



# Memoria del Proyecto

## “Fast-Food”



**Autoría:** Ezequiel Matías, Maggio Ares

**Tutor:** Pedro Prieto

**Curso:** 23/24

**Fecha:** Junio 23/24



<b>1 Introducción</b>	3
1.1 Breve descripción del proyecto	3
1.2 Exposición de los objetivos que se persiguen	3
<b>2 Análisis</b>	4
2.1 Identificación de necesidades del sector productivo	4
2.2 Análisis de la competencia, tanto a nivel local como en Internet	4
2.3 Actividad de la empresa y su ubicación en el sector de desarrollo software	4/5
2.4 Viabilidad técnica y oportunidad del proyecto	6
2.5 Características del proyecto: pliego de condiciones	7
2.6 Tecnologías a emplear	7
<b>3 Diseño</b>	8
3.1 Estructura general del proyecto. Diagrama de casos de uso	8/10
3.2 Elaboración de un guion de trabajo para el desarrollo del proyecto	10
3.3 Fases del proyecto. Tareas y plazos de ejecución	11
3.3.1 Metodología de trabajo	11
3.3.2 Fases de decremento	12
3.3.3 Tareas y Plazos de ejecución	13
3.4 Recursos materiales y personales	14
3.5 Estimación de gastos	15
3.6 Estimación de ingresos y precios de venta	16
3.7 Viabilidad económica	17
3.8 Necesidades de financiación	18
3.9 Definición y elaboración de la documentación del diseño Control de calidad	18/21
3.10 Revisión de la normativa aplicable	21/22
<b>4 Implementación</b>	22
4.1 Desarrollo del modelo de datos	22/26
4.2 Desarrollo de la programación del entorno de servidor	26/28
4.3 Desarrollo de la programación del entorno de cliente	28/32
<b>5 Validación</b>	32



5.1 Definición del procedimiento de evaluación, seguimiento y control del proyecto. Indicadores de calidad .....	32
5.2 Definición de procedimientos para la participación de los usuarios en la evaluación del proyecto. Documentos específicos .....	32/35
5.3 Registro de resultados .....	35
<b>6 Conclusiones</b> .....	35
6.1 Opinión personal .....	37
6.2 Futuras Mejoras .....	37

## 7 Bibliografía y referencias

# 1.Introducción

## 1.1 – Breve descripción del proyecto

Las personas que trabajan en oficina, generalmente el tiempo de descanso es de 1 hora, o 1 hora y media. Algunos usan ese tiempo para volver a su casa, otros comen en la oficina, o reservan en un restaurante para ir a comer. El proyecto se enfoca en esas personas que no vuelven a sus casas, por motivos de distancia del trabajo a su hogar.

Un cliente cuando llega al Restaurante el tiempo de perdida entre elegir su menú, la preparación de su menú entre una comida y otra puede ser mínimo unos 40 minutos, o 1 hora en el Restaurante.

“¿Cuál es mi idea de proyecto?” Reducir ese tiempo que un cliente pasa en el Restaurante

“¿Cómo funcionaria?” El usuario desde la aplicación podrá elegir el restaurante más cercano activando su ubicación, una vez seleccionado el lugar podrá armar su menú, y además elegir si lo pasa a buscar, o come en el lugar. Remarcando también la hora.

“¿Cuál sería el objetivo?” Cuando el cliente llegue al restaurante a la hora seleccionada, el restaurante tendrá el pedido terminado. Así en vez de estar 40 min o 1 hora, ese tiempo se reduciría a 20 min, ya que el restaurante sabe que va a comer.

## 1.2 – Exposición de los Objetivos que se persiguen



- ⇒ Acortar el tiempo de espera de un cliente en un restaurante, para aprovechar el tiempo de descanso
- ⇒ Darle también la oportunidad al restaurante, de saber los pedidos que le pueden entrar, para tener un orden. No es lo mismo que te lleguen 40 personas, que no vas a saber que van a comer, a que te lleguen esas 40, y tengas ya todos los pedidos.

## 2. Análisis

### 2.1 – Identificación de Necesidades del Sector Productivo

Poder sentarte en un lugar, en donde la comida se sirva rápido estando solo, o con compañeros cuando el tiempo de descanso es poco. Es una necesidad que en este sector puede venir muy bien. Aprovechando también que en España las a personas le gusta salir a comer fuera, y también que es uno de los rubros más importante en España.

Destacando también, que hoy en día la gente va más acelerada, y no le gusta esperar.

Tras esta situación, decidí llevar a cabo este proyecto en la cual se disminuirá el tiempo de espera, y un mejor orden en los restaurantes.

### 2.2 – Análisis de la competencia, tanto a Nivel local como en internet

Hoy en día todos los restaurantes tienen su propia página web, pero la idea de este servicio es no competir con eso, si no ayudar a que, a través de esta aplicación, el cliente vea el menú de su lugar preferido, y tenga la opción de comer en el lugar o llevar. El coste de este servicio sería igual para todos, y ayudaría a pequeños bares, o restaurantes a poder mostrar lo que venden. Este servicio sería cómodo tanto para el cliente y el dueño del lugar

A nivel de Internet, hay mucha competencia que publicitan lugares de comida para llevar como el caso de Uber o Glovo. Pero este servicio sería totalmente diferente ya que, al pedir a esos lugares, la persona tiene que esperar a que se lo lleven a la dirección que desee y además el tiempo de espera según la hora que lo pidas, habrá lugares que tarden 20min/30 min o incluso más.



Teniendo en cuenta lo comentado anteriormente nuestra aplicación se diferencia, ofreciendo un servicio que cuando se llegue al lugar, el tiempo de espera sea mucho menor

## 2.3 Actividad de la empresa y su ubicación en el sector de desarrollo software

### El nombre:

El nombre de la empresa será “Fast-Food”, es un nombre que va a representar la idea de la aplicación, y el servicio que va a ofrecer

### El logotipo:

El logotipo es una imagen simple, que se ve que es un reloj que va acelerado, que las agujas son un tenedor y cuchillo. Y abajo el nombre.



En el logotipo el color principal es el naranja que representa entusiasmo, confianza, éxito, generosidad y creatividad. Es un color más cálido, y no llega a ser tan agresivo como el color rojo.



### Forma jurídica:

Con respecto a la forma jurídica, optaría por Autónomo Individual que creo que sería la elección más segura para una empresa que recién se inicia, además sabiendo que empezaría por una sola persona. Es la forma que en gran parte requiere menos gestiones y tramites, lo que permite ahorrarte los costes de asesoramiento.

### La ubicación:

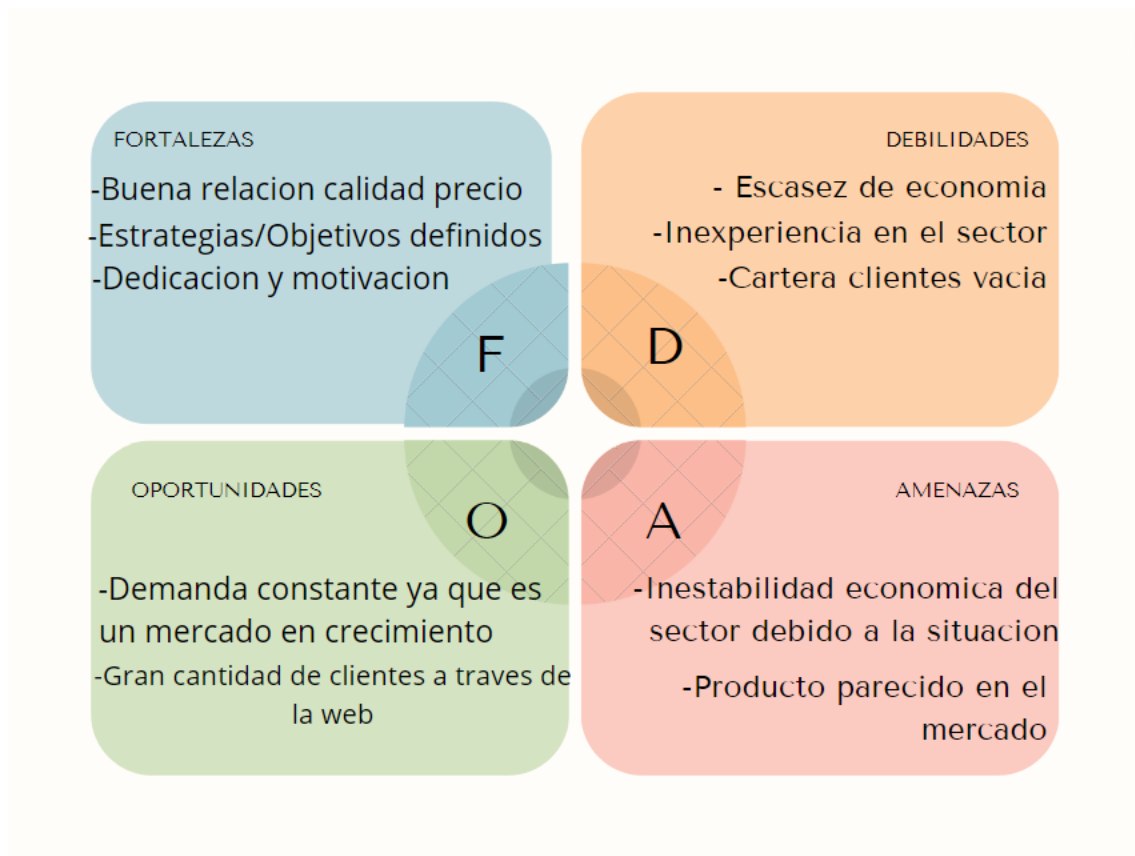
La ubicación de la empresa se establecerá en unas oficinas de Coworking, ya que permite a profesionales independientes, o empresas de nuevo inicio a compartir un espacio de trabajo, tanto virtual como físico.

Una de las ventajas más importantes que veo es tener un lugar de trabajo completo y exclusivo, que cubre todas las necesidades, además de los gastos que te ahorras, al comenzar con este proyecto

## 2.4 – Viabilidad técnica y oportunidad del proyecto

Este proyecto es viable a simple vista, ya que no se necesita una gran inversión. Según avanza el proyecto se podría optar por un software libre, para que el impacto de los gastos se reduzca.

### Análisis DAFO:



Mirando el análisis DAFO, mirando las Fortalezas, Amenazas, Debilidades, y Oportunidades, se puede deducir que la idea de negocio es buena, ya que el servicio es diferenciado de la competencia

## 2.5 – Características de Proyecto: Pliego de Condiciones

La estructura de esta aplicación va a tener tres perfiles de usuarios, diferenciados en cada caso:

Usuario sin registrar:

- Listado de restaurantes disponibles
- Datos contacto
- Ubicaciones de los restaurantes

Usuario registrado:

- Listado de restaurantes disponibles
- Datos contacto
- Ubicaciones de los restaurantes
- Elegir los platos del restaurante seleccionado



- Pagar con tarjeta
- Dejar su comentario sobre su experiencia
- Marcar como favorito

#### Usuario Administrador

- Gestión de usuario
- Gestión de los restaurantes

#### Usuario Empresa

- Gestionará la información de su carta (precios, platos, menús, etc.

## 2.6 – Tecnologías a Emplear

1. ASP .NET
2. HTML
3. CSS
4. Bootstrap
5. SQL Server

El cliente recibirá una página hecha con HTML Y CSS, mostrando los datos que se guardaran en una base de datos, que se mostrará con ASP .NET y esos datos se podrán actualizar, eliminar, crear nuevos clientes.

## 3. – Diseño

### 3.1 – Estructura general del proyecto. Diagrama casos de uso

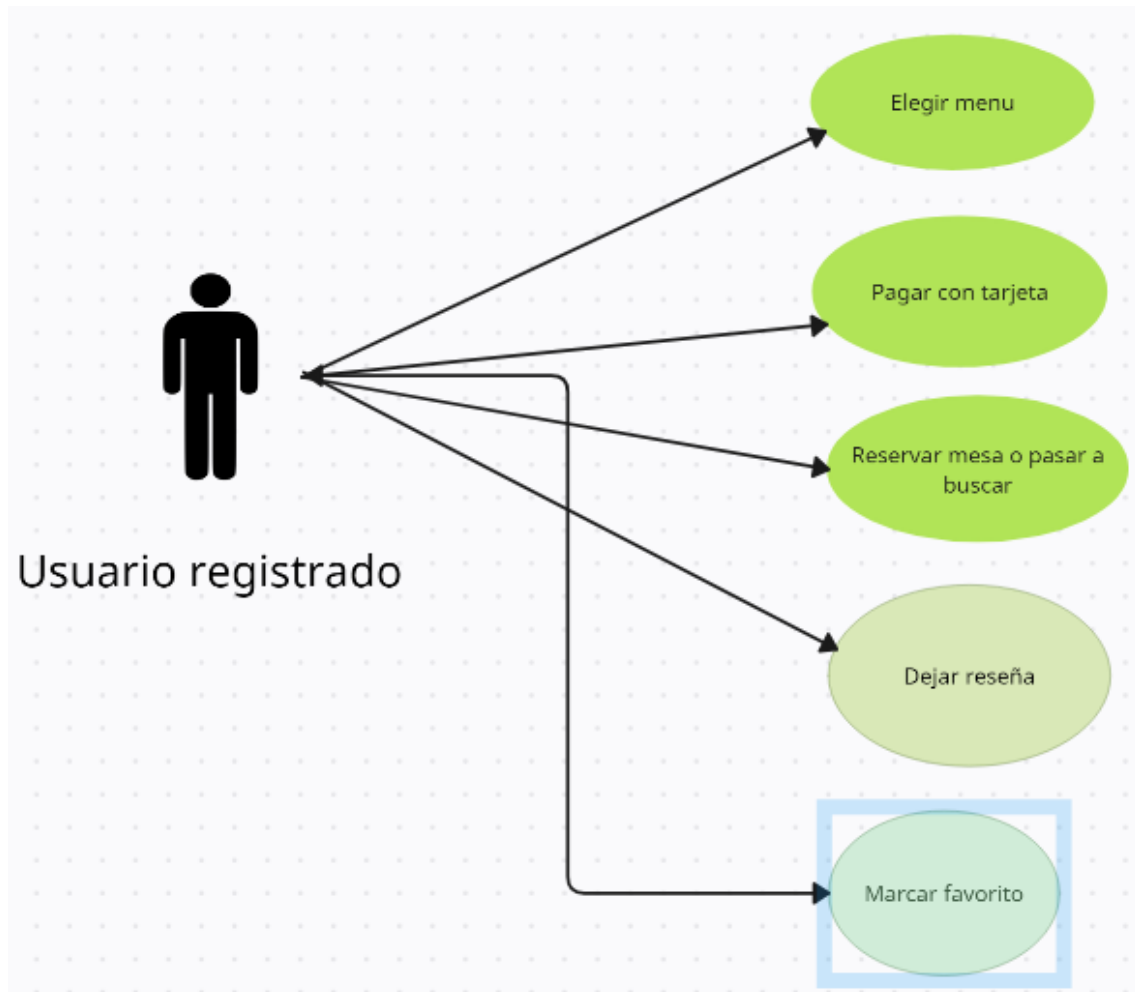
En esta aplicación vamos a identificar 3 actores: Usuario sin Registrar, Administrador, y Usuario Empresa. El usuario Registrado es una ampliación de usuario sin registrar, es decir que tendrá los mismos casos de uso que el usuario sin registrar añadiendo los nuevos casos



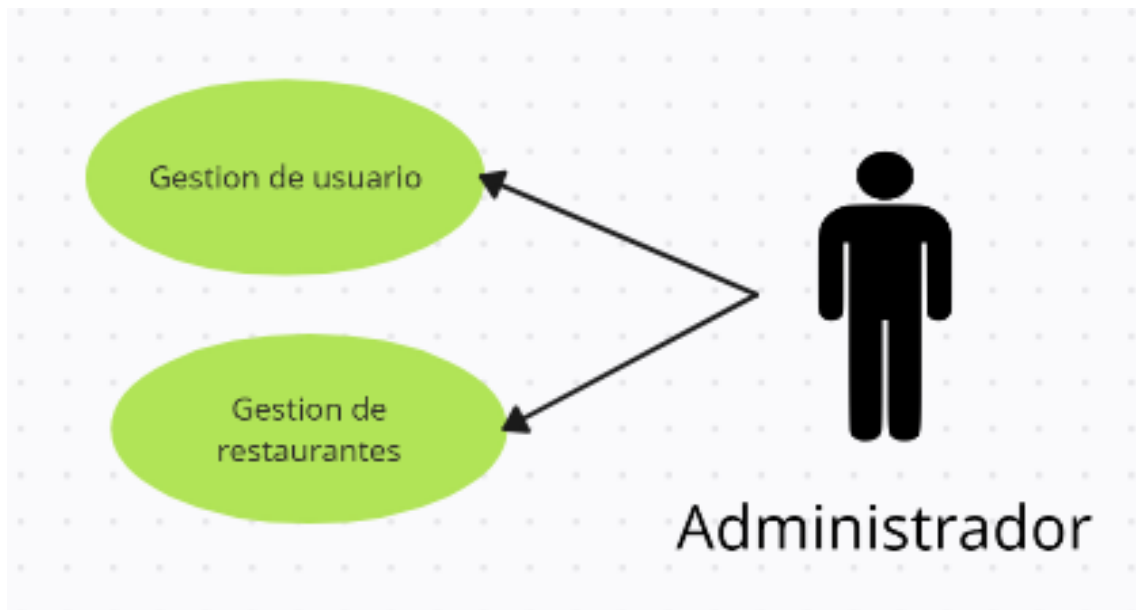
El usuario sin registrar que acceda a la aplicación tendrá acceso a las información de la aplicación, como Información del Restaurante, y también su ubicación, obviamente podrá registrarse o iniciar sesión



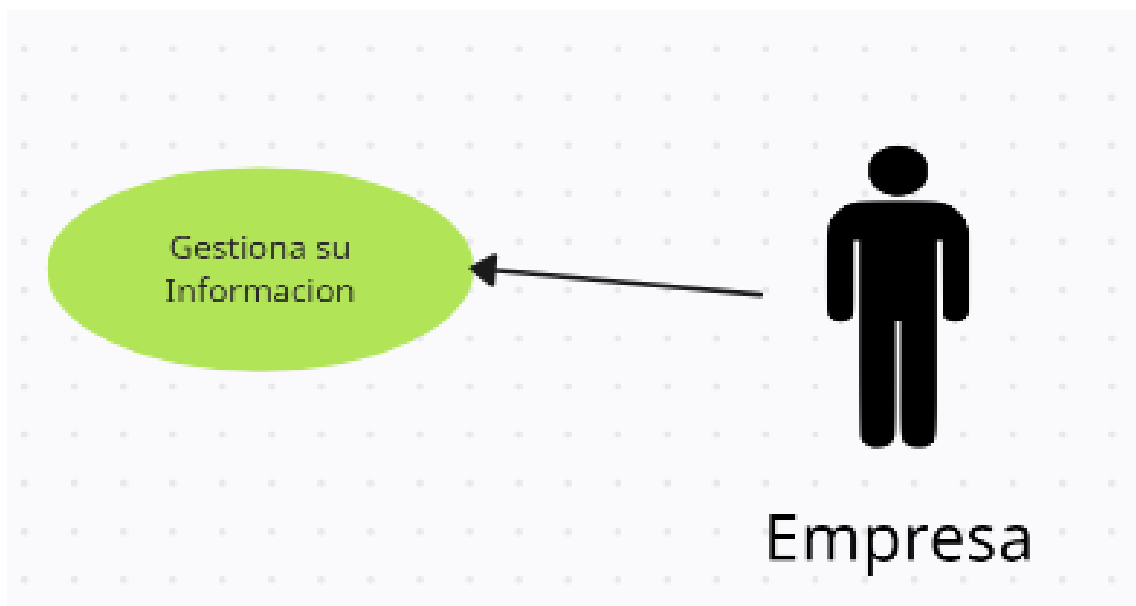
El usuario que este registrado, además de las que tiene el NO registrado, se le va a añadir nuevas funcionalidades, una de ellas cuando ya seleccioné el restaurante será armar o elegir su menú, luego el usuario podrá seleccionar si comer en el lugar o pasarlo a buscar. Otra funcionalidad es la de pagar con tarjeta. Luego podrá dejar su comentario, y marcar al restaurante como favorito para ayudar el restaurante, y facilitar su búsqueda en el próximo pedido.



En el caso del Administrador podrá gestionar los restaurantes que aparecen al igual que añadirlos, o eliminarlos. También gestionara los usuarios registrados de la aplicación.



En el caso del usuario Empresa, solamente podrá gestionar la información que mostrará en su menú



### 3.2 – Elaboración de un guion de trabajo para el desarrollo del proyecto

Para realizar este proyecto voy a llevar a cabo los siguientes pasos:



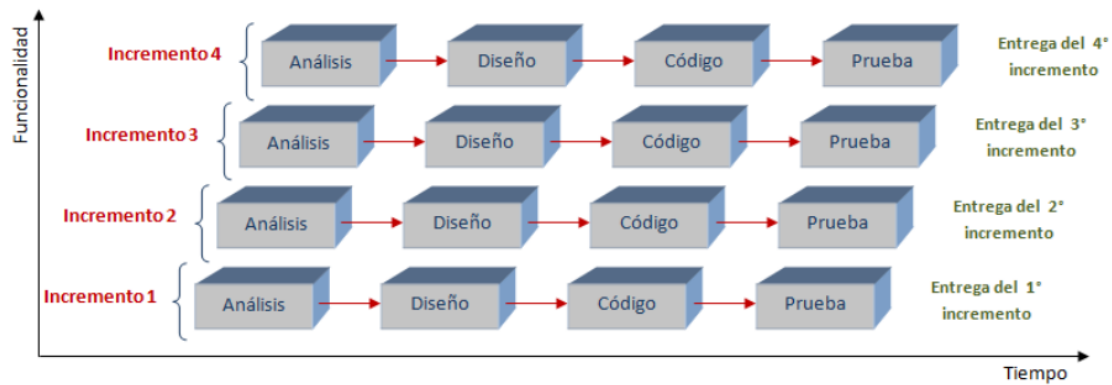
- Especificación de requisitos: Hare un breve estudio de las funcionalidades de la aplicación para saber los requisitos necesarios
- Elegir las tecnologías adecuadas para llevar a cabo el proyecto
- Preparar el entorno de trabajo, digamos, instalar el entorno de desarrollo, preparar repositorios GitHub, etc.
- Realizar los estudios de casos de uso de la aplicación
- Diseñar la interfaz gráfica del conjunto de vistas que formaran la aplicación
- Diseño y creación de la base de datos
- Desarrollar el código lógico
- Implementación de servicios
- Pruebas unitarias
- Realizar las pruebas de usabilidad del proyecto
- Plan de marketing: Estudio del mercado, y publicitarse en las redes sociales

### 3.3 – Fases del Proyecto. Tareas y Plazos de Ejecución

#### 3.3.1 – Metodología de trabajo

Para realizar este proyecto se va a utilizar la metodología de trabajo Incremental, ya que los requisitos se dividen en varios módulos independientes de un ciclo de desarrollo software.

Como podemos observar en la imagen, se separa de diferentes incrementos en los que se tendrán en cuenta funcionalidades que se irán añadiendo a las anteriores con cada incremento. Para saber con qué se debe comenzar, hay que valorar cual es la necesidad de mayor prioridad.



### 3.3.2 – Fases de cada incremento

El desarrollo de este proyecto se va a realizar mediante un número determinado de incrementos. En cada una de ellas se tendrá en cuenta cuatro de ellas: Análisis ,Diseño ,Código ,Prueba

#### **Análisis:**

En esta fase se determinarán los requisitos necesarios para llevar a cabo cada funcionalidad nueva a implementar

Primero se realizará una investigación del mercado actual, así como competencia y usuarios. Con esto podremos determinar los conceptos básicos de la aplicación.

Tenemos que también determinar la compatibilidad con los navegadores, así como los dispositivos, tanto a los usuarios con dispositivos móviles y ordenadores. Otro caso es comprobar la compatibilidad del código HTML y CSS en los diferentes navegadores.

#### **Diseño:**

Se llevarán a cabo todas las acciones relacionadas con la parte visual. Que se separe la estructura de la aplicación en diferentes partes. Esto permite ahorrar tiempo y trabajo ya que se reutiliza código que tengan el mismo diseño, como cabecera y pie de la aplicación.

#### **Implementación del código:**

Se realizará el desarrollo de los diferentes componentes, en el lenguaje de programación correspondiente especificado anteriormente

Para el control de versiones se utilizará GitHub, mediante la creación de un repositorio privado.

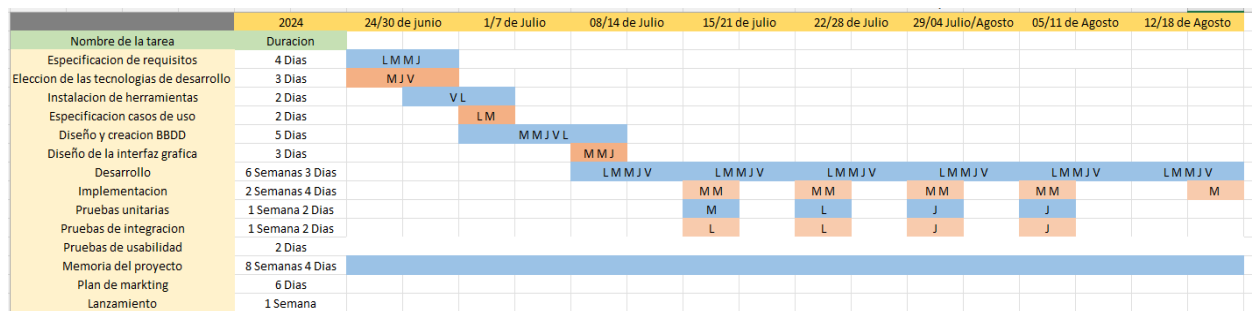
## Pruebas:

En esta fase se van a realizar todas las pruebas de test unitarios, tanto lo de integración como los de usabilidad para comprobar el correcto funcionamiento.

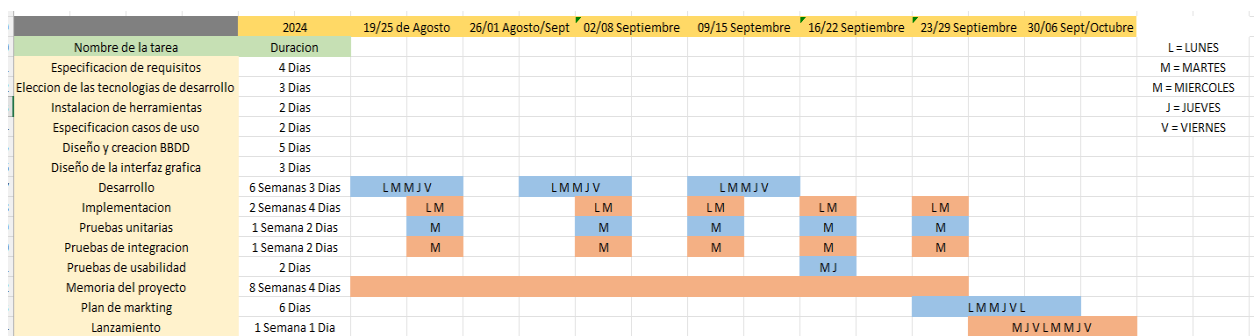
### 3.3.3 – Tares y Plazo de Ejecución

Para que sea un proyecto más organizado y poder facilitar el cálculo del tiempo estipulado, se utilizara el diagrama de Gantt. Comenzaremos por la semana del 24 de junio, y la finalización del proyecto será el 6 de octubre.

#### Inicio del Proyecto:



#### Finalización del Proyecto:



L = LUNES  
M = MARTES  
M = MIERCOLES  
J = JUEVES  
V = VIERNES



Como se observa en el diagrama, los procesos que más tiempo requieren son el Desarrollo y la documentación. El desarrollo se prevé que sean 6 semanas. El plan de marketing es importante, ya que es fundamental para llegar al mayor número de usuarios posibles, al igual que a mayor número posibles de restaurantes.

### 3.4 – Recursos Materiales y Personales

Este proyecto contará con dos fases, la primera de desarrollo de la aplicación web y la puesta en marcha de la empresa.

En la primera fase, con respecto a los recursos personales y materiales, será necesario un ordenador y un desarrollador de aplicaciones web que cuente con las herramientas necesarias para llevar a cabo el desarrollo del proyecto sin problemas. El desarrollador de aplicaciones web será el promotor principal para el proyecto.

Con respecto a la segunda fase del proyecto, la puesta en marcha de la empresa, se necesitarán tantos recursos materiales e inmateriales como por ejemplo un alojamiento web o una oficina para realizar los trabajos de gestión del proyecto.

Con respecto a la administración de la empresa, la llevará a cabo el propio promotor del proyecto.

#### **Recursos personales:**

Como se ha comentado anteriormente, para el desarrollo de la aplicación web y la administración de la empresa se requiere de una persona para llevar a cabo ambas partes del proyecto. En este caso será el promotor del proyecto quien se encargará de ambas acciones.

En el caso que la aplicación web, tenga un repunte muy alto en el transcurso de los años, se contratara personal para así poder repartir las tareas.

#### **Recursos Materiales:**

Con respecto a los recursos materiales, uno de los más importantes será la oficina de trabajo. Por eso se optó como se mencionó en los apartados de más arriba que la mejor opción es utilizar un coworking, porque aunque se puede realizar por teletrabajo, estará mejor preparado para el trabajo diario, y además el ahorro de costos que tendrías con el teletrabajo.

Un punto muy positivo es que dispondrá de una sala de reuniones, en el que se podrá realizar reuniones con posibles clientes, o trabajadores que se tengan en un futuro. También será necesario un ordenador para llevar a cabo el desarrollo de la aplicación web.

### Recursos Inmateriales:

Con respecto a estos recursos, será necesario el alojamiento web de la aplicación, para que cualquier usuario pueda acceder al sitio web.

## 3.5 – Estimación de Gastos

Con lo comentado anteriormente, se puede hacer una estimación de los gastos necesarios para que la aplicación Web se mantenga abierta durante el primer año

Costes fijos (total anual)	Euros (total anual)
• Alquiler (alojamiento web)	1080
• Gestor	1620
• Internet (móvil)	228
• SS	3600
• Hosting	60
• Plan marketing	145
• Ordenador	600
	TOTAL:7.333

## 3.6 – Estimación de ingresos y Precios de venta



La obtención de ingresos de la aplicación será la venta de servicios de publicidad, en la que puede variar el coste de la membresía según el tiempo que permanezca cada empresa en la aplicación. Esta aplicación ofrecerá, 2 tipos de membresías que las opciones serán de un mes, o todo el año

Duración de membresía	Precio
• 1 mes	100
• 1 año	1000

Una vez que tengamos los precios de cada membresía, hay que calcular el umbral de rentabilidad, para ver si es rentable o no tener las membresías a este precio

$$\text{Umbral de rentabilidad} = \frac{CF(\text{total})}{(\text{Precio})}$$

Con respecto a la primera membresía, que sería de 100 por mes, se calculó que habría que vender 73 membresías en total el primer año, lo que significa que al mes habría que vender un total de 6 membresías. Con esto se llegarían a cubrir todos los gastos del primero año, sin contar con la membresía del año completo.

Con respecto a la segunda membresía, que sería de 1000 por año, se calculó que será necesario vender 7 membresías de este tipo para cubrir gastos del primer año. En este caso no se tiene en cuenta la membresía del solamente un mes.

Sabiendo que la empresa es nueva, se puede estimar que la membresía que en un principio se va a usar más será la del mes (100 ), ya que como dije es una nueva empresa, y los restaurantes al ver que es nueva empresa van a probar en un principio con la del mes que con la de un año.

### 3.7 – Viabilidad económica

Teniendo en cuenta los costes calculados anteriormente y los precios de las diferentes membresías, faltaría detallar el plan de financiación de la empresa antes de abordar el plan de tesorería.

El promotor del proyecto aportara una cantidad inicial de 2000€, a la que se le sumara el préstamo que se pedirá al banco BBVA que serán 4000€ con un interés de un 8,70% que te quedaría un total de 4348€ a devolver en 2 años. Lo ideal es no pedir un préstamo tan grande. Por mes quedaría en 181,16€/mes

También un dato importante siendo cliente en el BBVA con tu nomina domiciliada, ahorras un 46,03€

Ingresos	Cantidad	Especificaciones
Promotor	2.000 €	Inicio del proyecto
Prestamo	4.500 €	Cuota 84,92€/mes

Una vez calculado el plan de financiación. Se calculará el plan de tesorería con los datos calculados anteriormente. Este plan consta de 2 tablas, la primera que incluye el total de ingresos, y la segunda incluirá el total de gastos cada mes, así como los valores mensuales, y anuales.

## TOTAL, INGRESOS

Entradas	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Total
Prestamo	4.500 €	-	-	-	-	-	-	-	-	-	-	-	4500
Aportacion Soc	2.000 €	-	-	-	-	-	-	-	-	-	-	-	2000
Ventas	0 €	1200	1600	1600	2100	2100	2100	1700	1600	1300	1200	800	17.300
Total Mes	6.000 €	1200	1600	1600	2100	2100	2100	1700	1600	1300	1200	800	23.800

## TOTAL, DE GASTOS

Gastos	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Total
Capital amortizado del prestamo	66 €	66,45	66,72	67	67,28	67,56	67,84	68,12	68,41	68,69	68,98	69,24	812,49
Interes del prestamo	18,75 €	18,47	18,02	17,92	17,64	17,36	17,08	16,8	16,51	16,23	15,94	15,65	206,55 €
Alquiler	90 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	90,00 €	1.080,00 €
Compra ordenadores	-	-	-	300,00 €	-	-	-	-	-	300,00 €	-	-	600,00 €
Material Oficina	100 €	-	-	100,00 €	-	-	100,00 €	-	-	100,00 €	-	-	400,00 €
Publicidad	-	-	-	150,00 €	-	-	-	-	150,00 €	-	-	-	300,00 €
Suministros	50 €	-	50,00 €	-	-	-	50,00 €	-	-	50,00 €	-	-	200,00 €
Estudio del mercado	250 €	-	-	-	-	-	-	-	-	-	-	-	250,00 €
SS	300 €	300,00 €	300,00 €	300,00 €	300,00 €	300,00 €	300,00 €	300,00 €	300,00 €	300,00 €	300,00 €	300,00 €	3.600,00 €
Gestor	135 €	135,00 €	135,00 €	135,00 €	135,00 €	135,00 €	135,00 €	135,00 €	135,00 €	135,00 €	135,00 €	135,00 €	1.620,00 €
Internet movil	19 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	228,00 €
Hosting	5 €	5,00 €	5,00 €	5,00 €	5,00 €	5,00 €	5,00 €	5,00 €	5,00 €	5,00 €	5,00 €	5,00 €	60,00 €
Internet movil	19 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	228,00 €
Internet movil	19 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	19,00 €	228,00 €
Total GASTOS	1.072 €	671,92	721,74	1221,92	671,92	671,92	821,92	671,92	821,92	1121,92	671,92	671,89	9813,04

Como podemos ver, los ingresos estimados obtenidos en primer año son de 23.800, y el total de gastos en el año son de 9813,04€

### 3.8 – Necesidad de Financiación

A continuación, detallamos las necesidades de Financiación iniciales para el proyecto según la estimación de cálculos anteriormente.

Fuente financiación	Cantidad
Promotor	2.000 €
Prestamo	4.500 €
Necesidad total	6.500 €

En el proyecto contamos con varias formas de financiación para que el proyecto pueda ponerse en marcha. La primera es la aportación económica del promotor que será de 2000 euros. Por otra parte, se solicitará un préstamo bancario a largo plazo “5 años” al banco BBVA que se especifica a continuación.

PRESTAMO BANCARIO	
Entidad	BBVA
Capital	4.500 €
Condiciones	Cuotas : 60
	Interes fijo: 5%
	Plazo: 5 Años

### 3.9 – Definición y Elaboración de la Documentación del Diseño Control de Calidad

Con respecto a la parte del diseño de la aplicación, vamos a encontrar que, en el cuerpo, tenemos una lista de los restaurantes que mejor valoración tienen por parte de los clientes. Además, el usuario podrá hacer una búsqueda mediante su ubicación y que se le muestren los restaurantes más cercanos

El diseño de la página web de “Fast-Food” va a tener la siguiente estructura

Menú / Cabecera / Zona de registro - Autenticación / Contenido / Pie de pagina

Las zonas deberán ser claramente definidas para poder realizar cualquier cambio de forma rápida sin necesidad de modifica el resto del contenido.

La estructura es la siguiente:

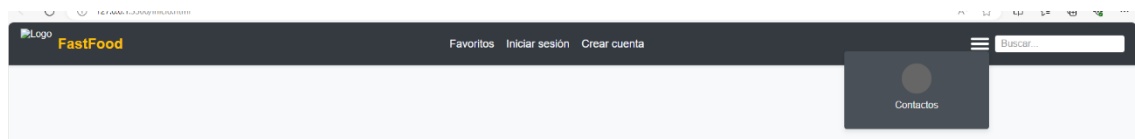
1. Menú	3. Zona de registro / autenticación
2. Cabecera	
4. Contenido	
5. Pie de página	

## Cabecera

En la aplicación nos encontraríamos con el logo arriba a la izquierda, junto con el nombre de la empresa. En el menú de opciones habrá un enlace que le mostrará los restaurantes que el usuario guardo como favorito.

El menú de opciones tendrá 2 enlaces más para Iniciar sesión, y otra para crear su cuenta. Si el usuario inicia sesión arriba a la derecha le aparecerá el nombre del usuario.

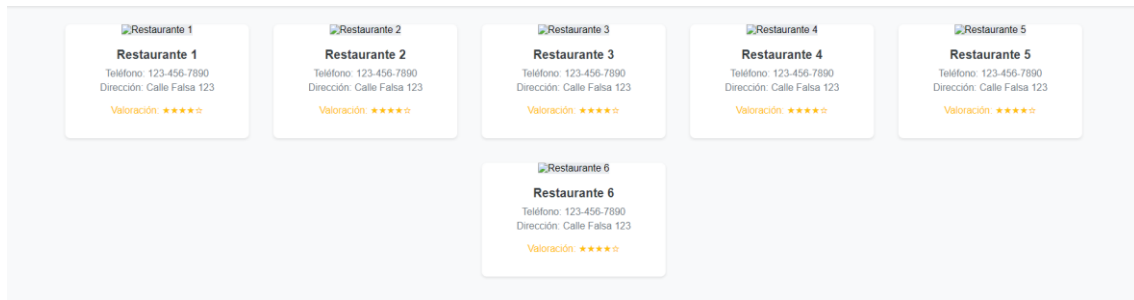
En el menú tendrá un botón de hamburguesa, que al desplegar tendrá su foto de perfil, con su nombre. Encontrará un enlace llamado “Contáctanos” que contendrá información de la aplicación, correo electrónico, y preguntas frecuentes sobre la aplicación.



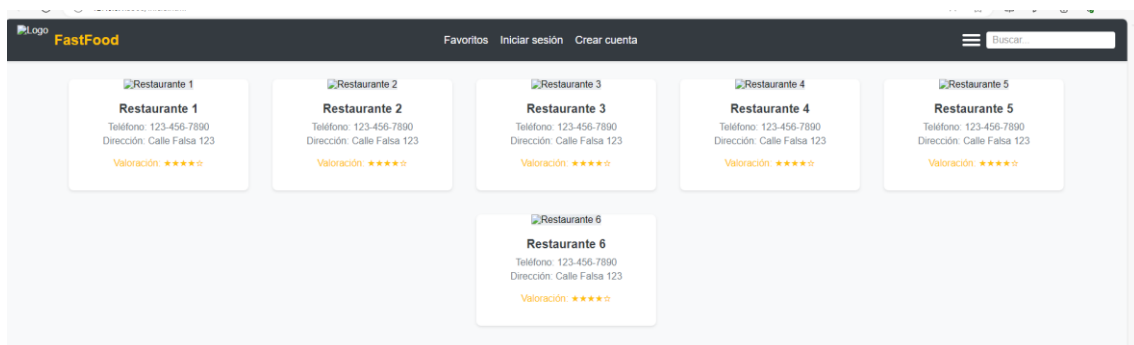
## Página principal

En la página principal podremos ver una lista de restaurantes con mejor valoración, que el usuario podrá interactuar con ellos. Cada restaurante tendrá información, como su teléfono, dirección, etc.

Los restaurantes se van a mostrar en una fila, de una manera más cómoda para el usuario, y de fácil lectura. Ya que a esta aplicación los usuarios interactuaran más con ella en sus móviles.

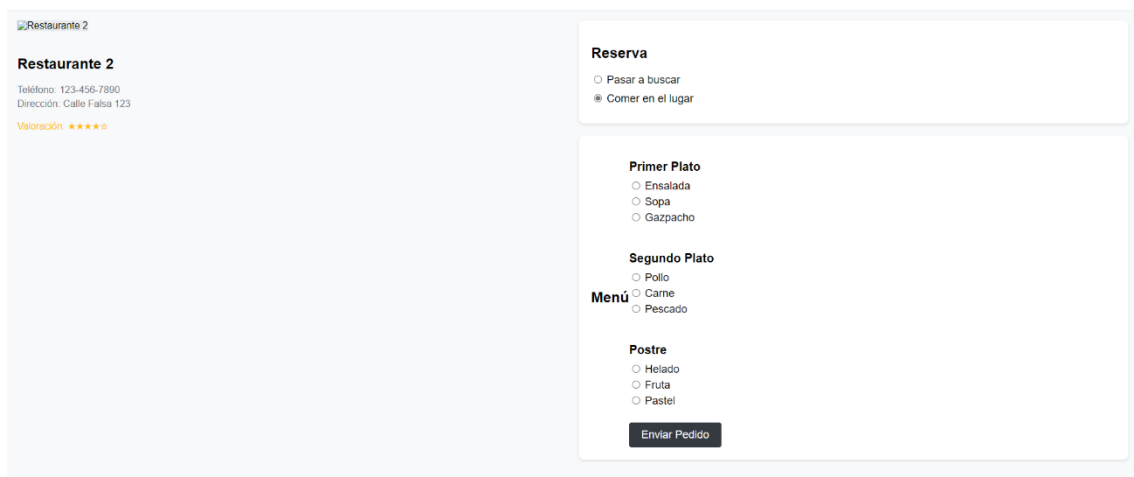


Esta sería una demostración del menú con los restaurantes, agregándole la cabecera.

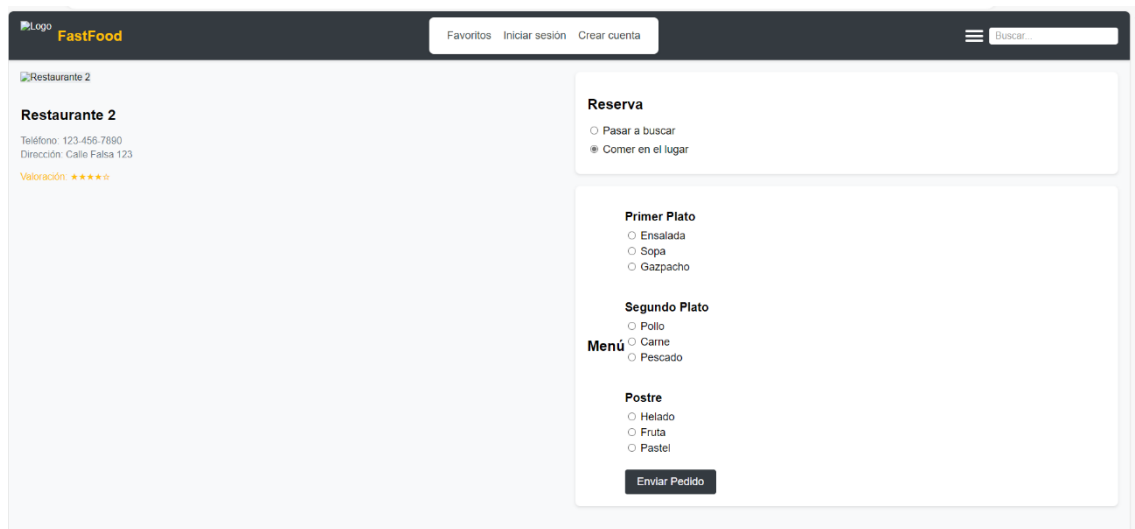


“¿Qué pasa cuando elijo un restaurante?” La aplicación te llevara a otra ventana en la que el usuario pueda armar su menú, también al usuario darle la opción de comer en el lugar, o pasarlo a buscar.

A la hora de seleccionar platos, solamente podrá seleccionar uno por cada plato.



Este menú, más la cabecera se vería de esta manera



The screenshot shows the FastFood website interface. On the left, there's a restaurant profile for 'Restaurante 2' with contact information and a 5-star rating. On the right, there's a 'Reserva' (Reservation) form with options to 'Pasarlo a buscar' or 'Comer en el lugar'. Below the reservation options, there's a menu section with categories: 'Primer Plato' (Ensalada, Sopa, Gazpacho), 'Segundo Plato' (Pollo, Carne, Pescado), and 'Postre' (Helado, Fruta, Pastel). An 'Enviar Pedido' button is at the bottom of the menu section.

## Footer

Estableceremos dirección de copyright de la página y enlaces a apartados legales. Tendremos un numero de contacto, y una breve introducción sobre nosotros



The footer section is divided into three columns: 'Sobre Nosotros' (About Us), 'Enlaces Rápidos' (Quick Links), and 'Contacto' (Contact). 'Sobre Nosotros' describes the company's mission. 'Enlaces Rápidos' lists links to Inicio, Menú, Contactos, and Acerca de. 'Contacto' provides the phone number, email, and address. A copyright notice '© 2024 FastFood. Todos los derechos reservados.' is at the bottom.

## Control de Calidad

Para llevar a cabo el control de calidad, se divide en 2 partes:

- Pruebas Unitarias que se comprobaba el correcto funcionamiento de los formularios de registro de los usuarios, y para comprobar que la información mostrada es la correcta.
- Prueba de Integración hay que asegurar que cada usuario, según el rol que tenga, pueda acceder a unas opciones determinadas.

## 3.10 – Revisión de la Normativa Aplicable

Para poder llevar a cabo este proyecto y no tener ningún problema legal en un futuro este proyecto cumplirá con las siguientes normativas:

### 1. Ley Orgánica de Protección de Datos y Garantía de los Derechos Digitales (LOPDGDD)

- Es muy importante que se protejan los datos, tanto de los usuarios como los datos de las empresas, para asegurar la seguridad a la hora del tratamiento de los datos.
-

## 2. Normativa de Cookies, Directiva 2009 / 136 / CE

- A través de esta normativa, se establecen a nivel europeo los usos permitidos y la forma en la que se debe solicitar el consentimiento de los usuarios al uso de “cookies”.

## 3. Ley de Sociedad de Servicios de Información y Comercio Electrónico

- En esta ley se definen las reglas necesarias para que el uso y la actividad económica en la red sea una experiencia positiva, segura y confiable

## 4. Normativa de Consumidores y Usuarios

- Son un conjunto de normas a nivel europeo que tratan de proteger los derechos de los consumidores y usuarios. Algunos derechos de los consumidores y usuarios son la información correcta, la protección de datos y la protección de sus legítimos intereses económicos y sociales

## 5. Ley de Propiedad Intelectual

- Esta ley protege la Propiedad Intelectual mediante las patentes, el derecho de autor y las marcas.

# 4. – Implementación

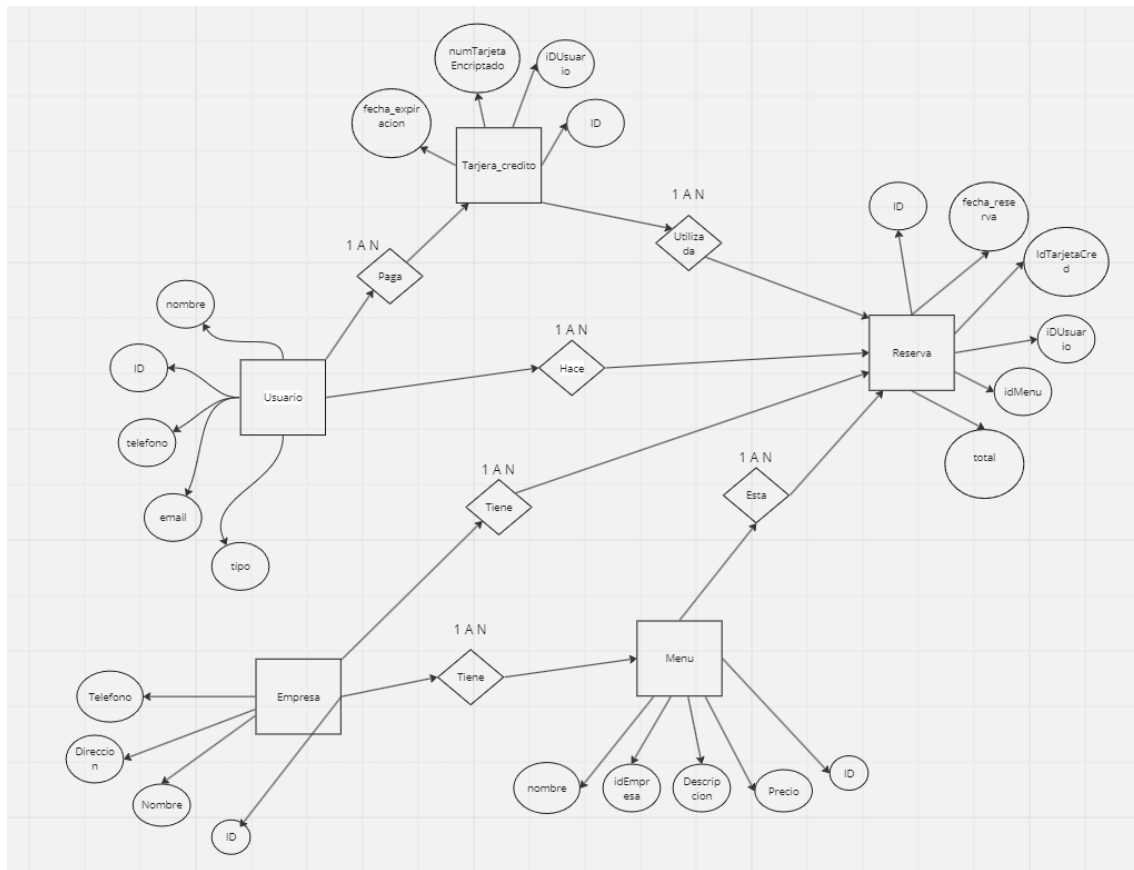
## 4.1 – Desarrollo del modelo de datos

Con respecto al modelo de datos, se muestran las siguientes entidades que formaran la aplicación

Entidad	Descripcion CP	Nombre CP
Usuario	Id del usuario, tanto cliente como administrados	IdCliente
Empresa	Id de la empresa	IdEmpresa
Menu	Id del Menu	IdMenu
Tarjeta credito	Id Tarjeta de credito	IdTarCre
Reserva	Id de la reserva	IdReserva

El modelo entidad relación que vemos, explica la idea de objetivo de empresa

En próximos apartados explicaremos más detalladamente el concepto y el trabajo de cada tabla



## RELACION ENTRE TABLAS

**Usuario y Tarjeta de crédito:** Relación uno a muchos. Un usuario puede tener múltiples tarjetas de crédito.

**Empresa y Menú:** Relación uno a muchos. Una empresa (restaurante) puede tener múltiples menús.

**Usuario y Reserva:** Relación uno a muchos. Un usuario puede hacer múltiples reservas

**Menú y Reservas:** Relación uno a muchos. Un menú puede estar en múltiples reservas.

**Tarjeta de crédito y Reserva:** Relación uno a muchos. Una tarjeta de crédito puede ser utilizada en múltiples reservas

**Empresa y Reserva:** Un Restaurante tiene múltiples reservas



Entonces el usuario con tarjeta de crédito puede tener múltiples tarjetas asociadas para realizar pagos. El restaurante puede ofrecer muchos menús a los usuarios. Y la tabla Reserva conecta usuarios, menús, y tarjeta de crédito indicando que usuario reservo que menú y con que tarjeta se realizó el pago.

Este modelo de datos asegura que las relaciones entre las entidades están bien definidas, permitiendo una gestión eficiente y segura de la información de usuarios, empresas, menús y pagos, mediante tarjetas de créditos

Tabla Usuario		
IdUser	INT PRIMARY KEY	AUTO_INCREMENT
NomUser	NVARCHAR(70)	
TelUser	NVARCHAR(15)	
DniUser	NVARCHAR(10)	
EmailUser	NVARCHAR(50)	UNIQUE
CodPUser	NVARCHAR(8)	
TipoEnum	INT (0 user y 1 admin)	

Con respecto a la entidad o tabla Usuario vemos que tiene varias propiedades, de las cuales serán datos privados para el correcto funcionamiento.

Podemos ver que el “IdUser” es la clave primaria y es auto incrementable. La propiedad TipoEnum muestra que si el Usuario es 0 es Cliente, y si es 1 Administrador.

Tabla Empresa		
IdEmp	INT PRIMARY KEY	AUTO_INCREMENT
NomEmp	NVARCHAR(70)	
TelEmp	NVARCHAR(15)	
EmailEmp	NVARCHAR(50)	
DirEmp	NVARCHAR(50)	

La tabla o entidad Empresa, también almacenara datos importantes, como teléfono “TelEmp”, correo “EmailEmp”, dirección del restaurante “DirEmp”.

Podemos ver que el id de la tabla es “IdEmp” que es la clave primaria y será auto incrementable.

Tabla Menu		
IdMenu	INT PRIMARY KEY	AUTO_INCREMENT
Id_Emp	INT	
NomMenu	NVARCHAR(30)	
DescMenu	NVARCHAR(100)	
PreMenu	Decimal(10,2)	
FOREIGN KEY (Id_Emp ) REFERENCES Empresa(IdEmp)		

La tabla Menú es diferente a las demás ya que representa a un elemento de la aplicación, es dicho caso a la tabla Empresa. Tenemos el “Id\_Emp” que representa a la tabla Empresa, haciendo la referencia con el Foreign Key. Luego tenemos la descripción del menú, y la propiedad de abajo el precio de dicho menú.

Tabla Tarjeta de credito		
IdTarCre	INT PRIMARY KEY	AUTO_INCREMENT
Id_User	INT	
NumTarCre_encryptado	VARBINARY(256)	
FecExpiTarCre	DATE	
CvvTarCre	VARCHAR(4)	
FOREIGN KEY (Id_Emp ) REFERENCES Empresa(Id_User)		

La tabla Tarjeta de Crédito vemos que además de tener la Primary Key en “IdTarCre” con auto incremento, vamos a necesitar el Id\_Emp para poder asociar la tarjeta de crédito de una Usuario. Podemos notar que la propiedad del número de la tarjeta “NumTarCre\_encryptado” estará encriptado, luego propiedades necesarias para la tarjeta de crédito, la Fecha de caducidad, y el CVV.

Tabla Reserva		
IdReserva	INT PRIMARY KEY	AUTO_INCREMENT
Id_User	INT	
Id_menu	INT	
Id_tarjeta_credito	INT	
Id_Empresa	INT	
FechaReserva	DATETIME	
TotalReserva	DECIMAL(10,2)	
FOREIGN KEY (Id_User ) REFERENCES Usuario(Id_User)		
FOREIGN KEY (Id_menu ) REFERENCES Menu(IdMenu)		
FOREIGN KEY (Id_tarjeta_credito ) REFERENCES Tarjeta de credito(IdTarCre)		



La última tabla Reserva contiene su IdReserva, pero además a través del FOREIGN KEY enlazamos esta tabla, con las tablas de Usuario, Menú, Tarjeta de crédito.

## 4.2 – Desarrollo de la programación del entorno de servidor

En lo referido a la arquitectura de la aplicación en el entorno servidor, esta se va a utilizar con la tecnología ASP .NET Core MVC (Modelo, Vista, Controlador), y el lenguaje que vamos a utilizar para este proyecto es C#, todo esto dentro del entorno de desarrollo Visual Studio.

Este patrón de arquitectura del controlador de viste del modelo separa en tres grupos de componentes principales, modelo, vista, y controlador. Las solicitudes del usuario se enrutan al controlador que se encargará de trabajar con el modelo para llevar a cabo las acciones deseadas por el usuario o recuperar determinados resultados de una consulta. También el controlador, una vez realizada la acción será el que elegirá la vista para mostrar al usuario los datos proporcionados del modelo. Esta separación de componentes aporta diversos beneficios como la sencillez a la hora de la escalabilidad de la aplicación, la facilidad de codificar, depuraciones y también a la hora de probar diferentes tareas.

- Responsabilidad del modelo: representa el estado de la aplicación y proporciona los medios para consultar y modificar la información. Así, el modelo es la representación de la información que maneja la aplicación en un momento dado y gestiona los accesos a dicha información: CRUD. Esta constituido por una serie de clases que representan las entidades de una base de datos

```
public class Usuario
{
    public int Id { get; set; }
    public string Nombre { get; set; }
    public string Email { get; set; }
    public string Contraseña { get; set; }
    public DateTime FechaRegistro { get; set; }
}
```



- Responsabilidad de controlador: es la encargada de responder a las acciones solicitadas por el usuario, traduciéndolas en alteraciones sobre la vista o sobre el modelo. Se podría decir que es la intermediaria entre el modelo y la vista. El controlador maneja la interacción con el usuario, actúa sobre el modelo, y selecciona la vista a desplegar. Esto puede estar constituido por una clase que dentro lleva una serie de métodos

```
[ApiController]
[Route("api/[controller]")]
public class UsuariosController : ControllerBase
{
    private readonly ApplicationDbContext _context;

    public UsuariosController(ApplicationDbContext context)
    {
        _context = context;
    }

    // GET: api/usuarios
    [HttpGet]
    public async Task<ActionResult<IEnumerable<Usuario>>> GetUsuarios()
    {
        return await _context.Usuarios.ToListAsync();
    }

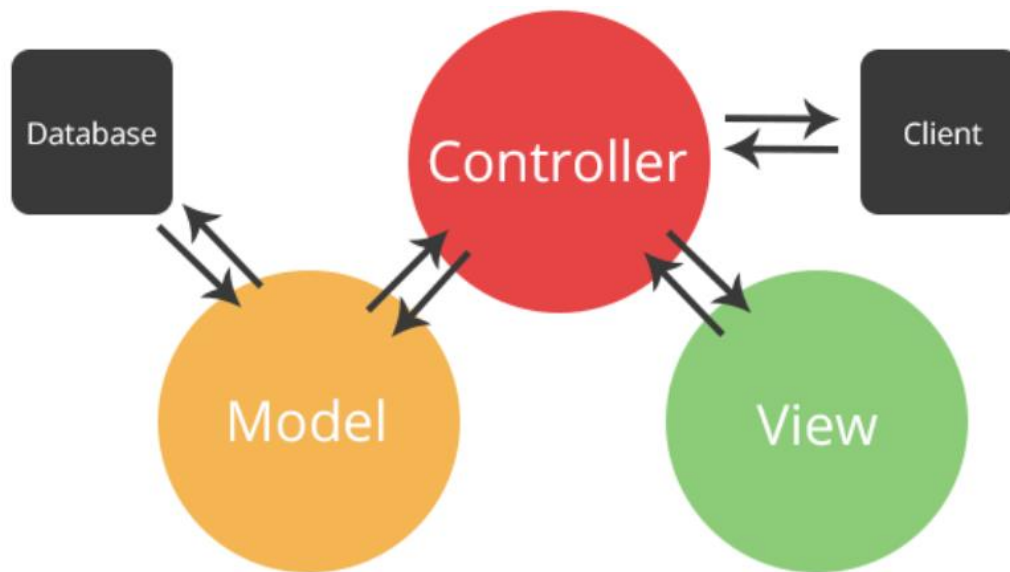
    // GET: api/usuarios/{id}
    [HttpGet("{id}")]
    public async Task<ActionResult<Usuario>> GetUsuario(int id)
    {
        var usuario = await _context.Usuarios.FindAsync(id);

        if (usuario == null)
        {
            return NotFound();
        }

        return usuario;
    }
}
```

- Responsabilidad de la vista: genera o pinta la interfaz a partir de la información obtenida del modelo. La vista cambiara a demanda del controlador, cuando una acción de usuario así lo requiera. Habitualmente

se utiliza un lenguaje o motor de vistas para generar las representaciones visuales.



#### 4.3 – Desarrollo de la Programación del Entorno de Cliente.

Para el desarrollo en el entorno cliente la intención es la de crear una imagen agradable de cara al usuario e intentando llevar una coherencia en todo el conjunto de la aplicación, para que de esta manera le sea más sencilla la navegación de los distintos usuarios de la web.

En este caso al usar la tecnología basada en ASP.NET Core MVC, las páginas de la aplicación web se generan de manera automática, estas vistas se forman con el lenguaje de vistas Razor y HTML. El código de vistas Razor se diferencia claramente por que este se encuentra entre llaves {} y precedido del signo arroba @. El lenguaje de vistas Razor admite el uso del lenguaje C# en las vistas de la aplicación, pudiendo combinar este lenguaje con el código de marcado HTML.

Se hará uso del lenguaje de marcado CSS3, con las hojas de estilo se le dará el aspecto deseado al proyecto web. Este lenguaje de marcado permite, además de otras cosas, modificar el color de diferentes elementos e incluso crear gradientes, rotar elementos, cambiar su tamaño y moverlos, cambiar el formato de las fuentes que se utilicen en la aplicación, etc.



- **HTML y Razor:** Se utilizarán para generar las páginas web dinámicas en el servidor. Razor permite incrustar código C# en las vistas, facilitando la generación de contenido dinámico.
- **CSS3 y Bootstrap:** CSS3 se utilizará para estilizar la interfaz de usuario, permitiendo modificar colores, fuentes y disposición de elementos. Bootstrap se integrará para utilizar componentes predefinidos y estilos responsivos, mejorando la coherencia y la apariencia del sitio web.

### Detalles del Proyecto

- **Página Maestra:** Utilizaremos una página maestra (`_Layout.cshtml`) para definir la estructura común de las páginas, como la cabecera, pie de página y navegación. Esto asegura una apariencia y navegación coherentes en toda la aplicación.
- **Vistas Específicas:** Cada funcionalidad, como listar usuarios, ver detalles, crear, editar y eliminar, tendrá su propia vista Razor. Por ejemplo, `Index.cshtml` listará los usuarios, `Details.cshtml` mostrará detalles de un usuario, y así sucesivamente.
- **Vistas de Inicio:** `_ViewStart.cshtml` y `_ViewImports.cshtml` configurarán directivas y layouts comunes a todas las vistas. `_ViewStart.cshtml` especificará el layout a usar, mientras que `_ViewImports.cshtml` incluirá las directivas necesarias.
- **Estilos y Scripts:** Los archivos CSS y JavaScript personalizados se incluirán en la carpeta `wwwroot`. Se utilizarán archivos como `site.css` para estilos personalizados y `site.js` para scripts.

### Explicación de un ejemplo de flujo de trabajo:

El usuario solicita ver una lista de restaurantes. Luego de esa solicitud le llega al controlador manejando así la solicitud, interactúa con el modelo para obtener la lista de restaurantes y selecciona la vista '`Index.cshtml`' para devolvérsela al usuario. Ahí entra Razor para generar el HTML dinámico basado en los datos del modelo. El servidor envía el HTML generado al cliente, que se renderiza en el navegador del Usuario.

### Beneficios de la Arquitectura MVC:

Los beneficios de esta arquitectura es que la lógica de la aplicación, la interfaz del usuario y el control de las interacciones se gestionan por separado, lo que facilita la escalabilidad, mantenimiento y pruebas.

## Estructura de clases y Controladores en ASP.NET Core MVC

Para mantener la aplicación organizada y eficiente, la estructura de clases y controladores seguirá una jerarquía clara que permite manejar las peticiones y respuestas de manera coherente

Modelos: Representan las entidades de la base de datos. También incluyen propiedades que mapean a las columnas de la base de datos. Un ejemplo claro de un modelo seria:

```
public class Usuario
{
    public int Id { get; set; }
    public string Nombre { get; set; }
    public string Email { get; set; }
    public string Contraseña { get; set; }
    public DateTime FechaRegistro { get; set; }
}
```

Controladores: Van a manejar las peticiones HTTP, interactuando con los modelos y seleccionan las vistas

```
public class UsuariosController : Controller
{
    private readonly ApplicationDbContext _context;

    public UsuariosController(ApplicationDbContext context)
    {
        _context = context;
    }

    // GET: Usuarios
    public async Task<IActionResult> Index()
    {
        var usuarios = await _context.Usuarios.ToListAsync();
        return View(usuarios);
    }

    // GET: Usuarios/Details/5
    public async Task<IActionResult> Details(int? id)
    {
        if (id == null)
        {
            return NotFound();
        }
        var usuario = await _context.Usuarios.FirstOrDefaultAsync(m => m.Id == id);
        if (usuario == null)
        {
            return NotFound();
        }
        return View(usuario);
    }

    // GET: Usuarios/Create
    public IActionResult Create()
    {
        return View();
    }
}
```

Vistas: Se generan usando Razor y se organizan en carpetas correspondientes.



## Página Maestra:

Para asegurar una apariencia coherente y una estructura uniforme, se utilizará una página maestra (`_Layout.cshtml`). Esta plantilla definirá la estructura general de la aplicación, incluyendo la cabecera, pie de página y navegación. Todas las demás vistas se derivarán de esta página maestra.

`Layout.cshtml`: Define la estructura básica de las páginas, incluyendo enlaces a archivos CSS y JavaScript comunes, como Bootstrap y los estilos personalizados.

Cada vista específica (como `Index.cshtml`, `Create.cshtml`, `Edit.cshtml`, `Details.cshtml`, `Delete.cshtml`) utilizará la página maestra `_Layout.cshtml` y se centrará en el contenido específico para cada operación.

`Index.cshtml`: Muestra la lista de usuarios.

`Create.cshtml`: Muestra el formulario para crear un nuevo usuario.

`Edit.cshtml`: Muestra el formulario para editar un usuario existente.

`Details.cshtml`: Muestra los detalles de un usuario específico.

`Delete.cshtml`: Muestra la confirmación para eliminar un usuario.

Al utilizar ASP.NET Core MVC con Razor, se crea una estructura clara y organizada que separa la lógica de la aplicación (controladores y modelos) de la presentación (vistas). Esta arquitectura facilita el mantenimiento y la escalabilidad de la aplicación, asegurando una experiencia de usuario coherente y una navegación intuitiva a través de la página maestra y las vistas específicas.

Además, con la ayuda del Framework de Bootstrap, nos va a ayudar a crear elementos tales como botones, tablas y menús entre otros elementos. Es una potente herramienta de código abierto la cual es de gran ayuda a la hora de diseñar los sitios y aplicaciones web. Dentro podemos encontrar muchas plantillas de diseño con formularios, tipografías, menús de navegación, botones, uso de columna en la cual son de gran ayuda a la hora del Responsive y una multitud de plantillas para poder realizar un sitio web de manera coherente y uniforme





Al usar estas tecnologías ASP .NET Core MVC con Razor y HTML, CSS3 y Bootstrap, crearemos una aplicación moderna, coherente y responsiva. Estas tecnologías permiten desarrollar una interfaz de usuario intuitiva y estéticamente agradable, mejorando así la experiencia del usuario y facilitando la navegación en toda la aplicación. La combinación de estas herramientas asegura un desarrollo eficiente y un mantenimiento sencillo del proyecto. Las páginas estarán diseñadas de una forma muy estructurada, para que el uso de ella sea lo más sencillo posible y no resulte una navegación compleja o difícil de entender.

## 5. Validación

### 5.1 – Definición del procedimiento de evaluación, seguimiento y control del proyecto. Indicadores de calidad.

Como ya se comentó en apartados anteriores, el proyecto se realizará con una metodología incremental, en la que el proyecto se realizara en módulos, en las cuales, al final de cada incremento, se ejecutaran las pruebas unitarias necesarias para el correcto funcionamiento de cada módulo.

Para realizar el seguimiento del proyecto, se utilizará la planificación plasmada en el diagrama de Gannt, en el cual se marcan los plazos de realización y desarrollo de cada fase, así también como los plazos de entrega.

Hablando del desarrollo, cuando terminamos cada módulo tendremos en cuenta la compatibilidad de cada componente con los navegadores más utilizados hasta ahora, así como el correcto funcionamiento de las diferentes características implementadas. Si no se cumple el visto bueno, no se dará el visto bueno

También es importante ponerse en el lugar del usuario a la hora de probar la aplicación, para así encontrar los errores de visualización o elementos y que la aplicación funciones lo mejor posible.

Como último punto, hablando de los indicadores de calidad, el administrador tendrá acceso al número de usuario y empresas que forman parte de la aplicación, podrá saber al 100% si la aplicación está teniendo éxito, o hay cosas que cambiar

## 5.2 – Definición del procedimiento para la participación de los usuarios en la evaluación del proyecto. Documentos específicos.

Con el objetivo de comprobar el correcto desempeño de la aplicación y para poder tener una visión objetiva es necesario llevar a cabo las pruebas de aceptación, que serán realizadas por los diferentes tipos de usuarios que formarán parte de la aplicación.

Con respecto a los usuarios que no se han registrado en la aplicación, se harán las siguientes comprobaciones:

- Comprobar que el contenido se muestra correctamente la información estática de la empresa
- Comprobamos que la información de los diferentes restaurantes se muestre correctamente.
- Comprobar que la visualización de la información de cada menú se muestra correctamente. Así como imágenes



- Comprobamos también el correcto funcionamiento de los enlaces en diferentes webs
- Comprobar que se puede registrar un usuario en el registro
- Comprobar que no se puede reservar sin estar registrado
- Comprobar el funcionamiento del inicio de sesión
- Comprobar que no aparecen opciones ocultas pertenecientes a otros usuarios

Con respecto al usuario que, si han iniciado sesión, comprobamos los siguientes puntos:

- Comprobar que puede iniciar sesión correctamente
- Comprobamos que los datos personales se pueden modificar
- Comprobar que se puede marcar una empresa
- Comprobar que se muestran las opciones de menú que sea correspondiente a su tipo de rol
- Comprobar que podemos ver las listas de favoritos marcadas por el usuario
- Comprobar que la ubicación nos muestra los restaurantes más cercanos

Con respecto al usuario empresa que está registrado en la aplicación, comprobamos los siguientes puntos

- Comprobar que podemos actualizar el menú de nuestra carta
- Comprobar que los pedidos de los clientes llegan correctamente
- Comprobar con la dirección es la correcta
- Comprobar que el horario de inicio de jornada, y finalización esta correctamente
- Comprobar que se muestra el menú correctamente

Por último, con respecto a los usuarios de tipo administrador, se comprobará el funcionamiento de los siguientes puntos:



- Comprobar que puede iniciar sesión y registrarse correctamente
- Comprobar que puede gestionar los usuarios registrados en la aplicación
- Comprobar que puede gestionar las empresas registradas en la aplicación
- Comprobar que se muestra las opciones de menú correspondientes

### 5.3 – Registro de Resultado

Con el objetivo de mejorar la experiencia del usuario con nuestra aplicación, se ofrecerá un correo electrónico en la cual podrá valorar la página, reportar errores que se pueda encontrar en la web. De este modo, la empresa estará al tanto de cualquier problema que aparezca, y así poder mejorar la experiencia de todos los usuarios que utilicen la aplicación web. No será necesario que un usuario este registrado en nuestra web, ya que cualquier usuario puede tener una mala experiencia con la aplicación web.

## 6.- Conclusión

Luego de este recorrido en el que se ha abordado las posibilidades de poder crear un proyecto dedicado especialmente a reservas para clientes en el área de la hostelería, comprobamos que podemos tener en cuenta que puede ser un negocio sostenible, robusto y seguro económicamente, remarcando que este proyecto no supondría grandes costes. Este proyecto que a futuro puede seguir creciendo ya que la tecnología es el punto más importante de este proyecto

Este proyecto puede tener un hueco grande en el mercado actual, teniendo en cuenta que España es un país en donde la hostelería es un punto muy fuerte, y debido a baja de empleados que hoy en día están teniendo los restaurantes, y lo difícil que se le hace para encontrar personal, la idea de que la hostelería a futuro no muy lejano comienza a trabajar de esta manera utilizando la tecnología, no es algo que pueda tardar en llegar.

### 6.1 – Opinión personal

A nivel personal, veo que adquiriré conocimientos que no imagine que podría conseguir, ya no hablo como proyecto final si no como salida profesional en un



futuro. Decidí hacer este proyecto porque trabaje en hostelería y los tiempos son muy importante, tanto como trabajador o cliente, al cliente no le gusta esperar y el trabajo quiere que el cliente sea rápido al decidir. Creo que el uso de una aplicación para armar los menús para el cliente es algo muy efectivo. Todavía hay muchos lugares que usan papel y bolígrafo, y creo que poder instalar este tipo de forma de trabajo, facilitaría el trabajo de muchas personas.

## 6.2 – Futuras Mejoras

1. El cliente pueda reservar no solamente para él, si no para más personas
2. Agregar detalles de los platos, y fotos para que el cliente pueda armar el menú en el mismo lugar, y no depender de un camarero.
3. Mejoras de accesibilidad y personalización de las vistas.
4. Indicar promociones para incentivar a los clientes
5. Crear la manera de que el cliente pueda ver el tiempo de la preparación de su menú, para que vaya al restaurante y pueda comer ni bien tome asiento

## 7.-Bibliografía y referencia

Porcentaje de restaurantes Comunidad valenciana:

<https://www.lasprovincias.es/gastronomia/restaurantes-nunca-valencia-127-decada-impulsados-turismo-20240307132433-nt.html?ref=https%3A%2F%2Fwww.lasprovincias.es%2Fgastronomia%2Frestaurantes-nunca-valencia-127-decada-impulsados-turismo-20240307132433-nt.html>

### Normativa de Cookies:

[https://www.cookiebot.com/es/?utm\\_source=google&utm\\_medium=pc&utm\\_term=&utm\\_campaign=cb\\_dm\\_ww\\_spa\\_dsa\\_search&utm\\_ad\\_group=ww-es-blog-es&gclid=CjwKCAjwjeuyBhBuEiwAJ3vuockgTgHxxHYBFj\\_Ef2O1-9k6ST1I5aul7zilx4749ZMgl8SY93i0\\_hoCL7lQAvD\\_BwE](https://www.cookiebot.com/es/?utm_source=google&utm_medium=pc&utm_term=&utm_campaign=cb_dm_ww_spa_dsa_search&utm_ad_group=ww-es-blog-es&gclid=CjwKCAjwjeuyBhBuEiwAJ3vuockgTgHxxHYBFj_Ef2O1-9k6ST1I5aul7zilx4749ZMgl8SY93i0_hoCL7lQAvD_BwE)

### Cumplimiento Legal:

<https://www.incibe.es/empresas/que-te-interesa/cumplimiento-legal>

### Análisis DAFO:

[https://www.canva.com/es\\_es/graficos/analisis-dafo/](https://www.canva.com/es_es/graficos/analisis-dafo/)

### Logotipo:

<https://monday.com/lang/es>

### Forma Jurídica (Autónomo):

[https://plataformapyme.es/es-es/creacion/formas-juridicas/Paginas/formas-juridicas-detalle.aspx?nombre=Empresario+Individual+\(Aut%C3%B3nomo\)](https://plataformapyme.es/es-es/creacion/formas-juridicas/Paginas/formas-juridicas-detalle.aspx?nombre=Empresario+Individual+(Aut%C3%B3nomo))

### Diagrama de casos de uso:

<https://www.lucidchart.com/pages/es/plantillas/diagrama-de-caso-de-uso>

### Ley de protección de datos:

<https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673>



Modelo Entidad-Relación:

<https://www.lucidchart.com/pages/es/ejemplos/herramienta-ERD>

Modelo-Vista-Controlado:

<https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>

ASP .NET Core MVC:

<https://learn.microsoft.com/es-es/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-8.0&tabs=visual-studio>