**PROYECTO**

**SITIO WEB RESTAURANTE CHIFA**

**RESTCHIFA**

****

**CICLO FORMATIVO DE GRADO SUPERIOR:**

**IFCS03 - Desarrollo de Aplicaciones Web**

**AUTOR:**

**ANTONIO HRISTOV LÓPEZ**

**TUTORA:**

**ASCENSION SANCHEZ MARTINEZ**

**Licencia**

**Esta obra está bajo una licencia Reconocimiento-Compartir bajo la misma licencia 3.0 España de Creative Commons. Para ver una copia de esta licencia, visite http://creativecommons.org/licenses/by-sa/3.0/es/ o envie una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.**

**RESUMEN**

La web, tanto en la evolución tecnológica como en la evolución social, representa un avance por su uso. En cierto modo, esto ha provocado la aparición de un nuevo “grupo” de individuos: los internautas.

Estos usuarios del Internet utilizan cada vez más la web para trabajar, viajar, ir de compras, vacaciones, etc. Unos cuantos ejemplos que muestran el alcance que ha tomado este fenómeno, tanto a efectos profesionales, como de ocio o de la vida cotidiana.

Por lo tanto, se conectan constantemente a los sitios web según sus necesidades o centros de interés para buscar, obtener información, consultar y elegir. Para ello necesitan diferentes sitios, informativos, comerciales u otros.

En cambio, si uno está interesado en el sector económico/comercial, rápidamente comprende que una agencia, un negocio, una tienda, un restaurante … físico ya no es suficiente y que es necesario una “tienda virtual”, para tener mayor visibilidad, aumentar y retener los clientes.

Este proyecto se encuentra en la encrucijada de estos dos aspectos, al tiempo que aborda el lado del placer y la cultura. Se trata de desarrollar una web de restaurantes, cuyos platos resulten de la fusión entre las cocinas de dos países y así despertar el interés de los internautas/consumidores.

El proyecto, aunque presenta lo esencial, no es exhaustivo y puede evolucionar. Para llevarlo a cabo se realizó un análisis de los requisitos requeridos, con el fin de obtener una expresión precisa de las necesidades. Este análisis permitió desarrollar las diferentes partes de manera más efectiva.

Luego, se realizaron pruebas con el fin de comparar los requisitos esperados con los resultados obtenidos y mejorar estos últimos.

**ABSTRACT**

Traducción al inglés del resumen realizado en el punto anterior.

**Índice de contenido**

[1 INTRODUCCIÓN 7](#_Toc137499461)

[2 ESQUEMAS Y GRÁFICOS 8](#_Toc137499462)

[2.1 PÁGINAS WEB (wireframe) 8](#_Toc137499463)

[3 PROGRAMACIÓN DEL PROYECTO 9](#_Toc137499464)

[3.1 ÁREA DE CONOCIMIENTO 9](#_Toc137499465)

[3.2 FRAMEWORKS 9](#_Toc137499466)

[3.3 ESTRUCTURA BASE DE DATOS 10](#_Toc137499467)

[3.4 EJEMPLO DE BASE DE DATOS 11](#_Toc137499468)

[3.5 BASE DE DATOS 12](#_Toc137499469)

[3.6 PÁGINAS WEB 20](#_Toc137499470)

[*3.6.1 Página Admin* 20](#_Toc137499471)

[*3.6.2 Página Inicio* 20](#_Toc137499472)

[*3.6.3 Página Platos* 21](#_Toc137499473)

[*3.6.4 Página Menus* 22](#_Toc137499474)

[*3.6.5 Página Reservas* 22](#_Toc137499475)

[*3.6.6 Página Contacto* 23](#_Toc137499476)

[*3.6.7 Página Próximas Reservas* 23](#_Toc137499477)

[*3.6.8 Página Plato Detallado* 24](#_Toc137499478)

[*3.6.9 Página Menú Detallado* 24](#_Toc137499479)

[*3.6.10 Página Próximas Reservas Detallada* 25](#_Toc137499480)

[3.7 VALIDACIÓN DE RESERVAS 25](#_Toc137499481)

[4 JUSTIFICACIÓN DE LA VIABILIDAD 27](#_Toc137499482)

[5 PREVENCIÓN DE RIESGOS 28](#_Toc137499483)

[6 FUENTES 28](#_Toc137499484)

[7 ANEXOS 30](#_Toc137499485)

[El origen de este estilo gastronómico 30](#_Toc137499486)

[7.1.1.1 ¿Chifa o Chaufa? 30](#_Toc137499487)

[*7.1.2 Pruebas* 32](#_Toc137499488)

[8 PLANIFICACIÓN DE LA EJECUCIÓN DEL PROYECTO 32](#_Toc137499489)

[8.1 Fase de Análisis 32](#_Toc137499490)

[8.2 Fase de diseño 32](#_Toc137499491)

[8.3 Fase de Implementación 32](#_Toc137499492)

[8.4 Fase de pruebas 32](#_Toc137499493)

[9 DEFINICIÓN DE PROCEDIMIENTOS DE CONTROL Y EVALUACIÓN 32](#_Toc137499494)

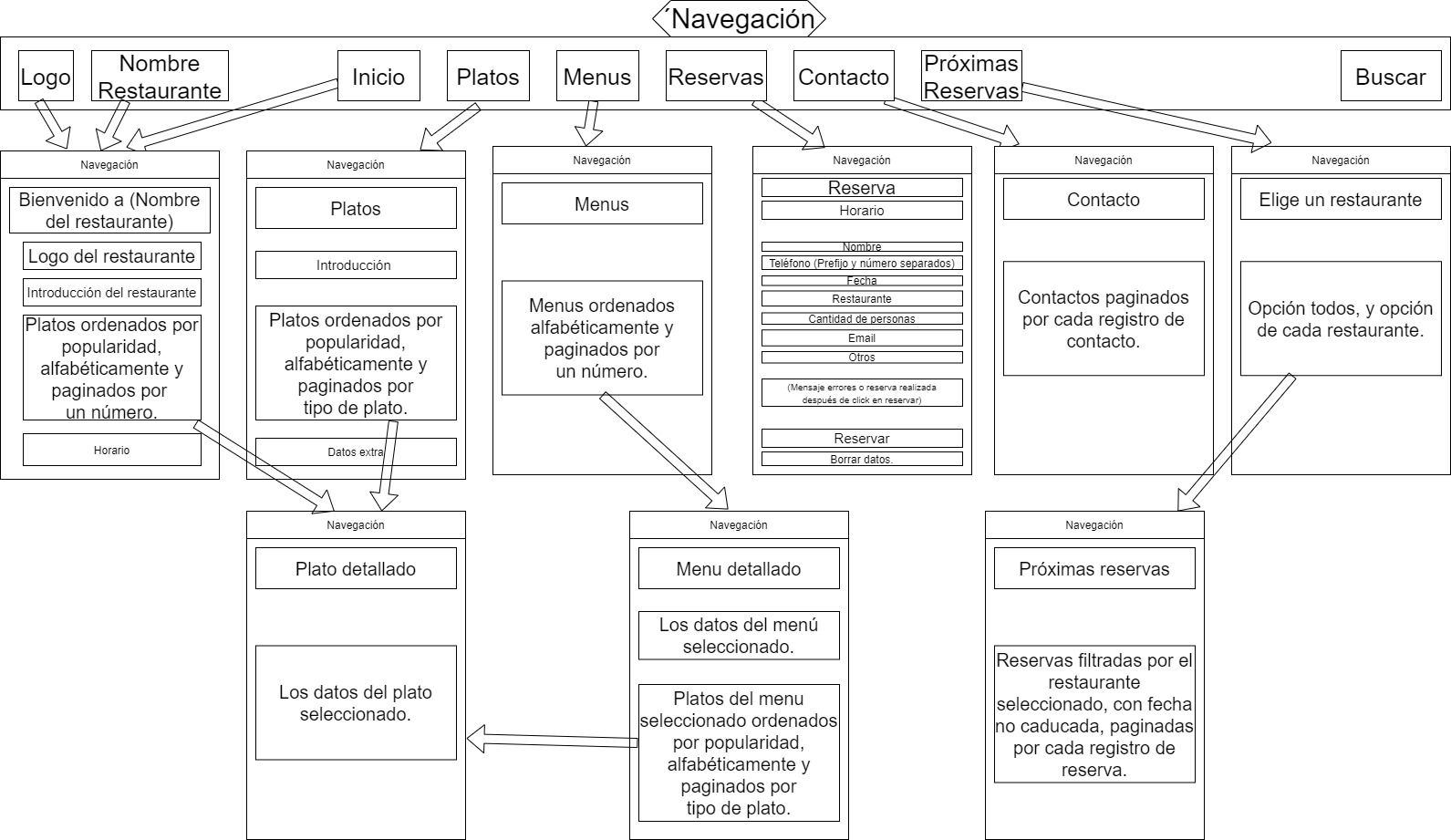
Índice de figuras

# INTRODUCCIÓN

* Propósito del proyecto.
* Enumerar las principales fases de desarrollo del mismo.

# ESQUEMAS Y GRÁFICOS

## PÁGINAS WEB (wireframe)



**Páginas web (wireframe)**

# PROGRAMACIÓN DEL PROYECTO

## ÁREA DE CONOCIMIENTO

Diseño Interfaces: Bootstrap, wireframe, css.

Servidor: Django.

Cliente: Html, javascript.

Empresa: Viabilidad del proyecto y prevención de riesgos.

Inglés: Nombres de variables y comentarios del código en inglés.

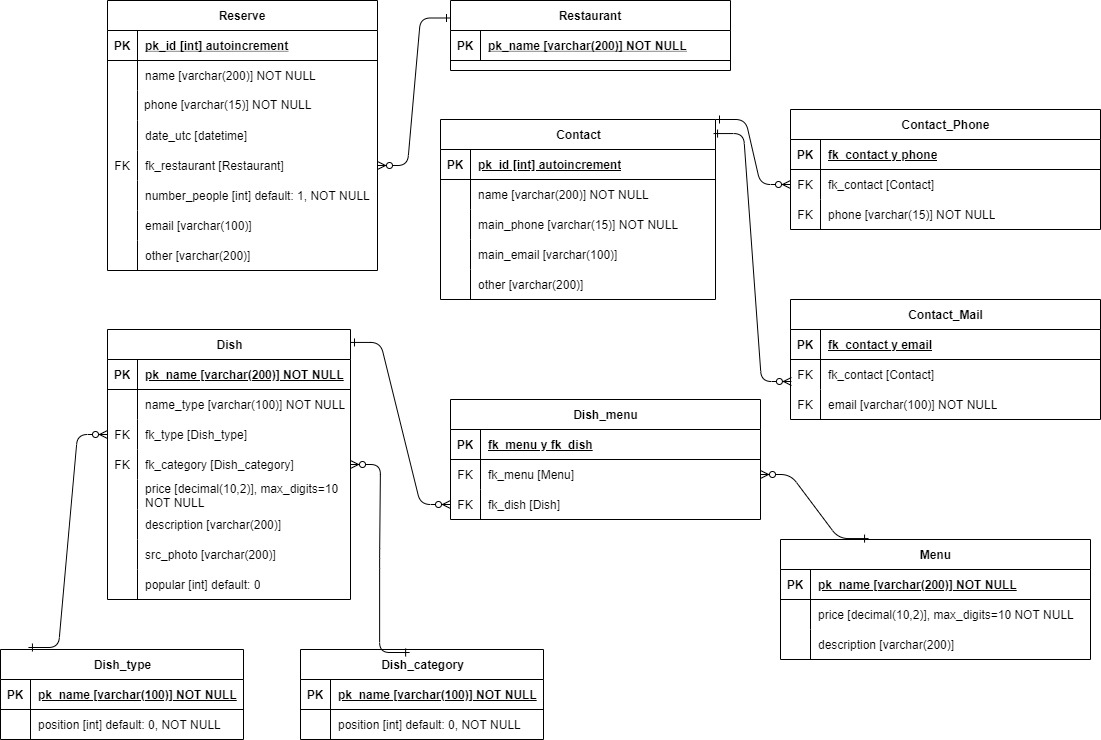
## FRAMEWORKS

Desarrollé el sitio web con Django (Servidor) y Bootstrap (Cliente) porque son los frameworks que mejor conozco y son fáciles de usar.

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo–vista–controlador (MVC) que en el diseño de software se utiliza comúnmente para implementar interfaces de usuario, datos y lógica de control.

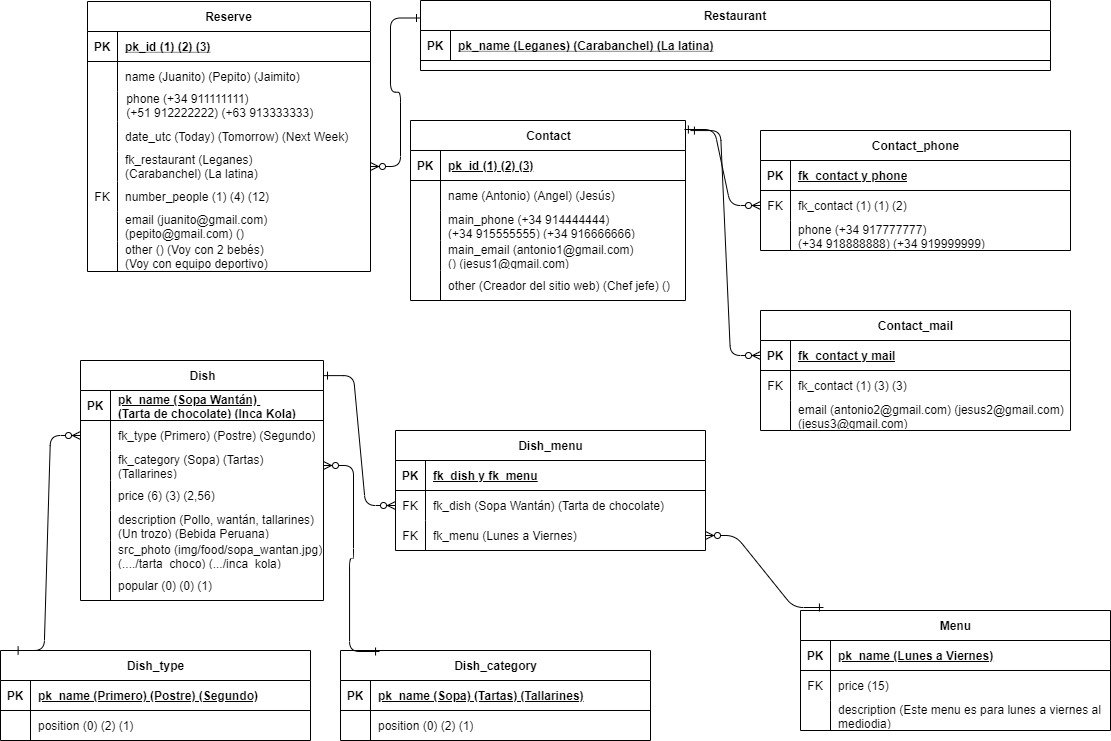
Bootstrap es una herramienta que permite crear una interfaz web, a través de lenguajes de programación como HTML, CSS o JavaScript, los cuales permiten adaptar el contenido de la web a los diferentes tipos de pantallas (diseño responsive) y mejorando, por tanto, la experiencia de usuario.

## ESTRUCTURA BASE DE DATOS



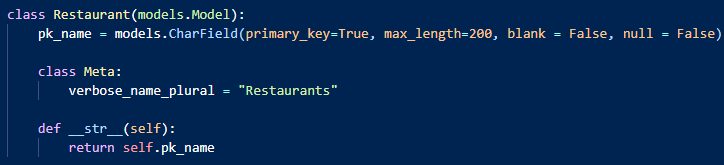
**Estructura Base de Datos**

## EJEMPLO DE BASE DE DATOS



**Ejemplo Base de Datos**

## BASE DE DATOS



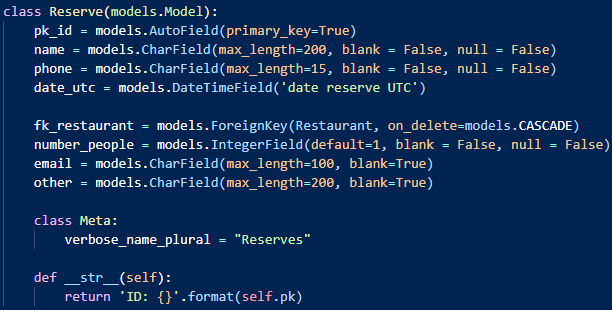
Se creó una entidad llamada “**Restaurant**” para guardar los restaurantes que el sitio web pueda tener porque el negocio podría crecer con la posibilidad de abrir más restaurantes sin la necesidad de un sitio web por cada restaurante, así que las reservas se filtran por restaurante.

Cada restaurante tiene:

**pk\_name**: Es el nombre del restaurante, de tipo texto que no se puede repetir, no puede estar vacío ni tener solo espacio o tabulaciones, el máximo número de caracteres es 200 pero puede variar al gusto del administrador.

El nombre en plural que aparece en la página de administración es “Restaurants” pero también puede variar al gusto del administrador.

Si no hay referencia a ningún campo por defecto devuelve el campo pk\_name.



Se creó una entidad llamada “**Reserve**” para guardar las reservas que el sitio web pueda tener.

Cada reserva tiene:

**pk\_id**: Es un número autoincremental que no se puede repetir.

**name**: Es el nombre de la reserva, de tipo texto, no puede estar vacío ni tener solo espacio o tabulaciones, el máximo número de caracteres es 200 pero puede variar al gusto del administrador.

**phone**: Es el número de teléfono del cliente, de tipo texto, no puede estar vacío ni tener solo espacio o tabulaciones, el máximo número de caracteres es 15 pero puede variar al gusto del administrador.

**date\_utc**: Es la fecha elegida para la reserva en formato UTC (Universal, no es local), de tipo fecha con hora y minuto, no debería estar vacío, el nombre del campo en la página de administración es ‘date reserve UTC’ pero puede variar al gusto del administrador.

**fk\_restaurant**: Es la instancia de un restaurante, de tipo entidad Restaurant, no debería estar vacío, si el identificador del restaurante instanciado se elimina se hace en cascada.

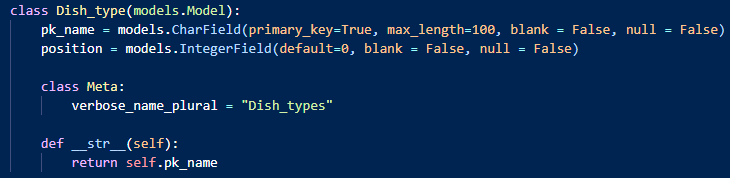
**number\_people**: Es el número de personas de la reserva, de tipo número, por defecto valdría 1 pero puede variar al gusto del administrador, no puede estar vacío ni tener solo espacio o tabulaciones.

**email**: Es el email del cliente, de tipo texto, es opcional así que puede estar vacío, pero el número máximo de caracteres es 100.

**other**: Son datos extra que el cliente quiera dar, por ejemplo: llevo un bebé, equipo deportivo, etc. De tipo texto, es opcional así que puede estar vacío, pero el número máximo de caracteres es 200.

El nombre en plural que aparece en la página de administración es “Reserves” pero también puede variar al gusto del administrador.

Si no hay referencia a ningún campo por defecto devuelve el texto ID: y el campo pk\_id.



Se creó una entidad llamada “**Dish\_type**” para guardar los tipos de platos que el sitio web pueda tener, ejemplo: Primero, segundo, postre, bebida.

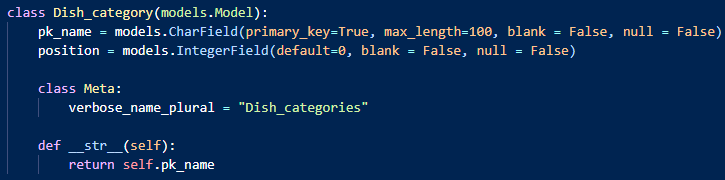
Cada tipo de plato tiene:

**pk\_name**: Es el nombre del tipo de plato, de tipo texto que no se puede repetir, no puede estar vacío ni tener solo espacio o tabulaciones, el máximo número de caracteres es 100 pero puede variar al gusto del administrador.

**position**: Es un número que sirve para ordenar, es decir, elegir el orden a la hora de mostrar los pk\_name, cuanto menor sea este número antes se muestra, de tipo número, por defecto valdría 0 pero puede variar al gusto del administrador, no puede estar vacío ni tener solo espacio o tabulaciones.

El nombre en plural que aparece en la página de administración es “Dish\_types” pero también puede variar al gusto del administrador.

Si no hay referencia a ningún campo por defecto devuelve el campo pk\_name.



Se creó una entidad llamada “**Dish\_category**” para guardar las categorías de platos que el sitio web pueda tener, ejemplo: Sopas, ensaladas. tallarines, pollos.

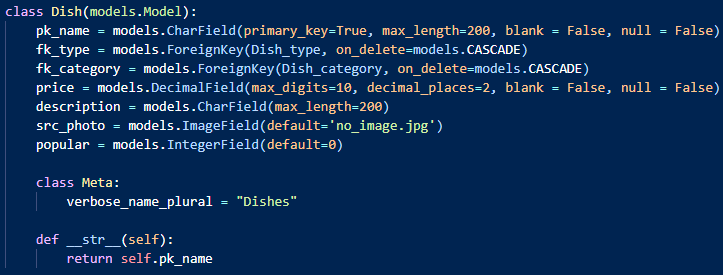
Cada categoría de plato tiene:

**pk\_name**: Es el nombre del tipo de plato, de tipo texto que no se puede repetir, no puede estar vacío ni tener solo espacio o tabulaciones, el máximo número de caracteres es 100 pero puede variar al gusto del administrador.

**position**: Es un número que sirve para ordenar, es decir, elegir el orden a la hora de mostrar los pk\_name, cuanto menor sea este número antes se muestra, de tipo número, por defecto valdría 0 pero puede variar al gusto del administrador, no puede estar vacío ni tener solo espacio o tabulaciones.

El nombre en plural que aparece en la página de administración es “Dish\_categories” pero también puede variar al gusto del administrador.

Si no hay referencia a ningún campo por defecto devuelve el campo pk\_name.



Se creó una entidad llamada “**Dish**” para guardar los platos que el sitio web pueda tener.

Cada plato tiene:

**pk\_name**: Es el nombre del plato, de tipo texto que no se puede repetir, no puede estar vacío ni tener solo espacio o tabulaciones, el máximo número de caracteres es 200 pero puede variar al gusto del administrador.

**fk\_type**: Es la instancia de un tipo de plato, de tipo entidad Dish\_type, no debería estar vacío, si el identificador del tipo de plato instanciado se elimina se hace en cascada.

**fk\_category**: Es la instancia de una categoría de plato, de tipo entidad Dish\_category, no debería estar vacío, si el identificador de la categoría de plato instanciada se elimina se hace en cascada.

**price**: Es el precio del plato, de tipo número con 2 decimales con un máximo de 10 dígitos, no puede estar vacío ni tener solo espacio o tabulaciones.

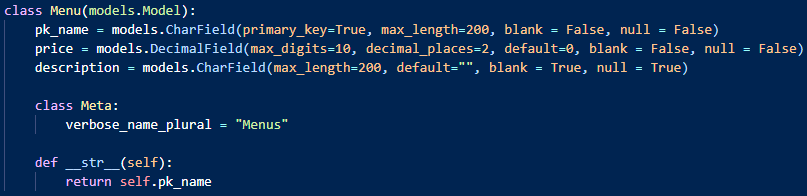
**description**: Es la descripción del plato en el que puedes añadir datos como los ingredientes que lleva. de tipo texto, el número máximo de caracteres es 200.

**src\_photo**: Es la imagen del plato. de tipo imagen (guarda el nombre y extensión de la imagen), si no hay una imagen o no detecta una recibe por defecto el valor “no\_image.jpg” que es una imagen elegida por defecto que puede variar al gusto del administrador.

**popular**: Es un número que sirve para almacenar las veces que se ha pedido cada plato y en base a eso ordenarlos por su popularidad, cuanto mayor sea este número antes se muestra, de tipo número, por defecto valdría 0 pero puede variar al gusto del administrador.

El nombre en plural que aparece en la página de administración es “Dishes” pero también puede variar al gusto del administrador.

Si no hay referencia a ningún campo por defecto devuelve el campo pk\_name.



Se creó una entidad llamada “**Menu**” para guardar los menús que el sitio web pueda tener.

Cada menú tiene:

**pk\_name**: Es el nombre del menú, de tipo texto que no se puede repetir, no puede estar vacío ni tener solo espacio o tabulaciones, el máximo número de caracteres es 200 pero puede variar al gusto del administrador.

**price**: Es el precio del menú, de tipo número con 2 decimales con un máximo de 10 dígitos, no puede estar vacío ni tener solo espacio o tabulaciones.

**description**: Es la descripción del menú en el que puedes añadir datos como los días y las horas del menú. de tipo texto, el número máximo de caracteres es 200.

El nombre en plural que aparece en la página de administración es “Menus” pero también puede variar al gusto del administrador.

Si no hay referencia a ningún campo por defecto devuelve el campo pk\_name.



Se creó una entidad llamada “**Menu\_Dish**” para guardar y asignar los platos que tiene cada menú en el sitio web.

Cada menú plato tiene:

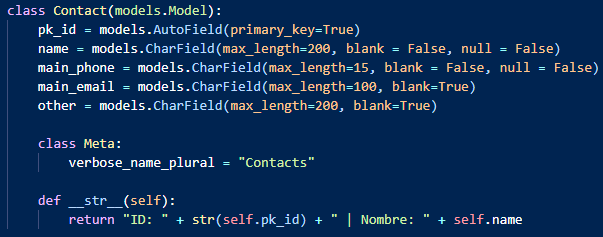
**fk\_menu**: Es la instancia de un menú, de tipo entidad Menu, no debería estar vacío, si el identificador del menú instanciado se elimina se hace en cascada.

**fk\_dish**: Es la instancia de un plato, de tipo entidad Dish, no debería estar vacío, si el identificador del plato instanciado se elimina se hace en cascada.

**unique\_fk\_menu\_fk\_dish\_combination**: Es el identificador generado por las instancias fk\_menu y fk\_dish, que no se debería repetir.

El nombre en plural que aparece en la página de administración es “Menu\_Dishes” pero también puede variar al gusto del administrador.

Si no hay referencia a ningún campo por defecto devuelve el texto Menu: (El nombre del menú) | Plato: (El nombre del plato)



Se creó una entidad llamada “**Contact**” para guardar los datos de contacto del personal del sitio web.

Cada contacto tiene:

**pk\_id**: Es un número autoincremental que no se puede repetir.

**name**: Es el nombre de la persona de contacto, de tipo texto, no puede estar vacío ni tener solo espacio o tabulaciones, el máximo número de caracteres es 200 pero puede variar al gusto del administrador.

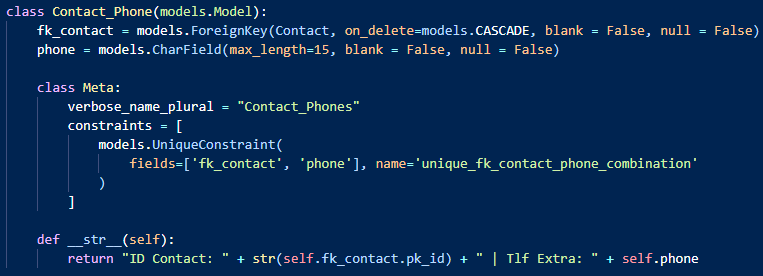
**phone**: Es el principal número de teléfono de la persona de contacto, de tipo texto, no puede estar vacío ni tener solo espacio o tabulaciones, el máximo número de caracteres es 15 pero puede variar al gusto del administrador.

**email**: Es el principal email de la persona de contacto, de tipo texto, es opcional así que puede estar vacío, pero el número máximo de caracteres es 100.

**other**: Son datos extra que la persona de contacto quiera dar, por ejemplo: disponibilidad, cargo, etc. De tipo texto, es opcional así que puede estar vacío, pero el número máximo de caracteres es 200.

El nombre en plural que aparece en la página de administración es “Contacts” pero también puede variar al gusto del administrador.

Si no hay referencia a ningún campo por defecto devuelve el texto ID: (El pk\_id de la persona de contacto) | Nombre: (El nombre de la persona de contacto)



Se creó una entidad llamada “**Contact\_Phone**” para guardar los números de teléfono adicionales de una persona de contacto del sitio web.

Cada teléfono contacto tiene:

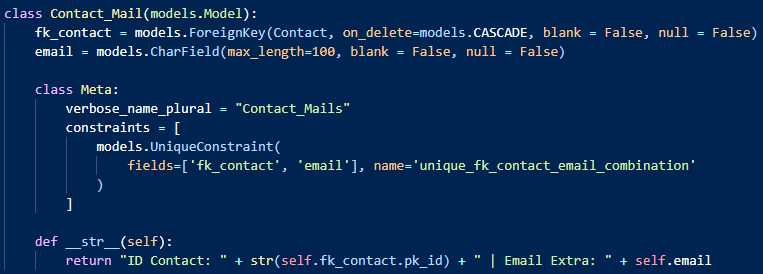
**fk\_contact**: Es la instancia de un contacto, de tipo entidad Contact, no debería estar vacío, si el identificador del contacto instanciado se elimina se hace en cascada.

**phone**: Es un número de teléfono adicional de la persona de contacto, de tipo texto, no puede estar vacío ni tener solo espacio o tabulaciones, el máximo número de caracteres es 15 pero puede variar al gusto del administrador.

**unique\_fk\_contact\_phone\_combination**: Es el identificador generado por los campos fk\_contact y phone, que no se debería repetir.

El nombre en plural que aparece en la página de administración es “Contact\_Phones” pero también puede variar al gusto del administrador.

Si no hay referencia a ningún campo por defecto devuelve el texto ID Contact: (El id del fk\_contact) | Tlf Extra: (El campo phone)



Se creó una entidad llamada “**Contact\_Mail**” para guardar los correos adicionales de una persona de contacto del sitio web.

Cada correo contacto tiene:

**fk\_contact**: Es la instancia de un contacto, de tipo entidad Contact, no debería estar vacío, si el identificador del contacto instanciado se elimina se hace en cascada.

**email**: Es un correo adicional de la persona de contacto, de tipo texto, no puede estar vacío ni tener solo espacio o tabulaciones, el máximo número de caracteres es 100 pero puede variar al gusto del administrador.

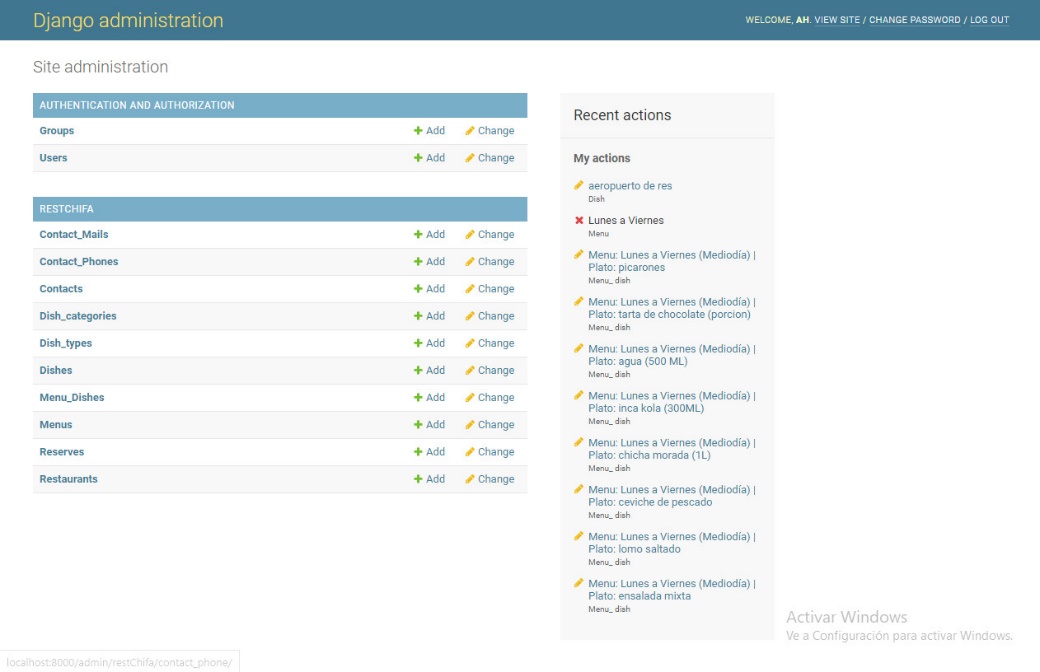
**unique\_fk\_contact\_email\_combination**: Es el identificador generado por los campos fk\_contact y email, que no se debería repetir.

El nombre en plural que aparece en la página de administración es “Contact\_Mails” pero también puede variar al gusto del administrador.

Si no hay referencia a ningún campo por defecto devuelve el texto ID Contact: (El id del fk\_contact) | Email Extra: (El campo email).

## PÁGINAS WEB

### *Página Admin*



Esta página es solo para administradores del sitio web, tiene un sistema de inicio de sesión en el que te pide usuario y contraseña.

Al iniciar sesión tienes acceso a la página Próximas reservas (lo veremos después) y a los datos de la base de datos, con la posibilidad de buscar datos por pk (clave única), crear datos, modificar datos existentes (ejemplo: Platos, reservas, etc) y por cada dato (ejemplo: un plato) poder ver el histórico de operaciones realizadas (quien ha creado, modificado ese dato y a que hora)

### *Página Inicio*



Arriba tiene un menú de navegación con el logo del restaurante, el nombre del restaurante, botones para ir a las páginas web del sitio (Inicio, Platos, Menus, Reservas, Contacto), si el usuario inició sesión en la página web de administrador aparece también la opción de ir a la página “Próximas Reservas” (Solo los administradores pueden entrar), también hay un buscador para filtrar resultados.

Tiene un título de bienvenida, el logo en grande, una introducción del restaurante, los platos ordenados por popularidad, alfabéticamente y paginados por un número de platos por cada página (ese número lo elige el administrador, por defecto es 10), al final está el horario del restaurante.

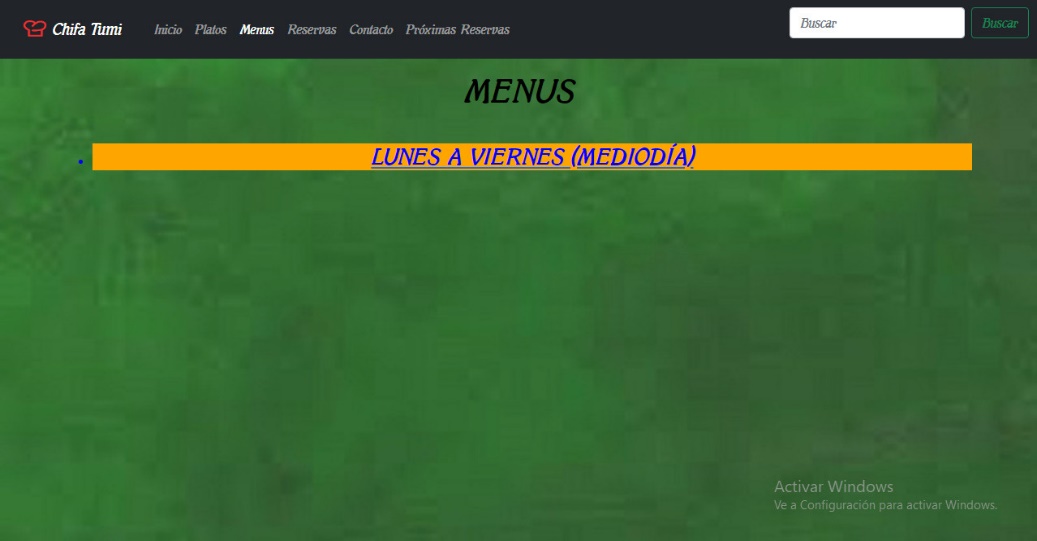
### *Página Platos*



Arriba tiene un menú de navegación con el logo del restaurante, el nombre del restaurante, botones para ir a las páginas web del sitio (Inicio, Platos, Menus, Reservas, Contacto), si el usuario inició sesión en la página web de administrador aparece también la opción de ir a la página “Próximas Reservas” (Solo los administradores pueden entrar), también hay un buscador para filtrar resultados.

Tiene un título, una introducción, y los platos ordenados por popularidad, alfabéticamente y paginados por tipos de platos (primero, segundo, postre, etc), al final hay información extra.

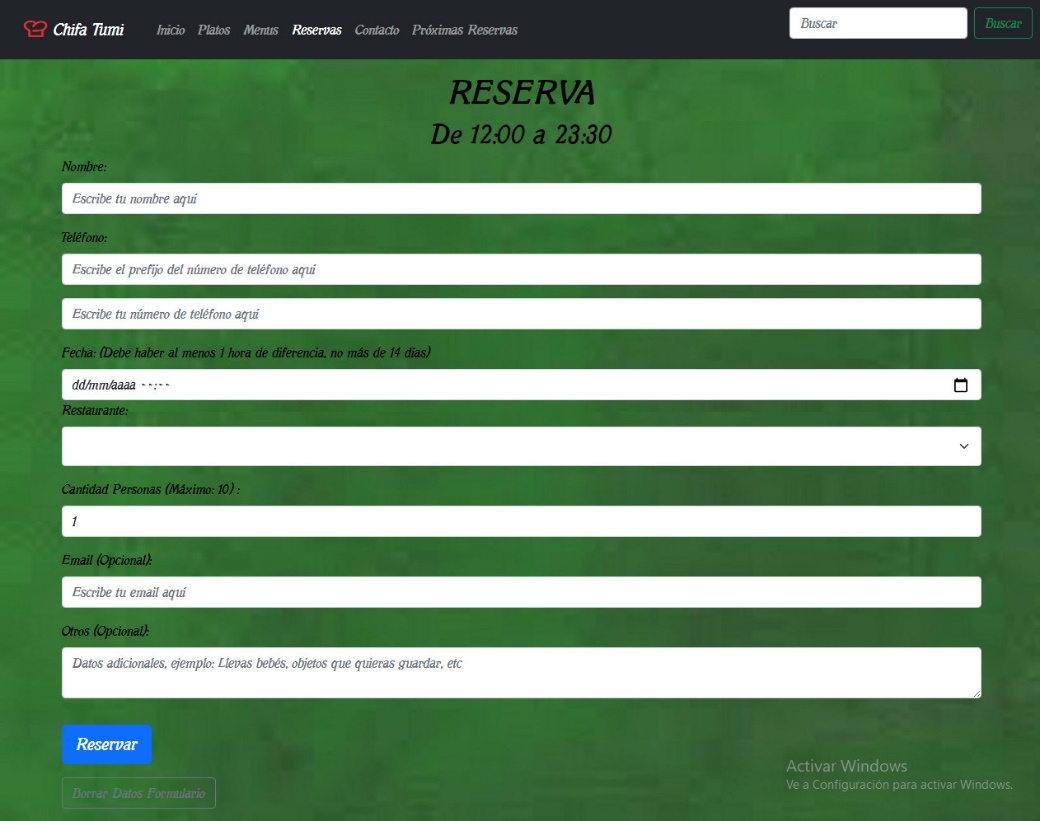
### *Página Menus*



Arriba tiene un menú de navegación con el logo del restaurante, el nombre del restaurante, botones para ir a las páginas web del sitio (Inicio, Platos, Menus, Reservas, Contacto), si el usuario inició sesión en la página web de administrador aparece también la opción de ir a la página “Próximas Reservas” (Solo los administradores pueden entrar), también hay un buscador para filtrar resultados.

Tiene un título, una introducción, y los menús paginados por un número de menús por cada página (ese número lo elige el administrador, por defecto es 10).

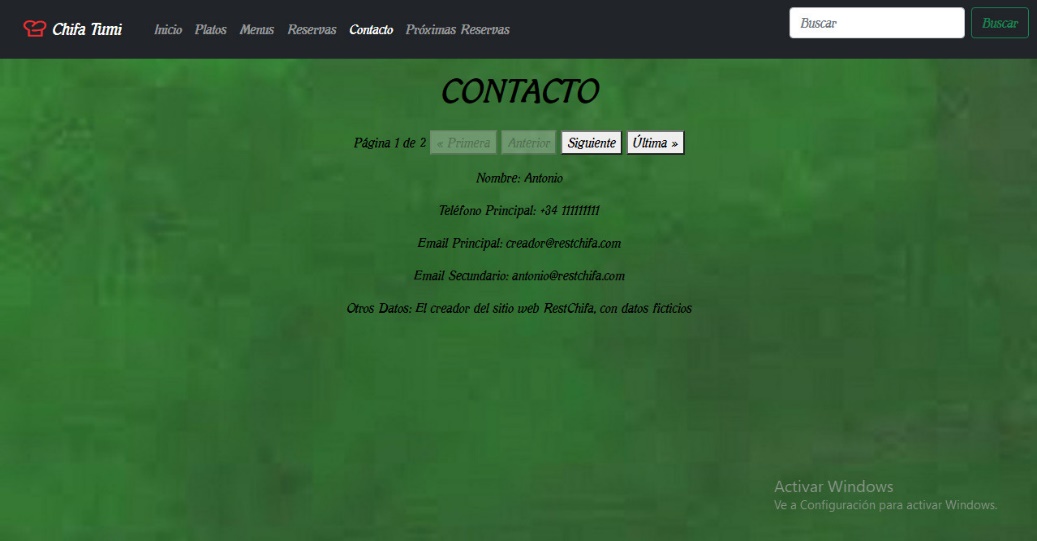
### *Página Reservas*



Arriba tiene un menú de navegación con el logo del restaurante, el nombre del restaurante, botones para ir a las páginas web del sitio (Inicio, Platos, Menus, Reservas, Contacto), si el usuario inició sesión en la página web de administrador aparece también la opción de ir a la página “Próximas Reservas” (Solo los administradores pueden entrar), también hay un buscador para filtrar resultados.

Tiene un título, el horario de reserva, un campo para el nombre, un campo para el prefijo del teléfono, un campo para el número de teléfono, un campo para la fecha, un campo para el restaurante, un campo para cantidad de personas, un campo email, un campo para otros datos, un botón reservar y un botón para vaciar los datos.

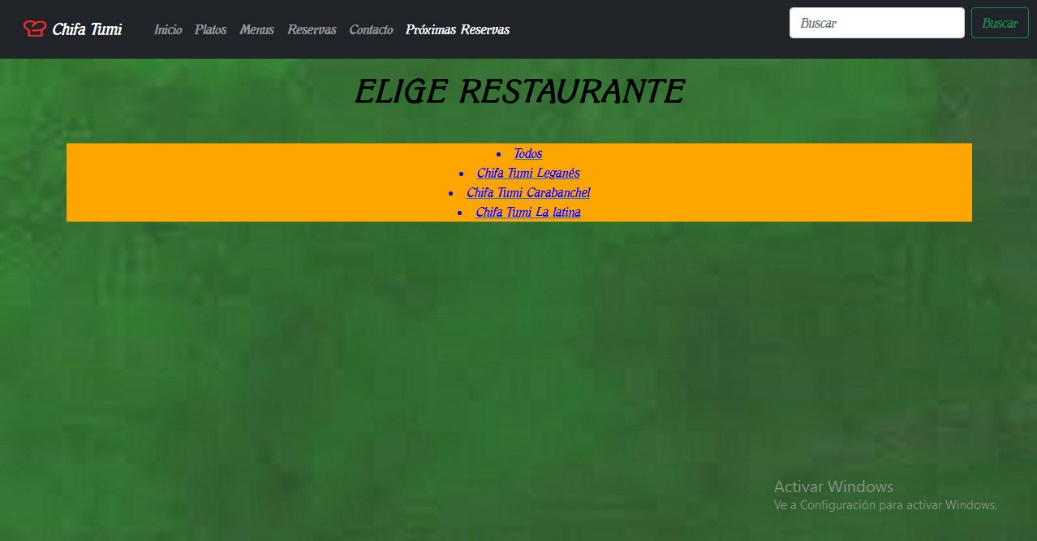
### *Página Contacto*



Arriba tiene un menú de navegación con el logo del restaurante, el nombre del restaurante, botones para ir a las páginas web del sitio (Inicio, Platos, Menus, Reservas, Contacto), si el usuario inició sesión en la página web de administrador aparece también la opción de ir a la página “Próximas Reservas” (Solo los administradores pueden entrar), también hay un buscador para filtrar resultados.

Tiene un título y los contactos paginados por cada registro de contacto.

### *Página Próximas Reservas*



Arriba tiene un menú de navegación con el logo del restaurante, el nombre del restaurante, botones para ir a las páginas web del sitio (Inicio, Platos, Menus, Reservas, Contacto), si el usuario inició sesión en la página web de administrador aparece también la opción de ir a la página “Próximas Reservas” (Solo los administradores pueden entrar), también hay un buscador para filtrar resultados.

Solo pueden entrar los administradores, tiene un título, te pide elegir un restaurante, tiene la opción “todos” y los restaurantes paginados por un número de restaurantes por cada página (ese número lo elige el administrador, por defecto es 10).

### *Página Plato Detallado*



Arriba tiene un menú de navegación con el logo del restaurante, el nombre del restaurante, botones para ir a las páginas web del sitio (Inicio, Platos, Menus, Reservas, Contacto), si el usuario inició sesión en la página web de administrador aparece también la opción de ir a la página “Próximas Reservas” (Solo los administradores pueden entrar), también hay un buscador para filtrar resultados.

Tiene un título y los datos del plato elegido.

Si se entra por la página menú detallado, no se muestra el precio.

### *Página Menú Detallado*



Arriba tiene un menú de navegación con el logo del restaurante, el nombre del restaurante, botones para ir a las páginas web del sitio (Inicio, Platos, Menus, Reservas, Contacto), si el usuario inició sesión en la página web de administrador aparece también la opción de ir a la página “Próximas Reservas” (Solo los administradores pueden entrar), también hay un buscador para filtrar resultados.

Tiene un título, los datos del menú y los platos ordenados por popularidad, alfabéticamente y paginados por tipos de platos (primero, segundo, postre, etc).

### *Página Próximas Reservas Detallada*



Arriba tiene un menú de navegación con el logo del restaurante, el nombre del restaurante, botones para ir a las páginas web del sitio (Inicio, Platos, Menus, Reservas, Contacto), si el usuario inició sesión en la página web de administrador aparece también la opción de ir a la página “Próximas Reservas” (Solo los administradores pueden entrar), también hay un buscador para filtrar resultados.

Solo pueden entrar los administradores, tiene un título y las reservas (con fecha no caducada) del restaurante elegido, paginados por cada registro de reserva.

## VALIDACIÓN DE RESERVAS

Para hacer una reserva se le pide al usuario:

**Nombre**: Es el nombre de la reserva, lo puede elegir el cliente, se valida que no esté vacío ni tenga más de 200 caracteres al enviar la reserva, si falla la validación no se crea la reserva y da un mensaje de error.

**Prefijo teléfono**: Es el prefijo del número de teléfono del cliente, de tipo texto, se valida que de forma opcional se pueda añadir un “+” al inicio, sigue 1 o 2 números entre el 1 y 9 incluídos, sigue de 0 a 5 veces un número del 0 al 9 incluídos o el carácter “-“ al enviar la reserva, si falla la validación no se crea la reserva y da un mensaje de error.

**Teléfono**: Es el número de teléfono del cliente, de tipo número, se valida que no esté vacío, sea un número (por si el usuario cambia el código del cliente), no tenga menos de 9 cifras ni tenga más de 14 cifras al enviar la reserva, si falla la validación no se crea la reserva y da un mensaje de error.

**Fecha**: Es la fecha elegida para la reserva en formato UTC (Universal, no es local), de tipo fecha con hora y minuto, se valida que sea formato fecha (por si el usuario cambia el código del cliente), no esté vacío y respete los rangos de días, horas y minutos elegidos por el administrador al enviar la reserva, si falla la validación no se crea la reserva y da un mensaje de error.

**Restaurante**: Es el restaurante al que se va a hacer la reserva, de tipo select option (un desplegable), se valida que no esté vacío y sea un valor válido de los que aparecen en el desplegable (por si el usuario cambia el código del cliente) al enviar la reserva, si falla la validación no se crea la reserva y da un mensaje de error.

**Cantidad Personas**: Es el número de personas de la reserva, de tipo número, se valida que sea un número entero (por si el usuario cambia el código del cliente) y no sea un número menor a 1 y mayor al número que elija el administrador al enviar la reserva, si falla la validación no se crea la reserva y da un mensaje de error.

**Email**: Es el email del cliente, de tipo email (texto), se valida en cliente que respete el formato “texto@texto.texto” y en servidor que no tenga más de 100 caracteres al enviar la reserva, si falla la validación no se crea la reserva y da un mensaje de error.

**Otros**: Son datos extra que el cliente quiera dar, por ejemplo: llevo un bebé, equipo deportivo, etc. De tipo texto, se valida que no tenga más de 200 caracteres al enviar la reserva, si falla la validación no se crea la reserva y da un mensaje de error.

**Botón Reservar**: Al pulsarlo pide confirmación (Por si se pulsó por accidente o no estás seguro de los datos) y si se acepta se envían los datos de la reserva, se hacen las validaciones y si alguna validación falla no crea la reserva y muestra el/los mensajes de error.

Al enviar los datos si no hay ningún error de validación se crea la reserva en la base de datos y se muestra un número identificador único (el pk\_id de la base de datos) al usuario y se le pide que lo guarde para poder identificar la reserva, editar los datos que no estén bien (lo edita un administrador), etc.

**Botón Borrar Datos Formulario**: Al pulsarlo pide confirmación (Por si se pulsó por accidente o quieres vaciar los campos del formulario) y si se acepta se vacían los campos del formulario como estaban por defecto.

Se vacía también con javascript porque si se intentó enviar los datos anteriormente, por defecto no se vacían los datos con la funcionalidad estándar del botón tipo reset.

# JUSTIFICACIÓN DE LA VIABILIDAD

# PREVENCIÓN DE RIESGOS

# FUENTES

<https://docs.djangoproject.com/en/4.1/howto/windows/>

<https://docs.djangoproject.com/en/4.1/intro/tutorial01/>

https://docs.djangoproject.com/en/4.2/topics/pagination/

https://ourcodeworld.com/articles/read/236/how-to-setup-your-first-django-project-and-say-hello-world-in-windows

https://projectsplaza.com/how-to-save-form-data-in-database-in-django/

https://stackoverflow.com/questions/24710233/python-convert-time-to-utc-format

https://www.1001freefonts.com/

# ANEXOS

### El origen de este estilo gastronómico

Sin querer adentrarnos en la historia, si es necesario situarse a finales del siglo XIX, cuando tras la independencia, se atrae a población china para trabajos duros. Esta población china se va integrando entre la población peruana, que es una mezcla de criollos, mestizos e indígenas, pero no así su estilo de cocinar.

Cuando estos chinos, logran una cierta independencia económica**,** empiezan a regentar casas de comida.En la mayor parte, importan desde China los ingredientes originales para sus creaciones, aunque al cabo de un tiempo empiezan a cultivar algunos vegetales chinos en Perú, o bien se adaptan a los ingredientes locales**, l**os platos de origen chino.

[](https://kachkaniraqmii.blogspot.com/2018/01/chifa-peruanismo-culinario.html)

Esta permeabilidad de ingredientes hace que se separe de la cocina cantonesa, adquiriendo una personalidad propia. De la forma de llamar la atención a los viandantes sobre los platos que ofrecían estas casas de comidas, usando los términos cantoneses «chi» y «fan» (饎飯), que significan, respectivamente «comer» y «arroz», se fue generando el término ***Chifa***.

#### ¿Chifa o Chaufa?

Estos dos términos están muy relacionados, aunque son distintos en cuanto a sus matices.

La traducción de «Chi fan» sería «comer arroz», y que luego se transformó en **Chifa**; mientras que «Chau fan» sería «arroz frito», y que los peruanos han adaptado como **Chaufa**. Por tanto, a la gastronomía en general fusión chino-peruana, se la conoce como chifa, mientras que algunos platos son chaufa.

### *Pruebas*

Son muchas pruebas que pueden realizarse en un proyecto, para eliminar los posibles errores y garantizar su correcto funcionamiento. Los casos de prueba establecen las condiciones/variables que permitirán determinar si los requisitos establecidos se cumplen o no. A continuación se detallan algunos de los casos de prueba que se ejecutarán para comprobar la correcta construcción de este proyecto.

# PLANIFICACIÓN DE LA EJECUCIÓN DEL PROYECTO

A continuación se detallan las actividades/tareas/procedimientos por cada una de las fases del proyecto previamente establecidas.

[Estas tareas podrán variar en función del proyecto]

## Fase de Análisis

1. Estudio de las necesidades a cubrir
2. Estudio de la situación actual
3. Establecimiento de los requisitos del proyecto
4. Valoración comparativa de las posibles soluciones
5. Identificación de las necesidades que implica el nuevo proyecto en la empresa.

Para solventar los problemas que plantea el proyecto puede ser necesario contratar personal, formarlo en determinadas metodologías/herramientas, comprar equipos...

1. Estudio de viabilidad de la solución elegida teniendo en cuenta no solo los beneficios económicos.
2. Corrección de posibles errores

## Fase de diseño

1. Preparación del entorno de diseño
2. Diseño de la arquitectura
3. Diseño de los interfaces
4. Diseño de los datos
5. Diseño de los procedimientos
6. Corrección de posibles errores

## Fase de Implementación

1. Preparación del entorno de implementación
2. Desarrollo de la arquitectura
3. Desarrollo de los interfaces, los datos y los procedimientos
4. Corrección de posibles errores

## Fase de pruebas

1. Preparación del entorno de pruebas
2. Ejecución de las pruebas y reporte de los errores encontrados

# DEFINICIÓN DE PROCEDIMIENTOS DE CONTROL Y EVALUACIÓN

* Definir el procedimiento de evaluación de las **incidencias** que puedan presentarse durante la realización de las diferentes actividades.
* Definir el procedimiento para gestionar los posibles **cambios** en los recursos y en las actividades.