

# PUBLIC CLASS HAMBURGUESA



Daniel Fernández Valcarce.  
Antonio Feliu de Cabrera Salas.  
Jorge Martín Menéndez.

CIFP "Borja Moll". 1º de CFGS Desarrollo de Aplicaciones Web. Año 2022.

# ÍNDICE

<b>ÍNDICE.....</b>	<b>Págs. 1-2</b>
<b>1. INTRODUCCIÓN.....</b>	<b>Págs. 3-4</b>
a. Descripción del problema y solución propuesta	
b. Diagrama inicial del conjunto del proyecto	
<b>2. METODOLOGÍA.....</b>	<b>Págs. 5-8</b>
a. Fase de desarrollo de software	
i. Estudio de viabilidad del proyecto	
ii. Análisis del sistema	
iii. Diseño del sistema	
iv. Construcción del sistema	
v. Implementación y aceptación del sistema	
b. Metodología seleccionada	
c. Planning	
<b>3. ANÁLISIS.....</b>	<b>Págs. 9-16</b>
a. Partes interesadas	
b. Requisitos funcionales	
c. Requisitos no funcionales	
d. Requisitos del sistema	
e. Diagrama de casos de uso	
f. Matriz de requisitos/tecnologías	
g. Modelo entidad-relación	
<b>4. DISEÑO.....</b>	<b>Págs. 17-21</b>
a. Diagrama de clases	
b. Diagrama de secuencia	
c. Modelo relacional	
d. Diseño de la web	
<b>5. IMPLEMENTACIÓN.....</b>	<b>Págs. 22-24</b>
a. Diagrama de Gantt	
b. Herramientas Utilizadas	
c. Git	
d. Versionado semantico	
e. Diagrama de ramas	

<b>6. PRUEBAS Y RESULTADOS.....</b>	<b>Págs. 25-26</b>
a. Tipo de pruebas y resultados	
<b>7. GESTIÓN TEMPORAL.....</b>	<b>Págs. 27-32</b>
a. Herramientas	
b. Tiempo previsto	
c. Tiempo real	
d. Comparación tiempo real - previsto	
e. Comparación tiempo real - previsto por fases	
f. Gestión temporal por persona - Clockify	
<b>8. CONCLUSIONES.....</b>	<b>Págs. 33</b>
<b>9. ANEXOS.....</b>	<b>Págs. 34-35</b>
a. Croquis de la web	
b. Diagrama de actividades	
c. Diagrama de flujo	

# **1 INTRODUCCIÓN**

## **1. a. DESCRIPCIÓN DEL PROBLEMA Y SOLUCIÓN PROPUESTA AL MISMO**

La empresa dedicada a la comercialización y venta de comida rápida Public Hamburguesa con sede en Palma de Mallorca, la cual lleva en el negocio de la alimentación desde 1990 ha contactado con la empresa tecnológica Class dirigida por Daniel, Antonio y Jorge. El cliente considera que su empresa se está quedando obsoleta con respecto a la competencia, sobretodo en el ámbito tecnológico, no disponen de teléfono de contacto ni reservas, lo cual hace que entre las 20:00 y las 22:00 el local quede saturado, los pedidos tardan en salir y los camareros y recepcionistas no den a basto, provocando varias quejas de los clientes del establecimiento por tardanzas a la hora de pedir.

La empresa tecnológica Class le propone la solución de crear una página web llamada Public Class Hamburguesa la cual permitirá a los clientes realizar pedidos, para tomar en el local y recoger, ver un tiempo aproximado de espera, su número de pedido, obtener una factura digital reducida en la cual se pueda observar los productos comprados, el descuento aplicable a ellos y el importe pagado.

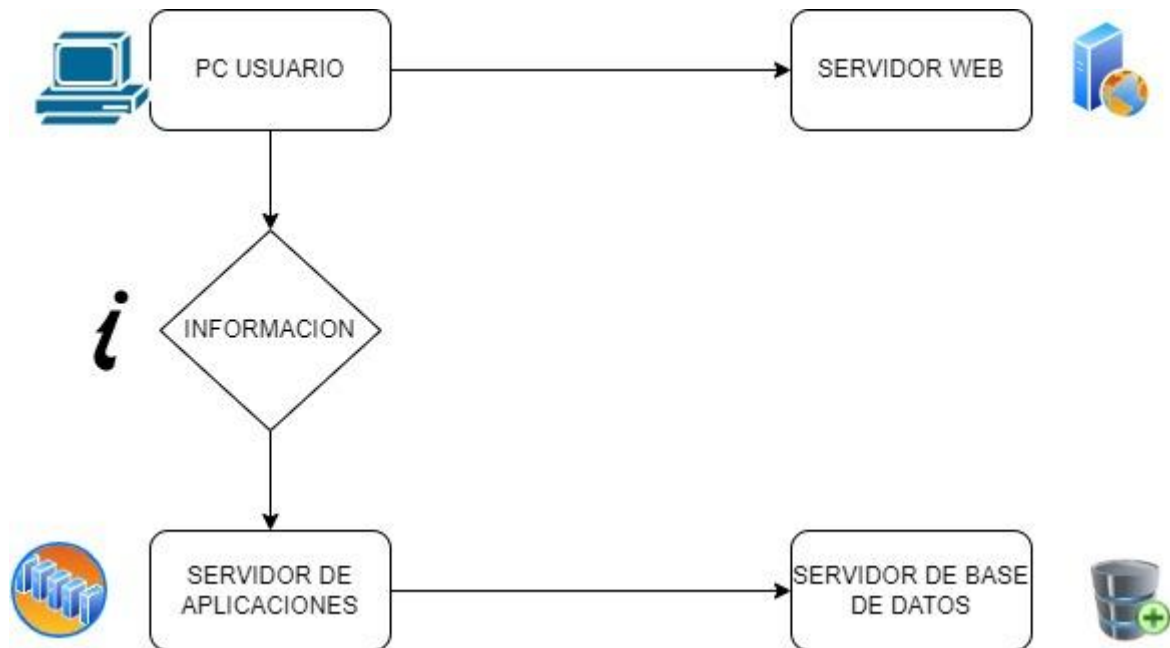
La empresa Class implementará en la empresa Public Hamburguesa, varios paneles interactivos conectados con el dominio de [www.PublicClassHamburguesa.com](http://www.PublicClassHamburguesa.com) para que los clientes puedan realizar los pedidos en el local, o realizarlos desde cualquier dispositivo con conexión a internet antes de llegar al local.

La página Public Class Hamburguesa, le permitirá a la empresa Public Hamburguesa conocer los pedidos realizados, cobrar a los clientes, crear un orden de pedidos en la cocina, establecer un tiempo medio para realizarlos, así como saber las cantidades remanentes de productos que quedan en el almacén sabiendo el número de pedidos realizados y alertar al administrador de la escasez de productos.

Con todo esto se pretende agilizar el proceso de pedido y producción de productos, establecer unos tiempos medios de producción que el personal deberá aproximarse, gestionar el almacén, haciendo con todo esto más fácil de gestionar el negocio para el administrador / gerente.

## 1. b.DIAGRAMA DEL CONJUNTO DEL PROYECTO

El diagrama del conjunto del proyecto



Resumidamente:



## **2 METODOLOGÍA**

### **2. a. FASES DEL DESARROLLO DEL SOFTWARE**

El Proceso para el desarrollo de software, es una estructura aplicada al desarrollo de un producto de software. Hay varios modelos a seguir para el establecimiento de un proceso para el desarrollo de software, cada uno de los cuales describe un enfoque diferente para diferentes actividades que tienen lugar durante el proceso.

Una metodología de desarrollo de software se refiere a un marco de trabajo, que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información.

#### **2. a. I. ESTUDIO DE VIABILIDAD DEL PROGRAMA**

El Estudio de Viabilidad, tiene por objeto comprobar si en un proyecto existen soluciones que cumplan los objetivos definidos en el planteamiento inicial y determinar que son viables física, legal, social, económica y financieramente.

En esta fase del proyecto se persigue obtener una visión de conjunto del proyecto, siendo, por tanto, sus objetivos fundamentales los siguientes:

- Conocer el proyecto en su conjunto.
- Obtener posibles soluciones.
- Estudiar la viabilidad de estas soluciones.

Mediante el estudio de viabilidad realizado, se han obtenido los siguientes resultados:

1. Ante el problema planteado, existen varias soluciones tecnológicas aplicables por la empresa Class.
2. Los objetivos a cumplir en el proyecto son los siguientes:
  - a. Implementación de tecnologías informáticas en la empresa Public Hamburguesa
  - b. Las tecnologías deben permitir al cliente un acceso fácil e interactivo a los productos ofrecidos.
  - c. Las tecnologías implementadas deben controlar el proceso de realización del producto, informando al cliente de en qué estado se encuentra, también se debe de marcar unos tiempos aproximados de realización del producto para los empleados.
  - d. La tecnología debe controlar el estado de el almacén haciendo un conteo entre los productos remanentes y los vendidos, y avisar a el administrador en caso de que haya prevista una escasez de productos
3. Ante las posibles opciones de solución encontradas se considera que la que tiene una mayor probabilidad de éxito a la hora de dar solución es la creación de un dominio World Wide Web que contenga los productos ofrecidos en Public Class así como la instalación de varios paneles en el establecimiento con acceso a el dominio.

## ANÁLISIS DEL SISTEMA

El propósito de este proceso es conseguir la especificación detallada del sistema de información, por medio de un catálogo de requisitos y de una serie de modelos que cubran las necesidades de información de los usuarios para los que se desarrollará el sistema de información y que serán la entrada para el proceso de Diseño del sistema de información.

## DISEÑO DEL SISTEMA

El propósito del diseño es obtener la definición de la arquitectura del sistema y del entorno tecnológico que le apoyará, junto con la especificación detallada de los componentes del sistema de información. A partir de esta información, se generan todas las especificaciones de construcción relativas al propio sistema, así como la especificación técnica del plan de pruebas, la definición los requisitos de implantación y el diseño de los procedimientos de migración y carga inicial.

## CONSTRUCCIÓN DEL SISTEMA

La construcción del sistema de información tiene como objetivo final la construcción y la prueba de los diferentes componentes del sistema de información, a partir de su conjunto de especificaciones lógicas y físicas, obtenido en la fase de diseño. Se desarrollan los procedimientos de operación y de seguridad, y se elaboran los manuales de usuario final y de explotación, estos últimos cuando proceda.

## IMPLEMENTACIÓN Y ACEPTACIÓN DEL SISTEMA

Este proceso tiene como objetivo principal la entrega y la aceptación del sistema en su totalidad, que puede comprender varios sistemas de información desarrollados de manera independiente, y un segundo objetivo, que es llevar a cabo las actividades oportunas para el paso a producción del sistema.

## METODOLOGÍA SELECCIONADA

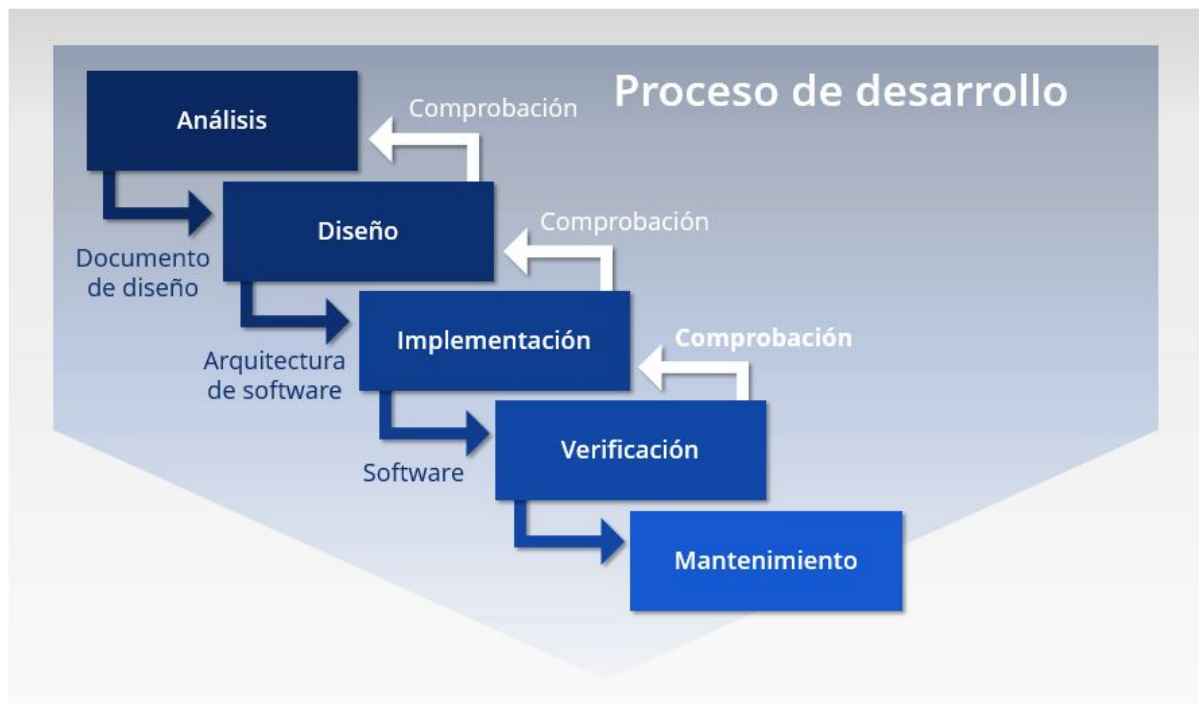
Durante la realización de este proyecto se ha hecho uso de la metodología en Cascada, que debido a que es nuestro primer proyecto, nos ha permitido saber si el proyecto está listo para avanzar a la siguiente fase del proyecto gracias a su fase de comprobación.

Las ventajas se presentan en esta metodología son:

- Es un modelo fácil e intuitivo de implementar y entender.
- Realiza un buen funcionamiento en equipos débiles y productos maduros, por lo que se requiere de menos capital y herramientas para hacerlo funcionar de manera óptima.
- Es un modelo conocido y utilizado con frecuencia.
- Promueve una metodología de trabajo efectiva: Definir antes que diseñar, diseñar antes que codificar.

Las principales desventajas que se presentan en esta metodología son:

- Cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo.
- Una etapa determinada del proyecto no se puede llevar a cabo a menos de que se haya culminado la etapa anterior.
- El proceso de creación del software tarda mucho tiempo ya que debe pasar por el proceso de prueba y hasta que el software no esté completo no se opera.





## PLANNING

DIAGRAMA DE GANTT													
PUBLIC CLASS HAMBURGUESA													
Nº	FASE DEL PROYECTO	INICIO	FINAL	4/04/2022	11/04/2022	18/04/2022	25/04/2022	02/05/2022	09/05/2022	16/05/2022	23/05/2022	30/05/2022	06/06/2022
				SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 5	SEMANA 6	SEMANA 7	SEMANA 8	SEMANA 9	SEMANA 10
1	PROPUESTA	04/04/2022	07/04/2022										
2	ANALISIS	07/04/2022	15/04/2022										
3	DISEÑO	15/04/2022	10/05/2022										
4	IMPLEMENTACIÓN	11/05/2022	27/05/2022										
5	PRUEBAS	28/05/2022	02/06/2022										
6	RESULTADOS	03/06/2022	06/06/2022										

DIAGRAMA DE GANTT					
PUBLIC CLASS HAMBURGUESA					
FASE DEL PROYECTO	FECHA PREVISTA INICIO	DÍAS TRABAJADOS	FECHA FINAL PREVISTA	SITUACIÓN	DÍAS PARA EL FINAL
PROPUESTA	04/04/2022	5	07/04/2022	FINALIZADO	0
ANALISIS	07/04/2022	15	15/04/2022	FINALIZADO	0
DISEÑO	15/04/2022	25	10/05/2022	FINALIZADO	0
IMPLEMENTACIÓN	11/05/2022	16	27/05/2022	FINALIZADO	0
PRUEBAS	28/05/2022	5	02/06/2022	FINALIZADO	0
RESULTADOS	03/06/2022	5	08/06/2022	EN CURSO	1

REALIZADO
EN PROCESO
SIN HACER

## **3 ANÁLISIS**

### **PARTES INTERESADAS**

Las partes interesadas son las siguientes:

- Empresa de comida rápida: Public Hamburguesa
- Empresa de desarrollo software: Class
- Los clientes de la empresa Public Hamburguesa

### **REQUISITOS FUNCIONALES**

A la hora de realizar el proyecto ciertos requisitos funcionales deberán ser cumplidos:

- CLIENTE\_RF01: El usuario debe poder iniciar un pedido
- CLIENTE\_RF02: El usuario debe poder añadir uno o varios productos a su pedido.
- CLIENTE\_RF03: El usuario debe poder cancelar su pedido siempre y cuando no lo haya finalizado.
- CLIENTE\_RF04: El usuario debe poder confirmar y finalizar su pedido.
- EMPLEADO\_RF01: El empleado debe poder visualizar los pedidos en cola.
- EMPLEADO\_RF02: El empleado debe poder marcar un pedido como “listo”.

### **REQUISITOS NO FUNCIONALES**

Los requisitos no funcionales son los siguientes:

- RnF01: La aplicación debe ofrecer ofertas al usuario.
- RnF02: La aplicación debe ser confiable.
- RnF03: Los usuarios deben indicar si son mayores de 18 años.
- RnF04: En caso de no ser mayores de 18 años los productos de venta que contengan alcohol deben desaparecer de la aplicación.
- RnF05: La aplicación debe hacer uso de la tecnología de código abierto.

## REQUISITOS DEL SISTEMA

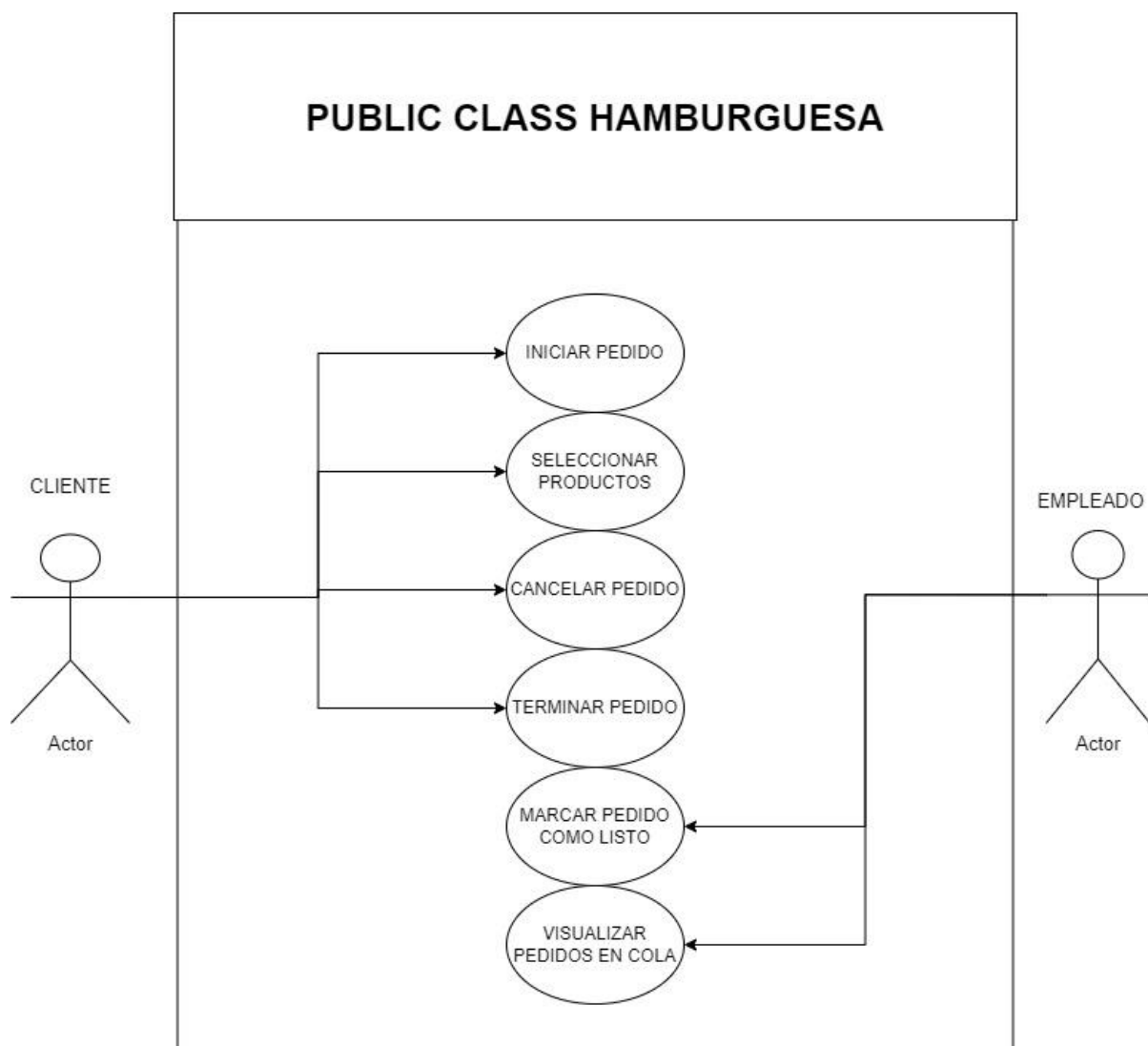
Los requisitos del sistema son los siguientes:

- El acceso informativo a la aplicación debe de estar disponible las 24 horas.
- Los pedidos solo deberán poder ser realizados en horario laboral.
- La implementación debe realizarse desde un dispositivo con conexión a internet o desde los paneles disponibles en el establecimiento.

## DIAGRAMA DE CASOS DE USO

Un caso de uso es un comportamiento ejecutante del sujeto. Un caso de uso se representa mediante un verbo que indica una acción, operación o tarea concreta. Esta operación se activará a partir de una orden dada por un agente externo al caso de uso. Podrá ser un actor que haga una petición y genere el caso de uso, u otro caso de uso que le invoque.

El diagrama de casos de uso es el que se presenta en la imagen a continuación:



## DOCUMENTACIÓN DETALLADA DE REQUISITOS FUNCIONALES

En este apartado veremos los casos de uso según los requisitos funcionales anteriormente nombrados, en detalle:

IDENTIFICADOR	CLIENTE_RF01	
NOMBRE	El usuario debe poder iniciar un pedido	
DESCRIPCIÓN	El cliente que lo desee, debe poder iniciar un pedido a través de la página web	
ACTORES	Clientes, Class	
PRECONDICIONES	El cliente debe tener acceso a internet y acceder a la url	
SECUENCIA NORMAL	nº	ACCION
	1	El cliente accede al sitio web
	2	El cliente empieza a seleccionar productos para su carrito
	3	-
SECUENCIA ALTERNATIVA	nº	ACCION
	1	-
	2	-
	3	-
POSTCONDICIONES	El cliente ha seleccionado productos para el carrito de la compra	
NOTAS	-	

IDENTIFICADOR	CLIENTE_RF02	
NOMBRE	Usuario añade productos a su pedido	
DESCRIPCIÓN	El usuario debe poder añadir uno o varios productos a su pedido	
ACTORES	Cliente	
PRECONDICIONES	Estar con una sesión iniciada	
SECUENCIA NORMAL	nº	ACCION
	1	Elige producto/s
	2	Confirma que los añade
	3	Terminal el pedido para que llegue al empleado
SECUENCIA ALTERNATIVA	nº	ACCION
	1	-
	2	-
	3	-
POSTCONDICIONES	Se muestra la comanda con sus datos en texto	
NOTAS	-	

IDENTIFICADOR	CLIENTE_RF03	
NOMBRE	Cancelar pedido	
DESCRIPCIÓN	El usuario debe poder cancelar un pedido antes de haberlo finalizado	
ACTORES	Cliente, Class	
PRECONDICIONES	No haber finalizado el pedido aún.	
SECUENCIA NORMAL	nº	ACCION
	1	El usuario selecciona que desea cancelar el pedido
	2	El usuario confirma que desea cancelar el pedido
	3	-
SECUENCIA ALTERNATIVA	nº	ACCION
	1	CLIENTE_RF04
	2	-
	3	-
POSTCONDICIONES	La cesta de la compra está vacía y no hay productos para tramitar	
NOTAS	Vease documento CLIENTE_RF04 para acciones alternativas	

IDENTIFICADOR	<b>CLIENTE_RF04</b>	
NOMBRE	El usuario debe poder confirmar y finalizar su pedido	
DESCRIPCIÓN	Una vez el usuario ha decidido los productos que quiere debe poder finalizar el pedido	
ACTORES	Cliente, Class, Public Hamburguesa	
PRECONDICIONES	Haber seleccionado los productos que desea	
SECUENCIA NORMAL	nº	<b>ACCION</b>
	1	El usuario decide que ya no quiere mas productos
	2	El usuario finaliza el pedido y lo tramita
	3	-
SECUENCIA ALTERNATIVA	nº	<b>ACCION</b>
	1	CLIENTE_RF03
	2	-
	3	-
POSTCONDICIONES	El pedido ha sido finalizado y la comanda enviada a el restaurante	
NOTAS	Vease documento CLIENTE_RF03 para acciones alternativas	

IDENTIFICADOR	<b>EMPLEADO_RF01</b>	
NOMBRE	El empleado debe visualizar los pedidos en cola	
DESCRIPCIÓN	El empleado que lo desee debe de visualizar los pedidos que los clientes han realizado y todavia no han salido de la cocina	
ACTORES	Cliente, Class, Public Hamburguesa	
PRECONDICIONES	EL cliente debe haber realizado un pedido con exito	
SECUENCIA NORMAL	nº	<b>ACCION</b>
	1	El empleado selecciona "Ver los pedidos en cola"
	2	-
	3	-
SECUENCIA ALTERNATIVA	nº	<b>ACCION</b>
	1	-
	2	-
	3	-
POSTCONDICIONES	El empleado ha visto los pedidos que están en cola	
NOTAS	-	

IDENTIFICADOR	<b>EMPLEADO_RF02</b>	
NOMBRE	El empleado debe poder marcar un pedido como listo	
DESCRIPCIÓN	El empleado al ver que el pedido ha salido de la cocina y ha sido entregado a el cliente debe marcar el pedido como "listo"	
ACTORES	Cliente, Class, Public Hamburguesa	
PRECONDICIONES	El pedido debe haber sido cocinado y preparado en su totalidad	
SECUENCIA NORMAL	nº	<b>ACCION</b>
	1	El pedido sale de la cocina preparado en su totalidad
	2	El pedido se le entrega al cliente
	3	-
SECUENCIA ALTERNATIVA	nº	<b>ACCION</b>
	1	-
	2	-
	3	-
POSTCONDICIONES	El pedido ha sido entregado al cliente	
NOTAS	-	

## LISTADO DE TECNOLOGÍAS

- **JAVA:** es un lenguaje de programación y plataforma informática desarrollado originalmente por James Gosling, de Sun Microsystems y adquirida por ORACLE en 2010, está orientado a objetos y el diseño de Java, su robustez, el respaldo de la industria y su fácil portabilidad han hecho de Java uno de los lenguajes con un mayor crecimiento y amplitud de uso en distintos ámbitos de la industria de la informática.
- **JAVASCRIPT (JS):** es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,<sup>2</sup> basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.
- **HTML:** siglas en inglés de HyperText Markup Language, en castellano ‘lenguaje de marcado de hipertexto’, hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C).
- **CSS:** siglas en inglés de Cascading Style Sheets, en castellano ‘Hojas de estilo en cascada’, es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS... Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y para muchas aplicaciones móviles.
- **MySQL:** es un sistema de gestión de bases de datos relacional por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo,<sup>12</sup> y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

## MATRIZ REQUISITOS/TECNOLOGIAS

### MATRIZ REQUISITOS

MATRIZ REQUISITOS		
PUBLIC CLASS HAMBURGUESA		
nº	ID	DESCRIPCION
1	CLIENTE_RF01	El usuario debe poder iniciar un pedido
2	CLIENTE_RF02	El usuario debe poder añadir uno o varios productos a su pedido.
3	CLIENTE_RF03	El usuario debe poder cancelar su pedido siempre y cuando no lo haya finalizado.
4	CLIENTE_RF04	El usuario debe poder confirmar y finalizar su pedido
5	EMPLEADO_RF01	El empleado debe poder visualizar los pedidos en cola
6	EMLEADO_RF01	El empleado debe poder marcar un pedido como "listo"

### MATRIZ TECNOLOGÍAS

Para la realización de este proyecto hemos utilizado las siguientes tecnologías:

- Lógica, operaciones y procesamiento en el backend: Java
- Recogida, transferencia y procesamiento de datos en el frontend: Javascript
- Presentación del sitio web en el front end: html y css
- Almacenamiento de datos: MySQL

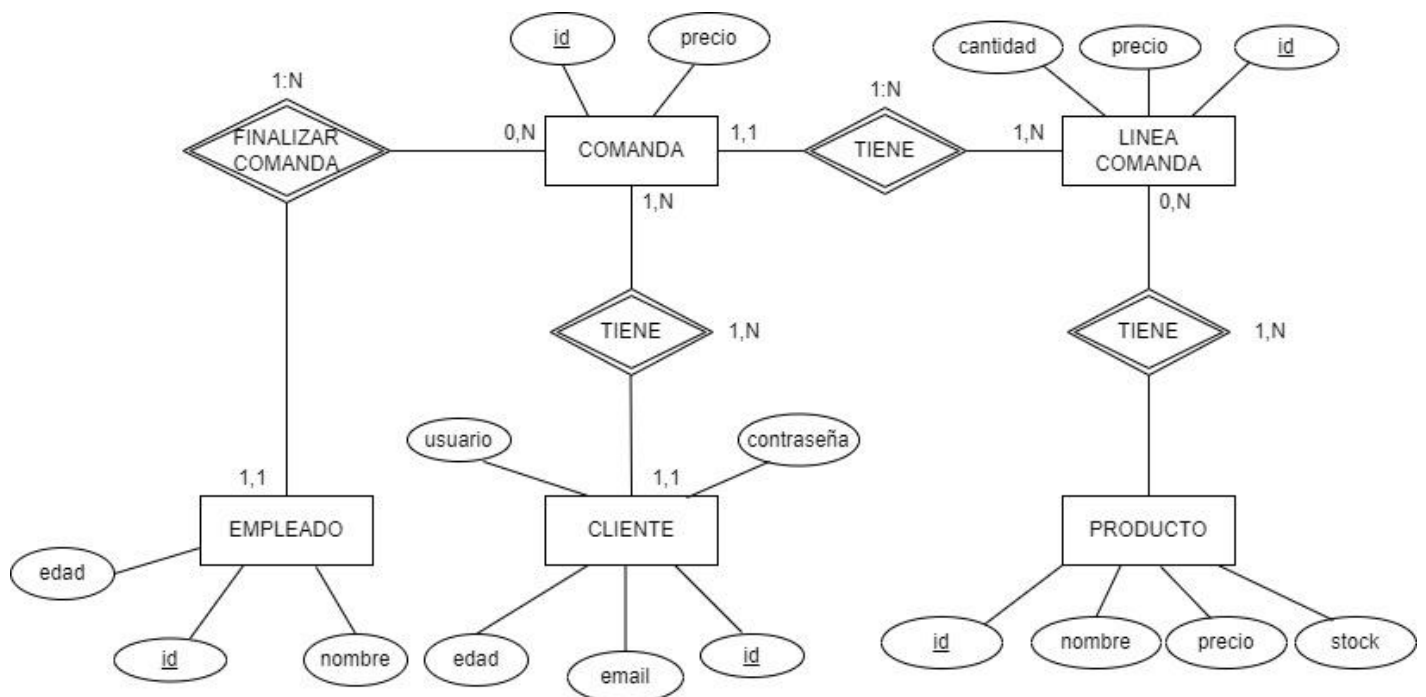
MATRIZ TECNOLOGIAS	
PUBLIC CLASS HAMBURGUESA	
nº	TECNOLOGIA
1	JAVA
2	JAVASCRIPT
3	HTML
4	CSS
5	MYSQL

## MATRIZ REQUISITOS/TECNOLOGIAS

En esta matriz se van a asociar los requisitos planteados con las tecnologías necesarias para su realización.

MATRIZ REQUISITOS / TECNOLOGIAS		
PUBLIC CLASS HAMBURGUESA		
nº	ID REQUISITO	TECNOLOGIA
1	CLIENTE_RF01	JAVA,JAVASCRIPT, HTML, CSS, MYSQL
2	CLIENTE_RF02	JAVA,JAVASCRIPT, HTML, CSS, MYSQL
3	CLIENTE_RF03	JAVA,JAVASCRIPT, HTML, CSS, MYSQL
4	CLIENTE_RF04	JAVA,JAVASCRIPT, HTML, CSS, MYSQL
5	EMPLEADO_RF01	JAVA,JAVASCRIPT, HTML, CSS, MYSQL
6	EMLEADO_RF01	JAVA,JAVASCRIPT, HTML, CSS, MYSQL

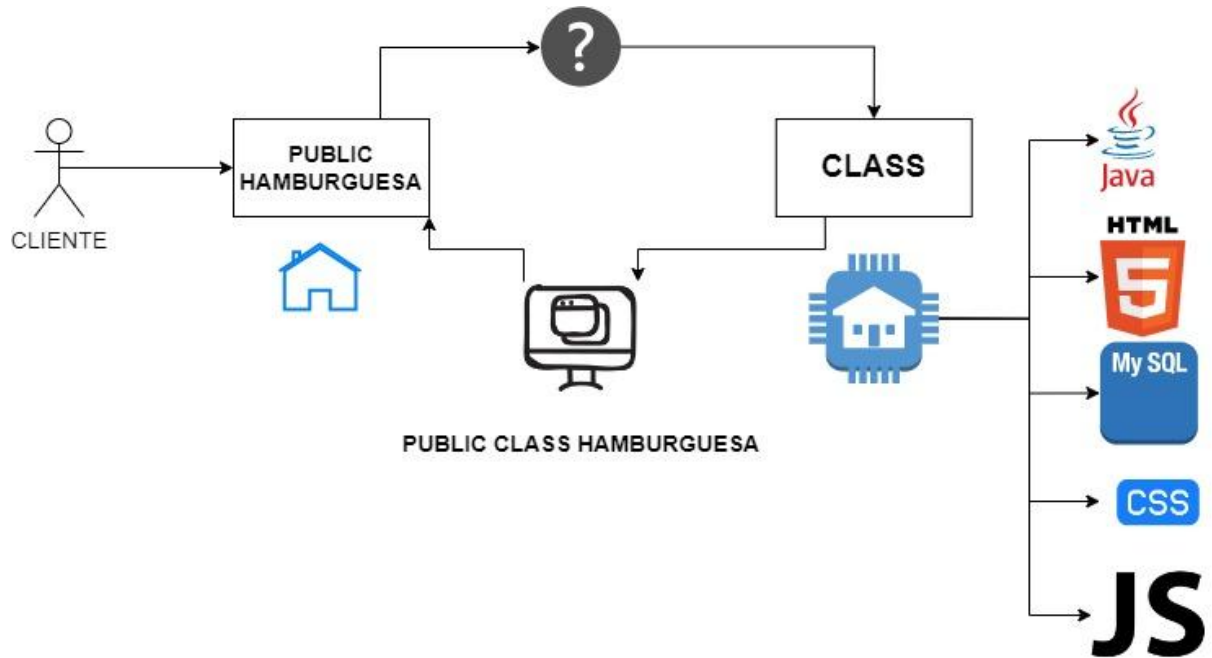
## MODELO ENTIDAD RELACIÓN



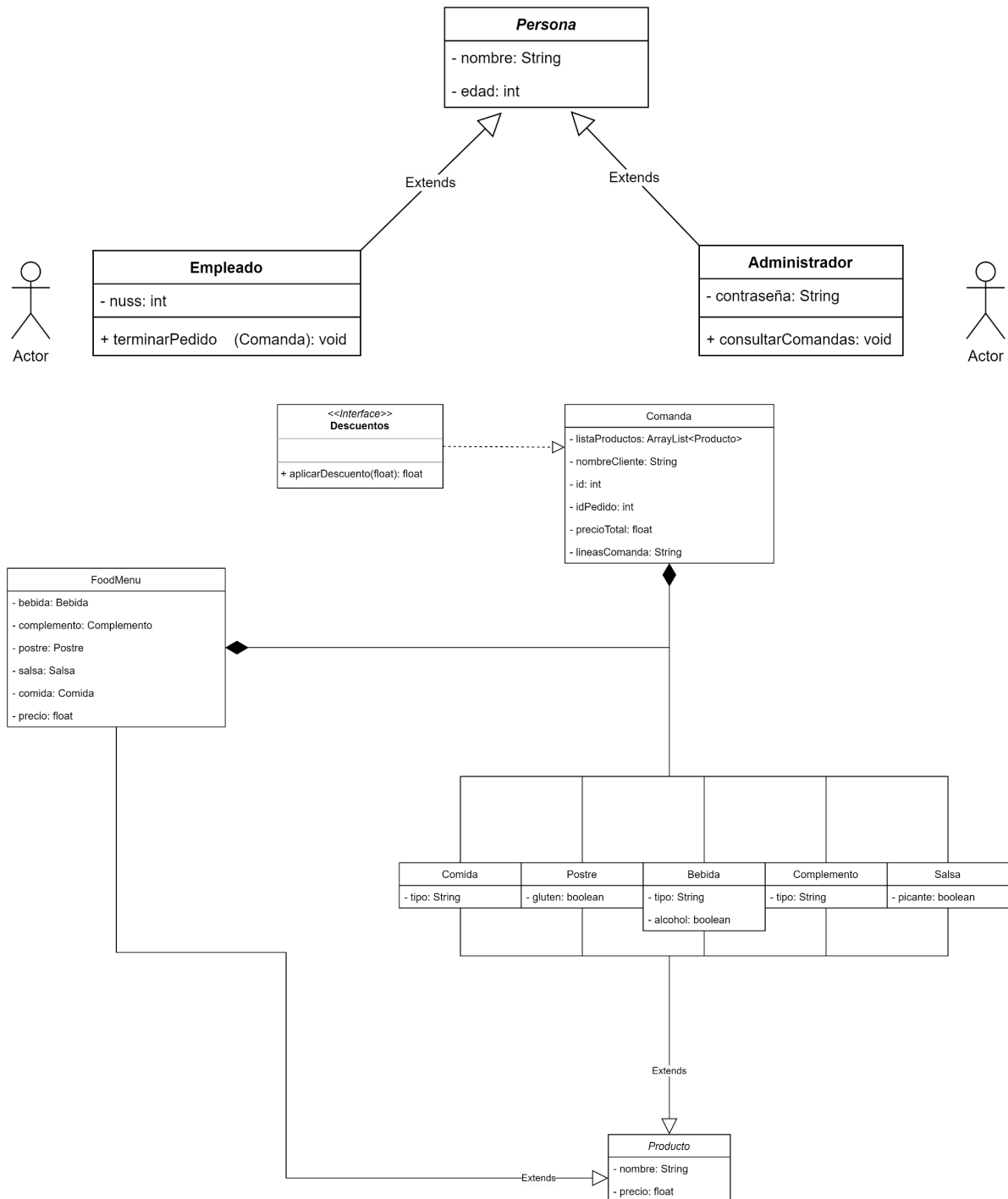


## 4 DISEÑO

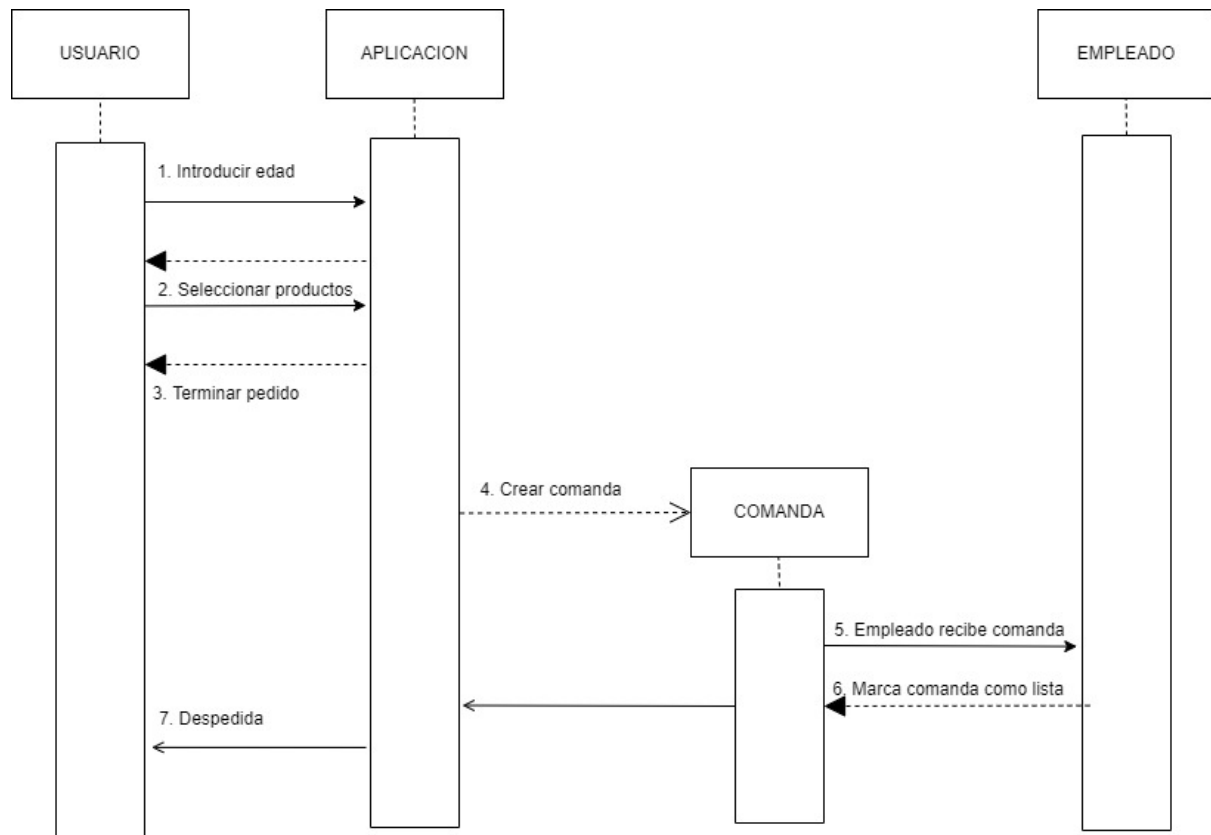
### DIAGRAMA DE DISEÑO GENERAL



# DIAGRAMA DE CLASES



## DIAGRAMA DE SECUENCIA



## MODELO RELACIONAL

- empleado(id, nombre, edad);  
EMPLEADO\_PK(id);
- cliente(id, edad, email, usuario, contraseña);  
CLIENTE\_PK(id);
- comanda(id, idEmpleado, idCliente precio);  
COMANDA\_PK(id, idEmpleado, idCliente);  
COMANDA\_FK(idEmpleado) REFERENCES (empleado.id);  
COMANDA\_FK(idCliente) REFERENCES (cliente.id);
- producto(id, precio, nombre, stock);  
COMANDA\_PK(id);
- lineaComanda(id, idComanda, idProducto precio, cantidad);

COMANDA\_PK(id, idComanda, idProducto);  
COMANDA\_FK(idComanda) REFERENCES (cliente.id);

## NORMALIZACIÓN

### Tercera forma normal (3FN):

empleado(id, nombre, edad);

EMPLEADO\_PK(id);

cliente(id, edad, email, usuario, contraseña);

CLIENTE\_PK(id);

comanda(id, idEmpleado, idCliente precio);

COMANDA\_PK(id, idEmpleado, idCliente);

COMANDA\_FK(idEmpleado) REFERENCES (empleado.id);

COMANDA\_FK(idCliente) REFERENCES (cliente.id);

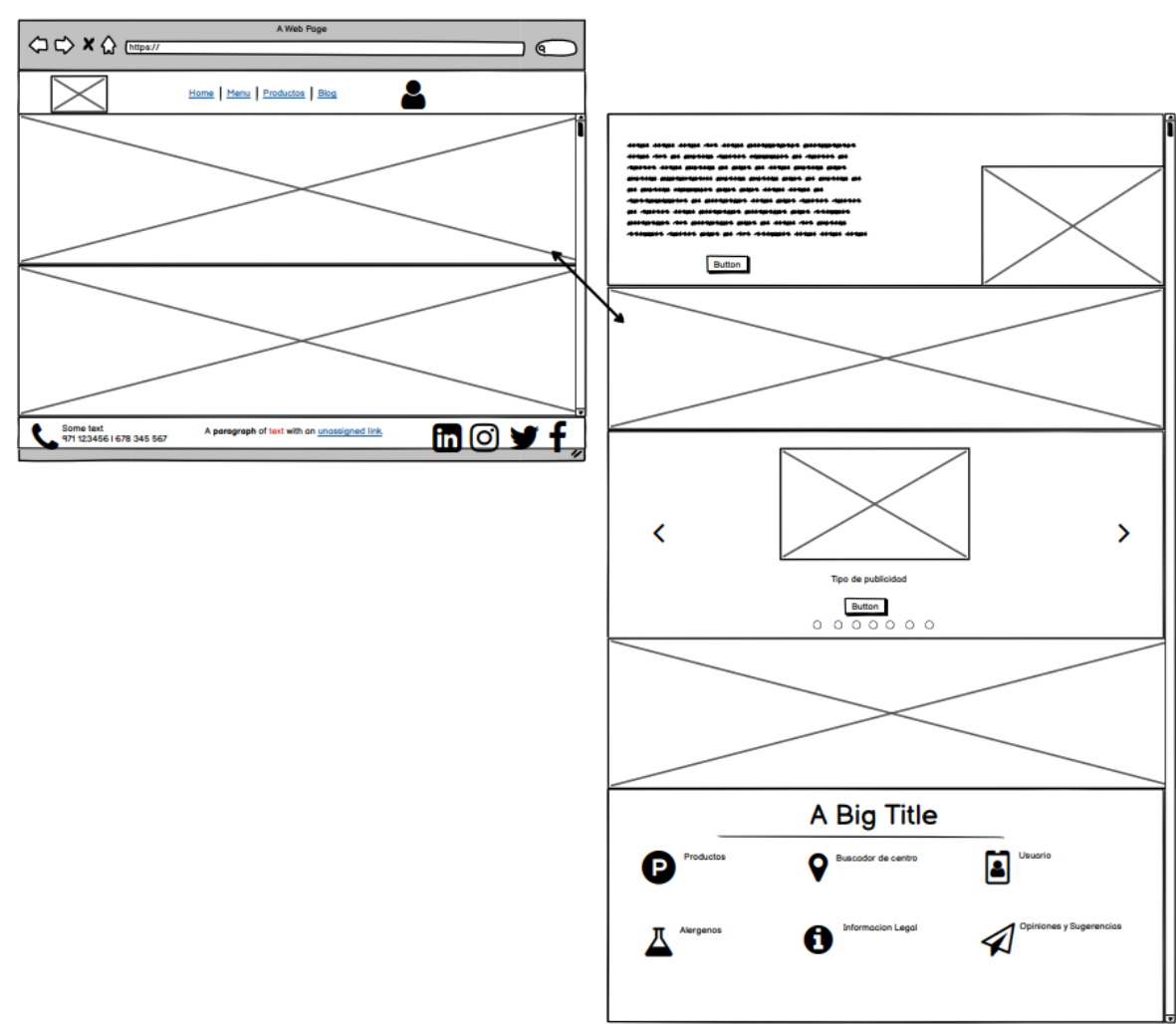
producto(id, precio, nombre, stock);

COMANDA\_PK(id);

lineaComanda(id, idComanda, idProducto precio, cantidad);

COMANDA\_PK(id, idComanda, idProducto);

# CROQUIS DE LA WEB



## 5 IMPLEMENTACIÓN

### DIAGRAMA DE GANTT

DIAGRAMA DE GANTT													
PUBLIC CLASS HAMBURGUESA													
Nº	FASE DEL PROYECTO	INICIO	FINAL	4/04/2022	11/04/2022	18/04/2022	25/04/2022	02/05/2022	09/05/2022	16/05/2022	23/05/2022	30/05/2022	06/06/2022
				SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 5	SEMANA 6	SEMANA 7	SEMANA 8	SEMANA 9	SEMANA 10
1	PROPUESTA	04/04/2022	07/04/2022										
2	ANÁLISIS	07/04/2022	15/04/2022										
3	DISEÑO	15/04/2022	10/05/2022										
4	IMPLEMENTACIÓN	11/05/2022	27/05/2022										
5	PRUEBAS	28/05/2022	02/06/2022										
6	RESULTADOS	03/06/2022	06/06/2022										

DIAGRAMA DE GANTT						
PUBLIC CLASS HAMBURGUESA						
FASE DEL PROYECTO	FECHA PREVISTA INICIO	DÍAS TRABAJADOS	FECHA FINAL PREVISTA	SITUACIÓN	DÍAS PARA EL FINAL	
PROPUESTA	04/04/2022	5	07/04/2022	FINALIZADO	0	
ANÁLISIS	07/04/2022	15	15/04/2022	FINALIZADO	0	
DISEÑO	15/04/2022	25	10/05/2022	FINALIZADO	0	
IMPLEMENTACIÓN	11/05/2022	16	27/05/2022	FINALIZADO	0	
PRUEBAS	28/05/2022	5	02/06/2022	FINALIZADO	0	
RESULTADOS	03/06/2022	5	08/06/2022	EN CURSO	1	

### HERRAMIENTAS UTILIZADAS

- **JAVA:** es un lenguaje de programación y plataforma informática desarrollado originalmente por James Gosling, de Sun Microsystems y adquirida por ORACLE en 2010, está orientado a objetos y el diseño de Java, su robustez, el respaldo de la industria y su fácil portabilidad han hecho de Java uno de los lenguajes con un mayor crecimiento y amplitud de uso en distintos ámbitos de la industria de la informática. En este proyecto la herramienta utilizada para el manejo del código en java es **Eclipse**: es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.
- **JAVASCRIPT (JS):** es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,2 basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo. En este proyecto la herramienta utilizada para el manejo de código en javascript es **Visual Studio Code**: es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para

la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

- **HTML:** siglas en inglés de HyperText Markup Language, en castellano ‘lenguaje de marcado de hipertexto’, hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C). En este proyecto la herramienta utilizada para el manejo de código en HTML es **Visual Studio Code**: es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.
- **CSS:** siglas en inglés de Cascading Style Sheets, en castellano ‘Hojas de estilo en cascada’, es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS... Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y para muchas aplicaciones móviles. En este proyecto la herramienta utilizada para el manejo de código en CSS es **Visual Studio Code**: es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.
- **MySQL:** es un sistema de gestión de bases de datos relacional por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo,<sup>12</sup> y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web. En este proyecto la herramienta utilizada para el manejo de código en MySQL es MySQL Workbench: es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL.

## GIT

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

## VERSIONADO SEMANTICO

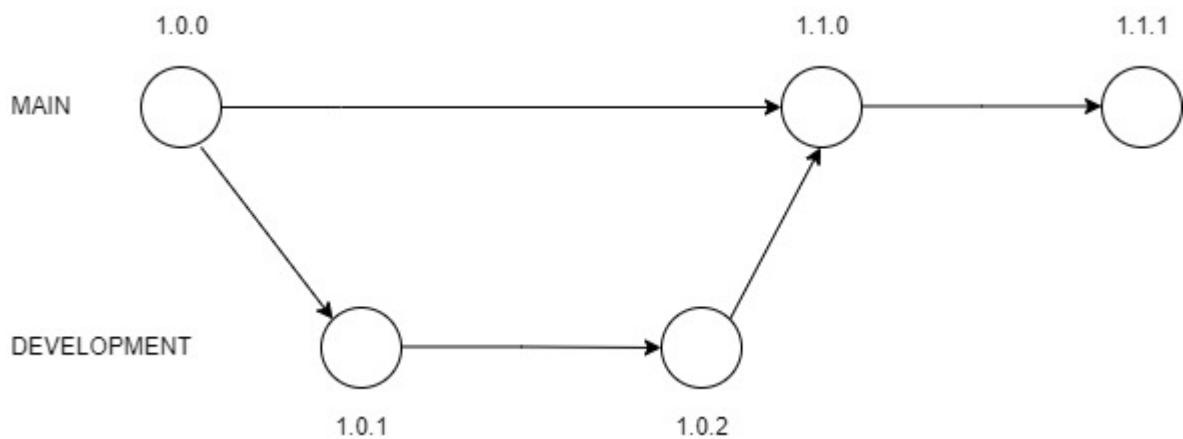
Dado un número de versión MAYOR.MENOR.PARCHE, se incrementa:

- La versión MAYOR cuando realizas un cambio incompatible en el API,
- La versión MENOR cuando añades funcionalidad que compatible con versiones anteriores, y
- La versión PARCHE cuando reparas errores compatibles con versiones anteriores.

## DIAGRAMA DE RAMAS

Las ramas utilizadas para el desarrollo de la aplicación han sido:

- **MAIN:** para versiones avanzadas o grandes cambios en la aplicación.
- **DEVELOPMENT:** para parches sobre ella





## **6 PRUEBAS Y RESULTADOS**

En cada una de las fases del ciclo de vida de un proyecto, será necesario que el trabajo llevado a término sea validado y verificado.

El objetivo de las pruebas es la evaluación de la calidad del software desarrollado durante todo su ciclo de vida, validando que hace lo que debe hacer y que lo hace tal y como se diseñó, a partir de los requerimientos.

### **TIPO DE PRUEBAS Y RESULTADOS**

El tipo de pruebas que se van a llevar a cabo en esta fase son las siguientes:

#### **PRUEBAS UNITARIAS**

Tienen como objetivo la detección de errores en los datos, en los algoritmos y en la lógica de éstos. El método utilizado en este tipo de pruebas es el de la caja blanca o el de caja negra.

Para la cual hemos introducido distintos valores en el carrito del pedido, siempre obteniendo el producto que hemos seleccionado dentro del mismo

#### **PRUEBAS DE INTEGRACIÓN**

Se encargan de detectar errores de las interfaces y en las relaciones entre los componentes.

Para la cual hemos navegado por todas las interfaces del sitio web siempre llegando a la página deseada

#### **PRUEBAS DE CARGA**

Se comprueba el rendimiento y la integridad de la aplicación ya terminada con datos reales y en un entorno que también simula el entorno real.

Para la cual se hacen cargas de distintos usuarios y distintos pedidos, obteniendo un resultado satisfactorio.

## PRUEBAS DE ACEPTACIÓN

Su objetivo es la validación o aceptación de la aplicación por parte de los usuarios. pruebas alfa y las pruebas beta.

En este tipo de pruebas nos hemos reunido con el cliente, en nuestro caso, los profesores periódicamente, para acercarnos lo más posible a el resultado final satisfactorio, finalmente hemos dado a probar la aplicación a nuestros profesores, por su cuenta.

## PRUEBAS DE SISTEMA

Su objetivo es comprobar que la integración es correcta.

Para las cuales se despliega el proyecto en red, a través de docker y docker-compose, con un resultado satisfactorio.

## 7 GESTIÓN TEMPORAL

### HERRAMIENTAS

La herramienta que hemos usado para la gestión temporal se llama Clockify.

Clockify es la aplicación gratuita más popular de seguimiento de tiempo y para equipos de trabajo de todos los tamaños. A diferencia de todos los demás rastreadores de tiempo, Clockify está disponible para un número ilimitado de usuarios de forma gratuita.

Clockify es una aplicación que funciona en un navegador, pero también puede instalarse en su computadora o teléfono para mayor comodidad.

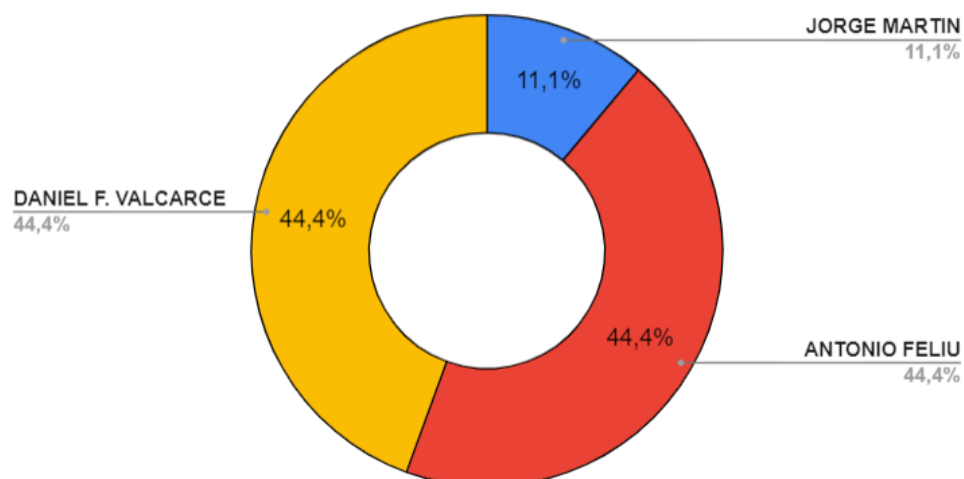
Clockify es utilizado por todos, desde autónomos, pequeñas empresas y agencias de consultoría, hasta instituciones gubernamentales, ONG, universidades y empresas Fortune 500.

### TIEMPO PREVISTO

El tiempo previsto para la realización del proyecto son 16 horas por semana, por persona, lo que en total hacen 576 horas en total. la distribución teórica inicial que se llevó a cabo fue la siguiente:

#### PUBLIC CLASS HAMBURGUESA

##### GESTION TEMPORAL HORAS

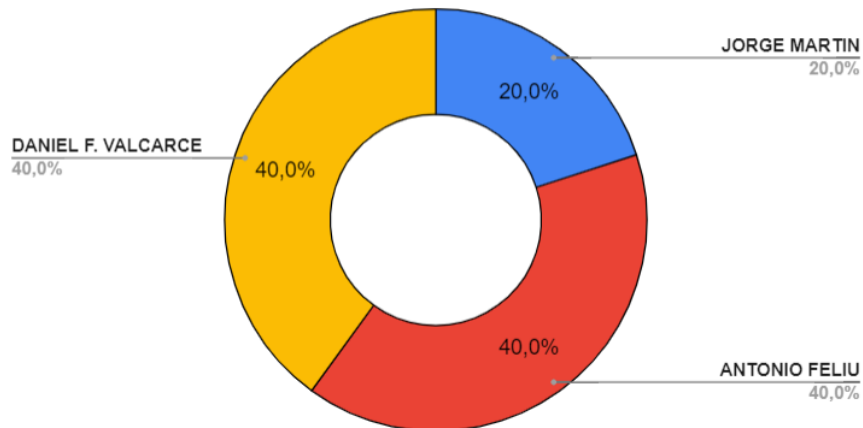


## TIEMPO REAL

La distribución del tiempo final utilizada para la realización del proyecto ha sido la siguiente:

### PUBLIC CLASS HAMBURGUESA

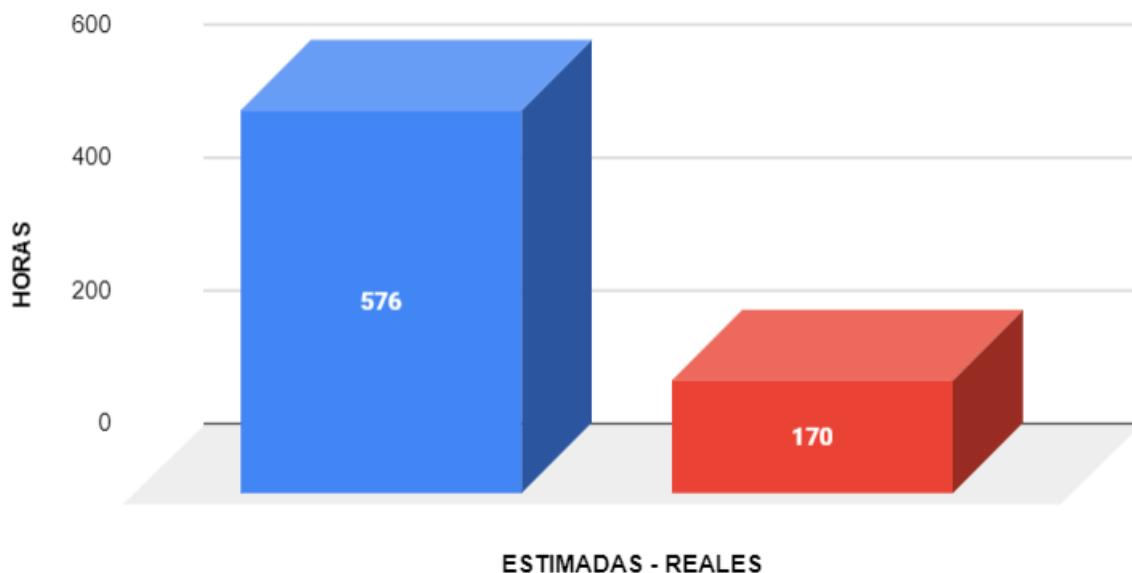
GESTION TEMPORAL HORAS



## COMPARACIÓN TIEMPO REAL - PREVISTO

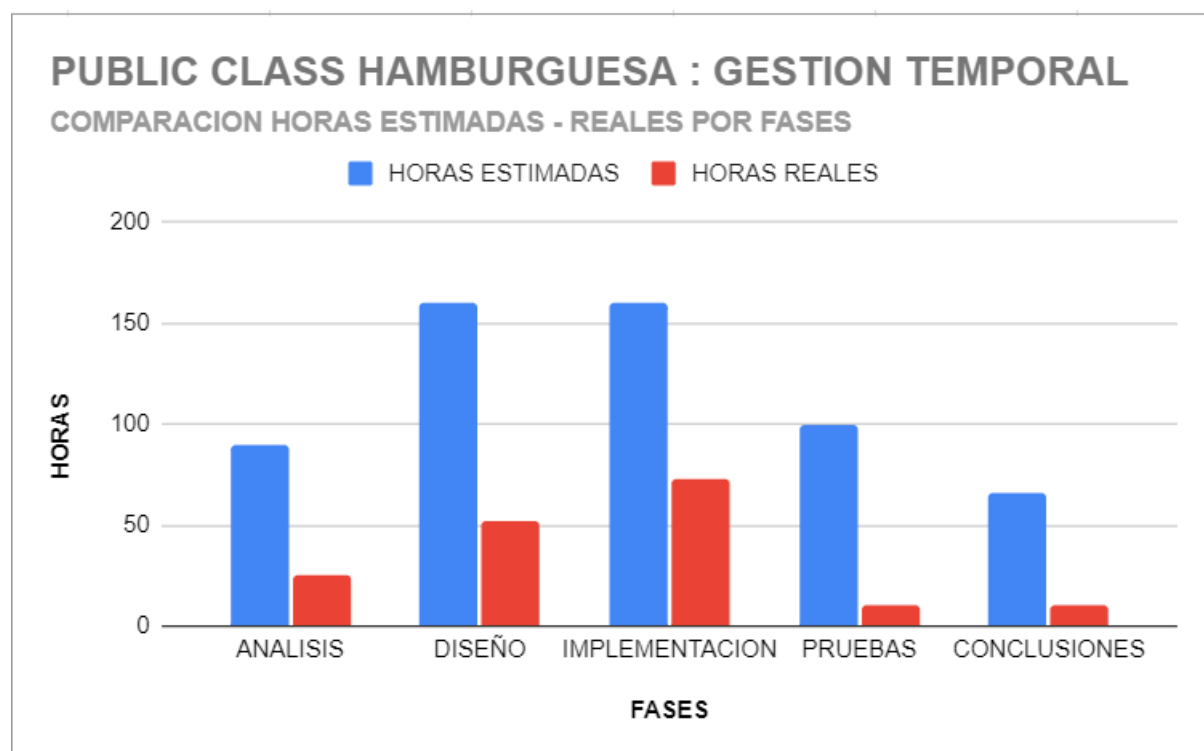
### PUBLIC CLASS HAMBURGUESA : GESTION TEMPORAL

COMPARACION HORAS ESTIMADAS - REALES



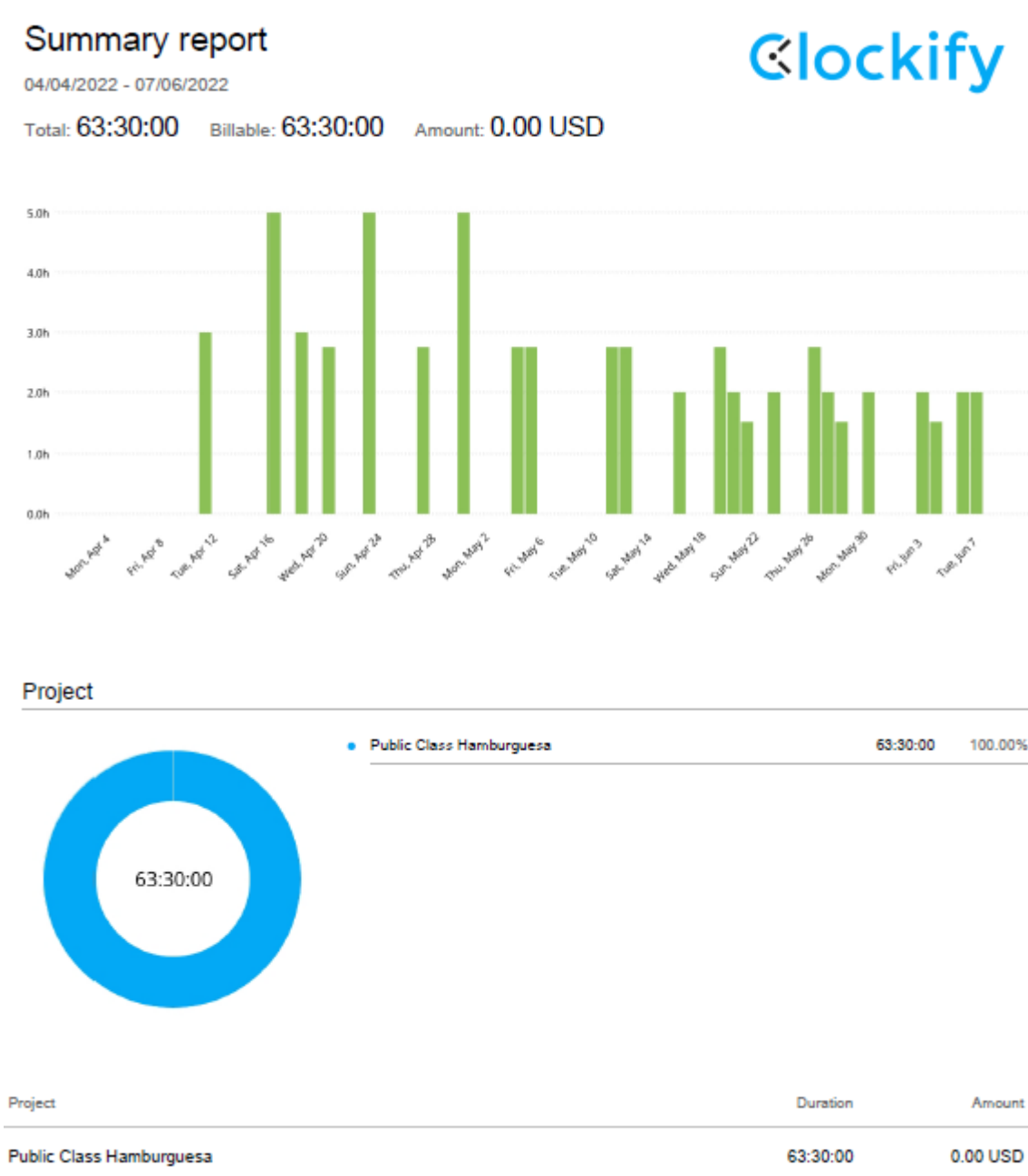
## COMPARACIÓN TIEMPO REAL - PREVISTO POR FASES

PUBLIC CLASS HAMBURGUESA		TIEMPO EN HORAS	
		PREVISTO	REAL
FASES	ANALISIS	90	25
	DISEÑO	160	52
	IMPLEMENTACION	160	73
	PRUEBAS	100	10
	CONCLUSIONES	66	10



# GESTIÓN TEMPORAL POR PERSONA. CLOCKIFY

DANIEL FERNÁNDEZ VALCARCE



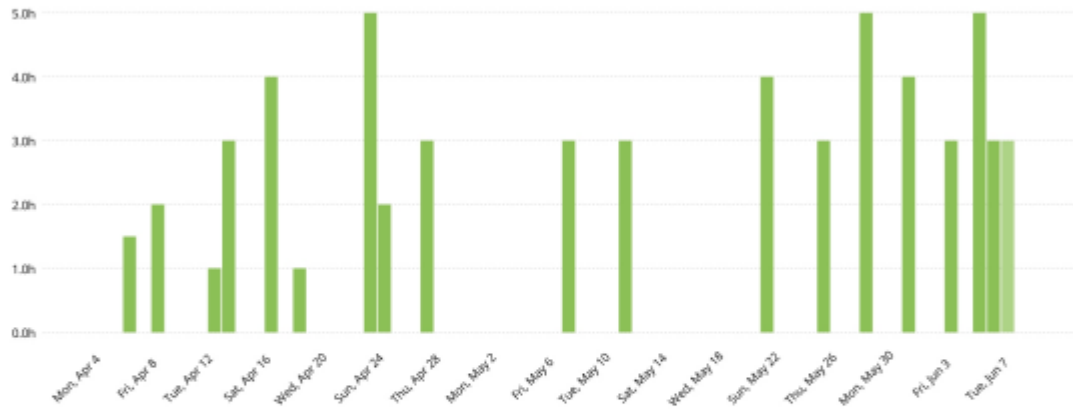
## ANTONIO FELIU DE CABRERA SALAS

### Summary report

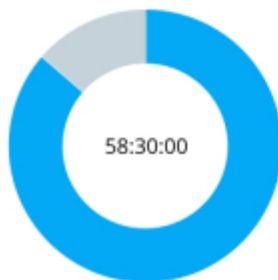
04/04/2022 - 08/06/2022



Total: 58:30:00 Billable: 55:30:00 Amount: 0.00 USD



### Project



• (Without Project)	08:00:00	13.68%
• Public Class Hamburguesa	50:30:00	86.32%

## JORGE MARTÍN MENÉNDEZ

### Summary report

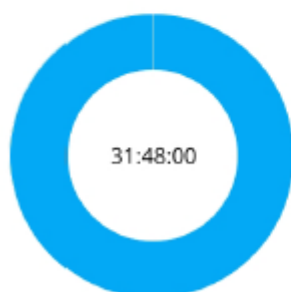
04/04/2022 - 07/06/2022



Total: 31:48:00 Billable: 31:48:00 Amount: 0.00 USD



### Project



Public Class Hamburguesa 31:48:00 100.00%

Project	Duration	Amount
Public Class Hamburguesa	31:48:00	0.00 USD



## TRANSFORMACIÓN A UNIDADES MONETARIAS

Según lo que hemos visto anteriormente, se han trabajado 170 horas, contando que la unidad monetaria usada es el euro, y la localización es España, en donde el salario medio de un programador es 28.500€ al año, lo cual nos hace un gasto por hora de 14,62€ el coste total de la aplicación en este punto es de unos 2485,4€.

## 8 CONCLUSIONES

La evaluación de programas es un instrumento de gestión. Es un proceso de duración determinada que trata de valorar de manera sistemática y objetiva la pertinencia, el rendimiento y el éxito de los programas y proyectos concluidos y en curso. La evaluación se realiza con carácter selectivo para dar respuesta a determinadas preguntas e impartir orientación a los encargados de tomar decisiones y los administradores de programas, así como para obtener información que permita determinar si las teorías e hipótesis básicas que se utilizaron al formular el programa resultaron válidas, qué surtió efecto o no, y por qué.

## MATRIZ REQUISITOS REALIZADOS

En esta matriz se exponen los diferentes requisitos funcionales, y si han sido realizados.

PUBLIC CLASS HAMBURGUESA			
nº	ID	DESCRIPCION	REALIZADO
1	CLIENTE_RF01	El usuario debe poder iniciar un pedido	SI
2	CLIENTE_RF02	El usuario debe poder añadir uno o varios productos a su pedido.	SI
3	CLIENTE_RF03	El usuario debe poder cancelar su pedido siempre y cuando no lo haya finalizado.	SI
4	CLIENTE_RF04	El usuario debe poder confirmar y finalizar su pedido	SI
5	EMPLEADO_RF01	El empleado debe poder visualizar los pedidos en cola	NO
6	EMLEADO_RF01	El empleado debe poder marcar un pedido como "listo"	NO

## VALORACIÓN FINAL

El Proyecto ha mostrado que el producto implementado es factible y viable y que el éxito en términos de cumplimiento de metas y resultados guardan una relación directa con el diseño, el presupuesto (en cuanto a alcances y disponibilidad), el tiempo y la metodología.

## POSIBLES MEJORAS

En este punto, las posibles mejoras pueden ser completar los siguientes requisitos funcionales:

EMPLEADO_RF01	El empleado debe poder visualizar los pedidos en cola
EMLEADO_RF01	El empleado debe poder marcar un pedido como "listo"

## LECCIONES APRENDIDAS

En cuanto a lecciones aprendidas, lo primero a destacar, sería que la gestión temporal ha sido demasiado optimista, la previsión a largo plazo se ha de cumplir.

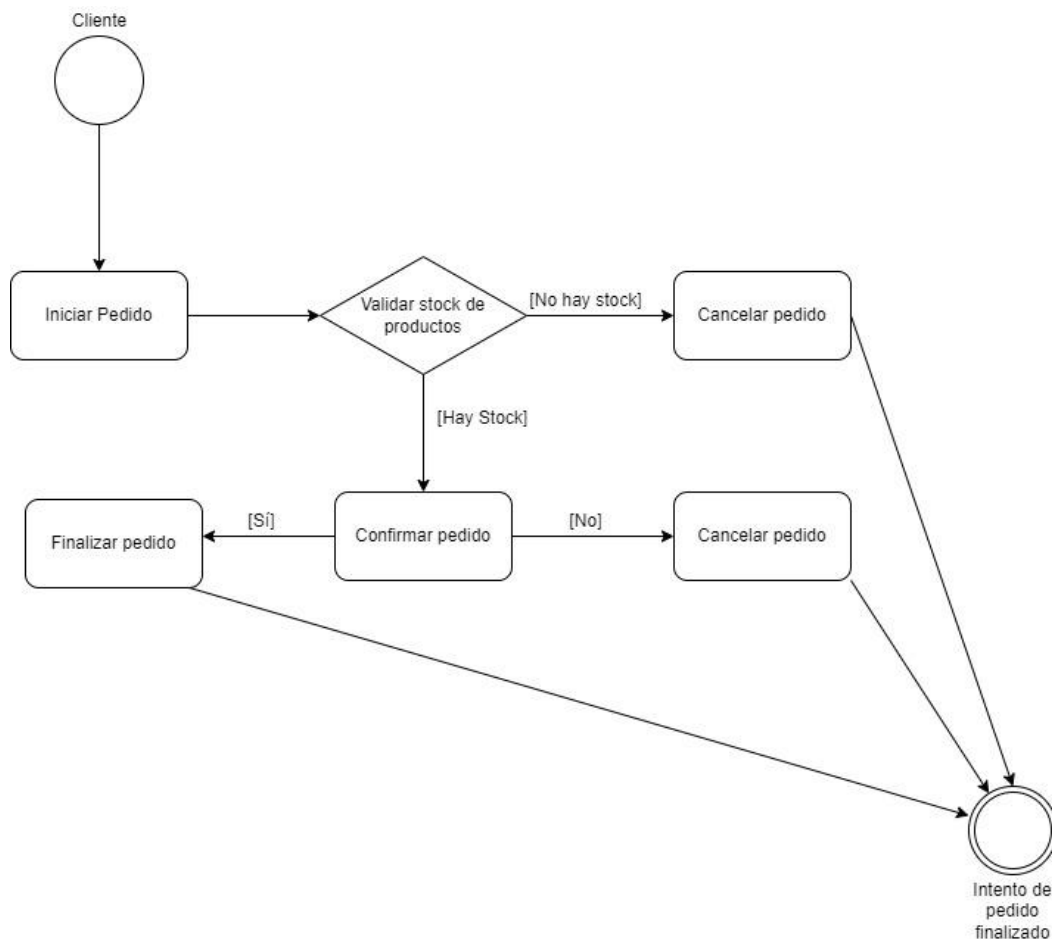
## 8 ANEXOS

### ENLACE AL GITHUB

<https://github.com/Public-Class-Hamburguesa/Public-Class-Hamburguesa>

### DIAGRAMA DE ACTIVIDADES

Un diagrama de actividades es un diagrama de comportamiento que ilustra el flujo de actividades a través de un sistema. Los diagramas de actividad UML también pueden utilizarse para representar un flujo de eventos en un proceso comercial. Pueden utilizarse para examinar los procesos comerciales a fin de identificar su flujo y sus necesidades.



### DIAGRAMA DE FLUJO

Un Diagrama de Flujo representa la esquematización gráfica de un algoritmo, el cual muestra gráficamente los pasos o procesos a seguir para alcanzar la solución de un problema. Su correcta construcción es sumamente importante porque, a partir del mismo se escribe un programa en algún Lenguaje de Programación.

