| Flujo normal | 1. El administrador ingresa a la web a. Digita los datos para ingresar a su cuenta 2. La aplicación le ofrecerá la opción de ingresar al CRUD y realizar las opciones deseadas con las recetas en la BBDD |
| Flujos alternativos | |
| Excepciones | E1. No se cargaron todos los datos requeridos. E1.1. El sistema le indica al usuario qué datos son requeridos E1.2. Vuelve al flujo principal |
| Prioridad | Alta |
| Frecuencia de uso | Alta en un principio, baja en la medida en que se deseen agregar menos recetas a la aplicación |
| Reglas de negocio | Se ingresan, modifican o eliminan recetas de la aplicación. Las recetas quedarán disponibles para agregar a los menús, mientras que las eliminadas no |
| Requerimientos especiales | Tener permisos de administrador |



| Obtener menú de recetas semanal | |
|---------------------------------|--|
| Actores | Usuario |
| Descripción | Permite al usuario crear un menú semanal con las recetas que se quieran realizar en una semana determinada |
| Disparador | Entrar a la página web con permisos con la intención de planear su menú semanal de recetas |
| Precondiciones | <ol style="list-style-type: none"> 1. Tener conexión a internet e ingresar a la url de la web 2. Iniciar sesión en su cuenta <ol style="list-style-type: none"> a. Haber creado una cuenta previamente, de lo contrario crearla |
| Postcondiciones | <ol style="list-style-type: none"> 1. El usuario tendrá su menú semanal 2. Podrá también modificar el menú semanal de acuerdo con sus preferencias 3. El menú semanal quedará listo para extraer los ingredientes y su correspondiente lista de compras |
| Flujo normal | <ol style="list-style-type: none"> 1. Ingresar a la url de la web |

| | |
|----------------------------|--|
| | <p>2. Haber creado una cuenta previamente, de lo contrario crearla</p> <p>3. Elegir las opciones que la aplicación va indicando para diseñar el menú semanal que el usuario desee</p> |
| Flujos alternativos | |
| Excepciones | <p>E1. El usuario ingresa credenciales incorrectas</p> <p>E1.1. El sistema genera el error correspondiente y le indica al usuario que intente nuevamente</p> <p>E1.2. De ser necesario, el sistema ofrecerá la opción para restaurar la contraseña</p> |
| Prioridad | Alta |
| Frecuencia de uso | Alta |
| Reglas de negocio | El menú semanal del usuario quedará creado, podrá consultar recetas e ingredientes |



| Generar lista de compras | |
|----------------------------|---|
| Actores | Usuario, desarrollador |
| Descripción | Permite al usuario generar una lista de compras con todos los ingredientes necesarios para cocinar las recetas seleccionadas para la semana. |
| Disparador | Luego de haber generado el menú semanal un botón llevará al usuario a la lista de compra. |
| Precondiciones | <ol style="list-style-type: none"> 1. El usuario debe haber generado un menú semanal desde su respectiva cuenta. |
| Postcondiciones | El usuario podrá consultar su lista de compras al ingresar a la aplicación, puede enviarla a su correo electrónico o generar una imagen descargable con la lista de compras. |
| Flujo normal | <ol style="list-style-type: none"> 1. Ingresar a la url de la web. 2. Haber creado una cuenta previamente, de lo contrario crearla. 3. Elegir las opciones que la aplicación va indicando para diseñar el menú semanal que el usuario desee. 4. Una vez planeado el menú semanal se puede redirigir el usuario clicando el botón para proseguir a la lista de compra de ingredientes. |
| Flujos alternativos | |
| Excepciones | E1. El usuario no ha seleccionado ninguna receta para la semana. |

| | |
|--------------------------|---|
| | E1.1. El botón para redirigirse a la lista de compras estará deshabilitado hasta que se seleccione al menos una receta. |
| Prioridad | Alta |
| Frecuencia de uso | Alta |
| Reglas de negocio | La lista de compra de los ingredientes para el menú semanal quedará lista y el usuario podrá utilizarla para mercar. |

7. Manual técnico (Backend)

Importante: antes de iniciar la aplicación web es necesario importar la base de datos alojada en la carpeta raíz del proyecto '127_0_0_1.sql' en el gestor de bases de datos MySQL. Así mismo, asegurarse tener instalado en el computador Python 3.8+ y ejecutar desde la terminal el comando desde la carpeta raíz del proyecto el comando \$pip install -r requirements.txt

- forms.py

El siguiente archivo contiene las clases RecipeForm y MenuForm. En ambos casos, estas clases ayudan a contrastar la información obtenida a través de los formularios y los datos asignados a los modelos construidos para las recetas y para el menú.

Así mismo, contienen una subclase Meta que identifica cómo la clase opera, al final obtiene un objeto basado en el modelo creado para las recetas y los menús junto a los datos obtenidos de los formularios.

```
1 from django import forms
2 from .models import Recipe, Menu
3
4 class RecipeForm(forms.ModelForm):
5
6     """RecipeForm class helps validating the information introduced onto the forms to validate it matches the model created for Recipes."""
7
8     class Meta:
9
10         """Metaclass for Recipeform class shows how it behaves: it takes Recipe as the model and validates form fields against the model created."""
11
12         model = Recipe
13         fields = '__all__'
14
15 class MenuForm(forms.ModelForm):
16
17     """Menuform class helps validating the information introduced onto the forms to validate it matches the model created for Menus."""
18
19     class Meta:
20
21         """Metaclass for Menuform class shows how it behaves: it takes Recipe as the model and validates form fields against the model created."""
22
23         model = Menu
24         fields = '__all__'
```

- models.py

El código en models.py es usado por el framework Django para crear el modelo por el cual se construye las tablas en la base de datos y la estructura de los datos allí almacenados.

Se encuentra la clase Recipe que estructura el modelo que van a tener las recetas y sus correspondientes campos en la base de datos. Contiene un id como llave primaria, un título, el tipo de receta, los ingredientes, las instrucciones y una imagen.

Por otra parte, contiene dos métodos `__str__(self)` que devuelve un string del objeto que constituye la clase, en este caso el título.

El método `delete(self)` se usa para borrar la imagen del almacenamiento de la aplicación y los datos de la correspondiente base de datos, toda vez que se pretenda borrar una receta de la aplicación.

```
hera > models.py > Recipe > __str__
1  from django.db import models
2
3  # Create your models here.
4
5  class Recipe(models.Model):
6
7      """Class Recipe constructs the model on which the correspondent database table is created."""
8
9      id = models.AutoField(primary_key=True)
10     title = models.CharField(max_length=100, null=True, verbose_name="Título")
11     type = models.CharField(max_length=20, null=True)
12     ingredients = models.TextField(max_length=10000, null=True)
13     instructions = models.TextField(max_length=10000, null=True)
14     photo = models.ImageField(upload_to='images/', null=True)
15
16     def __str__(self) -> str:
17
18         """
19         It defines the data type of the object associated to this class. It will return the specified parameter (title) as a string.
20         -type: str -> this method returns a string.
21         """
22
23         title = self.title
24         return title
25
26     def delete(self):
27
28         """
29         This method is created to delete both the picture and the data from the DB when deleting a row from the CRM app.
30         It takes self, the object, as a parameter to capture image's path and data row to delete it.
31         """
32
33         self.photo.storage.delete(self.photo.name)
34         super().delete()
```

- views.py

El archivo views.py contiene las funciones por las cuales se logra visualizar el contenido de la aplicación. Se importan los métodos render y redirect para renderizar la información obtenida por los formularios u obtenida a través de consultas sql y ordenarla de acuerdo al modelo en una plantilla html particular.

De la misma manera, se importan los modelos Recipe y Menu, y las clases RecipeForm y MenuForm para así poder renderizar la información de acuerdo con los modelos creados y la información obtenida de los correspondientes formularios.

Se importa la librería os que se utiliza para interactuar con el sistema operativo, en este caso para eliminar una imagen de la aplicación, en el momento de actualizar una receta. Así mismo se importan los decorators @login_required y @permission_required para ejecutarlos previo a ciertas funciones y así prevenir que se pueda ingresar a la aplicación sin iniciar sesión y evitar que ciertas funciones se ejecuten a menos que se haya iniciado sesión como administrador.

```
8
9
10 """
11 views.py includes all the necessary functions for the app to work and show the correspondant information to the user.
12
13 imports:
14     render and redirect functions to help visualize the content from the app, rendering information included from the forms.
15     Recipe and Menu models to be able to render the data sent through the form according to the model created and the fields on the DB.
16     RecipeForm and MenuForm to validate the data sent through the app to contrast it against the model.
17     os package to interact with the operative system, in this case, in order to delete an image.
18     login_required decorator to ensure that the user is logged in before accessing the app.
19     permission_required decorator to ensure that the user has the permission to access the app.
20 """
```

Las siguientes líneas de código muestran las funciones que se utilizan para renderizar la información y mostrarla en sus correspondientes plantillas html. Se usa request como parámetro cuando el usuario solicita visualizar dicha plantilla y en determinados casos se realizan consultas sql y se envían los datos en un objeto para ser renderizado por el método render() y poder ser visualizado por el usuario.

```

20
21 @Login_required(login_url='/login/')
22 def index(request):
23     """Render the index page per user request, which is used as parameter for this function."""
24     return render(request, 'pages/index.html')
25
26 @Login_required(login_url='/login/')
27 def aboutus(request):
28     """Render the about us page per user request, which is used as parameter for this function."""
29     return render(request, 'pages/aboutus.html')
30
31 @Login_required(login_url='/login/')
32 @permission_required('menu.view_recipe', login_url='/login/')
33 def recipes_list(request):
34     """Render the list of recipes availables at this time per user request, which is used as parameter for this function."""
35     #The following line queries the DB for all recipes, parse them onto objects based on the Recipe model and send it as a parameter for the render function
36     recipes = Recipe.objects.all()
37     return render(request, 'recipes/recipes_list.html', {'recipes': recipes})
38
39 @Login_required(login_url='/login/')
40 @permission_required('menu.add_recipe', login_url='/login/')
41 def add_recipe(request):
42     """Render the page to add a recipe to the DB per user request, which is used as parameter for this function."""
43     #This function takes the data sent through the form, validate it based on the model, if it's correct it'll save the recipe, otherwise it'll render the page again.
44     form = RecipeForm(request.POST or None, request.FILES or None)
45     if form.is_valid():
46         form.save()
47         return redirect('recipes')
48     return render(request, 'recipes/add.html', {'form': form})

```

A continuación, se observan las funciones para editar y borrar recetas de la aplicación. En el caso de la función de borrado, se requieren el parámetro de request y el id para determinar cuál receta es la que se va a eliminar.

En el caso de la función de edición, se reciben los parámetros de request y de id para determinar cuál es la receta por modificar y precargar la información directamente en el formulario. Así mismo, cuando el formulario recibe una foto se precisa eliminar la anterior foto del almacenamiento de la aplicación previo a actualizar la información de dicha receta.

```

50 @Login_required(login_url='/login/')
51 @permission_required('menu.change_recipe', login_url='/login/')
52 def modify(request, id):
53     """Render the recipe edition page per admin request. It takes request from the user as parameters and the id of the recipe the user'd like to edit. """
54     #Get everything from the recipe
55     recipe = Recipe.objects.get(id=id)
56     #Get data from the form
57     form = RecipeForm(request.POST or None, request.FILES or None, instance=recipe)
58     if len(request.FILES) != 0:
59         #When the image is being updated, the previous image needs to be deleted first
60         if len(recipe.photo) > 0:
61             os.remove(recipe.photo.path)
62             recipe.photo = request.FILES['photo']
63             recipe.title = request.POST.get('title')
64             recipe.ingredients = request.POST.get('ingredients')
65             recipe.instructions = request.POST.get('instructions')
66             recipe.save()
67             messages.success(request, "¡Receta actualizada correctamente!")
68             return redirect('recipes')
69     return render(request, 'recipes/modify.html', {'form': form})
70
71 @Login_required(login_url='/login/')
72 @permission_required('menu.delete_recipe', login_url='/login/')
73 def delete(request, id):
74     """This function deletes a recipe from the app. It takes request parameter from the user and the recipe's id so it can be deleted from the DB. """
75     #Capture id of the recipe we want to delete
76     recipe = Recipe.objects.get(id=id)
77     recipe.delete()
78     return redirect('recipes')
79

```

La siguiente imagen muestra el código de las funciones para renderizar las páginas para generar menú semanal y el menú una vez generado. Se toman los datos del formulario, en el primer caso, y en el segundo, se realizan consultas sql del menú creado para poder mostrarlo al usuario.

```

88 @Login_required(login_url='/login/')
89 def menu(request):
90     """Render the menu creation page per user request, which is used as parameter for this function."""
91     #Get all available recipes and throw them to the user so they can select the ones they want
92     sql = "SELECT * FROM menu_recipe"
93     recipes = Recipe.objects.raw(sql)
94     if request.method == "POST":
95         form = MenuForm(request.POST)
96         if form.is_valid():
97             frm = form.save(commit=False)
98             frm.user = request.user
99             frm.save()
100             return redirect('weekly_menu')
101     return render(request, 'recipes/menu.html', {'recipes': recipes})
102
103 @Login_required(login_url='/login/')
104 def weekly_menu(request):
105     """Render the page where the user can visualize the menu they created. Request parameter for when user request it. """
106
107     #Query data from the recipes the user selected
108     mon_sql = "SELECT menu_menu.id, title, photo, type, ingredients, instructions FROM menu_recipe join menu_menu WHERE monday_breakfast_id = menu_recipe.id or monday_lunch_id = menu_recipe.id or monday_dinner_id = menu_recipe.id"
109     mon_menu = Menu.objects.raw(mon_sql)
110     tue_sql = "SELECT menu_menu.id, title, photo, type, ingredients, instructions FROM menu_recipe join menu_menu WHERE tuesday_breakfast_id = menu_recipe.id or tuesday_lunch_id = menu_recipe.id or tuesday_dinner_id = menu_recipe.id"
111     tue_menu = Menu.objects.raw(tue_sql)
112     wed_sql = "SELECT menu_menu.id, title, photo, type, ingredients, instructions FROM menu_recipe join menu_menu WHERE wednesday_breakfast_id = menu_recipe.id or wednesday_lunch_id = menu_recipe.id or wednesday_dinner_id = menu_recipe.id"
113     wed_menu = Menu.objects.raw(wed_sql)
114     thu_sql = "SELECT menu_menu.id, title, photo, type, ingredients, instructions FROM menu_recipe join menu_menu WHERE thursday_breakfast_id = menu_recipe.id or thursday_lunch_id = menu_recipe.id or thursday_dinner_id = menu_recipe.id"
115     thu_menu = Menu.objects.raw(thu_sql)
116     fri_sql = "SELECT menu_menu.id, title, photo, type, ingredients, instructions FROM menu_recipe join menu_menu WHERE friday_breakfast_id = menu_recipe.id or friday_lunch_id = menu_recipe.id or friday_dinner_id = menu_recipe.id"
117     fri_menu = Menu.objects.raw(fri_sql)
118     sat_sql = "SELECT menu_menu.id, title, photo, type, ingredients, instructions FROM menu_recipe join menu_menu WHERE saturday_breakfast_id = menu_recipe.id or saturday_lunch_id = menu_recipe.id or saturday_dinner_id = menu_recipe.id"
119     sat_menu = Menu.objects.raw(sat_sql)
120     sun_sql = "SELECT menu_menu.id, title, photo, type, ingredients, instructions FROM menu_recipe join menu_menu WHERE sunday_breakfast_id = menu_recipe.id or sunday_lunch_id = menu_recipe.id or sunday_dinner_id = menu_recipe.id"
121     sun_menu = Menu.objects.raw(sun_sql)
122
123     #Recipes' data is stored on a dictionary which is sent as a context for the app to renderize the data and present it to the user
124     context = {'mon_menu': mon_menu, 'tue_menu': tue_menu, 'wed_menu': wed_menu, 'thu_menu': thu_menu, 'fri_menu': fri_menu, 'sat_menu': sat_menu, 'sun_menu': sun_menu}
125     return render(request, 'recipes/weekly_menu.html', context)

```

La última función muestra la plantilla con la correspondiente lista de compras y las consultas sql para obtener esa información y mostrarla al usuario.

```

119 @Login_required(login_url='/login/')
120 def shopping_list(request):
121     """Render the shopping list page per user request, which is used as parameter to this function."""
122
123     #Query the ingredients from the recipes the user selected
124     mon_sql = "SELECT menu_menu.id, ingredients from menu_recipe join menu_menu where monday_breakfast_id = menu_recipe.id or monday_lunch_id = menu_recipe.id or monday_dinner_id = menu_recipe.id"
125     mon_ingredients = Menu.objects.raw(mon_sql)
126     tue_sql = "SELECT menu_menu.id, ingredients from menu_recipe join menu_menu where tuesday_breakfast_id = menu_recipe.id or tuesday_lunch_id = menu_recipe.id or tuesday_dinner_id = menu_recipe id"
127     tue_ingredients = Menu.objects.raw(tue_sql)
128     wed_sql = "SELECT menu_menu.id, ingredients from menu_recipe join menu_menu where wednesday_breakfast_id = menu_recipe.id or wednesday_lunch_id = menu_recipe.id or wednesday_dinner_id = menu_recipe id"
129     wed_ingredients = Menu.objects.raw(wed_sql)
130     thu_sql = "SELECT menu_menu.id, ingredients from menu_recipe join menu_menu where thursday_breakfast_id = menu_recipe.id or thursday_lunch_id = menu_recipe.id or thursday_dinner_id = menu_recipe id"
131     thu_ingredients = Menu.objects.raw(thu_sql)
132     fri_sql = "SELECT menu_menu.id, ingredients from menu_recipe join menu_menu where friday_breakfast_id = menu_recipe.id or friday_lunch_id = menu_recipe.id or friday_dinner_id = menu_recipe id"
133     fri_ingredients = Menu.objects.raw(fri_sql)
134     sat_sql = "SELECT menu_menu.id, ingredients from menu_recipe join menu_menu where saturday_breakfast_id = menu_recipe.id or saturday_lunch_id = menu_recipe.id or saturday_dinner_id = menu_recipe id"
135     sat_ingredients = Menu.objects.raw(sat_sql)
136     sun_sql = "SELECT menu_menu.id, ingredients from menu_recipe join menu_menu where sunday_breakfast_id = menu_recipe.id or sunday_lunch_id = menu_recipe.id or sunday_dinner_id = menu_recipe id"
137     sun_ingredients = Menu.objects.raw(sun_sql)
138
139     #Store ingredients on a dictionary which is sent as a context for the app to renderize the data and present it to the user
140     context = {'mon_ingredients': mon_ingredients, 'tue_ingredients': tue_ingredients, 'wed_ingredients': wed_ingredients, 'thu_ingredients': thu_ingredients, 'fri_ingredients': fri_ingredients, 'sat_ingredients': sat_ingredients, 'sun_ingredients': sun_ingredients}
141     return render(request, 'recipes/shopping_list.html', context)

```

- urls.py

El archivo urls.py contiene una lista con las rutas establecidas para cada una de las páginas a las que se tiene acceso en la aplicación. Cada campo en la lista se constituye

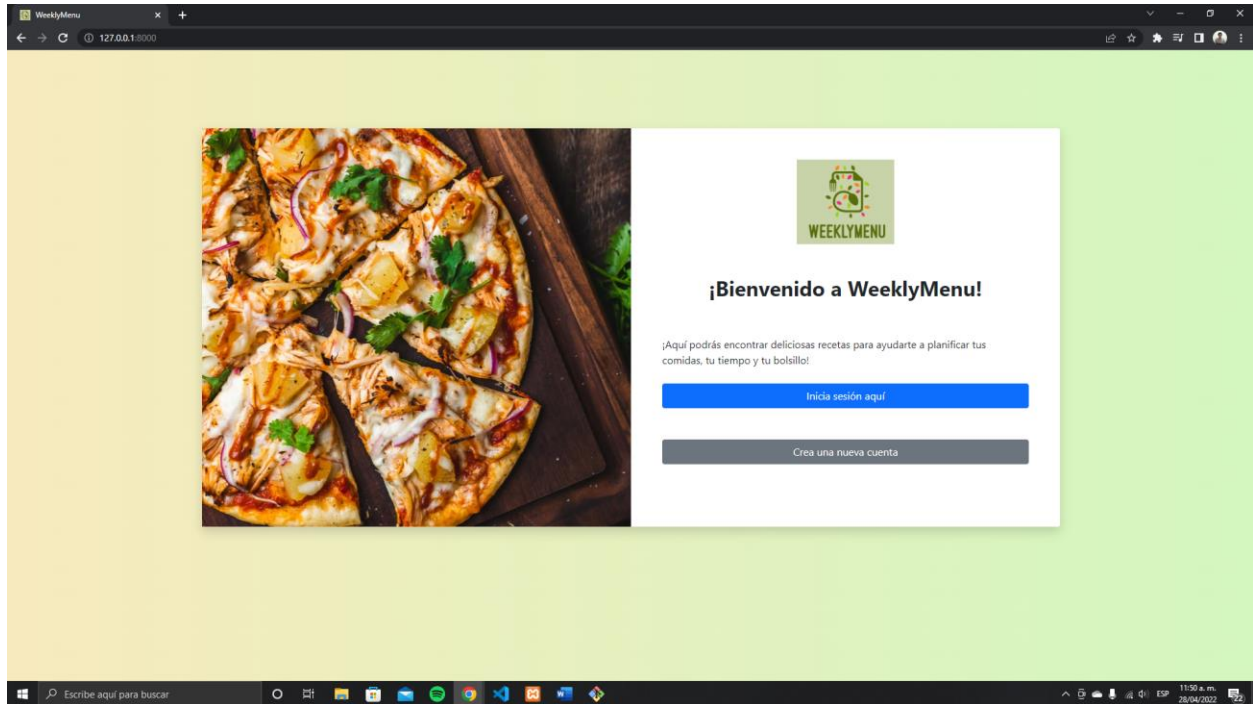
de un método que toma como parámetros la dirección url, la función importada del archivo views.py y el nombre que recibe dicha función para ser llamada desde las plantillas html o al renderizar desde una función específica del archivo views.py.

```
1 from django.urls import path
2 from . import views
3 from django.conf import settings
4 from django.contrib.staticfiles.urls import static
5
6 """
7 the following list captures the necessary data in order to route across different pages of the app.
8 'path' takes as parameters the url, the function imported from views.py and the name used to call the specific path on the html templates and rendered view.
9 """
10
11 urlpatterns = [
12     path("", views.index, name="index"),
13     path("about", views.about, name="about"),
14     path("recipes", views.recipes_list, name="recipes"),
15     path("recipes/add", views.add_recipe, name="add"),
16     path("recipes/modify/ctrl:id", views.modify, name="modify"),
17     path("recipes/delete/ctrl:id", views.delete, name="delete"),
18     path("menu", views.menu, name="menu"),
19     path("weekly_menu", views.weekly_menu, name="weekly_menu"),
20     path("shopping_list", views.shopping_list, name="shopping_list"),
21 ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

8. Manual de Usuario (Frontend)

1. Experiencia de Usuario de administrador

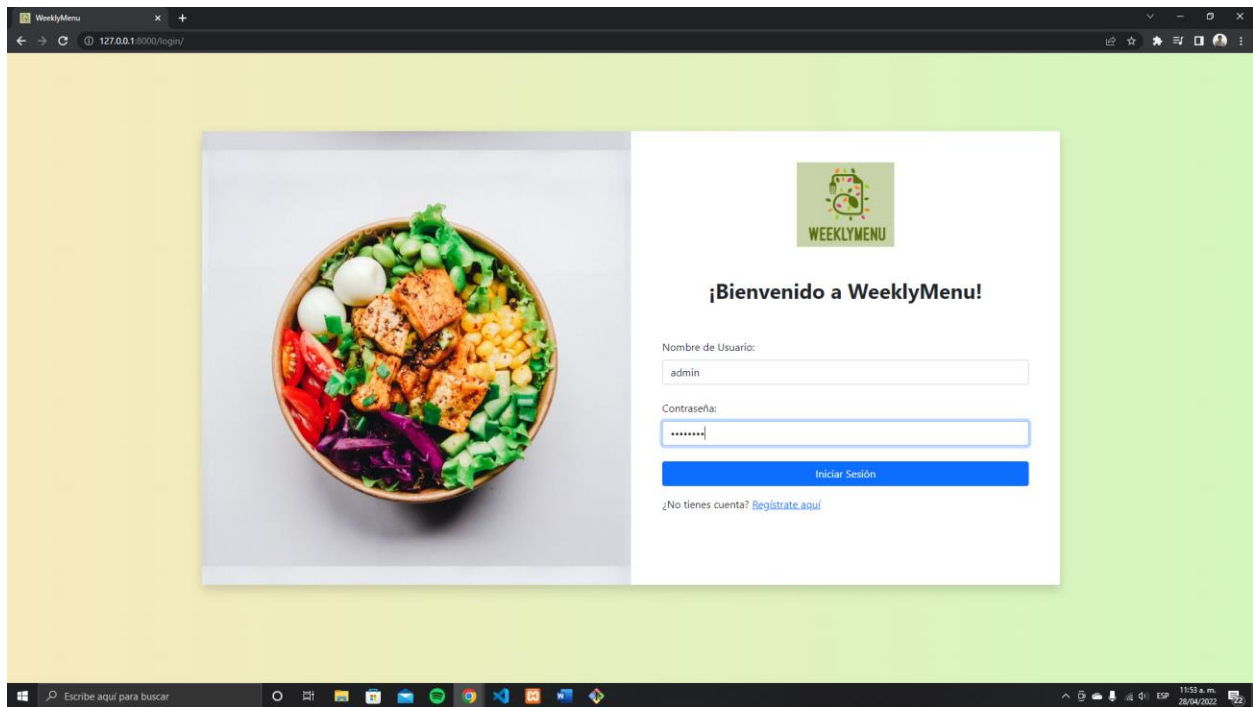
1.1. Primero, elegir la opción de inicio de sesión



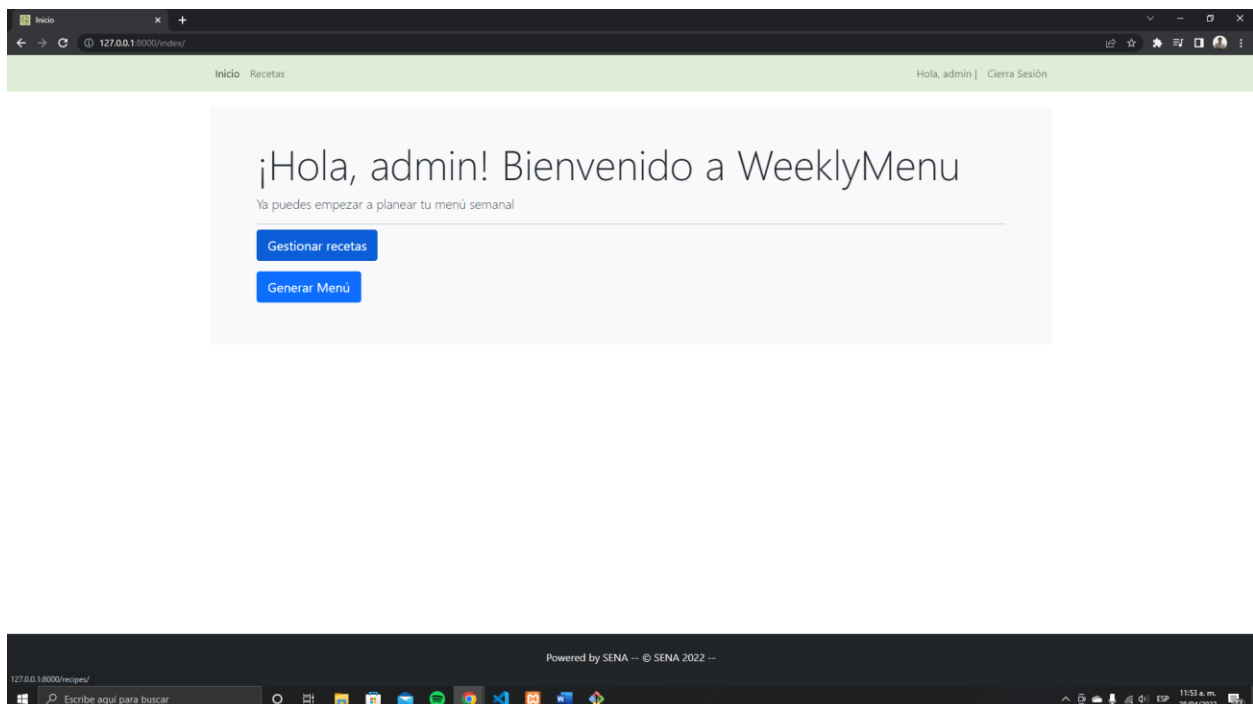
1.2. Iniciar sesión como administrador con las credenciales:

Usuario: 'admin'

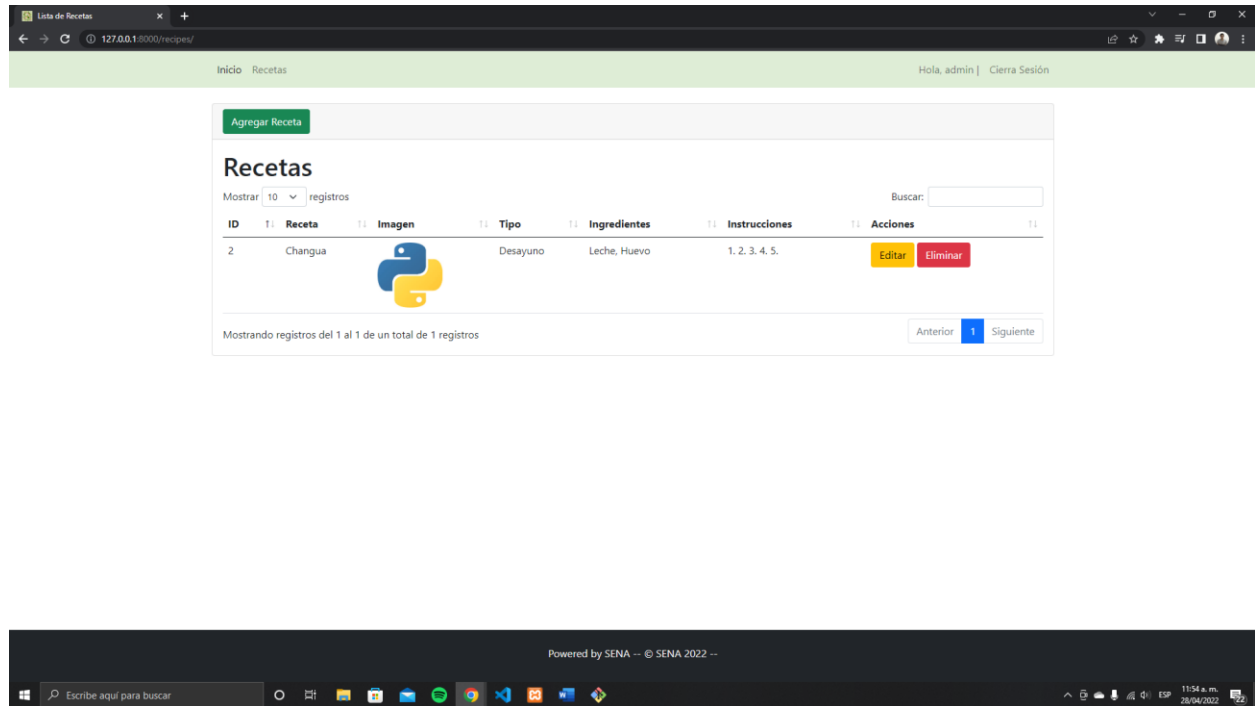
Contraseña 'adsi1234'



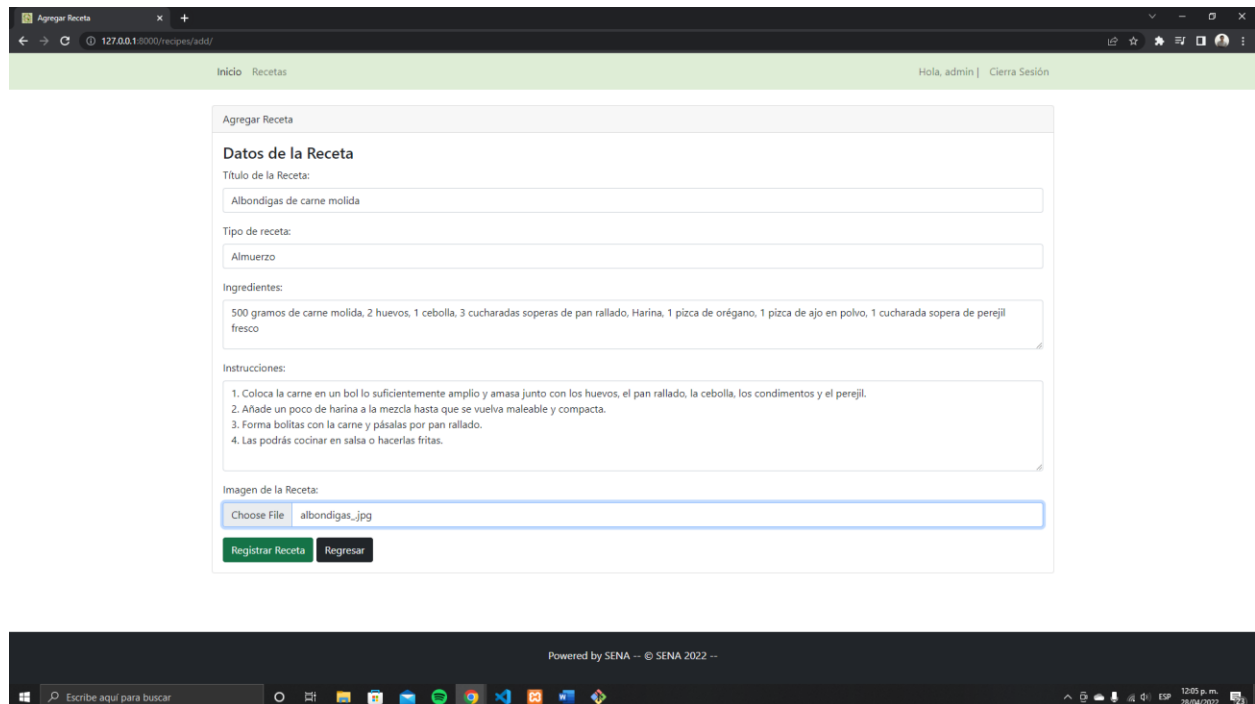
1.3. Clicar en gestionar recetas



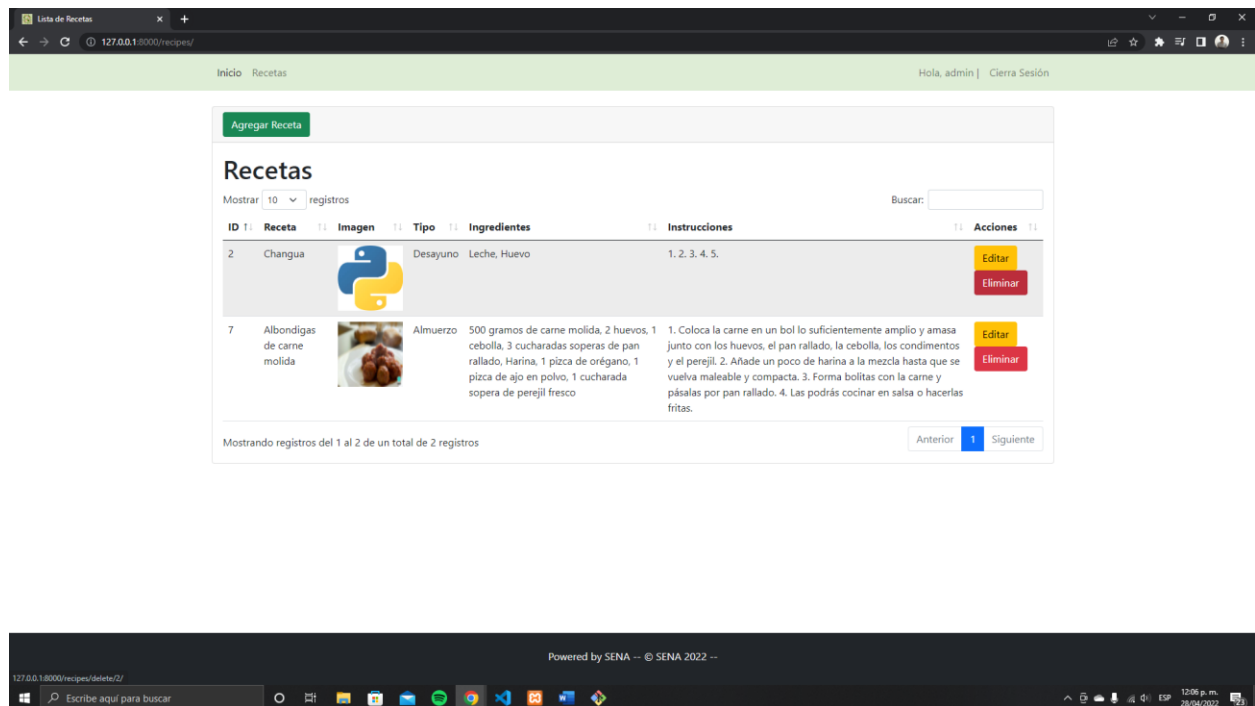
1.4. Se ven las recetas actuales en la aplicación



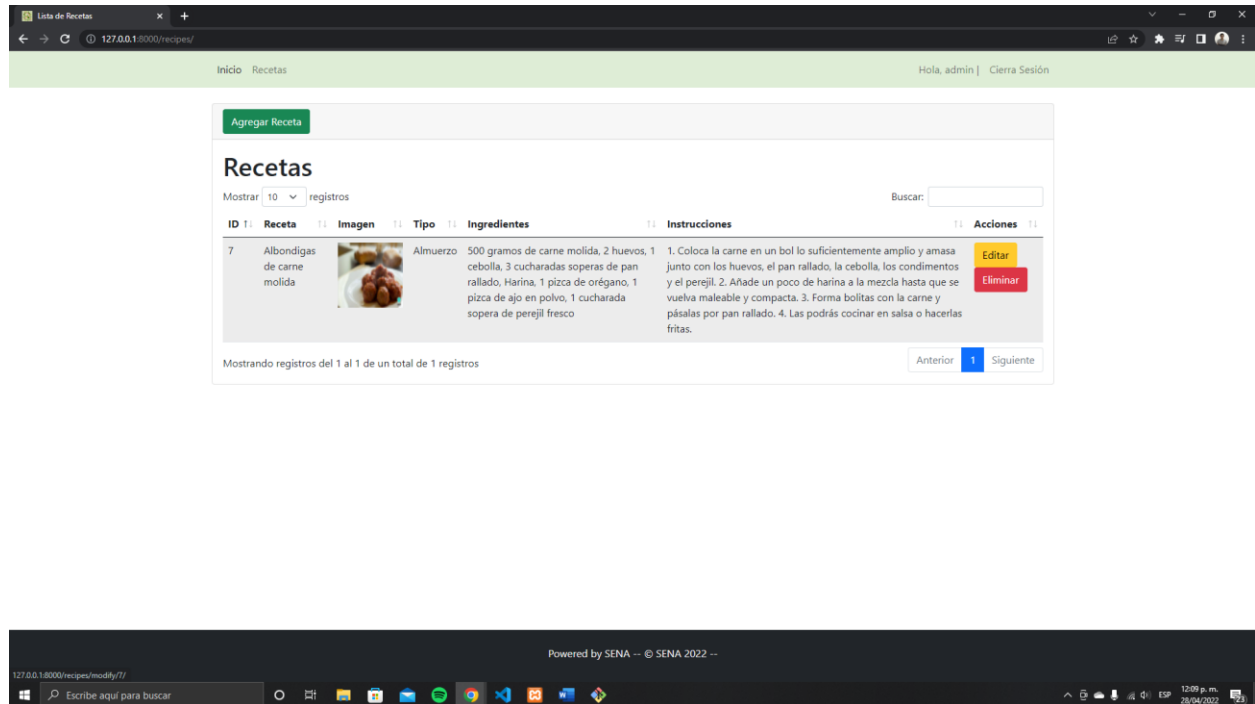
1.5. Para agregar una receta se hace clic en Agregar Receta y rellenamos el formulario



1.6. Volvemos a la lista de recetas, verificamos la nueva receta. Ahora procedemos a eliminar una receta



1.7. Ahora podemos modificar la receta que nos queda



1.8. Modificamos los datos y registramos el cambio

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/recipes/modify/1/'. The page title is 'Modificar Receta'. The navigation bar includes 'Inicio', 'Recetas', and a user profile 'Hola, admin | Cierre Sesión'. The main content area is titled 'Editar Receta' and contains the following fields:

- Datos de la Receta:**
 - Título de la Receta: Albondigas de Ternera
 - Tipo de receta: Almuerzo
 - Ingredientes: 500 gramos de carne molida, 2 huevos, 1 cebolla, 3 cucharadas soperas de pan rallado, Harina, 1 pizca de orégano, 1 pizca de ajo en polvo, 1 cucharada sopera de perejil fresco
 - Instrucciones:
 - Coloca la carne en un bol lo suficientemente amplio y amasa junto con los huevos, el pan rallado, la cebolla, los condimentos y el perejil.
 - Añade un poco de harina a la mezcla hasta que se vuelva maleable y compacta.
 - Imagen de la Receta: [Image of a meatball dish]
- Buttons: 'Registrar Receta' (green) and 'Regresar' (black).

The footer of the browser window shows 'Powered by SENA -- © SENA 2022 --' and the system clock indicates '13:12 p.m. 28/04/2022'.

2. Experiencia de usuario regular

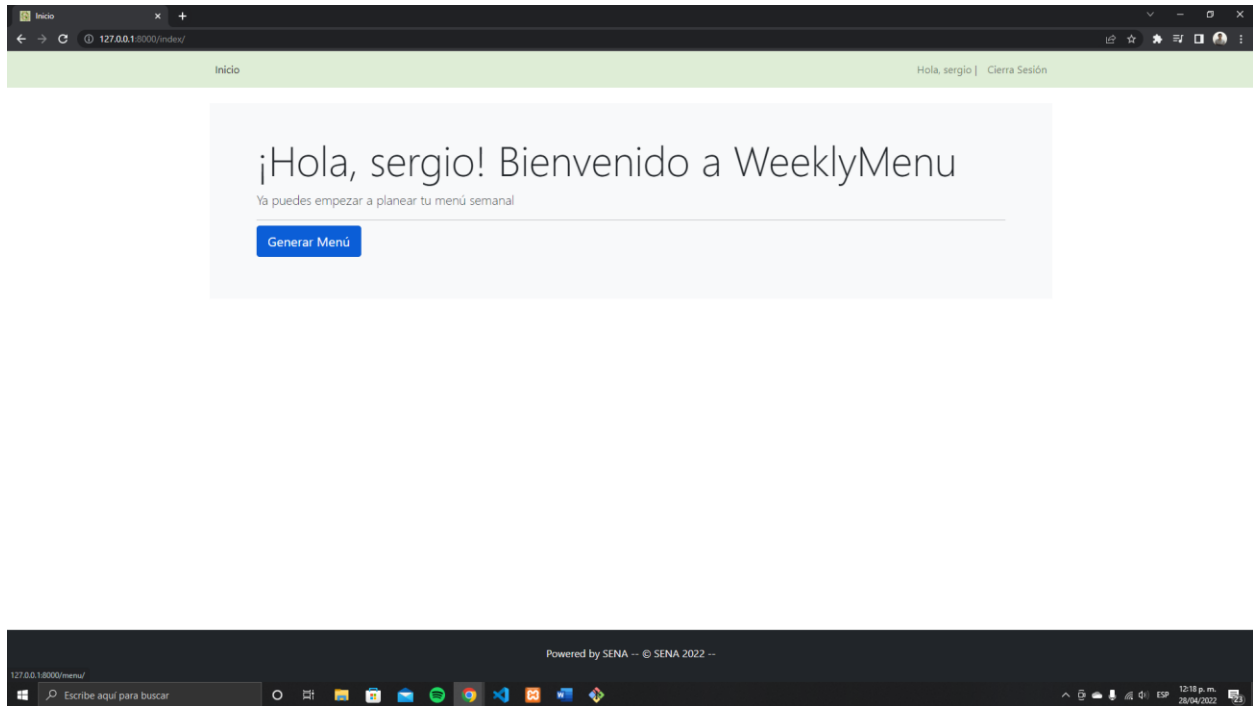
2.1. En lugar de iniciar sesión, ahora creamos una cuenta nueva

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/sign_up/'. The page title is 'WeeklyMenu'. The background features a large image of a bowl of ramen. The registration form is titled 'Crea tu cuenta en WeeklyMenu' and includes the following fields:

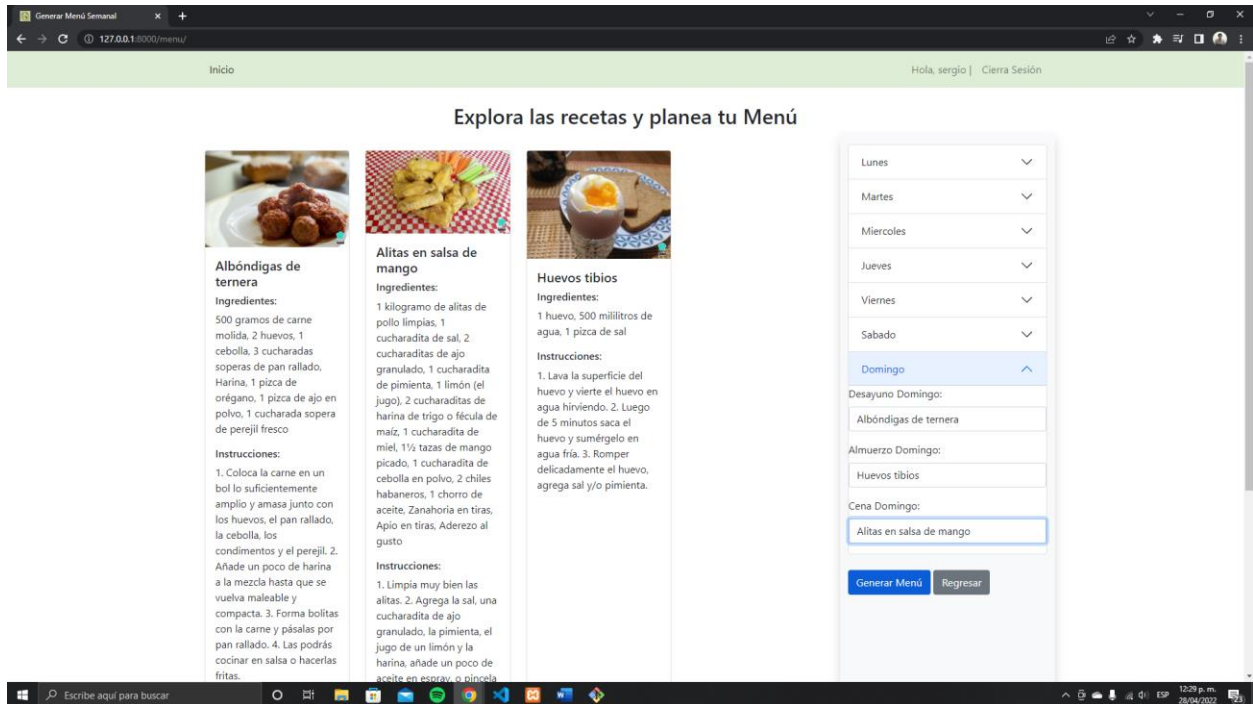
- Nombre de Usuario: sergio
- Correo electrónico: seragonzalez11@misena.edu.co
- Contraseña: [masked]
- Confirma tu Contraseña: [masked]
- Buttons: 'Registrar Datos' (blue) and 'Regresar' (black).

The footer of the browser window shows the system clock indicating '13:18 p.m. 28/04/2022'.

2.2. Luego de registrar nuestros datos podemos generar un nuevo menú semanal



2.3. Exploramos las recetas disponibles y en el menú del costado derecho se planifica el menú relleno los campos del formulario



2.4. Se genera el menú semanal y se muestra por tablas de acuerdo al día

The screenshot shows a web browser displaying the 'Menú Semanal' application. The page has a green header with 'Inicio' on the left and 'Hola, sergio | Cierra Sesión' on the right. The main content area displays two sections: 'Menú Lunes' and 'Menú Martes'. Each section contains a table with columns: 'Tipo', 'Receta', 'Imagen', 'Ingredientes', and 'Instrucciones'.

Menú Lunes

| Tipo | Receta | Imagen | Ingredientes | Instrucciones |
|----------|--------------------------|--------|--|--|
| Almuerzo | Albóndigas de ternera | | 500 gramos de carne molida, 2 huevos, 1 cebolla, 3 cucharadas soperas de pan rallado, Harina, 1 pizca de orégano, 1 pizca de ajo en polvo, 1 cucharada sopera de perejil fresco | 1. Coloca la carne en un bol lo suficientemente amplio y amasa junto con los huevos, el pan rallado, la cebolla, los condimentos y el perejil. 2. Añade un poco de harina a la mezcla hasta que se vuelva maleable y compacta. 3. Forma bolitas con la carne y pásalas por pan rallado. 4. Las podrás cocinar en salsa o hacerlas fritas. |
| Almuerzo | Alitas en salsa de mango | | 1 kilogramo de alitas de pollo limpias, 1 cucharadita de sal, 2 cucharaditas de ajo granulado, 1 cucharadita de pimienta, 1 limón (el jugo), 2 cucharaditas de harina de trigo o fécula de maíz, 1 cucharadita de miel, 1½ tazas de mango picado, 1 cucharadita de cebolla en polvo, 2 chiles habaneros, 1 chorro de aceite, Zanahoria en tiras, Apio en tiras, Aderezo al gusto | 1. Limpia muy bien las alitas. 2. Agrega la sal, una cucharadita de ajo granulado, la pimienta, el jugo de un limón y la harina, añade un poco de aceite en espray, o pincela las alitas, y hornéalas a 180 °C con calor por arriba y por abajo por 15-20 minutos. 3. Asa los chiles habaneros y luego lícuolos con el mango por tres minutos junto con miel, cebolla y ajo en polvo. 4. Cocina la mezcla y reserva para cubrir las alitas posterior a que las alitas estén horneadas. |
| Desayuno | Huevos tibios | | 1 huevo, 500 mililitros de agua, 1 pizca de sal | 1. Lava la superficie del huevo y vierte el huevo en agua hirviendo. 2. Luego de 5 minutos saca el huevo y sumérgelo en agua fría. 3. Romper delicadamente el huevo, agrega sal y/o pimienta. |

Menú Martes

| Tipo | Receta | Imagen | Ingredientes | Instrucciones |
|----------|-----------------------|--------|---|---|
| Almuerzo | Albóndigas de ternera | | 500 gramos de carne molida, 2 huevos, 1 cebolla, 3 cucharadas soperas de pan rallado, Harina, 1 pizca de orégano, 1 pizca de ajo en polvo, 1 cucharada sopera de perejil fresco | 1. Coloca la carne en un bol lo suficientemente amplio y amasa junto con los huevos, el pan rallado, la cebolla, los condimentos y el perejil. 2. Añade un poco de harina a la mezcla hasta que se vuelva maleable y compacta. 3. Forma bolitas con la carne y pásalas por pan rallado. 4. Las podrás cocinar en salsa o hacerlas fritas. |

2.5. La siguiente página muestra la lista de ingredientes

The screenshot shows a web browser displaying the 'Lista de Compra' application. The page has a green header with 'Inicio' on the left and 'Hola, sergio | Cierra Sesión' on the right. The main content area displays a section titled 'Lista de Compra' which contains a list of ingredients organized into groups. Each group is separated by a horizontal line and includes a small image of the ingredient.

Lista de Compra

- 500 gramos de carne molida, 2 huevos, 1 cebolla, 3 cucharadas soperas de pan rallado, Harina, 1 pizca de orégano, 1 pizca de ajo en polvo, 1 cucharada sopera de perejil fresco
- 1 kilogramo de alitas de pollo limpias, 1 cucharadita de sal, 2 cucharaditas de ajo granulado, 1 cucharadita de pimienta, 1 limón (el jugo), 2 cucharaditas de harina de trigo o fécula de maíz, 1 cucharadita de miel, 1½ tazas de mango picado, 1 cucharadita de cebolla en polvo, 2 chiles habaneros, 1 chorro de aceite, Zanahoria en tiras, Apio en tiras, Aderezo al gusto
- 1 huevo, 500 mililitros de agua, 1 pizca de sal
- 500 gramos de carne molida, 2 huevos, 1 cebolla, 3 cucharadas soperas de pan rallado, Harina, 1 pizca de orégano, 1 pizca de ajo en polvo, 1 cucharada sopera de perejil fresco
- 1 huevo, 500 mililitros de agua, 1 pizca de sal
- 500 gramos de carne molida, 2 huevos, 1 cebolla, 3 cucharadas soperas de pan rallado, Harina, 1 pizca de orégano, 1 pizca de ajo en polvo, 1 cucharada sopera de perejil fresco
- 1 kilogramo de alitas de pollo limpias, 1 cucharadita de sal, 2 cucharaditas de ajo granulado, 1 cucharadita de pimienta, 1 limón (el jugo), 2 cucharaditas de harina de trigo o fécula de maíz, 1 cucharadita de miel, 1½ tazas de mango picado, 1 cucharadita de cebolla en polvo, 2 chiles habaneros, 1 chorro de aceite, Zanahoria en tiras, Apio en tiras, Aderezo al gusto
- 1 huevo, 500 mililitros de agua, 1 pizca de sal
- 500 gramos de carne molida, 2 huevos, 1 cebolla, 3 cucharadas soperas de pan rallado, Harina, 1 pizca de orégano, 1 pizca de ajo en polvo, 1 cucharada sopera de perejil fresco
- 1 kilogramo de alitas de pollo limpias, 1 cucharadita de sal, 2 cucharaditas de ajo granulado, 1 cucharadita de pimienta, 1 limón (el jugo), 2 cucharaditas de harina de trigo o fécula de maíz, 1 cucharadita de miel, 1½ tazas de mango picado, 1 cucharadita de cebolla en polvo, 2 chiles habaneros, 1 chorro de aceite, Zanahoria en tiras, Apio en tiras, Aderezo al gusto
- 1 huevo, 500 mililitros de agua, 1 pizca de sal
- 500 gramos de carne molida, 2 huevos, 1 cebolla, 3 cucharadas soperas de pan rallado, Harina, 1 pizca de orégano, 1 pizca de ajo en polvo, 1 cucharada sopera de perejil fresco
- 1 kilogramo de alitas de pollo limpias, 1 cucharadita de sal, 2 cucharaditas de ajo granulado, 1 cucharadita de pimienta, 1 limón (el jugo), 2 cucharaditas de harina de trigo o fécula de maíz, 1 cucharadita de miel, 1½ tazas de mango picado, 1 cucharadita de cebolla en polvo, 2 chiles habaneros, 1 chorro de aceite, Zanahoria en tiras, Apio en tiras, Aderezo al gusto