

## Programación Orientada a Objetos – Unidad 5

### Caso de estudio: Aplicación web de recetas de cocina

#### Descripción

El restaurante “Sancho Panza” plantea como estrategia de marketing una aplicación web donde los usuarios registrados compartan recetas. Al final de cada mes, el usuario que compartió la receta con más “Me gusta” recibe como premio una cena gratis.

Un usuario registrado puede cargar una nueva receta, consultar el ranking mensual de recetas, consultar recetas, dar “Me gusta” a recetas que no sean propias.

#### **FUNCIONALIDAD #1: Acceso a la aplicación**

##### Descripción

<b>COMO</b>	Usuario Registrado
<b>NECESITO</b>	Acceder a la aplicación
<b>PARA</b>	Hacer uso de las funcionalidades que la misma provee

##### Escenario

#1		Comentarios
	El usuario ingresa su correo electrónico y la contraseña.	Los caracteres de la contraseña no se deben ver al momento del ingreso.  Los usuarios deben estar previamente registrados.
	La aplicación verifica los datos ingresados.	
	Si el correo y la contraseña introducidos corresponden a un usuario registrado, la aplicación da acceso a las funcionalidades.	
	Si el correo y la contraseña introducidos no corresponden a un usuario registrado, la aplicación muestra un mensaje de error.	

#### **FUNCIONALIDAD #2: Ingresar una receta**

##### Descripción

<b>COMO</b>	Usuario Registrado
<b>NECESITO</b>	Acceder a la aplicación

<b>PARA</b>	Registrar una receta
-------------	----------------------

#### Escenario

#2		Comentarios
	El usuario selecciona la opción "Compartir receta".	
	El usuario ingresa el nombre de la receta, el tiempo estimado para realizarla, la descripción del proceso de elaboración, y de cada ingrediente que se requiere el nombre del ingrediente, la cantidad y la unidad de medida.	El tiempo estimado se ingresa en minutos.
	El usuario pulsa "Guardar receta"	
	La aplicación crea la receta, carga el valor 0 en la cantidad de "Me gusta" y la fecha actual.	La fecha actual se obtiene del sistema.

### **FUNCIONALIDAD #3: Consultar ranking**

#### Descripción

<b>COMO</b>	Usuario Registrado
<b>NECESITO</b>	Acceder a la aplicación
<b>PARA</b>	Consultar el ranking de recetas

#### Escenario

#3		Comentarios
	El usuario selecciona la opción "Consultar ranking".	
	La aplicación muestra las 5 recetas con más "Me gusta".	El listado debe visualizarse ordenado descendientemente por cantidad de "Me gusta"  Los datos a mostrar son: Nombre de la receta, Nombre del usuario que la compartió y cantidad de "Me gusta"

### **FUNCIONALIDAD #4: Consultar una receta por tiempo de elaboración**

#### Descripción

<b>COMO</b>	Usuario Registrado
<b>NECESITO</b>	Acceder a la aplicación
<b>PARA</b>	Consultar una receta cuyo tiempo de elaboración sea menor a un tiempo dado

#### Escenario

#4		Comentarios
	El usuario selecciona la opción “Consultar recetas por tiempo”.	
	La aplicación solicita que ingrese el tiempo.	
	La aplicación muestra un listado con el nombre de cada receta cuyo tiempo de elaboración sea menor al ingresado.	
	El usuario selecciona una receta.	
	La aplicación muestra los datos de la receta, los datos de los ingredientes y el nombre del usuario que la compartió. También debe aparecer el botón “Me gusta” si el usuario que está consultando la receta no es el mismo que la ha compartido.	
	Si el usuario dio clic al botón “Me gusta” la aplicación incrementa la cantidad de “Me gusta” de la receta.	

#### **FUNCIONALIDAD #5: Consultar una receta por ingrediente**

##### **Descripción**

<b>COMO</b>	Usuario Registrado
<b>NECESITO</b>	Usuario Registrado
<b>PARA</b>	Consultar las recetas que contengan un ingrediente determinado

##### **Escenario**

#5		Comentarios
	El usuario selecciona la opción “Consultar recetas por ingrediente”.	
	La aplicación solicita que ingrese el ingrediente.	
	La aplicación muestra un listado con el nombre de las recetas que necesitan el ingrediente ingresado.	La coincidencia entre el ingrediente ingresado y el de la receta puede ser total o parcial. Ej. Si se ingresa “queso”, se debería obtener recetas tengan como ingrediente: queso, queso rallado, queso parmesano, etc.
	El usuario selecciona una receta.	
	La aplicación muestra los datos de la receta, los datos de los ingredientes y el nombre del usuario que la compartió. También debe aparecer el botón “Me gusta” si el usuario que está consultando la receta no es el mismo que la ha compartido	

Si el usuario dio clic al botón “Me gusta” la aplicación incrementa la cantidad de “Me gusta” de la receta	
--	--

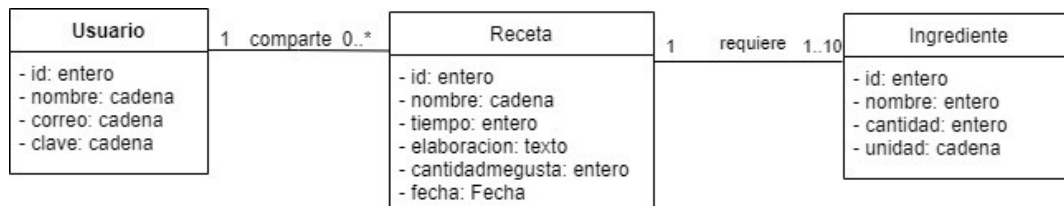
## Usuarios

Los usuarios de la aplicación son los usuarios registrados.

## Reglas de negocio globales

1. Un usuario puede compartir muchas recetas.
2. Un usuario registrado puede no compartir recetas.
3. Una receta puede tener como máximo 10 ingredientes.
4. En todas las páginas de la aplicación debe figurar el nombre del restaurante.

## Diagrama de clases



## Tablas de base de datos

### Usuario

id	nombre	correo	clave
----	--------	--------	-------

Nota: la clave de los usuarios que ya están guardados en la base de datos, es el nombre de usuario del correo electrónico (anaestrada, luchicortez, martosj, gimeaballay67).

### Receta

id	nombre	tiempo	elaboracion	cantidadmegusta	fecha	usuarioid
----	--------	--------	-------------	-----------------	-------	-----------

### Ingrediente

id	nombre	cantidad	unidad	recetaid
----	--------	----------	--------	----------

## Consideraciones para la resolución

1. Definir las clases correspondientes al diseño planteado.
2. Diseñar páginas fáciles de usar, comprensibles y agradables para el usuario.
3. Para ingresar y mostrar datos utilizar los controles apropiados.
4. Usar Flask y SQLAlchemy para manejar desde la aplicación los datos almacenados en la base de datos.

5. Verificar la validez de los datos (fechas, números, etc.) ingresados y de los datos que se almacenan en la base de datos.
6. Las contraseñas de los usuarios se deben almacenar cifradas por razones de seguridad.

Para cifrar las contraseñas pueden utilizar el siguiente código:

```
import hashlib
if __name__ == "__main__":
    # cifrado de la clave utilizando md5
    clave = 'julia459'
    result = hashlib.md5(bytes(clave, encoding='utf-8'))
    # muestra la clave cifrada en hexadecimal, esta es la que se guarda en base de
    datos
    print(result.hexdigest())
```

**Nota:** las contraseñas **no se descifran**. Se ingresa una contraseña, se cifra, el valor obtenido se compara con el valor registrado en la base de datos.