

# INSTITUTO TECNOLÓGICO DE NUEVO LAREDO

CON LA CIENCIA POR LA HUMANIDAD

## INGENIERÍA EN SISTEMAS COMPUTACIONALES



### Programación móvil 2



#### DOCENTE

Ing. Humberto Peña Valle M.T.I.



#### INTEGRANTES

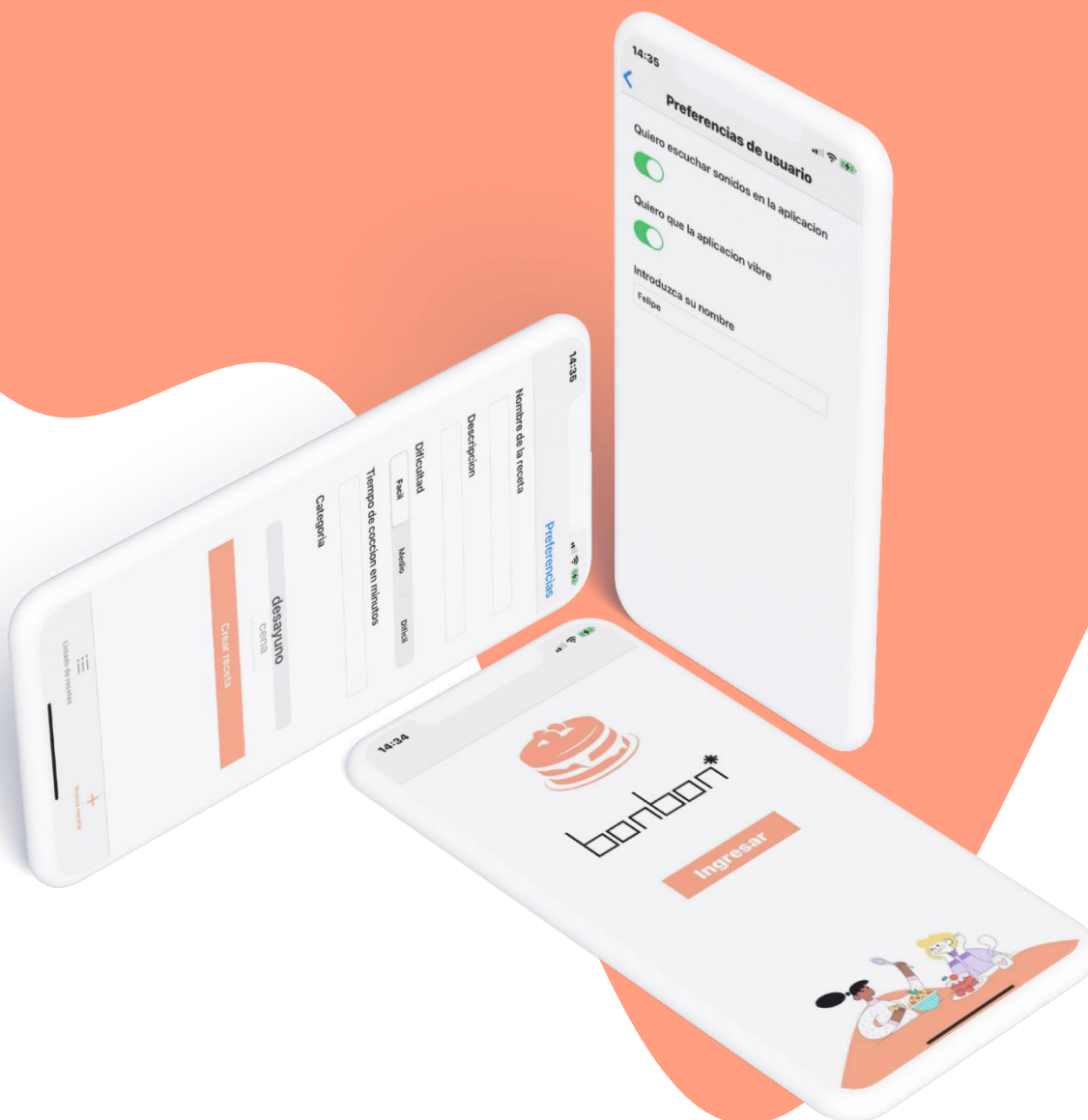
Jessica Janet Grajeda Castellanos	17100229
Juan Felipe Garza Sánchez	17100218
Miguel Ángel Méndez Cruz	17100254

# CONTENIDO

1.	<u>EXPLICACIÓN DEL TEMA</u>	5
2.	<u>CAPTURAS DE PANTALLA</u>	6
3.	<u>EXPLICACIÓN DEL MANEJO DE DATOS</u>	10
3.1	<u>BASES DE DATOS</u>	10
3.2	<u>WEB SERVICES O WEB APP</u>	11
3.3	<u>DATOS DE CONFIGURACIÓN</u>	15
	<u>CONCLUSIONES</u>	16

bonbon\*



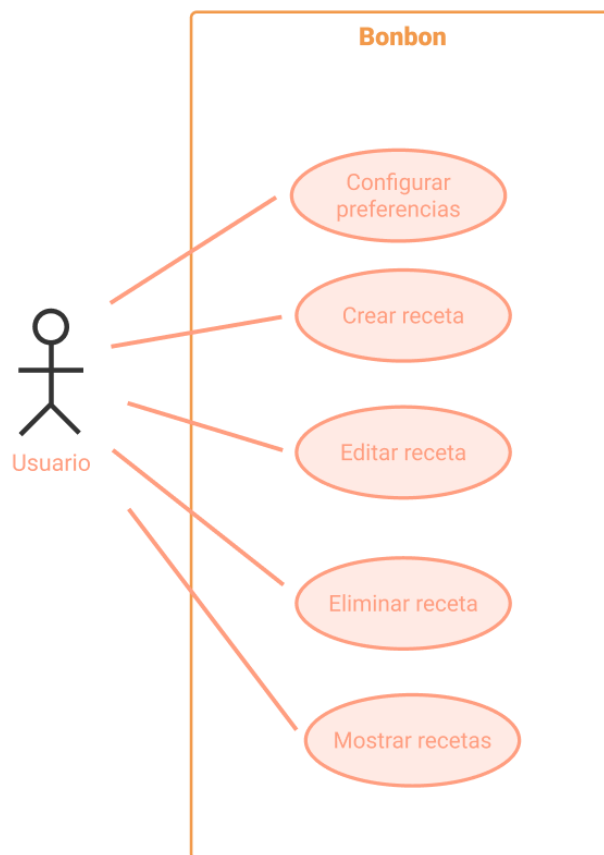


# Explicación

## ¿Qué es Bonbon?

Bonbon es una aplicación que ha sido creada con el propósito de compartir con otros usuarios la experiencia que han tenido al llevar a cabo una receta de cocina, de tal forma que puedan dar su punto de vista a otras personas dando a conocer los siguientes datos:

- Nombre de la receta
- Una breve descripción o comentarios de la receta
- La dificultad que tuvieron para implementar la receta
- Categoría a la que pertenece la receta de cocina
- Tiempo de cocción promedio dado en minutos

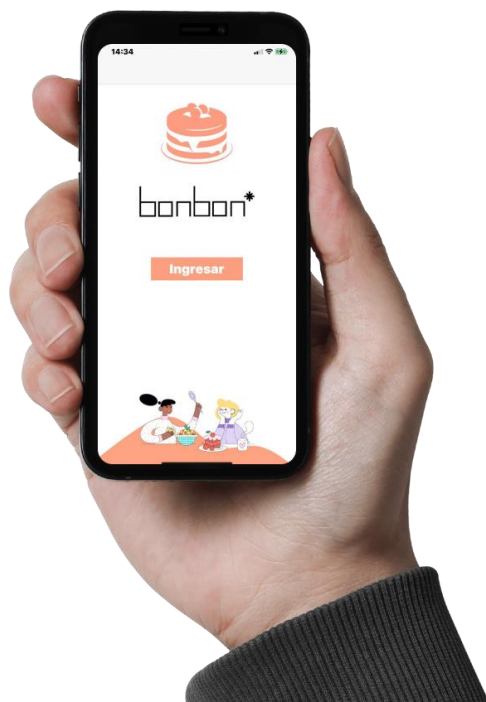


La aplicación como tal puede consultar, dar de alta, modificar y eliminar recetas.

## Pantalla de bienvenida

Como primera pantalla tenemos la bienvenida al usuario al sistema, dando un acceso directo a este por medio de un sencillo botón centrado.

Como dato adicional, al seleccionar el botón de ingresar se reproducirá un sonido característico de bienvenida, en donde dicho sonido puede ser omitido si el usuario desea asignarlo en el apartado de preferencias de usuario.

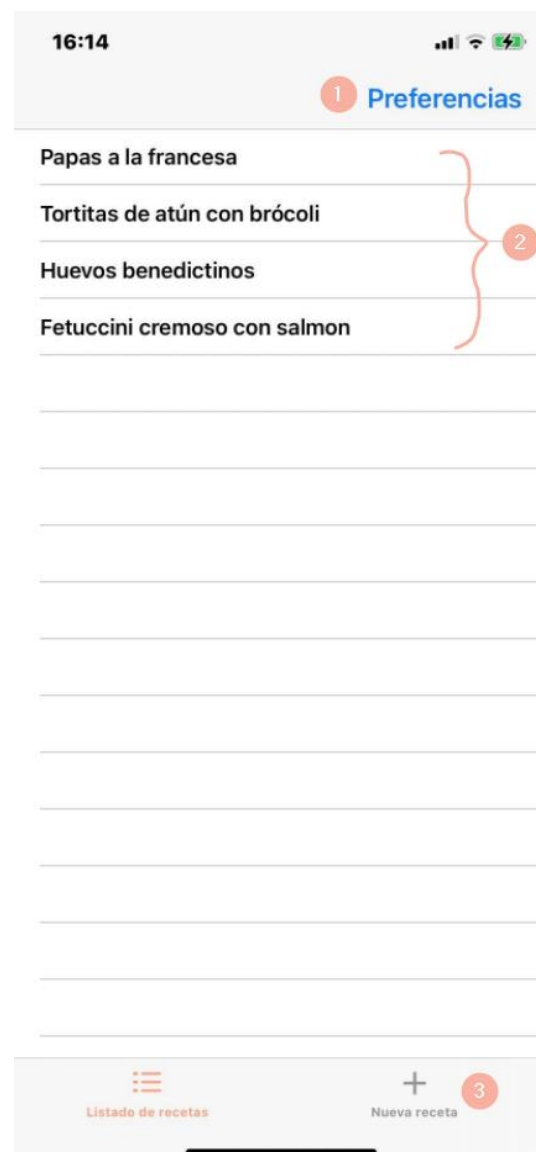
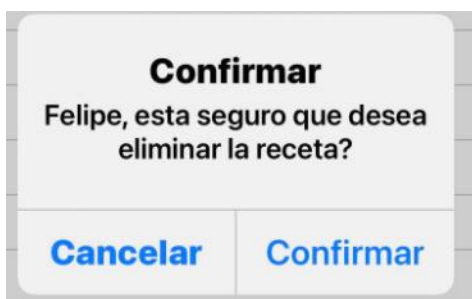


## Pantalla de listado

Es aquí en donde podemos ver todas las recetas agregadas al sistema (punto 2), así como posibilidad de acceder a una en específico para editarla (dando tap rápido sobre el registro), eliminándolo (presionando el registro hasta que aparezca una alerta de confirmación), o incluso crearla desde la opción que aparece en el tab bar menú de abajo (punto 3).

También se tiene opción de acceder a las preferencias de usuario en la parte superior derecha (punto 1)

Al intentar eliminar un registro presionando sobre el aparecerá la siguiente alerta:



14:35



## Preferencias

Nombre de la receta

Descripcion

Dificultad

Facil

Medio

Dificil

Tiempo de coccion en minutos

Categoria

desayuno

cena

almuerzo

Crear receta



Listado de recetas



Nueva receta

## Pantalla de creado

Aparecerá un formulario tal cual se muestra en la imagen de la izquierda, es aquí en donde debemos llenar todos los campos sin excepción alguna, en caso de que hayamos omitido alguno en específico, el sistema lanzara un mensaje de error con aquellos campos a los cuales nos faltó llenar.







De lo contrario, saldrá un mensaje satisfactorio y se reiniciarán todos los campos para poder crear una nueva receta si así se desea.



## Pantalla de edición

Aparecerá un formulario precargado con los datos de la receta a modificar, tal cual se muestra en la imagen de la izquierda, es aquí en donde debemos llenar todos los campos sin excepción alguna, en caso de que hayamos omitido alguno en específico, el sistema lanzará un mensaje de error con aquellos campos a los cuales nos faltó llenar.

14:36




 Back
Editar receta

Nombre de la receta

Descripcion

Dificultad

Facil
Medio
Dificil

Tiempo de coccion en minutos

Categoria

desayuno
cena
almuerzo
postre

GUARDAR

**Ha ocurrido un error!**  
Felipe, se han encontrado los siguientes errores:  
- hace falta introducir el nombre de la receta  
- no se ha dado una descripcion de la receta  
- no se ha introducido el tiempo de coccion

Aceptar

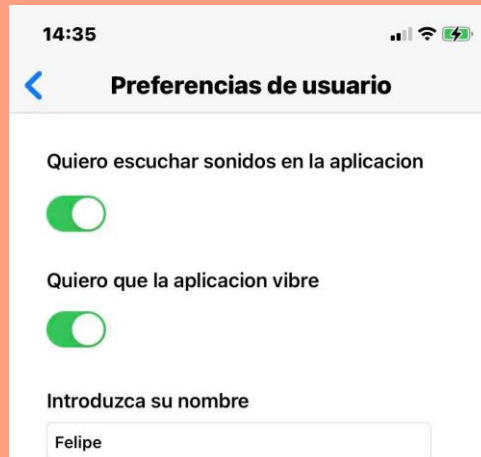
De lo contrario, saldrá un mensaje satisfactorio.

**Aviso**  
Felipe, se ha creado la receta con exito

OK



## Pantalla de preferencias de usuario



Es aquí en donde se guardan las preferencias del usuario, podremos definir si queremos escuchar sonidos dentro de la aplicación (como ejemplo esta al pulsar el botón de ingresar en la pantalla de inicio, o cuando se manda una alerta de error).

También se puede definir si se quiere o no se quiere activar la vibración a la hora de mandar una alerta de error.

Por ultimo tenemos una opción para que la aplicación sepa nuestro nombre y se nos indique cuando sea necesario, ejemplo:

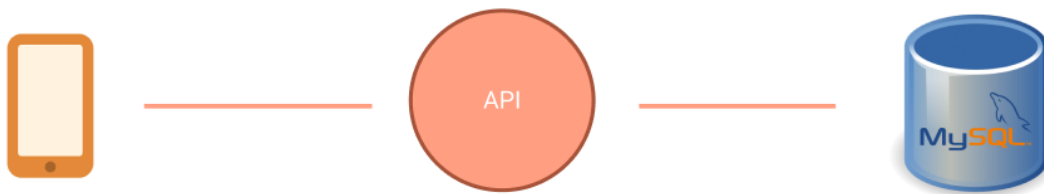


Para ver reflejado los cambios basta con simplemente modificar los controles.

# Manejo de datos

Para el guardado de las recetas la aplicación utiliza una API ubicada en el siguiente punto de acceso:

<http://fegarza.com/api/>

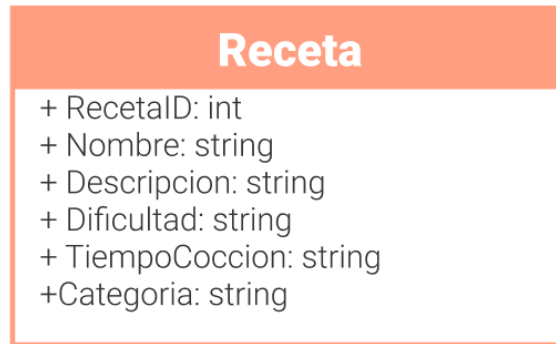


Dicha API esta creada con PHP y tiene acceso al motor de base datos de MySQL para manejar las distintas operaciones de las recetas.

La tabla receta luce de la siguiente forma, en donde su llave primaria es un entero incrementable (RecetaID).



La representación de la receta como clase sería la siguiente:



En donde las operaciones definidas para las recetas son las siguientes:

- Dar de alta
- Editar
- Eliminar
- Consultar

Para definir el tipo de operación basta con agregar el parámetro “operación” sobre el mismo punto de acceso definido con anterioridad y un número que caracterice la operación.

Valores del parámetro “operación”:

- Para una alta usar el valor “1”  
<http://fegarza.com/api/?operacion=1>
- Para editar un registro el valor “2”  
<http://fegarza.com/api/?operacion=2>
- Para una consulta usar el valor “3”  
<http://fegarza.com/api/?operacion=3>
- Para una eliminación usar el valor “4”  
<http://fegarza.com/api/?operacion=4>

Al consumir la API sin importar cual sea la operación se devolverá un contenido de tipo JSON a quien consuma la API, ya que es la manera en que tanto el cliente que consume el API y la API se comuniquen entre sí.

## Operación de alta (1)

Esta operación trabaja con un tipo de petición POST y necesita de los siguientes parámetros en el cuerpo de la consulta para funcionar.

	KEY	VALUE
<input type="checkbox"/>	RecetaID	35
<input checked="" type="checkbox"/>	Nombre	ejemplo nombre de la receta
<input checked="" type="checkbox"/>	Dificultad	Medio
<input checked="" type="checkbox"/>	Tiempo Coccion	22
<input checked="" type="checkbox"/>	Categoria	facil

Una vez enviada la petición y siendo correcta la operación, devolverá como tal el registro insertado en una representación JSON.

```

1  {
2    "RecetaID": 70,
3    "Nombre": "ejemplo nombre de la receta",
4    "Descripcion": "",
5    "Dificultad": "Medio",
6    "TiempoCoccion": "",
7    "Categoria": "facil",
8    "NUsuario": "APP IOS"
9  }
10
11

```

Status: 200 OK Time: 281 ms Size: 396 B

## Operación de editado (2)

Funciona similar que la operación de alta, sin embargo, aquí es necesario agregar el parámetro RecetalD y su respectivo valor, ya que es con este con el cual haremos referencia a que registro queremos editar.

<https://fegarza.com/api/?operacion=2>

POST <span>▼</span> <a href="https://fegarza.com/api/?operacion=2">https://fegarza.com/api/?operacion=2</a>		
Params <span>●</span>	Authorization	Headers (8)
Body <span>●</span>	Pre-request Script	Tests
Settings		
<input type="radio"/> none <input checked="" type="radio"/> form-data <input type="radio"/> x-www-form-urlencoded <input type="radio"/> raw <input type="radio"/> binary <input type="radio"/> GraphQL		
KEY	VALUE	
<input checked="" type="checkbox"/> RecetalD	70	
<input checked="" type="checkbox"/> Nombre	ejemplo de editado	
<input checked="" type="checkbox"/> Dificultad	Medio	
<input checked="" type="checkbox"/> Tiempo Coccion	22	
<input checked="" type="checkbox"/> Categoria	facil	
<input checked="" type="checkbox"/> NUsuario	anonimo	

Dando como resultado el registro editado en una representación JSON.

Body

Cookies

Headers (9)

Test Results

Status: 200 OK

Time: 279 ms

Size: 376 B

Pretty

Raw

Preview

Visualize

JSON



1

2

3

4

5

6

7

1

2

3

4

5

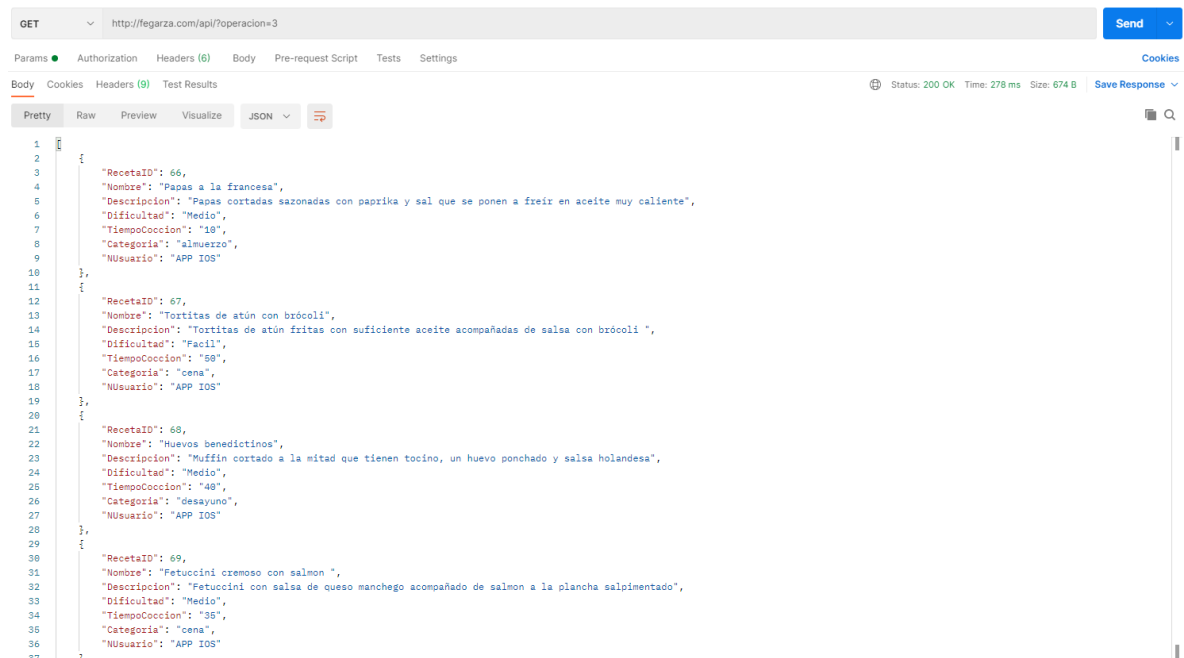
6

7

```
{
  "id": "70",
  "nombre": "ejemplo de editado",
  "descripcion": "",
  "dificultad": "Medio",
  "tiempoCoccion": "22",
  "categoria": "facil",
}
```

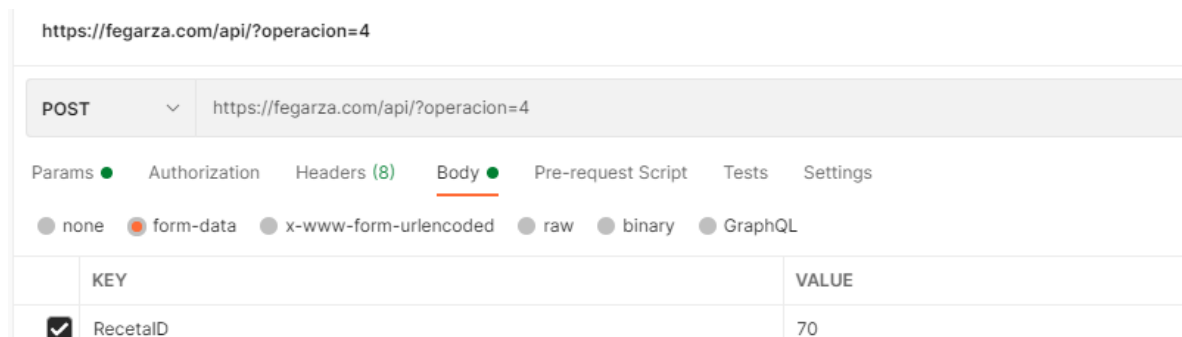
### Operación de consulta (3)

Esta operación trabaja con un tipo de petición GET y no necesita un parámetro en especial más que su tipo de operación.



### Operación de eliminado (4)

Es de tipo POST y basta con definir el parámetro RecetaID



## ¿Cómo se maneja el consumo de la API dentro de la aplicación? (Configuración)

Para agilizar el proceso de consulta, sabiendo que la aplicación solo hace 4 operaciones como lo es alta, creado, listado y baja de recetas se creó un enumerador.

```
enum Operacion: String{
    case alta = "1"
    case edicion = "2"
    case consulta = "3"
    case baja = "4"
}
```

De ahí se optó por hacer una extensión de la clase URLRequest, de tal manera que los parámetros se asignaran dependiendo del tipo de operación y la instancia de la clase recibo, quedando de la siguiente forma:

```
extension URLRequest{
    init(endpoint: String, operacion: Operacion, receta: Receta?) throws {
        guard let url = URL(string: endpoint + "?operacion=\(String(operacion.rawValue))") else{
            throw ErrorManager.runtimeError("No se ha logrado crear la instancia de URL")
        }
        self.init(url: url)

        if(operacion == Operacion.consulta && receta == nil){
            self.httpMethod = "GET"
        }else{
            self.httpMethod = "POST"
        }

        self.httpBody = self.obtenerParametros(operacion: operacion, receta: receta).data(using: .utf8)
    }

    private func obtenerParametros(operacion: Operacion, receta: Receta?) -> String{
        if(receta == nil){
            return ""
        }

        var cadenaDeParametros = ""

        switch operacion{
            case Operacion.alta, Operacion.edicion:
                cadenaDeParametros = "Nombre=\(receta!.Nombre!)&Descripcion=\(
                    receta!.Descripcion!)&Dificultad=\(receta!.Dificultad!)&TiempoCoccion=\(
                    receta!.TiempoCoccion!)&Categoria=\(receta!.Categoria!)"
                if(operacion == Operacion.edicion){
                    cadenaDeParametros += "&RecetaID="+String(receta!.RecetaID!)
                }
            case Operacion.baja, Operacion.consulta:
                if(receta?.RecetaID != nil){
                    cadenaDeParametros = "RecetaID="+String(receta!.RecetaID!)
                }
        }

        return cadenaDeParametros
    }
}
```

## Conclusiones

### **Jessica Janet Grajeda Castellanos**

A partir de los aprendizajes obtenidos durante este curso pudimos realizar el proyecto aquí presentado, personalmente el desarrollar Bonbon me ayudó mucho a aterrizar temas que no me habían quedado del todo claro, puedo destacar que si tuvimos obstáculos en el proceso ya que no teníamos la práctica que nos hubiese gustado pero de todas maneras al menos yo disfruté el desarrollo de este proyecto y me llevo conmigo el conocimiento para aplicarlo en el futuro en el ambiente laboral

### **Juan Felipe Garza Sánchez**

Implementando todo lo aprendido sobre el curso hemos logrado crear una aplicación con un conjunto de buenas prácticas recomendadas por el profesor, desde lo más básico de crear una variable en Swift hasta el uso de extensiones en los distintos controles que nos ofrece la librería UIKit, también considerando que la aplicación esté funcionando y trabajando con otras aplicaciones externas (en este caso una API), de tal manera que sea ya algo funcional para un usuario final.

Todo lo realizado en esta aplicación fue con un el propósito de practicar todo aquello que se ha visto durante las sesiones de clase, así como reforzar aquellos temas en los cuales llegue a tener confusión por medio de su implementación en una aplicación de producción, también lidiar con los distintos problemas que se dieron durante el desarrollo (bugs y desconocimiento de un tema en particular), todo ello fue de suma importancia para que se logrará una comprensión mayor de los temas, si bien el conocimiento adquirido es de gran utilidad es necesario siempre tener en cuenta y aceptar que jamás es suficiente con lo que sabemos o conocemos, sino que tenemos que tener una perspectiva de aprendizaje a modo que no nos quedemos estancados en el conocimiento adquirido que ya sabemos.

No me queda más que agradecer a mi profesor quien siempre estuvo constantemente impartiéndonos sus conocimientos y experiencias en lo que a desarrollo de aplicaciones en IOS respecta. Por lo menos por mi parte quiero dejar en claro que todo aquello aprendido no será en vano y marcará un antes y un después para mí como desarrollador de aplicaciones móviles.



**Miguel Ángel Méndez Cruz**

En conclusión sobre la aplicación me parece que elegimos bien la temática para cumplir los puntos necesarios donde la aplicación trato sobre recetas de cocina donde podemos agregar más recetas por medio de un formulario, todas las recetas agregadas se guardan en una API creada con php el cual tiene acceso al motor de la base de datos de MySQL, además de la aplicación se tuvo que arreglar por la situación actual con el covid, tuvimos poco para practicar en cuanto a programación me refiero, pero a pesar de eso se hizo lo posible para realizar el proyecto final, la solución a ese problema fue descargar máquinas virtuales. Fuera de la situación de "Covid" la materia es muy interesante aunque me hubiera gustado practicar como normalmente debió ser.