



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Trabajo fin de Grado

Grado en Ingeniería Informática – Ingeniería del Software

Portal Django Panadería

Autor/es:
José María García Berdejo

Tutor/es:
José Ángel Galindo Duarte

Primera convocatoria

Sevilla, *junio de 2024*

Calificación Obtenida: 8,5

Resumen

Este trabajo documenta la creación de una aplicación web para dispositivos móviles y ordenadores de sobremesa para la gestión de pedidos y productos de una panadería. Las funcionalidades de la aplicación incluyen la posibilidad de realizar pedidos por parte de los clientes de la panadería con dos métodos de pago y la adición, modificación y borrado de productos de la tienda por parte de los administradores.

Esta idea se basa en el desarrollo de una aplicación de base que sea aplicable a varias empresas que funcionen con un modelo de negocio similar a una panadería y que permita el envío de productos a domicilio, ya que está más orientada a facilitar este tipo de tareas en la gestión de las empresas. Partiendo de esta aplicación, se podría adaptar perfectamente, una solución personalizada que modificará la base mencionada de acuerdo con los requisitos impuestos por la parte interesada.

El desarrollo de este proyecto ha sido posible gracias a los conocimientos adquiridos en diferentes asignaturas, como Planificación y Gestión de Proyectos Informáticos en la que se nos enseñó no sólo a planificar sino a ejecutar la implementación de un proyecto en Django en el cual se base gran parte del desarrollo llevado a cabo. También cabe destacar la gran labor que se llevó a cabo en la implementación de pruebas software durante todo el grado, pero especialmente en la asignatura Evolución y Gestión de la Configuración, donde además de aprender a realizar multitud de pruebas sobre el software del proyecto dado, también se nos inculcó conceptos como la “dockerización” de las aplicaciones web para su posterior despliegue, que ha resultado muy útil para llevar a cabo un ejecutable de este proyecto.

Palabras clave: aplicación web, envíos, pedidos, gestión de productos, gestión de stock, gestión de existencias, django, vue.js, backend, frontend.

Abstract

This project documents the creation of a web application for mobile devices and desktop computers for the order and product management of a bakery. The functionalities of the application include the possibility of ordering by the bakery's customers with two payment methods and the addition, modification, and deletion of products from the shop by the administrators.

This idea is based on the development of a basic application that is applicable to several companies that operate with a business model like a bakery and that allows home delivery of products, as it is more oriented towards facilitating this type of tasks in the management of companies. Based on this application, a customised solution could be perfectly adapted to modify the aforementioned base according to the requirements imposed by the interested party.

The development of this project has been possible thanks to the knowledge acquired in different subjects, such as Planning and Management of IT Projects, in which we were taught not only to plan but also to execute the implementation of a project in Django, on which a large part of the development carried out is based. It is also worth mentioning the great work carried out in the implementation of software testing throughout the degree, but especially in the subject Evolution of Management and Configuration, where in addition to learning how to carry out a multitude of tests on the Django software, we also learnt how to use the Django software in the implementation of the Django project, which is the basis for much of the development carried out.

Keywords: web application, shipping, ordering, product management, stock management, stock management, django, vue.js, backend, frontend.

Agradecimientos

En este apartado quisiera agradecer a varias personas el gran apoyo que he recibido durante la confección de este gran proyecto.

En primer lugar, me gustaría dar las gracias a la Universidad de Sevilla por todos los conocimientos otorgados y por la gran labor docente que desempeña. De todas las etapas educativas que se viven desde la infancia hasta bien entrada la juventud, la universidad es de las pocas en las que realmente se trabaja sobre todo muchísimo la paciencia y es una virtud de la que pocas personas gozan. Gracias a la institución y los grandes docentes que brinda he adquirido los conocimientos necesarios para la realización de este trabajo.

También debo agradecer a los docentes que en el ámbito escolar, pero sobre todo en la etapa secundaria me han abierto puertas de mundos que desconocía, motivándome a seguir el camino que recorro y que me ha convertido en la persona que soy. En especial a mi profesor de TIC de cuarto de la ESO que con su gran entusiasmo en la informática fue uno de los que me inspiró a estudiar este grado.

Por supuesto, quiero agradecer a José Ángel Galindo Duarte todo el asesoramiento que me ha brindado no sólo durante este trabajo sino también en otras asignaturas que imparte en la escuela. Sin sus consejos, habría pensado demasiado algunas opciones que se recogen en el proyecto y no lo habría acabado a tiempo. Su papel como tutor del TFG ha sido espectacular.

Por último pero no menos importante, quiero agradecer especialmente el esfuerzo que ha realizado mi familia apoyándome en todo momento y sin dejar que el trabajo pudiera conmigo, motivándome y reflexionando cuáles serían los movimientos más adecuados a llevar a cabo para que el proyecto fuera un éxito. A mi amigo Álvaro Bernal le agradezco el interés y los conocimientos que hemos compartido para que el proyecto luzca y funcione aún mejor de lo que en principio lo hacía, brindando un punto de vista distinto y único que me ha hecho ser precavido y cuidadoso con el contenido de esta memoria en algunos aspectos. Ambos siempre me han apoyado y creído en mí. Sin vuestra ayuda y cariño nunca hubiera llegado a donde estoy hoy.

Índice general

Resumen	III
Agradecimientos	V
Índice de cuadros	1
Índice de figuras	3
Introducción	6
1.1. Introducción.....	6
1.2. Motivación	7
1.3. Alcance	8
1.4. Objetivos	8
1.4.1. Conocimientos previos.....	9
1.4.2. Formación Propia	10
1.5. Estructura del Documento	10
Materias Relacionadas	12
2.1. Herramientas de Gestión.....	12
2.2. Herramientas de Desarrollo.....	13
2.3. Herramientas de Pruebas.....	16
2.4. Metodologías.....	17
Planificación	18
3.1. Plan de Gestión de las Comunicaciones	18
3.2. Análisis de Requisitos	19
3.2.1. Clases y características de los usuarios	19
3.2.2. Funcionalidades del sistema.....	19
3.2.3. Descripción actual del sistema.....	20
3.2.4. Recopilación de Requisitos.....	21
3.2.5. Diseño	35
3.3. Definir el alcance.....	36
3.3.1. Enunciado del Alcance	36
3.3.2. Plan de Gestión del Alcance	37
3.3.3. EDT	39
3.3.4. Diccionario EDT	39
3.3.5. Secuenciación de las actividades	42
3.3. Plan de Gestión de los Recursos.....	44
3.4. Plan de Gestión del Cronograma.....	44
3.5.1. Estimaciones de tiempo.....	45
3.5.2. Cronograma.....	49
3.6. Estimación de costes.....	50

3.6.1.	Coste Humano.....	50
3.6.2.	Coste Material	50
3.6.3.	Coste de Servicios Externos.....	51
3.6.4.	Conclusión.....	51
3.7.	Plan de Gestión de Riesgos	52
3.8.	Plan de Gestión de la Calidad	53
	Sistema Desarrollado.....	56
4.1.	Modelo de datos.....	56
4.2.	Backend	57
4.2.1.	Aplicación de autenticación	58
4.2.2.	Aplicación two_factor	59
4.2.3.	Aplicación “base”	60
4.2.4.	Aplicación de pedidos.....	61
4.2.5.	Aplicación de pagos.....	62
4.2.6.	Aplicación de productos.....	62
4.2.7.	Dificultades encontradas.....	64
4.3.	Frontend.....	65
4.3.1.	Página de inicio	65
4.3.2.	Catálogo de productos.....	66
4.3.3.	Carrito de la compra	67
4.3.4.	Mis pedidos	69
4.3.5.	Administración de productos.....	69
4.3.6.	Administración de pedidos.....	70
4.3.7.	Estadísticas	71
4.3.8.	Página “Sobre Nosotros”	72
4.4.	Seguridad.....	72
	Implementación	74
5.1.	Implementación del backend	74
5.2.	Implementación del frontend	75
	Gestión del despliegue	77
	Pruebas	78
7.1.	Pruebas de backend.....	78
7.1.1.	Pruebas de Modelo.....	78
7.1.2.	Pruebas de vista con API.....	79
7.1.3.	Pruebas de carga	80
7.2.	Pruebas de frontend.....	81
7.2.1.	Pruebas unitarias.....	81
7.2.2.	Pruebas de vista dinámicas	81
	Manual de usuario	82

8.1.	Página principal.....	82
8.2.	Registro y acceso.....	83
8.3.	Manual del cliente	88
8.3.1.	Catálogo de productos.....	88
8.3.2.	Carrito de la compra	90
8.3.3.	Sobre Nosotros.....	94
8.4.	Manual del administrador	94
8.4.1.	Administración de productos.....	94
8.4.2.	Administración de categorías.....	99
8.4.3.	Administración de pedidos.....	102
8.4.4.	Estadísticas	105
	Licencia	107
	Conclusión.....	110
	Planes futuros.....	111
	Bibliografía.....	112
	Glosario de Términos.....	115
	Anexos	116
13.1.	Informe de Seguimiento de la Iteración 1	116
13.1.1.	Objetivo de la Reunión	116
13.1.2.	Evaluación del Código	116
13.1.3.	Acciones Correctivas y Sugerencias.....	117
13.1.4.	Plan de Acción.....	117
13.1.5.	Aprobación	117
13.2.	Informe de Seguimiento de la Iteración 2	117
13.2.1.	Objetivo de la Reunión	117
13.2.2.	Evaluación del Código	117
13.2.3.	Acciones Correctivas y Sugerencias.....	118
13.2.4.	Plan de Acción.....	118
13.2.5.	Aprobación	118
13.3.	Informe de Seguimiento de la Iteración 3	119
13.3.1.	Objetivo de la Reunión	119
13.3.2.	Evaluación del Código	119
13.3.3.	Acciones Correctivas y Sugerencias.....	120
13.3.4.	Plan de Acción.....	120
13.3.5.	Aprobación	120
13.4.	Informe de desempeño.....	120
13.4.1.	Actualización a mitad de la iteración 1 (15/1/2024)	120
13.4.2.	Actualización al final de la iteración 1 (23/1/2024)	121
13.4.3.	Actualización a mitad de la iteración 2 (15/2/2024)	123

13.4.4.	Actualización al final de la iteración 2 (10/3/2024)	124
13.4.5.	Actualización a mitad de la iteración 3 (25/3/2024)	125
13.4.6.	Actualización al final de la iteración 3 (3/4/2024)	127
13.5.	Seguimiento de la Documentación	129
13.5.1.	Borrador de la memoria (24/4/2024)	129
13.5.2.	Avances hasta la primera mitad de la memoria (8/5/2024)	129
13.5.3.	Avances hasta la segunda mitad de la memoria (13/5/2024)	130
13.5.4.	Finalización de la memoria (24/5/2024)	131
13.6.	Documentación adicional	131

Índice de cuadros

Cuadro 3.1: Información correspondiente a cada interesado	19
Cuadro 3.2: Leyenda de siglas	22
Cuadro 3.3: Registro de requisitos.....	25
Cuadro 3.4: Matriz de trazabilidad de requisitos	33
Cuadro 3.5: Lista de entregables del proyecto	37
Cuadro 3.6: Actividades de la Iteración 1.....	39
Cuadro 3.7: Actividades de la Iteración 2.....	40
Cuadro 3.8: Actividades de la Iteración 3.....	41
Cuadro 3.9: Actividades del Seguimiento.....	41
Cuadro 3.10: Actividades del Cierre.....	41
Cuadro 3.11: Procesos de gestión del cronograma	45
Cuadro 3.12: Estimaciones de tiempo de la ejecución.....	47
Cuadro 3.13: Estimación de pruebas y correcciones	47
Cuadro 3.14: Estimación del inicio	48
Cuadro 3.15: Estimaciones de la planificación	48
Cuadro 3.16: Estimaciones documentación	49
Cuadro 3.17: Estimación de costes materiales	50
Cuadro 3.18: Desglose del TCO para un año	52
Cuadro 3.19: Desglose del TCO para los cinco meses del desarrollo.....	52
Cuadro 3.20: Acciones del plan de calidad	53
Cuadro 13.1: Actividades completadas en la mitad de la iteración 1.....	120
Cuadro 13.2: Problemas encontrados en la mitad de la iteración 1.....	121
Cuadro 13.3: Seguimiento de riesgos en la mitad de la iteración 1.....	121
Cuadro 13.4: Rendimiento en la mitad de la iteración 1	121
Cuadro 13.5: Seguimiento de la calidad en la mitad de la iteración 1	121
Cuadro 13.6: Actividades completadas al final de la iteración 1.....	122
Cuadro 13.7: Problemas encontrados al final de la iteración 1.....	122
Cuadro 13.8: Seguimiento de riesgos al final de la iteración 1	122
Cuadro 13.9: Rendimiento al final de la iteración 1	122
Cuadro 13.10: Seguimiento de la calidad al final de la iteración 1	122
Cuadro 13.11: Actividades completadas en la mitad de la iteración 2.....	123
Cuadro 13.12: Problemas encontrados en la mitad de la iteración 2.....	123
Cuadro 13.13: Seguimiento de riesgos en la mitad de la iteración 2	124
Cuadro 13.14: Rendimiento en la mitad de la iteración 2	124
Cuadro 13.15: Seguimiento de la calidad en la mitad de la iteración 2	124

Cuadro 13.16: Actividades completadas al final de la iteración 2	124
Cuadro 13.17: Problemas encontrados al final de la iteración 2.....	125
Cuadro 13.18: Seguimiento de riesgos al final de la iteración 2	125
Cuadro 13.19: Rendimiento al final de la iteración 2	125
Cuadro 13.20: Seguimiento de la calidad al final de la iteración 2	125
Cuadro 13.21: Actividades completadas en la mitad de la iteración 3.....	126
Cuadro 13.22: Problemas encontrados en la mitad de la iteración 3.....	126
Cuadro 13.23: Seguimiento de riesgos en la mitad de la iteración 3.....	126
Cuadro 13.24: Rendimiento en la mitad de la iteración 3	127
Cuadro 13.25: Seguimiento de la calidad en la mitad de la iteración 3	127
Cuadro 13.26: Actividades completadas al final de la iteración 3.....	128
Cuadro 13.27: Problemas encontrados al final de la iteración 3.....	128
Cuadro 13.28: Seguimiento de riesgos en la mitad de la iteración 3.....	128
Cuadro 13.29: Rendimiento en la mitad de la iteración 3	128
Cuadro 13.30: Seguimiento de la calidad en la mitad de la iteración 3	128
Cuadro 13.31: Frecuencia de errores e índice de precisión de la documentación.....	129
Cuadro 13.32: Apartados mínimos exigidos y número de páginas que contienen.....	129
Cuadro 13.33: Frecuencia de errores e índice de precisión de la documentación.....	130
Cuadro 13.34: Apartados mínimos exigidos y número de páginas que contienen.....	130
Cuadro 13.35: Frecuencia de errores e índice de precisión de la documentación.....	130
Cuadro 13.36: Apartados mínimos exigidos y número de páginas que contienen.....	131
Cuadro 13.37: Frecuencia de errores e índice de precisión de la documentación.....	131
Cuadro 13.38: Apartados mínimos exigidos y número de páginas que contienen.....	131

Índice de figuras

Figura 3.1: EDT del proyecto	39
Figura 3.2: Secuenciación de las actividades del proyecto	43
Figura 3.3: Diagrama temporal real del proyecto	49
Figura 4.1: Diagrama UML de clases del backend	56
Figura 4.2: “Serializador” de la clase Customer	57
Figura 4.3: “ViewSet” de la clase Customer	58
Figura 7.1: Informe de cobertura de backend	79
Figura 7.2: Gráficos que muestran el número total de peticiones por segundo al sistema y los tiempos de respuesta de este.	80
Figura 8.1: Vista de la página principal desde un ordenador	82
Figura 8.2: Sección de productos destacados de la página principal	83
Figura 8.3: Sección de productos destacados y pie de página.....	83
Figura 8.5: Registro Completo visto desde el móvil	84
Figura 8.4: Formulario de registro visto desde el móvil	84
Figura 8.7: Menú de activación del doble factor (móvil)	84
Figura 8.6: Formulario de inicio de sesión (móvil)	84
Figura 8.8: Selección del método de autenticación (móvil)	85
Figura 8.9: Código QR del generador de códigos (visto desde un ordenador)	85
Figura 8.10: Éxito en la activación del doble factor de autenticación.....	86
Figura 8.11: Formulario de inicio de sesión.....	86
Figura 8.12: Código del doble factor en el campo “Token”	86
Figura 8.13: Menú del usuario en la barra de navegación.....	87
Figura 8.14: Formulario de cambio de contraseña	87
Figura 8.15: Opción de activación del doble factor.....	87
Figura 8.16: Activación de la autenticación en dos pasos	88
Figura 8.17: Catálogo de productos	88
Figura 8.18: Flechas para aumentar o disminuir la cantidad de un producto (indicadas con la flecha roja)	89
Figura 8.19: Mensaje de alerta mostrado con cantidad negativa	89
Figura 8.20: Cambio en el número de ítems del carrito (indicado en el círculo rojo) ...	90
Figura 8.21: Vista del carrito de la compra (fragmento).....	91
Figura 8.22: Selección de tipo de pago y entrega (fragmento)	91
Figura 8.23: Pedido realizado con éxito (independientemente del método de pago)...	91
Figura 8.24: Mis pedidos.....	92
Figura 8.25: Menú del usuario	92

Figura 8.26: Pago con Stripe	93
Figura 8.27: Correo electrónico de confirmación.....	93
Figura 8.28: Sobre Nosotros.....	94
Figura 8.29: Menú de Administración.....	95
Figura 8.30: Catálogo de productos del administrador	95
Figura 8.31: Formulario de creación de productos	96
Figura 8.32: Error en la introducción del precio de un producto	96
Figura 8.33: Mensaje mostrado al crear un producto con éxito (visto desde una pantalla con tamaño inferior a 992 píxeles, por eso aparece la hamburguesa, que está señalada en la figura con un círculo rojo, que contiene el menú recogido).....	97
Figura 8.34: Mensaje que informa de la actualización con éxito	98
Figura 8.35: Alerta de confirmación de eliminación del producto	98
Figura 8.36: Mensaje que informa de la correcta eliminación del producto	98
Figura 8.37: Menú de Administración (Categorías seleccionado)	99
Figura 8.38: Listado de categorías.....	99
Figura 8.39: Formulario de creación de categorías	100
Figura 8.40: Mensaje de creación exitosa de categoría	100
Figura 8.41: Formulario de edición de una categoría	100
Figura 8.42: Mensaje mostrado al actualizar satisfactoriamente una categoría.....	101
Figura 8.43: Alerta de confirmación de eliminación de una categoría	101
Figura 8.44: Mensaje que informa de la correcta eliminación de una categoría	102
Figura 8.45: Menú de Administración (Pedidos seleccionado)	102
Figura 8.46: Panel de administración de pedidos.....	103
Figura 8.47: Búsqueda por ID del pedido.....	103
Figura 8.48: Búsqueda por nombre del cliente.....	104
Figura 8.49: Botones para ir a la página siguiente o anterior en el listado de pedidos	104
Figura 8.50: Formulario de edición de un pedido	105
Figura 8.51: Alerta de confirmación de eliminación de un pedido	105
Figura 8.52: Vista de las estadísticas de la página web	106
Figura 9.1: Derechos de autor sobre el proyecto django-two-factor-auth	108

Primera Parte

Preámbulo

Introducción

1.1. Introducción

La idea detrás de este gran proyecto nace de una necesidad tan simple como la gestión de pedidos en una panadería. Desde que el negocio de las panaderías existe los clientes han acudido a la panadería a comprar el pan del día que el panadero había obrado horas antes en el horno. Sin embargo, en los tiempos que corren, hay auténticos monopolios de empresas panaderas que no pueden gestionar la cantidad de pedidos que reciben de una manera tan rudimentaria, fundamentalmente porque la clientela que tienen es mucho más variada. Desde pequeños negocios que venden pan precocinado hasta grandes restaurantes que necesitan proveerse de materias primas, la gestión de los pedidos podría convertirse en un auténtico infierno. Pero no sólo las grandes corporaciones, sino también las pequeñas panaderías tienen derecho a crecer y no hay nada mejor para darse a conocer al mundo hoy que Internet. Una página web vistosa, simple de utilizar y que sea igualmente útil para gestionar los productos de la tienda y los pedidos de los clientes es una oportunidad única para poder empezar la expansión de un negocio.

Así se comenzó a trabajar en la construcción de una página web que pretende acercar a estos negocios una herramienta asequible con la que se puede informatizar sus pedidos, control de existencias y gestionar los productos. Esta aplicación web estará diseñada tanto para ser utilizada en dispositivos móviles como en portátiles y ordenadores de sobremesa, con una interfaz intuitiva y fácil de utilizar tanto por usuarios experimentados con las tecnologías como personas que no gocen de tantos conocimientos con la informática.

La página desarrollada cuenta con una pantalla principal en la que los clientes pueden registrarse o iniciar sesión, agregar productos destacados de la página principal al carrito de la compra o dirigirse al catálogo de productos para seleccionar los que deseen adquirir. Se puede pagar con tarjeta o contra reembolso y se puede escoger entre recoger el pedido en la tienda o que te lo traigan a tu dirección. Además, cuenta con dos tipos de usuarios, los clientes que pueden hacer sus pedidos y llevar a cabo un seguimiento de estos en una página de fácil acceso; y los administradores, que tendrán la opción de crear, modificar o eliminar productos de la tienda, realizar un seguimiento de los pedidos de todos los clientes y acceso a un módulo de estadísticas en el que podrán analizar cuáles son los productos más vendidos y los clientes que más pedidos realizan.

Los datos de los pedidos, los productos y los clientes que dotan de funcionamiento a la aplicación web serán almacenados en una base de datos y el acceso a los pedidos por parte de los clientes estará restringido a aquellos usuarios que no hayan activado la doble verificación de seguridad que se pretende integrar en esta aplicación, para reforzar la seguridad de esta.

1.2. Motivación

La principal motivación que el autor ha tenido para llevar a cabo este proyecto parte de la necesidad de mejorar el sistema de pedidos que tenía su madre con su panadera. La historia real es la que sigue a continuación.

La madre del autor de esta memoria llevaba varios años pidiendo pan a su panadera que, afortunadamente, lo traía a casa todas las mañanas. Sin embargo, aunque lo más sencillo habría sido pedirle las mismas piezas de pan cada día, no siempre se consumían las mismas cantidades y tipos de pan todos los días. Es por eso por lo que cuando la tan conocida aplicación de mensajería instantánea “WhatsApp” llegó a nuestras vidas, comenzó a pedirle el pan que quería cada día por esa aplicación. El problema es que la panadera que nos traía el pan no se acordaba de cobrarle el pan a sus clientes, por lo que fácilmente podrías llegar a tener que pagarle cantidades desorbitadas tras varios meses sin poder hacerlo, debido al tiempo que transcurría entre pago y pago. Eso, evidentemente también le afectaba financieramente a la panadera que tenía que hacer lo imposible por ajustar los gastos con los ingresos que recibía. La memoria que tenía la panadera no daba abasto para todos los clientes, pedidos y cobros que tenía que llevar a cabo.

Como mi madre no quería que la suma de cantidades mensuales se dispararan debido a la escurridiza memoria de la panadera, cada mensaje que le enviaba por la noche, incluía las cantidades y tipos de piezas de pan que deseaba tener al día siguiente en casa; además la panadera, que vio con buenos ojos el hecho de que mi madre quisiera pagarle a su debido tiempo, le facilitó el precio de todos los productos que vendía en la panadería (picos, roscón de reyes en época navideña, etc...) y así mes a mes mi madre realizaba las cuentas manualmente y le pagaba en efectivo cuando la veía por las mañanas, que aunque en apariencia puede parecer ser sencillo, en realidad es más difícil de lo que es, teniendo en cuenta que por las mañanas es cuando reparte el pan y apenas pasan uno o dos minutos desde que llega a una casa, deja el pan y se dirige a la siguiente parada de su itinerario mañanero de reparto.

Entonces surgió “Bizum” con el que se podían hacer transacciones bancarias de una forma simple y rápida. Con su aparición, el trabajo de pago a la panadera fue más cómodo, ya que al llegar fin de mes, se le realizaba un pago con las cuentas de dicho mes. Sin embargo mi madre llegado este punto continuaba realizando las cuentas manualmente, situación que por supuesto era muy ineficiente; así que le propuse que como primer paso utilizara una hoja de cálculo en la nube (Google Drive) para que no tuviera que operar los cálculos manualmente, sino que pusiera la cantidad mensual que había pedido (y la que le había traído, ya que la panadera acostumbra a no traer todo el pan que se le pide sin notificarlo, aunque esto ya depende más de la panadera que de mi madre).

Aun así, este sistema seguía siendo rudimentario, teniendo en cuenta que era mi madre la que seguía teniendo que escribir en las celdas correspondientes las cantidades diarias de pan. Por tanto, como aspirante a ingeniero, decidí que quería ofrecer una solución sencilla para la gestión de los pedidos y los pagos a esta pequeña empresa de mi pueblo.

De cara a la realización del trabajo de fin de grado, quería que no fuera un proyecto que “sólo” sirviera para llenar una estantería, sino más bien para ayudar a mi familia y sobre todo a las panaderías a sobrellevar esta situación. He estado trabajando en otros

proyectos similares de carácter personal y sinceramente, creo que el software está diseñado para facilitarle la vida a las personas y por eso decidí dar el paso al desarrollo de tan importante proyecto.

Mi principal inspiración para realizar este proyecto se basa en la necesidad de automatizar lo máximo posible la vida de las personas y que puedan ser más eficientes en su vida diaria. La historia que se describió anteriormente demuestra que una gestión tan simple como pedir pan a una panadería puede convertirse en toda una odisea y bajo mi punto de vista, la tecnología aunque para muchos pueda ser una barrera, no sirve sino para llevarnos hacia un mundo más cómodo, en el que las tareas más repetitivas sean las máquinas las que las lleven a cabo y la humanidad pueda centrarse en tareas de razonamiento más importantes que un simple pedido de pan diario o tal vez en disfrutar de lo que queda de día viendo una serie en la televisión.

Otro de los pilares fundamentales para la realización de este proyecto, es la necesidad de mostrarle a los profesores de la escuela que las competencias que he adquirido durante estos cuatro años de estudio son más que suficientes para llevar a cabo un proyecto de esta magnitud y que tengo motivos de peso para demostrar que puedo ejercer como ingeniero de software cuando finalice este proyecto.

Por último, quería dejar a mis compañeros en un futuro un ejemplo bien implementado a nivel de seguridad y de UI para que tomen este proyecto si lo desean y experimenten con él con fines didácticos. En las secciones que vienen a continuación explicaré con detalle por qué he elegido las tecnologías de backend y frontend que sustentan el trabajo y la metodología de trabajo que he ido siguiendo para completar el proyecto en tiempo y coste.[JD1][JG2]

1.3. Alcance

Este proyecto está enfocado en desarrollar esta aplicación que sea capaz de satisfacer las necesidades para una panadería online y que pueda ser aplicable a cualquier panadería que reparta pan a domicilio como en nuestro caso. Asimismo, el objetivo de esta aplicación es que pueda servir como punto de partida para cualquier panadería y/o negocio que contemple el reparto a domicilio de productos.

1.4. Objetivos

Aunque se podría intuir al leer la **Introducción y Motivación** de esta memoria, el objetivo fundamental de este proyecto es ofrecer una solución integral para la gestión eficiente de productos y pedidos en la panadería. Esto permitirá optimizar sus procesos diarios de recepción de pedidos y contabilidad de productos, liberando recursos para otras áreas del negocio.

La necesidad de abordar problemas relacionados con la morosidad de algunos clientes también ha sido un factor determinante en el desarrollo de esta aplicación. Se busca mitigar, en la medida de lo posible, los retrasos en los pagos por parte de los clientes, si bien este objetivo se considera secundario.

Además de estos objetivos, el proyecto tiene como propósito aplicar y poner en práctica todas las técnicas de planificación, implementación y elicitación de requisitos adquiridas durante los cuatro años de estudios en la escuela.

Por último, la base en hechos reales de este proyecto aporta un valor significativo al trabajo que va a involucrar y las horas invertidas en el mismo, ya que una vez finalizado, se prevé ofrecer esta aplicación web a la panadería en cuestión y establecer un acuerdo comercial con ellos, como si se tratara de un cliente más.

1.4.1. Conocimientos previos

Como conocimientos previos al inicio del desarrollo se han de destacar los conocimientos que se han adquirido a lo largo de diversas asignaturas del grado en Ingeniería del Software. Algunas de estas asignaturas son:

- Ingeniería de Requisitos: Ha sido fundamental para discernir entre los requisitos funcionales y no funcionales de la aplicación, así como los requisitos de seguridad. Aunque no se han tratado unos requisitos de gran complejidad, la asignatura especifica claramente cómo diferenciarlos.
- IISSI 2: La aplicación desarrollada con React JS durante la parte práctica de la asignatura ha servido como inspiración para poder implementar la página utilizando otro framework en mi caso, pero sabiendo manejar desde un punto de visto de principiante, las promesas para obtener datos de una API y mostrarlos al usuario final.
- Planificación y Gestión de Proyectos Informáticos: Esta asignatura no ha sido sólo clave para la correcta planificación de este trabajo, sino que también ha sido vital en el desarrollo gracias a la aplicación Django que se llevó a cabo en esta. Gran parte del código utilizado en la aplicación se ha aplicado con sus modificaciones correspondientes para acelerar el proceso de implementación e invertir las horas correspondientes en el desarrollo del frontend con una tecnología que desconocía.
- Evolución y Gestión de la Configuración: Gracias a la implementación de pruebas de carga con Locust y el desarrollo de tests con Selenium, ha sido mucho más sencillo llevar a cabo la parte de las pruebas de software de la aplicación. Además, se nos enseñó a “dockerizar” aplicaciones web, autoconteniendo proyectos para su posterior despliegue de una forma muy intuitiva.

Aparte de estos conocimientos que he adquirido gracias a la Universidad de Sevilla, también han sido imprescindibles una serie de habilidades personales que ido desarrollando a lo largo de mi formación:

- La capacidad de gestionar y planificar los tiempos de un proyecto entregable asegurando que se entregará a tiempo y su desarrollo contemple pruebas y posibles correcciones antes de la entrega para finalizar el trabajo de la forma más completa posible.
- Uso de la herramienta Git. Muy importante para trabajar con proyectos. Una herramienta de control de versiones es clave para este tipo de proyectos, sobre todo por la tranquilidad que ofrece saber que se pueden deshacer cambios en el caso de que algún aspecto del código esté funcionando incorrectamente.

- Conocimiento sobre cómo estructurar correctamente el código que se escribe para facilitar la localización de errores o modificaciones.
- Conocimiento sobre depuración de código de cualquier proyecto software, sin importar el framework que se esté utilizando. Esto me ha permitido encontrar errores en menos tiempo e identificar funcionalidades mal implementadas al probarlas.
- Capacidad para aprender de una forma rápida el uso de una tecnología nueva, como pueda ser Vue.js. Aunque la adaptación a una nueva herramienta pueda ser difícil, saber interpretar la documentación y aprender a usarla para aplicarla en tus proyectos es crucial para llevar a cabo un desarrollo fluido.

1.4.2. Formación Propia

En este apartado expondré aquellas tecnologías que he debido aprender a utilizar para desarrollar el proyecto:

- **Two Factor Authentication:** la autenticación de doble factor es una tecnología que viene incorporada con el framework Django, haciendo uso de librería como “totp” y “django-otp”. Para integrarlo en mi proyecto he usado un proyecto open source de GitHub en el que viene una implementación de ejemplo que se puede aplicar en cualquier proyecto bajo una licencia, que está explicada más abajo en el apartado de **Licencia**.
- **Vue.js 3/Vite:** el framework que se ha empleado para implementar el frontend de la aplicación ha sido una novedad que quería incluir como parte de mi formación en la universidad y lo he utilizado como pretexto para incluirlo en este proyecto. Permite crear aplicaciones web “responsives” de una forma muy intuitiva, lo que ha facilitado el desarrollo del frontend en poco tiempo. Además, se adapta fácilmente a librerías de estilos como Bootstrap lo que permite un diseño rápido de la página.
- **JavaScript:** a pesar de haber cursado IISSI 2, por desgracia no recordaba con precisión cómo se maneja JavaScript para utilizarlo en el frontend, por lo que parte de este proyecto se ha dedicado al correcto uso de este lenguaje para desarrollar el trabajo.

1.5. Estructura del Documento

Este documento está dividido en varios apartados y en cada uno de ellos se expondrán contenidos relevantes de cada fase por las que ha pasado el proyecto. A medida que se vaya avanzando en esta memoria, se irá incrementando el carácter técnico del contenido explicado.

El primer apartado es la **Introducción**. En él se describe el resumen del trabajo, la motivación que lleva a su desarrollo, los objetivos y conocimientos previos, entre otros aspectos. Este apartado sirve fundamentalmente para introducir al lector a la problemática que pretende resolver este proyecto.

El segundo apartado estará centrado en las **Materias Relacionadas**, en el que se explicarán todas las herramientas que han sido adoptadas para la realización del proyecto, ya sean de desarrollo (por ejemplo Visual Studio Code) o de gestión (como GitHub). También se detallarán las metodologías que se han seguido durante la planificación del proyecto y la fase de ejecución de este.

Tras explicar las metodologías, en el tercer apartado se muestra la **Planificación** del proyecto, por tanto aquí se detalla el análisis temporal y se estiman los costes que ha conllevado el proyecto en base a las herramientas y recursos utilizados en el desarrollo. También se especifica el análisis de los requisitos del cliente para con la página web que se tiene que desarrollar y el diseño que va a ser aplicado.

En el cuarto apartado se explicará cuál ha sido el **Sistema Desarrollado**, es decir, cómo se ha construido la aplicación. Este apartado recogerá todos los problemas que han ido surgiendo a lo largo del desarrollo y la solución final que se adoptó para resolverlos. También se incluyen tres apartados más relativos al desarrollo, como son la **Implementación**, las **Pruebas** llevadas a cabo sobre el sistema y el **Despliegue** de las tres partes que conforman el trabajo.

El octavo apartado está dedicado a albergar el **Manual de usuario** que detalla cómo se debe utilizar la aplicación y las funcionalidades que contiene. Estos manuales están orientados para servir como guía a los usuarios del sistema.

Por último, el noveno apartado se reserva para el **Cierre** del proyecto, en el que se exponen las conclusiones del trabajo, donde se valorarán las lecciones aprendidas y daré mi opinión sobre el proyecto. Hay un espacio de este apartado en el que se encuentra la bibliografía y un glosario de términos, con los conceptos más usados en esta memoria.

Para distinguir aún mejor cada apartado y definir una estructura más clara, el trabajo estará dividido en tres partes. **Preámbulo**, que contiene los tres primeros apartados destinados a introducción, análisis y planificación; **Desarrollo**, que contendrá los apartados cuarto, quinto, sexto, séptimo y octavo dedicados a aspectos de la implementación y ejecución del proyecto; y **Desenlace**, que sólo dispondrá de un apartado, el cierre del proyecto.

Materias Relacionadas

En esta sección se presentan todas las herramientas que he utilizado a lo largo de la planificación y desarrollo del proyecto, diferenciando entre herramientas de gestión, herramientas de desarrollo, herramientas de pruebas y metodologías empleadas.

2.1. Herramientas de Gestión

La elección de herramientas que te ayuden a gestionar adecuadamente un proyecto durante su desarrollo es crucial. El éxito del trabajo está directamente influenciado por una buena gestión. De igual manera, existen herramientas de gestión que nos hacen la tarea mucho más fácil y así trabajar de una forma más cómoda.

Las herramientas de gestión utilizadas en este trabajo han sido:

- **Clockify:** herramienta que nos permite llevar un control exhaustivo del tiempo invertido en las tareas del proyecto y generar informes detallados con el tiempo que han llevado la realización de cada una de esas tareas.

Para utilizarla, primero debes crear un proyecto al que le irás asignando tareas y una descripción si es necesario, luego pulsas “Iniciar” y ya estarás registrando el tiempo. Una vez finalices la tarea o simplemente quieras dejar de contar el tiempo, pulsas en “Detener” y se creará una entrada de tiempo con las horas, minutos y segundos que ha estado funcionando el cronómetro. En caso de que desees modificar a posteriori el tiempo que has invertido, puedes hacerlo en cualquier momento haciendo click sobre el contador de horas totales de la tarea o cambiando la hora de inicio y fin de esta.

- **GitHub:** herramienta que nos permite alojar cualquier tipo de fuente de caracteres, ya bien sea código o documentación, que utiliza el sistema de control de versiones Git.

Este sistema de almacenamiento en la nube cuenta con tres factores muy importantes que hacen que destaque por encima de sus competidores:

- Puedes trabajar en remoto desde cualquier ubicación y en cualquier dispositivo, ya que GitHub está alojado en Internet.
- Contiene los últimos cambios que subiste desde la última vez que se sincronizaron los cambios con tu repositorio en local, por lo que puedes analizar cuáles han sido esos cambios y localizar errores en ficheros de una forma más rápida.
- Su función GitHub Actions es muy interesante para trabajos en los que hay varios miembros en un grupo y quieren comprobar que la aplicación se integra correctamente, por tanto, el proceso de integración continua configurable mediante archivos YAML de GitHub Actions es fundamental. Además se pueden programar despliegues automáticos en función de las etiquetas que se

asocian a las peticiones de fusión generadas en los repositorios, por lo que simplifica este proceso software.

2.2. Herramientas de Desarrollo

En este apartado citaré las tecnologías escogidas para el desarrollo de la aplicación, desde el backend (en forma de API REST) hasta el frontend (como una aplicación web responsive). Antes de comenzar las herramientas utilizadas, me gustaría aclarar cuál ha sido el entorno de desarrollo que se ha usado para implementar tanto el backend como el frontend:

- IDE: Visual Studio Code

Aunque se podría considerar más un editor de texto con múltiples funcionalidades y plugins que permiten ejecutar código, se puede admitir que este es uno de los editores más utilizados por la comunidad de desarrolladores por su interfaz sencilla y simple; y por la cantidad de extensiones que se le pueden agregar para hacer de la experiencia de la programación, una sensación más agradable.

En mi caso, lo he equipado con extensiones para Python y Vue.js, de forma que me ofrezca una asistencia inteligente en cada sentencia de código que he escrito. También he utilizado su sección de control de versiones para sincronizar los cambios del proyecto, haciéndolo de una forma más intuitiva que con la terminal integrada que provee y Git. Por último, otra de las herramientas incorporadas para el desarrollo ha sido GitHub Copilot que me ha ayudado mucho a aprender más rápido cómo trabajar con Vue.js y a solucionar errores que habría tardado horas o días en resolver manualmente; aunque cabe destacar que este asistente no es perfecto y más de una vez se ha equivocado con las respuestas que me ha ofrecido.

Seguidamente se detallarán las tecnologías y herramientas que se han empleado para el desarrollo del backend y la justificación de su elección:

- PostgreSQL

Es un sistema de base de datos [1] que ya ha sido utilizado en varias asignaturas en este último curso del grado y por la familiaridad y la forma tan simple en que funciona ha sido el sistema elegido. Además, es un sistema fácil de integrar en un proyecto Django, lo que lo hacía el candidato idóneo para la aplicación, teniendo en cuenta que en su posterior puesta en producción, es fácilmente mantenible y su gestión de las migraciones es personalizable en caso de modificar los archivos correspondientes.

Elegir una base de datos que no fuera relacional en el caso de nuestra aplicación tendría poco sentido, sabiendo a priori que no se necesitan un gran número de clases para definir los modelos; pero esto será definido más adelante en el apartado de **Sistema Desarrollado**.

- Django

Es el framework de backend [2] que se ha utilizado para gestionar toda la lógica de la aplicación en la parte del servidor, conectándose con la base de datos definida

en PostgreSQL y limitando el acceso a determinadas funciones de la aplicación mediante la integración del doble factor de autenticación.

El principal motivo por el que se ha elegido este y no otro framework es fundamentalmente el uso de este en varias asignaturas del primer cuatrimestre del cuarto curso y la implementación de varios proyectos con esta tecnología. Al comienzo del desarrollo de este proyecto pensaba utilizar Spring Boot como framework de backend debido a la familiaridad que suponía haber trabajado con él durante todo el tercer curso del grado; sin embargo mi tutor me recomendó utilizar Django ya que es una herramienta que aunque tenga una curva de aprendizaje un poco empinada al inicio, es una de las mejores opciones para implementar un proyecto si se dispone de relativamente poco tiempo para desarrollar.

Además de esto, los dos proyectos que se desarrollaron en Django durante el primer cuatrimestre del cuarto curso me han permitido avanzar a gran velocidad en el transcurso de la implementación, aportando conocimientos en la serialización de los datos y una estructura sólida sobre la que trabajar desde un principio, sin necesidad de implementar un proyecto desde cero que habría sido más tedioso, aumentando la dificultad de este trabajo.

- **JSON**

Es un formato de texto ligero para el intercambio de datos [3], la mayor ventaja que supone utilizar JSON para el intercambio de datos es que es sencillo de comprender para el programador y simple de interpretar para una máquina, por lo que lo hace la herramienta perfecta para el intercambio de información mediante una API como en el caso del proyecto. Se utiliza tanto en el frontend, como para gestionar los datos que se obtienen de la aplicación por la parte del cliente.

- **Django Two-Factor Authentication**

Aunque se ha integrado en el proyecto usando un proyecto de ejemplo de GitHub, la autenticación de doble factor de Django [4] ha sido una herramienta empleada para mejorar la seguridad de la aplicación desarrollada, siendo fundamental tenerla activada para poder realizar un pedido, ver los pedidos y realizar todo tipo de acciones esenciales en la aplicación, exceptuando la visualización de los productos disponibles.

La implementación de esta tecnología en mi proyecto se basa en una recomendación por parte de mi tutor, con el fin de mejorar la seguridad del proyecto y brindarme la oportunidad de aprender a implementar un nuevo mecanismo de seguridad que no había explorado durante mis estudios.

Considero que ha sido una decisión importante tener en cuenta este tipo de características en un proyecto software, sabiendo que la autenticación en una aplicación puede suponer un grave problema de seguridad y que a largo plazo puede conllevar sanciones económicas si ocurre algún tipo de filtración de los datos.[JD3]

La ventaja fundamental de utilizar un proyecto en el que ya está implementada la infraestructura necesaria para que funcione correctamente, radica en esencia en la facilidad de la personalización de ciertos aspectos, como la redirección tras iniciar sesión con el doble factor de autenticación, y en el tema de las plantillas HTML para poder configurarlas y darles el aspecto que se necesite adaptando la temática de los colores y las imágenes al gusto del desarrollador sin tener que crear toda la

infraestructura desde cero, que es lo que en un principio suscita la documentación oficial a llevar a cabo. De hecho, en un primer momento se intentó implementar el doble factor usando sólo la librería referenciada más arriba, pero fue prácticamente imposible saber en qué directorio tenía que modificar qué archivo para que las páginas se pudieran personalizar como quería. La documentación es más una guía para configurar e instalar el doble factor, pero no para montar desde cero la infraestructura si es lo que se necesita en el proyecto en el que se va a integrar.

- **Postman Agent**

Es una herramienta que se utiliza para lanzar peticiones HTTP a cualquier API. En ocasiones se ha usado para comprobar las respuestas a ciertas peticiones, sobre todo de inicio de sesión con el token de autenticación provisto por parte del doble factor de Django, que al realizar las pruebas de carga devolvía errores que no acababa de comprender, así que me dispuse a averiguar el porqué de dichos errores haciendo uso de Postman (los resultados de estas pruebas se analizan en la sección de **Pruebas de carga**).

Tras describir las herramientas que han intervenido en la creación del backend pasaré a detallar las herramientas utilizadas en la implementación del frontend:

- **Node.js**

Es un entorno de ejecución de un solo hilo, basado en el lenguaje JavaScript. Se utiliza para crear aplicaciones de red y del lado del servidor rápidas y escalables [4]. Aunque es posible crear un backend con Node.js, en mi caso lo he utilizado para ejecutar el framework de frontend que he empleado para el desarrollo de la vista de la página web.

- **Vue.js + Vite**

Es un framework “progresivo” de JavaScript utilizado para construir interfaces de usuario interactivas y de una sola página (SPA). Se centra en la capa de vista de una aplicación web y facilita la creación de componentes reutilizables, lo que permite una estructura modular y mantenible del código. Vue.js es ideal tanto para proyectos pequeños como para aplicaciones web de gran escala, debido a que utiliza una sintaxis fácil de entender y su curva de aprendizaje es relativamente suave. Se dice de Vue.js que es un framework “progresivo” porque es altamente adaptable y puede integrarse fácilmente con otras bibliotecas y herramientas. [6]

Por otro lado, Vite [7] es una herramienta de desarrollo de frontend ultrarrápida y minimalista diseñada para proyectos basados en Vue.js y React. Durante el desarrollo, los módulos ECMAScript de ESM se cargan rápidamente, lo que reduce los tiempos de compilación y recarga en comparación con otras herramientas como Webpack. Además, la experiencia de desarrollo de Vite es en tiempo real, lo que significa que los cambios en el código se muestran en el navegador sin recargar la página. Esto hace que sea una opción muy utilizada entre los desarrolladores que buscan programar de forma rápida y eficiente sus proyectos.

Las razones por las que se ha escogido Vue.js y no React o Angular son las siguientes:

- A la hora de implementar el frontend en mi opinión utilizar React no estaba entre mis opciones ya que se ha utilizado en otras asignaturas en el grado

(concretamente en IISSI 2) y buscaba un framework que no hubiera empleado antes para conocer nuevas tecnologías de desarrollo.

- Por otra parte, quería aprender un nuevo framework cuya curva de aprendizaje fuera lo más suave posible, teniendo en cuenta la magnitud de este proyecto y que cualquier traba que me pusieran las herramientas a la hora de plasmar mis ideas sobre la aplicación podía suponerme una gran inversión de tiempo que no estaba dispuesto a asumir. Al comentar las posibles opciones que había en el mercado con mi tutor, me recomendó utilizar Vue.js al ser un framework que trabaja en sus componentes con “todo en uno”, es decir, que en el mismo componente se pueden definir estilos en CSS, HTML y código en JavaScript para manejar los datos de backend; y sobre todo la facilidad con la que se integra con multitud de librerías lo hacía el framework perfecto para desarrollar el frontend del proyecto.
- Además, según estuve investigando de cara a tomar la decisión, en la mayoría de las páginas web insistían en la rigidez de Angular para ciertos aspectos en el desarrollo, por ejemplo, la variedad de estructuras que tiene como componentes, pipe e inyectables, además de una sintaxis compleja dificultan el aprendizaje. [8][9]
- Adicionalmente, el trabajo desempeñado en otras asignaturas con HTML ha facilitado en gran medida el uso de Vue.js en el proyecto, ya que la dificultad radicaba en recordar el uso de JavaScript más que en la comprensión de la estructura de Vue.js.

2.3. Herramientas de Pruebas

Para ir finalizando los apartados de herramientas, una vez desarrolladas ciertas partes de la aplicación, se describirán las tecnologías empleadas en la realización de pruebas que comprueben el correcto funcionamiento del sistema:

- **Locust**

Es una herramienta de prueba de carga/rendimiento de código abierto basado en Python para HTTP y otros protocolos [10]. En este proyecto se ha utilizado para comprobar la resistencia a peticiones de la API creada con Django Rest Framework.

- **Selenium**

Es una herramienta de automatización de pruebas para aplicaciones web. [11] Concretamente se centra en realizar pruebas de interfaz gráfica, haciendo clicks en los botones que se quieran pulsar o rellenando formularios, entre otras muchas tareas. Con este software se han realizado diversas pruebas sobre todo en el módulo de los pedidos, en el que también interviene la presencia de una persona para poder agregarle la información del doble factor necesaria para realizar las acciones que se necesitaban probar. Para más detalles acerca de estas pruebas, véase la sección de **Pruebas de vista dinámicas**. [JD4][JG5]

2.4. Metodologías

La metodología que se va a emplear para llevar a cabo la ejecución del proyecto, es decir, la parte de implementación de las tareas que se tendrán que realizar para entregar una solución funcional, va a ser SCRUM aunque adaptado a una sola persona. Normalmente SCRUM es una metodología ágil que se suele aplicar en grandes proyectos donde hay un equipo de trabajo formado por más de cinco personas; y como es evidente, en este trabajo sólo ha participado una, por lo que la mayoría de la documentación asociada a SCRUM (Sprint Review, Retrospective y Backlog) no formará parte de la documentación que se entregará al tutor del proyecto, sino que se le facilitarán informes de seguimiento durante las reuniones de seguimiento del trabajo, para informar de la evolución del trabajo.

Para poder desarrollar el proyecto de una forma más intuitiva y organizada se ha seguido una filosofía de ramas GitFlow muy simplificada, de manera que se ha creado una rama por tarea a implementar, entendiendo que cada tarea conllevaba varias subtareas que por pura simplificación se han omitido en la creación de “issues” y así mantener un repositorio más limpio para una sola persona. En el ámbito de los “commits” llevados a cabo, se ha seguido una estructura estandarizada para evitar que se fuera de control cualquier cambio relacionado con el código. Dicha estructura se representa a continuación:

[Título descriptivo] #Issue asociada al commit (si la hay)

Breve descripción (si es necesario)

Además, para la planificación se va a utilizar PMBOK [12], un conjunto de buenas prácticas en la ingeniería de software que contiene una hoja de ruta con la documentación imprescindible que un proyecto debería tener para que al ejecutarlo, sea un éxito en tiempo y coste. En ese libro se especifica que un proyecto debe tener un Plan de Dirección del Proyecto que a su vez incluye un Plan de Gestión de la Integración, del Alcance, del Cronograma, de los Costes, de la Calidad, de los Recursos, de las Comunicaciones, de los Riesgos, de las Adquisiciones y de los Interesados. Dado el alcance de este proyecto, se tendrán en cuenta aquellos planes que tengan sentido aplicar a un trabajo de este tamaño (contemplar por ejemplo el Plan de Gestión de las Adquisiciones en un proyecto como este no tiene mucho sentido).

La versión de PMBOK con la que se trabajará será la sexta edición, publicada en el año 2017 y con la que el autor del proyecto ha trabajado durante una parte de los estudios.

Planificación

En este capítulo se describirá cómo se ha organizado en todos los aspectos el proyecto, desde una recopilación de los requisitos hasta el análisis temporal y de costes que tendría este. Teniendo en cuenta que se va a emplear PMBOK, se seguirá en la medida de lo posible el orden en que esta metodología define un proyecto, desarrollando los planes de gestión correspondientes.

Desde un punto de vista muy amplio, se puede definir el esfuerzo temporal que se desea llevar a cabo para desarrollar el proyecto. En principio, se estima que el proyecto se realice durante cinco meses, siendo estos enero, febrero, marzo, abril y mayo. Los tres primeros meses estarían destinados a cubrir en su totalidad el código que resolverá la problemática que se ha establecido en el objetivo de este proyecto y los dos últimos están planteados para la realización de esta memoria.

También se pueden establecer a muy alto nivel los riesgos más importantes que se pueden registrar a priori, como lo son el hecho de no cumplir en tiempo y coste con las metas establecidas para con este proyecto debido a la gran inversión sobre todo en tiempo que requerirá el aprendizaje de la nueva tecnología y/o los posibles errores con los que se puede uno topar a lo largo del desarrollo. Por supuesto, otro de los grandes riesgos que conlleva un proyecto de esta magnitud es el cambio de los requisitos por parte del cliente. Sin embargo, al tratarse de un proyecto cuyo ámbito se puede restringir en principio por las funcionalidades que serán recopiladas más adelante (extraídos de la conversación que tuve con mi familiar), el proyecto se podría alargar todo lo que uno quisiera, si se cuentan con los recursos en tiempo y coste necesarios.

A continuación se intercalarán los planes de gestión que recoge PMBOK junto con la recopilación y análisis de los requisitos establecidos por el cliente.

3.1. Plan de Gestión de las Comunicaciones

El propósito de este plan es asegurar que la información relevante del proyecto se comunique de manera adecuada y efectiva a todas las partes interesadas, lo que contribuye a la gestión exitosa del proyecto y a la satisfacción de las necesidades de las partes interesadas.

En el siguiente cuadro se pueden observar los datos que se ofrecerán a las partes interesadas, la periodicidad de entrega de la información y los entregables o entregables que se esperan (véase el cuadro 3.1).

Nota: Entiéndase al Patrocinador, como al cliente de la aplicación, en el caso del proyecto sería el tutor, ya que es éste el que aprueba la evolución de la aplicación en cada seguimiento.

Información	Interesado(s)	Medio	Momento o Periodicidad	Entregable(s)	Observaciones
Grado de cumplimiento de objetivos del proyecto	Patrocinador	Correo electrónico	Cada 15 días	Documentación que requiere de aprobación.	La documentación será enviada por correo electrónico al patrocinador.
Contingencias detectadas	Patrocinador	Correo electrónico	Comunicación puntual		

Cuadro 3.1: Información correspondiente a cada interesado

3.2. Análisis de Requisitos

En este apartado se va a realizar una recopilación de los requisitos y su posterior análisis para definir el alcance del proyecto y comenzar a vislumbrar la planificación temporal y financiera que tendrá este. No obstante, cabe destacar la importancia de la definición del contexto en el que se fundamenta la aplicación, para realmente ensalzar el valor que aporta esta aplicación para con los objetivos que se han establecido previamente.

3.2.1. Clases y características de los usuarios

En la aplicación se distinguen claramente dos tipos de usuarios:

- Clientes

Como su propio nombre indica, los usuarios clientes harán uso de la aplicación para realizar pedidos según el stock que haya en función de los productos que pidan y también podrán consultar el estado de sus pedidos.

- Administrador(es)

Es el usuario que está destinado a gestionar la página web. Tendrá acceso a todos los datos tanto de los productos y pedidos, como de los clientes. El objetivo de la aplicación es que los administradores sean los que en esencia, creen y editen los productos y las categorías de los productos, y actualicen el estado de los pedidos una vez pagados (se contemplaría la opción de pago en efectivo, por lo que es necesario para llevar la contabilidad de la empresa que quede registrado el pago de las cuantías de los pedidos que no se han efectuado con tarjeta).

Está en plural, porque no necesariamente tiene por qué haber un solo administrador en la empresa, dando la posibilidad de que cada empresario administre los permisos de los usuarios administradores del sistema como desee.

3.2.2. Funcionalidades del sistema

Una vez definido cada perfil de usuario al que está destinado la aplicación, se va a proceder a la introducción de las funcionalidades que el sistema debe ofrecer a

cada uno de ellos a un nivel más profundo, sin entrar en detalles tan específicos (para ello estaría la sección de **Recopilación de Requisitos**).

- Clientes:
 - o Realizar pedidos.
 - o Consultar el historial de pedidos.
 - o Consultar el estado de los pedidos.
- Administradores:
 - o Crear nuevos productos y categorías.
 - o Modificar productos y categorías.
 - o Eliminar productos y categorías.
 - o Mostrar estadísticas del sistema.
 - o Realizar seguimiento de pedidos.
 - o Modificar pedidos.
 - o Eliminar pedidos.

3.2.3. Descripción actual del sistema

En el sistema que utilizan en la panadería, la gestión de pedidos se realiza exclusivamente mediante la comunicación directa con uno de los dos panaderos que reparten pan o bien en la propia panadería. Esto quiere decir que cualquier cambio que deseen realizar, se lo tienen que hacer saber a la panadera en persona o utilizando una aplicación de mensajería como “WhatsApp”.

Además la contabilidad la llevan a cabo mediante una libreta en la que tienen apuntadas todas las cuentas de la panadería, lo que hace muy difícil saber en tiempo real quién ha abonado los importes de sus pedidos y quién no; teniendo en cuenta que la panadera por sí misma no avisa de las cantidades que tienen que abonar los clientes, simplemente estos le facilitan una cantidad en efectivo, cuando se efectúa la entrega del pedido, que puede ser o no correspondida a la que deben a la panadera y de ahí va cobrando esta; o bien se tienen que acercar a la panadería y preguntarle cuánto le deben.

- Análisis DAFO del sistema actual

Es muy recomendable vislumbrar de forma esquemática cuáles son las deficiencias y amenazas que supone para la empresa el sistema de gestión actual, del que se puede asegurar que es muy rudimentario. Además, conocer cuáles son las fortalezas de nuestra solución software y las oportunidades que presenta, es una manera sumamente útil de prever algunos de los requisitos que formarán parte de la aplicación a posteriori. A continuación, se muestra el análisis realizado:

- Debilidades:
 - La comunicación de cambios en los pedidos y/o de realización de estos implica pérdidas de tiempo tanto para el cliente como para la panadera y puede provocar irregularidades o errores en las entregas.
 - Posibilidad de impagos por parte de los clientes.
- Amenazas:
 - Los clientes pueden querer cambiar de panadera, por el simple hecho de que otros proveedores faciliten la realización de los pedidos y la comunicación sea más sencilla.
 - Los impagos pueden provocar en ciertos momentos, que la panadería no reciba los ingresos adecuados al gasto que estén teniendo y llevarla a un punto financiero delicado, pudiendo llegar a cerrar si no realizan el pago del alquiler del local o de las materias primas a su debido tiempo.
 - Sensación de hastío de clientes y empleados.
- Fortalezas:
 - Es fácil implantar un nuevo sistema debido a que si se mejoran las gestiones en la panadería, los ingresos serían estables, la panadera no tendría que hacer un esfuerzo tan importante por cuadrar las cuentas usando métodos rudimentarios para ello y los clientes podrían realizar los pedidos de una forma más cómoda.
- Oportunidades:
 - La implementación de la aplicación desarrollada tratará de reforzar las debilidades, reduciendo en la medida de lo posible el número de amenazas.
 - Para aquellos clientes que tengan un buen manejo de las nuevas tecnologías, podrían recomendar esta panadería a otros conocidos, haciendo crecer el negocio de manera progresiva.

3.2.4. Recopilación de Requisitos

A continuación, con el objetivo de definir detalladamente las características del sistema final, se llevará a cabo un proceso de clasificación, documentación, priorización y validación de los requisitos recopilados que se desean incluir en el proyecto. Para su clasificación se ha optado por la creación de varias categorías, en concreto: Funcional, No Funcional, de Seguridad, de Privacidad y de Portabilidad.

Los requisitos se encuentran en el cuadro 3.3, clasificados según las categorías definidas, además de incluir la prioridad del requisito, su criterio de aceptación y el interesado. La priorización de los requisitos sería Baja (requisitos relacionados con el seguimiento y otros requisitos para mejorar la experiencia de usuario), Media (requisitos relacionados con el pago, el carrito de la compra y la administración de la página) y Alta (requisitos relacionados con la tienda y el perfil de usuario).

Para seguir la trazabilidad de requisitos con los entregables se han relacionado estos en la matriz de trazabilidad de requisitos, aportando además una métrica y un método de validación del requisito (véase el cuadro 3.4).

PD	Requisito del producto
PY	Requisito del proyecto

Cuadro 3.2: Leyenda de siglas

ID DEL REQUISITO	CATEGORÍA	REQUISITO	PRIORIDAD	CRITERIO DE ACEPTACIÓN	INTERESADO
PD01	Requisito Funcional	El carrito de la compra siempre estará visible (siempre que se pueda).	Media	Se ve el carrito de la compra.	Patrocinador
PD02	Requisito Funcional	El carrito de la compra dispondrá de un mecanismo simple para que el usuario pueda ampliar o reducir el número de unidades de los productos del pedido.	Media	Se pueden añadir o quitar productos del carrito.	Patrocinador
PD03	Requisito Funcional	Los productos de la tienda estarán organizados por las mismas categorías que utiliza el cliente en sus tiendas físicas.	Alta	Existen categorías dentro de la web para clasificar los productos.	Patrocinador
PD04	Requisito Seguridad	El cliente debe sentirse seguro durante la compra.	Media	Protección de los datos en todo momento.	Patrocinador
PD05	Requisito Funcional	El cliente será atendido en español.	Alta	Idioma de la página en español.	Patrocinador
PD06	Requisito Funcional	La identificación del usuario en el registro se hará utilizando un usuario y una clave.	Alta	Acceso mediante usuario y contraseña	Patrocinador
PD07	Requisito Funcional	El cliente tiene acceso directo al seguimiento de su pedido.	Baja	Posibilidad de seguimiento.	Patrocinador
PD08	Requisito Funcional	Los productos agotados están claramente marcados.	Media	Existe una marca que lo indica.	Patrocinador
PD09	Requisito Funcional	Cada producto o servicio vendido en la tienda dispondrá de sólo una imagen.	Alta	Sólo una imagen por producto.	Patrocinador
PD10	Requisito Funcional	En la tienda podremos enviar productos al carrito	Media	Posibilidad de añadir o quitar	Patrocinador

		de la compra, indicando la cantidad.		productos del carrito en la tienda.	
PD11	Requisito Funcional	Desde la tienda podremos revisar el estado del carrito.	Media	Carrito siempre disponible desde la tienda.	Patrocinador
PD12	Requisito Funcional	Desde el carrito de la compra podremos finalizar la compra.	Media	Finalizar la compra en el carrito.	Patrocinador
PD13	Requisito Funcional	Durante el proceso de registro se solicitarán los datos del cliente.	Media	Acceso a los datos del cliente para realizar la compra.	Patrocinador
PD14	Requisito Funcional	Durante el proceso de registro se solicitarán los datos de envío de todos los pedidos que se realicen.	Media	Acceso a datos de envío.	Patrocinador
PD15	Requisito Funcional	Durante el proceso de compra se solicitarán los datos de pago (directamente o iniciando sesión si está registrado).	Media	Acceso a datos de pago (pasarela de pago).	Patrocinador
PD16	Requisito Funcional	Al finalizar el proceso de compra el cliente recibe un correo con los datos del producto comprado, el importe y la dirección de entrega.	Baja	Llega un correo con la información del pedido una vez realizada la compra.	Patrocinador
PD17	Requisito Funcional	No se contempla la devolución de productos.	Baja	Una vez comprado, no se puede devolver el producto.	Patrocinador
PD18	Requisito Funcional	La marca corporativa de la empresa cliente debe reflejarse en el sitio web.	Media	Marca visible en el sitio web.	Patrocinador
PD19	Requisito Funcional	El usuario puede registrarse en la solución y facilitar la dirección de entrega de los pedidos, además del código postal y acceder utilizando un usuario y una clave de acceso.	Alta	Registro mediante correo y contraseña.	Patrocinador
PD20	Requisito Funcional	El usuario puede cambiar su contraseña.	Alta	Cambio de contraseña intuitivo.	Patrocinador
PD21	Requisito Funcional	Aquellos usuarios que no activen el mecanismo de seguridad para su cuenta durante el registro tendrán la	Alta	Los usuarios pueden activar el mecanismo desde su perfil.	Patrocinador

		posibilidad de activarlo a posteriori para que pueda realizar pedidos			
PD22	Requisito Funcional	Seguimiento de pedidos por ID de pedido.	Baja	Se puede hacer un seguimiento de los pedidos mediante su ID.	Patrocinador
PD23	Requisito Funcional	Seguimiento de pedidos, edición y borrado por el administrador.	Baja	El administrador puede hacer el seguimiento de los pedidos, editarlos (entregado, pagado, método de envío y de pago) y eliminarlos, sin posibilidad de devolución de los productos.	Patrocinador
PD24	Requisito Funcional	Creación, edición y borrado de productos por el administrador.	Media	El administrador puede crear, editar o eliminar productos de la tienda.	Patrocinador
PD25	Requisito Funcional	Creación, edición y borrado de categorías por el administrador.	Media	El administrador puede crear, editar o eliminar categorías de la tienda.	Patrocinador
PD26	Requisito Funcional	Visualización de estadísticas de la tienda (productos más vendidos, usuarios que más pedidos realizan).	Baja	El administrador puede analizar los productos más vendidos y los usuarios que más pedidos realizan.	Patrocinador
PD27	Requisito Funcional	Debe de haber una sección con los productos destacados de la tienda.	Baja	Se muestran los productos destacados en la página principal.	Patrocinador
PD28	Requisito Seguridad	Se desea que el sistema disponga de un mecanismo de seguridad reforzado para evitar suplantaciones de identidad por parte de los administradores o de los usuarios.	Alta	Se impide el acceso a los pedidos a los usuarios que no tengan activado el mecanismo de seguridad.	Patrocinador
PD29	Requisito Funcional	Métodos de pago: Pasarelas de pago.	Media	Se puede realizar el pago mediante una pasarela de pago.	Patrocinador
PD30	Requisito Funcional	Métodos de pago: pago contra-reembolso.	Media	Los clientes pueden seleccionar la opción de pagar contra-reembolso.	Patrocinador

PD31	Requisito Funcional	Formas de entrega: entrega estándar o recoger en tienda física.	Baja	Los clientes pueden seleccionar la forma de entrega al realizar un pedido.	Patrocinador
PD32	Requisito Funcional	Página con información de la empresa cliente.	Baja	Existe una página que muestra información de la empresa cliente.	Patrocinador
PD33	Requisito No Funcional	Los datos de contacto y ubicación de la empresa cliente se debe mostrar siempre.	Baja	Existe un espacio en toda la página donde se puede ver esta información claramente.	Patrocinador
PD34	Requisito No Funcional	La página web tiene que poder ser utilizable en dispositivos móviles.	Baja	Se puede manejar con facilidad la página en un móvil o tableta.	Patrocinador
PY01	Requisito Portabilidad	El producto final debe entregarse como un contenedor de aplicaciones con las instrucciones de instalación y puesta en producción.	Baja	Entrega de un contenedor con la aplicación y las instrucciones para la instalación.	Patrocinador

Cuadro 3.3: Registro de requisitos

Información de Requisito					Relación de Trazabilidad con...				
ID	Requisito	Prioridad	Categoría	Interesado	Objetivo(s) del Proyecto	Entregable(s) de EDT	Criterio(s) de Aceptación	Métrica(s)	Validación
PD 01	El carrito de la compra siempre estará visible (siempre que se pueda).	Mediana	Requisito Funcional	Patrocinador	Carrito de la compra	1.2	Se ve el carrito de la compra.	No aplica	Mediante Criterios de Aceptación
PD 02	El carrito de la compra dispondrá de un mecanismo simple para que el usuario pueda ampliar o reducir	Mediana	Requisito Funcional	Patrocinador	Carrito de la compra	1.2	Se pueden añadir o quitar productos del carrito.	No aplica	Mediante Criterios de Aceptación

	el número de unidades de los productos del pedido.								
PD 03	Los productos de la tienda estarán organizados por las mismas categorías que utiliza el cliente en sus tiendas físicas.	Alta	Requisito Funcional	Patrocinador	Catálogo de productos	1.2	Existen categorías dentro de la web para clasificar los productos.	No aplica	Mediante Criterios de Aceptación
PD 04	El cliente debe sentirse seguro durante la compra.	Mediana	Requisito Seguridad	Patrocinador	Seguridad	1.2	Protección de los datos en todo momento.	No aplica	Mediante Criterios de Aceptación
PD 05	El cliente será atendido en español.	Alta	Requisito Funcional	Patrocinador	Accesibilidad	1.1, 1.2, 1.3	Idioma de la página en español.	No aplica	Mediante Criterios de Aceptación
PD 06	La identificación del usuario en el registro se hará utilizando un usuario y una clave.	Alta	Requisito Funcional	Patrocinador	Gestión de usuarios	1.1	Acceso mediante usuario y contraseña	No aplica	Mediante Criterios de Aceptación
PD 07	El cliente tiene acceso directo al seguimiento de su pedido.	Baja	Requisito Funcional	Patrocinador	Seguimiento de pedidos	1.3	Posibilidad de seguimiento.	No aplica	Mediante Criterios de Aceptación

PD 08	Los productos agotados están claramente marcados.	Medi a	Requi sito Funci onal	Patroci nador	Catálog o de product os	1.2	Existe una marca que lo indica.	No aplica	Media nte Criteri os de Acept ación
PD 09	Cada producto o servicio vendido en la tienda dispondrá de sólo una imagen.	Alta	Requi sito Funci onal	Patroci nador	Catálog o de product os	1.2	Sólo una imagen por producto.	No aplica	Media nte Criteri os de Acept ación
PD 10	En la tienda podremos enviar productos al carrito de la compra, indicando la cantidad.	Medi a	Requi sito Funci onal	Patroci nador	Carrito de la compra	1.2	Posibilidad de añadir o quitar productos del carrito en la tienda.	No aplica	Media nte Criteri os de Acept ación
PD 11	Desde la tienda podremos revisar el estado del carrito.	Medi a	Requi sito Funci onal	Patroci nador	Carrito de la compra	1.2	Carrito siempre disponible desde la tienda.	No aplica	Media nte Criteri os de Acept ación
PD 12	Desde el carrito de la compra podremos finalizar la compra.	Medi a	Requi sito Funci onal	Patroci nador	Carrito de la compra	1.2	Finalizar la compra en el carrito.	No aplica	Media nte Criteri os de Acept ación
PD 13	Durante el proceso de registro se solicitarán los datos	Medi a	Requi sito Funci onal	Patroci nador	Compra	1.1	Acceso a los datos del cliente para realizar la compra.	No aplica	Media nte Criteri os de Acept ación

	del cliente.								
PD 14	Durante el proceso de registro se solicitará n los datos de envío de todos los pedidos que se realicen.	Medi a	Requi sito Funci onal	Patroci nador	Compra	1.1	Acceso a datos de envío.	No aplica	Media nte Criteri os de Acept ación
PD 15	Durante el proceso de compra se solicitará n los datos de pago (directa mente o iniciando sesión si está registrad o).	Medi a	Requi sito Funci onal	Patroci nador	Compra	1.3	Acceso a datos de pago (pasare la de pago).	No aplica	Media nte Criteri os de Acept ación
PD 16	Al finalizar el proceso de compra el cliente recibe un correo con los datos del producto comprad o, el importe y la dirección de entrega.	Baja	Requi sito Funci onal	Patroci nador	Compra	1.3	Llega un correo con la informa ción del pedido una vez realiza da la compra .	No aplica	Media nte Criteri os de Acept ación
PD 17	No se contemp la la devoluci ón de	Baja	Requi sito Funci onal	Patroci nador	Gestión de reclama ciones	1.3	Una vez compra do, no se puede	No aplica	Media nte Criteri os de Acept ación

	productos.						devolver el producto.		
PD 18	La marca corporativa de la empresa cliente debe reflejarse en el sitio web.	Mediana	Requisito Funcional	Patrocinador	Calidad	1.2	Marca visible en el sitio web.	No aplica	Mediante Criterios de Aceptación
PD 19	El usuario puede registrarse en la solución y facilitar la dirección de entrega de los pedidos, además del código postal y acceder utilizando un usuario y una clave de acceso.	Alta	Requisito Funcional	Patrocinador	Gestión de usuarios	1.1	Registro mediante correo y contraseña.	No aplica	Mediante Criterios de Aceptación
PD 20	El usuario puede cambiar su contraseña.	Alta	Requisito Funcional	Patrocinador	Gestión de usuarios	1.1	Cambio de contraseña intuitivo.	No aplica	Mediante Criterios de Aceptación
PD 21	Aquellos usuarios que no activen el mecanismo de seguridad para su cuenta durante el registro	Alta	Requisito Funcional	Patrocinador	Seguridad	1.1	Los usuarios pueden activar el mecanismo desde su perfil.	No aplica	Mediante Criterios de Aceptación

	tendrán la posibilidad de activarlo a posteriori para que pueda realizar pedidos								
PD 22	Seguimiento de pedidos por ID de pedido.	Baja	Requisito Funcional	Patrocinador	Seguimiento de pedidos	1.3	Se puede hacer un seguimiento de los pedidos mediante su ID.	No aplica	Mediante Criterios de Aceptación
PD 23	Seguimiento de pedidos, edición y borrado por el administrador.	Baja	Requisito Funcional	Patrocinador	Administración de pedidos	1.3	El administrador puede hacer el seguimiento de los pedidos, editarlos (entregado, pagado, método de envío y de pago) y eliminarlos, sin posibilidad de devolución de los productos.	No aplica	Mediante Criterios de Aceptación
PD 24	Creación, edición y borrado	Mediana	Requisito Funcional	Patrocinador	Administración de	1.3	El administrador puede	No aplica	Mediante Criterios de

	de productos por el administrador.				productos		crear, editar o eliminar productos de la tienda.		Aceptación
PD 25	Creación, edición y borrado de categorías por el administrador.	Mediana	Requisito Funcional	Patrocinador	Administración de productos	1.3	El administrador puede crear, editar o eliminar categorías de la tienda.	No aplica	Mediante Criterios de Aceptación
PD 26	Visualización de estadísticas de la tienda (productos más vendidos, usuarios que más pedidos realizan).	Baja	Requisito Funcional	Patrocinador	Estadísticas	1.3	El administrador puede analizar los productos más vendidos y los usuarios que más pedidos realizan.	No aplica	Mediante Criterios de Aceptación
PD 27	Debe de haber una sección con los productos destacados de la tienda.	Baja	Requisito Funcional	Patrocinador	Catálogo de productos	1.2	Se muestran los productos destacados en la página principal.	No aplica	Mediante Criterios de Aceptación
PD 28	Se desea que el sistema disponga de un mecanismo de seguridad reforzado para	Alta	Requisito Seguridad	Patrocinador	Seguridad	1.1	Se impide el acceso a los pedidos a los usuarios que no tengan activada	No aplica	Mediante Criterios de Aceptación

	evitar suplantaciones de identidad por parte de los administradores o de los usuarios.						o el mecanismo de seguridad.		
PD 29	Métodos de pago: Pasarelas de pago.	Mediana	Requisito Funcional	Patrocinador	Método de pago	1.3	Se puede realizar el pago mediante una pasarela de pago.	No aplica	Mediante Criterios de Aceptación
PD 30	Métodos de pago: pago contra-reembolso.	Mediana	Requisito Funcional	Patrocinador	Método de pago	1.3	Los clientes pueden seleccionar la opción de pagar contra-reembolso.	No aplica	Mediante Criterios de Aceptación
PD 31	Formas de entrega: entrega estándar o recoger en tienda física.	Baja	Requisito Funcional	Patrocinador	Compra	1.3	Los clientes pueden seleccionar la forma de entrega al realizar un pedido.	No aplica	Mediante Criterios de Aceptación
PD 32	Página con información de la empresa cliente.	Baja	Requisito Funcional	Patrocinador	Calidad	1.3	Existe una página que muestra a información de la empresa cliente.	No aplica	Mediante Criterios de Aceptación
PD 33	Los datos de contacto	Baja	Requisito No	Patrocinador	Calidad	1.2	Existe un espacio	No aplica	Mediante Criterios

	y ubicación de la empresa cliente se debe mostrar siempre.		Funcional				o en toda la página donde se puede ver esta información claramente.		os de Aceptación
PD 34	La página web tiene que poder ser utilizable en dispositivos móviles.	Baja	Requisito No Funcional	Patrocinador	Calidad	1.1, 1.2, 1.3	Se puede manejar con facilidad la página en un móvil o tableta.	Disposición adecuada de los elementos en pantallas con un tamaño o similar a la de un iPhone 14 Pro Max	Mediante Criterios de Aceptación
PY 01	El producto final debe entregarse como un contenedor de aplicaciones con las instrucciones de instalación y puesta en producción.	Baja	Requisito Portabilidad	Patrocinador	Entrega	1.3	Entrega de un contenedor con la aplicación y las instrucciones para la instalación.	No aplica	Mediante Criterios de Aceptación

Cuadro 3.4: Matriz de trazabilidad de requisitos

De igual modo el cliente nos ha facilitado los siguientes requisitos de información que debe proporcionar la aplicación:

- El sistema debe almacenar datos sobre los productos. Los datos almacenados serán:
 - Nombre del producto
 - Descripción
 - Precio: en el caso de los Picos el precio correspondería al precio por bolsa de 250 gramos.
 - Imagen (opcional)
 - Stock
 - Categoría: habrá cuatro categorías, “Normal”, “Integral”, “Pico” y “Otros”.

- El sistema ha de almacenar la siguiente información de los usuarios:
 - Nombre
 - Apellidos
 - Nombre de usuario
 - Correo electrónico
 - Dirección: preferiblemente la de entrega de los pedidos.
 - Código postal: para saber en qué pueblo o ciudad reside el cliente y realizar rutas de reparto óptimas.

- El sistema deberá almacenar información sobre los pedidos. Los datos que necesita almacenar son:
 - Fecha de creación
 - Fecha de modificación
 - Pagado: indicar de alguna manera si la cuantía total del pedido ha sido abonada o no.
 - Estado del pedido: puede encontrarse en estado “Pendiente”, “Enviado” o “Entregado”.
 - Método de entrega: en caso de que el cliente desee recoger en la tienda su pedido podría hacerlo, luego existen dos métodos de entrega, “Entrega estándar” (lo que se conoce como entrega a domicilio) y “Recogida en tienda”.
 - Método de pago: como se ha indicado anteriormente, se puede abonar el importe del pedido con tarjeta o haciendo uso de efectivo (contra reembolso), por lo que los métodos de pago serían “Tarjeta” y “Contra-reembolso”.
 - Además se requiere que en cada pedido aparezcan los productos que se han solicitado junto a la cantidad que se necesita de estos y su precio unitario, para tener un mayor control de los pedidos que se realizan.

Adicionalmente, el cliente nos ha trasladado cuáles serían sus reglas de negocio:

- Sólo los usuarios que tengan habilitado el doble factor de autenticación podrán realizar pedidos y consultar el estado de estos, para evitar que otros usuarios consulten dicha información.

- Sólo los administradores que tengan el doble factor de autenticación activado podrán modificar y crear productos y categorías, así como consultar el estado de los pedidos, modificarlos o eliminarlos. Tampoco podrán consultar las estadísticas generales de la aplicación si no activan el doble factor.

3.2.5. Diseño

En lo que se refiere al diseño de la aplicación cabe destacar que se ha partido de una estructura prefijada por uno de los proyectos que se desarrollaron a lo largo del primer cuatrimestre, por tanto, como tal no se han tenido unos mockups de la aplicación porque lo que se ha llevado a cabo en este proyecto ha sido más bien una adaptación del código previamente existente en HTML a su homólogo en Vue.js, invirtiendo cuantiosas horas en la configuración y personalización del CSS correspondiente para que la página fuera lo más similar posible a la organización previa que tenía la aplicación original Django en la que se ha basado este proyecto.

No obstante, existen varias restricciones que se han aplicado constantemente en el desarrollo de la interfaz gráfica en el transcurso de este. Las restricciones son las que siguen:

- Adaptabilidad a distintos dispositivos (responsividad)

Aunque ya ha sido mencionado en la recopilación de requisitos, esta restricción es una de las más importantes debido al uso que se le puede dar a la aplicación. El público objetivo que utilizará esta aplicación puede no estar muy familiarizado con las nuevas tecnologías y el manejo de un móvil per se les puede resultar complejo, por lo que se ha tenido en cuenta que la mayoría de los usuarios finales que utilizarán la aplicación, lo harán haciendo uso del teléfono móvil. Por ello, era de vital importancia que la aplicación se pueda utilizar tanto en dispositivos con una pantalla pequeña (de aproximadamente 4.5 pulgadas hasta 6.69 pulgadas) como en dispositivos con una pantalla de resolución de 15 pulgadas o más.

- Adaptabilidad en el cambio de temas y colores

Teniendo en cuenta que parte del alcance de este proyecto consiste en la fácil modificación de la aplicación para ser implantada en diversas panaderías que estén interesadas en su uso, es imprescindible que los colores, fotos y estilos de la aplicación sean fácilmente adaptables en función de la empresa.

Por tanto, se ha seguido un enfoque de desarrollo que permite intercambiar fácilmente fuentes, estilos e imágenes al modificar las propiedades de un componente central. Este componente sirve de base para gran parte de los demás, facilitando así la gestión y personalización de los elementos visuales en toda la aplicación.

En lo que respecta a la imagen del fondo de la página de inicio y al logotipo que aparece en la esquina superior izquierda, bastaría con agregar una nueva imagen en la carpeta de recursos de Vue.js (assets).

Gracias a estas facilidades, la aplicación permite una sencilla personalización para adaptarse a las necesidades visuales de cada empresa que esté interesada en su adquisición.

- Disposición de componentes

La disposición de los componentes en la interfaz gráfica ha de ser lo más intuitiva posible. Es por ello, que durante el desarrollo se decidió crear la barra de navegación de manera que el logo de la empresa permitiera volver constantemente a la página de inicio y los menús desplegables basta con dejar

el cursor sobre ellos para acceder a las distintas opciones que ofrecen. Por otro lado, el acceso al carrito de la compra es sencillo, así como añadir o eliminar productos del carrito.

Todo esto está acompañado de diversos mensajes y alertas que indican en todo momento si la acción que se está realizando está permitida o no, evitando posibles ambigüedades durante la interacción con la aplicación.

3.3. Definir el alcance

En esta sección de la memoria se procederá a detallar cuál será el alcance del proyecto que se ha desarrollado. Concretamente, se establecerá una base sólida para comprender y gestionar lo que se incluirá y lo que no se incluirá en el proyecto, lo que es esencial para garantizar que el proyecto se desarrolle de manera efectiva y que se alcancen sus objetivos.

3.3.1. Enunciado del Alcance

El proyecto tiene como objetivo fundamental crear una aplicación web para una panadería dentro de un plazo específico, cuyo éxito se mide por el cumplimiento de los requisitos acordados. Los entregables deben ser aprobados por el cliente y contienen una aplicación funcional, instrucciones y documentos de cierre REVISAR. La aplicación será tanto para portátiles y sobremesas como para dispositivos móviles, por lo que es imprescindible desarrollar una aplicación que se adapte a ambos tipos de sistema. El proyecto tiene un presupuesto de 12.000 € + IVA (basado en la **Estimación de costes**) y un plazo de tres meses para la implementación. Cualquier cambio que desee efectuar a lo largo del proyecto requerirá la aprobación por el cliente.

La lista de entregables del proyecto se puede ver en el cuadro 3.5.

Entregable 1	Entregable de proyecto: Este entregable se refiere a la generación de los informes de seguimiento y desempeño según lo especificado.
	Entregable de producto: Incluye la inicialización, autenticación con doble factor de autenticación y el modelo de datos. Este entregable se refiere a la fase inicial del proyecto y comprende la implementación de la parte de la autenticación fundamentalmente.
Entregable 2	Entregable de proyecto: Este entregable se refiere a la generación de los informes de seguimiento y desempeño según lo especificado.
	Entregable de producto: Incluye el catálogo y el carrito de la compra. Este entregable se centra en la funcionalidad principal de la aplicación relacionada con el catálogo de productos de la tienda y la gestión del carrito de la compra, además de las implementaciones previamente entregadas.
Entregable 3	Entregable de proyecto: Este entregable se refiere a la generación de los informes de seguimiento y desempeño según lo especificado.
	Entregable de producto: Incluye la pasarela de pago, el seguimiento de los pedidos y la administración del sistema

	(gestión de productos, pedidos y estadísticas). Este entregable se centra en la finalización de las características más importantes de la aplicación, relacionada con el seguimiento de los pedidos de los clientes, el proceso de pago y la administración del sistema por parte de los administradores.
Entrega final	Entregable de proyecto: Informe de Cierre, lecciones aprendidas y recopilación de informes anteriores.
	Entregable de producto: Aplicación Portal Django Panadería completa e instrucciones de despliegue.

Cuadro 3.5: Lista de entregables del proyecto

Los criterios de aceptación de este proyecto consisten en el cumplimiento de todos los requisitos acordados con el cliente, el tamaño del proyecto debe ser acorde con el alcance predefinido en esta sección y la entrega de la documentación completa del proyecto. En cuanto a las restricciones generales del mismo se encuentran el plazo límite de entrega del proyecto, sin la posibilidad de contraer retrasos significativos; y los recursos financieros asignados, que son limitados y se deben gestionar eficazmente para evitar sobrecostos que supongan exceder los fondos disponibles.

3.3.2. Plan de Gestión del Alcance

En esta subsección de la definición del alcance se tratará y definirá el alcance del proyecto, lo que incluye la aceptación formal de los entregables del proyecto y la gestión de cambios al enunciado del alcance.

Como se ha mencionado previamente en el **Enunciado del Alcance**, el proyecto tiene como objetivo principal crear una aplicación web que permita al cliente vender piezas de pan y cumplir con los plazos acordados. El trabajo se dividirá en iteraciones para producir entregables que el cliente debe aprobar. El éxito del proyecto se medirá por el cumplimiento de los requisitos y necesidades acordados. Para la aprobación de los entregables, la aplicación web debe funcionar según lo especificado y cumplir con los requisitos del proyecto. Los entregables incluirán un contenedor de aplicaciones con instrucciones de instalación y puesta en producción, junto con documentos para el seguimiento y cierre del proyecto. La aplicación web estará diseñada tanto para su uso en portátiles y sobremesas, como para dispositivos móviles, como lo ha solicitado el cliente. El proyecto se llevará a cabo con un presupuesto de 12.000 € + IVA y se completará en un máximo de tres meses, desde el 1 de enero de 2024 hasta el 1 de abril de 2024. Cualquier inconveniente o cambio durante el proyecto se comunicará al cliente y requerirá su aprobación antes de la implementación.

Para la estructura de la EDT (Estructura de Desglose del Trabajo) se ha definido en el caso del proyecto una estructura *Hybrid WBS o EDT Orientada a Proyecto* y se ha decidido utilizar las iteraciones como *paquetes de trabajo*. En esta subsección no se contempla el número de entregables a realizar, esto se especifica en el **Enunciado del Alcance** detalladamente. En la subsección Diccionario de la EDT se especifican los paquetes de trabajo con sus respectivas actividades. Además, los entregables se harán de manera unitaria por iteración, a menos que se tenga que modificar en alguna ocasión el entregable debido a la falta o exceso de recursos para un entregable. En el subapartado **Enunciado del Alcance** se detallan los entregables y se dividen por entregables de producto y de proyecto.

El mantenimiento de la línea base del alcance consistirá en la gestión adecuada del alcance, realizando las reuniones necesarias con el cliente para consensuar los cambios que no estén contemplados en este documento y que necesiten de su aprobación para la correcta continuación del proyecto. Para ello, se deberá decidir qué alternativas seguir. Además, también se contemplará la eficiencia del trabajo para cada uno de los requisitos según las expectativas previstas y su tiempo de ejecución.

Los cambios al alcance del proyecto deben ser consensuados y aceptados por el cliente, estos cambios surgen con la imposibilidad de cumplir con alguno de los requisitos mínimos impuestos por el cliente. En cualquier caso, cualquier imposibilidad o requisito con necesidad de modificación debe ser sustituido por alguna alternativa viable y equiparable que no suponga un cambio en la usabilidad final al usuario.

Para poder garantizar la aceptación de cada uno de los entregables o incrementos de valor sobre el producto se han determinado dos criterios esenciales.

- **Producto Software:** se debe asegurar el correcto funcionamiento del producto y el cumplimiento de todos los requisitos establecidos. Para asegurarse de este hecho, el producto debe haber sido testeado y superar los estándares de usabilidad y accesibilidad establecidos.
- **Documentación del proyecto:** se debe asegurar que se cumpla con los estándares establecidos en el plan de gestión calidad y seguir esta línea de desarrollo en cuanto a apariencia, accesibilidad y usabilidad mínima.

Los requisitos mínimos del proyecto indicados por el cliente han de ser realizados para la aceptación del producto. Como se ha descrito a lo largo de esta sección el alcance del proyecto contempla la casuística de la necesidad de modificación de los requisitos y la validación del cliente. En cualquier caso, el alcance debe recoger el conjunto de requisitos mínimos y/o modificaciones y como se gestionará a lo largo del proyecto.

Por último, el control del alcance del proyecto se gestionará de manera coordinada entre el realizador de este proyecto y el tutor de este (denominado cliente para asemejar el trabajo a un proyecto real), siendo este último el encargado de validar las modificaciones que surjan durante el ciclo de vida del proyecto.

3.3.3. EDT

La EDT que se ha creado sería la que aparece en la figura 3.1.

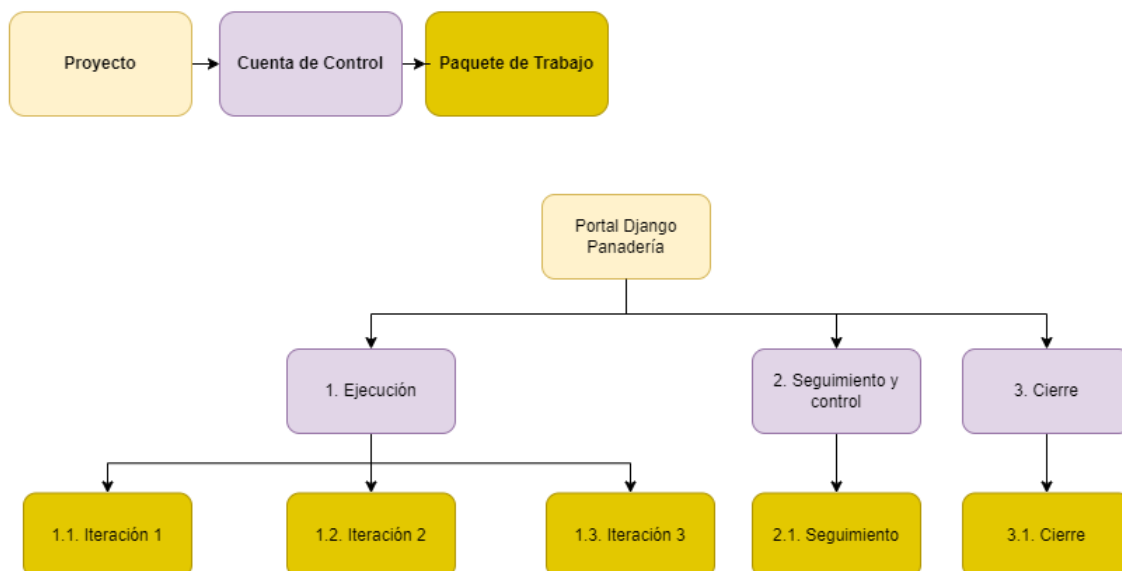


Figura 3.1: EDT del proyecto

3.3.4. Diccionario EDT

A continuación, serán definidas las actividades a realizar en base a la EDT. Para ello nos ayudaremos de los cuadros 3.6, 3.7, 3.8, 3.9 y 3.10. Teniendo en cuenta que el responsable de llevar a cabo las actividades y los recursos humanos disponibles serían la misma persona, es decir, el autor de esta memoria; no tiene sentido repetir en dichos campos el nombre una y otra vez, por lo que lo aclaro para evitar confusiones.

Nota: 1 día corresponde a 8 horas de trabajo.

Id: 1.1	Nombre Paquete: Iteración 1	
Descripción: Realizar el modelo de datos, armar la configuración inicial del proyecto, permitir el ingreso de usuarios a la aplicación e integrar el doble factor de autenticación en el proyecto.		
Entregables: Este paquete corresponde al entregable de producto del entregable 1, definido en el Enunciado del Alcance .		
Actividades/Hitos		Duración
1.1.1: Realizar modelo de datos		1 día
1.1.2: Inicialización		1 día
1.1.3: Gestión de usuarios		1 día
1.1.4: Integración del doble factor		2 días
Criterios de aceptación:		
- Los usuarios pueden registrarse, iniciar sesión y activar el doble factor para posteriormente realizar pedidos.		

Cuadro 3.6: Actividades de la Iteración 1

Id: 1.2	Nombre Paquete: Iteración 2
Descripción: Mostrar la marca de la empresa cliente en las vistas, realizar el catálogo de productos, marcar claramente los productos que estén agotados e implementar el carrito de la compra.	
Entregables: Este paquete corresponde al entregable de producto del entregable 2, definido en el Enunciado del Alcance .	
Actividades/Hitos	Duración
1.2.1: Mostrar marca corporativa	1 día
1.2.2: Catálogo de productos	3 días
1.2.3: Marca de productos agotados	1 día
1.2.4: Mostrar productos destacados en la página principal	1 día
1.2.5: Carrito de la compra	2 días
Criterios de aceptación: - Se puede ver el logo de la empresa cliente en cada una de las vistas. - Se pueden ver los productos en el catálogo. - Se pueden detectar claramente los productos que estén agotados. - Se puede añadir uno o varios productos a la cesta de la compra y aumentar/disminuir su cantidad o eliminarlos.	

Cuadro 3.7: Actividades de la Iteración 2

Id: 1.3	Nombre Paquete: Iteración 3
Descripción: Establecer el mecanismo de compra de productos mediante la pasarela de pago, permitir tanto a usuarios como al administrador gestionar todo lo relacionado al seguimiento de los pedidos y realizar el sistema de administración de la aplicación, incluyendo la gestión de productos y categorías, pedidos y estadísticas de la aplicación, sólo para los administradores.	
Entregables: Este paquete corresponde al entregable de producto del entregable 3, definido en el Enunciado del Alcance .	
Actividades/Hitos	Duración
1.3.1: Integrar la pasarela de pago	1 día
1.3.2: Seguimiento de pedidos por usuarios	1 día
1.3.3: Confirmación de compra por correo	1 día
1.3.4: Seguimiento de pedidos por el administrador	1 día
1.3.5: Creación de productos y categorías por el administrador	1 día
1.3.6: Edición y borrado de productos y categorías por el administrador	1 día
1.3.7: Edición y borrado de pedidos por el administrador	1 día
1.3.8: Estadísticas de productos y usuarios	1 día
1.3.9: Generar un contenedor de aplicaciones con el producto final	1 día
1.3.10: Elaborar instrucciones de instalación y puesta en producción	1 día
Criterios de aceptación: - Se puede efectuar el pago de un pedido utilizando la pasarela. - Todo usuario que realice un pedido puede ver el estado en el que se encuentra dicho pedido. - Todo usuario que realice un pedido recibe un correo de confirmación de la compra. - El administrador puede consultar el estado en el que se encuentran todos los pedidos. - El administrador puede crear productos y categorías dentro de la aplicación.	

-	El administrador puede editar y eliminar productos y categorías dentro de la aplicación.
-	El administrador tiene acceso a las estadísticas de los productos más vendidos y de los usuarios que realizan más pedidos.
-	Se puede instalar y poner en producción la aplicación a partir de un contenedor.

Cuadro 3.8: Actividades de la Iteración 3

Id: 2.1	Nombre Paquete: Seguimiento
Descripción: Generar informes que permitan consultar el desempeño y seguimiento del proyecto por parte del autor de esta memoria durante la ejecución del trabajo.	
Entregables: Este paquete corresponde a los entregables de proyecto de los entregables 1, 2 y 3, definidos en el Enunciado del Alcance .	
Actividades/Hitos	Duración
2.1.1: Generar Informe de Desempeño	1 h
2.1.2: Actualizar desempeño (mitad de la iteración 1)	1 h
2.1.3: Actualizar desempeño (final de la iteración 1)	1 h
2.1.4: Actualizar desempeño (mitad de la iteración 2)	1 h
2.1.5: Actualizar desempeño (final de la iteración 2)	1 h
2.1.6: Actualizar desempeño (mitad de la iteración 3)	1 h
2.1.7: Actualizar desempeño (final de la iteración 3)	1 h
2.1.8: Generar Informe de Seguimiento de la iteración 1	1 h
2.1.9: Generar Informe de Seguimiento de la iteración 2	1 h
2.1.10: Generar Informe de Seguimiento de la iteración 3	1 h
Criterios de aceptación:	
- Los informes generados muestran correctamente el desempeño y seguimiento del autor de la memoria durante la ejecución del proyecto.	

Cuadro 3.9: Actividades del Seguimiento

Id: 3.1	Nombre Paquete: Cierre
Descripción: Entrega de la documentación y del producto.	
Entregables: Este paquete corresponde al entregable final, definido en el Enunciado del Alcance .	
Actividades/Hitos	Duración
3.1.1: Revisar que los documentos a entregar cumplan con las normas de formato impuestas	2 h
3.1.2: Empaquetar todos los documentos a entregar	1 h
3.1.3: Comprobar que el producto implementa todos los requisitos funcionales	2 h
3.1.4: Empaquetar el producto final a entregar	1 h
Criterios de aceptación:	
- La entrega de la documentación se ha realizado a tiempo y cumple con los mínimos. - La entrega del producto se ha realizado a tiempo y cumple con los mínimos.	

Cuadro 3.10: Actividades del Cierre

3.3.5. Secuenciación de las actividades

En este apartado se secuenciarán las actividades de manera que se sepa en qué orden se deben abordar las tareas que tenemos que llevar a cabo. Para secuenciarlas se ha utilizado el programa MS Project 2021, ya que el autor de esta memoria se encuentra más familiarizado con esta herramienta que con versiones de código abierto como Project Libre, además de contar con licencia por parte de la Universidad de Sevilla para su uso estrictamente académico.

Suponemos que las actividades se realizan durante una jornada laboral de ocho horas, desde las 08:00 horas de la mañana hasta las 17:00 horas de la tarde con un descanso de una hora para almorzar, comprendiendo las cuarenta horas semanales legales que un trabajador puede efectuar un trabajo. Además, aquellas que tienen una duración de menos de un día (una hora o similar) se ajustan a un solo día, teniendo en cuenta que si la siguiente actividad dura un día o jornada completa no podría realizarse porque se excederían las horas legales de trabajo para su realización.

Por desgracia, el planteamiento de esta secuenciación es utópico, esto quiere decir que en la práctica es muy difícil que en un día (que corresponde a 8 horas de trabajo) se terminen las tareas que se tienen asignadas. Como se podrá observar en el informe de tiempo del Clockify REVISAR, el autor del trabajo no es lo suficientemente rápido trabajando para finalizar una tarea en un solo día de trabajo, tomando a veces incluso una semana para completar con éxito una actividad del proyecto. Por tanto, considérese más como una estimación la figura en la que aparecen secuenciadas las actividades a realizar en el proyecto (véase la figura 3.2).

Por último, quisiera destacar que el tiempo “restante” que habría entre el 8 de enero y el último día de marzo, se ocupará realizando actividades de apoyo a la implementación del proyecto, es decir, que en realidad la duración de las actividades, como se podrá posteriormente comprobar en el cronograma, se expandirá hasta el día 31 de marzo.

	①	Código de EDT	Modo de	Nombre de tarea	Duración	Comienzo	Fin	Predcesoras
1			✓	Realizar modelo de datos	1 día	lun 08/01/24	lun 08/01/24	
2			✓	Inicialización	1 día	mié 10/01/24	mié 10/01/24	
3			✓	Gestión de usuarios	1 día	jue 11/01/24	vie 12/01/24	2
19			✓	Generar Informe de Desempeño	1 hora	jue 11/01/24	jue 11/01/24	
4			✓	Integración del doble factor	2 días	lun 15/01/24	mar 16/01/24	3
20			✓	Actualizar desempeño	1 hora	mié 17/01/24	mié 17/01/24	
23			✓	Generar Informe de Seguimiento de la iteración 1	1 hora	mié 17/01/24	mié 17/01/24	
5			✓	Mostrar marca corporativa en las vistas	1 día	jue 18/01/24	jue 18/01/24	2
6			✓	Catálogo de productos	3 días	vie 19/01/24	mar 23/01/24	2
7			✓	Marca de productos agotados	1 día	mié 24/01/24	mié 24/01/24	6
8			✓	Mostrar productos destacados en la página principal	1 día	jue 25/01/24	jue 25/01/24	6
9			✓	Carrito de la compra	2 días	vie 26/01/24	lun 29/01/24	6
21			✓	Actualizar desempeño	1 hora	mar 30/01/24	mar 30/01/24	
24			✓	Generar Informe de Seguimiento de la iteración 2	1 hora	mar 30/01/24	mar 30/01/24	
25			✓	Integrar pasarela de pago	1 día	mié 31/01/24	mié 31/01/24	9
10			✓	Seguimiento de pedidos por usuarios	1 día	jue 01/02/24	jue 01/02/24	9,25
12			✓	Confirmación de compra por correo	1 día	vie 02/02/24	vie 02/02/24	9,25
11			✓	Seguimiento de pedidos por el administrador	1 día	mar 06/02/24	lun 05/02/24	10
13			✓	Creación de productos y categorías por el administrador	1 día	mar 06/02/24	mar 06/02/24	12
14			✓	Edición y borrado de productos y categorías por el administrador	1 día	mié 07/02/24	mié 07/02/24	
15			✓	Edición y borrado de pedidos por el administrador	1 día	jue 08/02/24	jue 08/02/24	
16			✓	Estadísticas de productos y usuarios	1 día	vie 09/02/24	vie 09/02/24	
17			✓	Generar un contenedor de aplicaciones con el producto final	1 día	lun 12/02/24	lun 12/02/24	3,4,5,6,7,8,9,25,10,12,11,13,14,15,16
18			✓	Elaborar instrucciones de instalación y puesta en producción	1 día	mar 13/02/24	mar 13/02/24	17
22			✓	Actualizar desempeño	1 hora	mié 14/02/24	mié 14/02/24	
26			✓	Generar Informe de Seguimiento de la iteración 3	1 hora	mié 14/02/24	mié 14/02/24	
27			✓	Revisar que los documentos a entregar cumplan con las normas de formato Impuestas	2 horas	mié 14/02/24	mié 14/02/24	
28			✓	Empaquetar todos los documentos a entregar	1 hora	mié 14/02/24	mié 14/02/24	
29			✓	Entrega de la documentación	1 hora	mié 14/02/24	mié 14/02/24	
30			✓	Comprobar que el producto implementa todos los requisitos funcionales	1 hora	mié 14/02/24	mié 14/02/24	
31			✓	Empaquetar el producto final a entregar	1 hora	mié 14/02/24	mié 14/02/24	
32			✓	Entrega del producto	1 hora	jue 15/02/24	jue 15/02/24	

Figura 3.2: Secuenciación de las actividades del proyecto

3.3. Plan de Gestión de los Recursos

Una vez se sabe cuál es nuestro alcance y las actividades que se tienen que desarrollar para dar forma al proyecto, se deben conocer los diferentes recursos de los que disponemos para nuestra causa.

En mi caso, el único recurso humano del que dispongo es el mío propio, aunque no se debe olvidar que también puedo apoyarme en compañeros de clase y mi tutor para resolver ciertas dudas, pero en última instancia, aquel que posee la potestad de avanzar con este proyecto es el redactor de la memoria (y que por consiguiente ha desarrollado todo el proyecto).

En cuanto a los recursos físicos, se ha contado con varios dispositivos que han facilitado el trabajo durante la realización del proyecto. Específicamente, se han utilizado los siguientes materiales:

- Lenovo Ideapad 330 15-ICH
- Monitor BenQ GL2780-B
- Ratón inalámbrico Trust Verto (ergonómico)
- Teclado Microsoft WS2-00012 Surface
- Smartphone Xiaomi Redmi Note 8 Pro
- iPad Air 4ª Generación

El coste asociado a estos materiales está recogido en el apartado de **Estimación de costes**.

El calendario de los recursos (en mi caso, el calendario de trabajo) se establece de lunes a viernes, aunque cabe destacar que los fines de semana se ha tenido que trabajar mucho más, debido al escaso tiempo que se le puede dedicar entre semana a la realización del proyecto por las otras asignaturas.

Por último, destacaría la visualización de múltiples tutoriales y cursos disponibles en Internet para la formación de cara a la realización del trabajo, fundamentalmente debido al empleo de nuevas tecnologías (concretamente Vue.js) como requisitos de capacitación.

3.4. Plan de Gestión del Cronograma

El objetivo de esta sección es gestionar cómo se planificará, desarrollará, controlará y gestionará el tiempo del proyecto. En otras palabras, trataremos de definir cómo se llevará a cabo la gestión del tiempo a lo largo de todo el ciclo de vida del proyecto.

La metodología que se va a seguir para la gestión del cronograma se compone de varios elementos y definiciones:

- Las actividades se identificarán siguiendo el siguiente esquema: C.T.A, donde la C indica la cuenta de control, la T indica el paquete de trabajo y la A el número de la actividad. Asimismo, se definirán en función del contenido que haya que llevar a cabo en la misma para cerrarla.

- Los recursos, que están definidos en el **Plan de Gestión de los Recursos** se asignarán según las necesidades del proyecto y/o de cada una de las actividades definidas para los paquetes de trabajo, y se han estimado de acuerdo a la duración estimada de las actividades de cada paquete de trabajo en la EDT (que vienen recogidas concretamente en el **Diccionario de la EDT**).
- Los esfuerzos se estimarán de acuerdo a la carga de trabajo que se supone que contiene una actividad y se definirán en horas de trabajo por persona.

El nivel de exactitud con el que se definirá el tiempo empleado para cada actividad será en horas que se tarde en realizar una actividad, los requisitos se medirán en tareas en las que se dividirán los paquetes de trabajo y el presupuesto disponible en euros. Como umbral de las varianzas de estos parámetros, se empleará el tiempo de realización de tareas, que podrá variar entre cuatro y cinco días para la finalización de una tarea, teniendo en cuenta que algunas se pueden retrasar en función del dominio de las tecnologías que se estén empleando en el momento.

El cronograma se mostrará en formato de diagrama de Gantt aunque su reporte se haga directamente en MS Project 2021 para una mayor facilidad en su interpretación (en PDF no se podría leer correctamente por el formato de exportación de este).

A continuación, se muestra un cuadro en el que aparecen los procesos de gestión del cronograma (véase el cuadro 3.11).

Identificación de las actividades	Se han identificado en función de los paquetes de trabajo previstos en el Diccionario de la EDT .
Secuenciación de las actividades	Se ha establecido un orden para la correcta realización de las actividades y que existan las menores dependencias entre ellas posibles (ver Secuenciación de las actividades).
Estimación de los recursos	La estimación de los recursos se ha recogido en el Plan de Gestión de los Recursos .
Estimación de esfuerzos y duraciones	Se establecen tanto en el Diccionario de la EDT y como en el Cronograma.
Actualización, monitoreo y control del cronograma	Conforme surjan nuevos cambios en las actividades del proyecto durante la fase de ejecución, éstos se recogerán en el cronograma para poder tenerlo actualizado.

Cuadro 3.11: Procesos de gestión del cronograma

3.5.1. Estimaciones de tiempo

En este apartado se detallan todas las estimaciones de tiempo consideradas para llevar a cabo las actividades planificadas, junto con el tiempo real empleado para completar cada una de ellas (véase el cuadro 3.12).

Nota: el porcentaje de variación de una actividad se calcula siguiendo esta fórmula:

$$\frac{100 \cdot (\text{Tiempo real invertido} - \text{Tiempo estimado})}{\text{Tiempo estimado}}$$

ID	Nombre Actividad	Tiempo estimado	Tiempo real invertido	Variación (%)
1.1	Iteración 1	14h	21h 45min	+55,36%
1.1.1	Realizar modelo de datos	3h	2h	-33,33%
1.1.2	Inicialización	1h	1h	0%
1.1.3	Gestión de usuarios	2h	3h 45min	+87,5%
1.1.4	Integración del doble factor	8h	15h	+87,5%
1.2	Iteración 2	17h	35h 30min	+108,82%
1.2.1	Mostrar marca corporativa en las vistas	1h	1h	0%
1.2.2	Catálogo de productos	4h	11h	+175%
1.2.3	Marca de productos agotados	1h	1h	0%
1.2.4	Mostrar productos destacados en la página principal	3h	3h 30min	+16,67%
1.2.5	Carrito de la compra	8h	19h	+137,5%
1.3	Iteración 3	39h	44h 30min	+14,1%
1.3.1	Integrar la pasarela de pago	2h	2h	0%
1.3.2	Seguimiento de pedidos por usuarios	4h	4h	0%
1.3.3	Confirmación de compra por correo	3h	4h 30min	+50%
1.3.4	Seguimiento de pedidos por el administrador	4h	4h 40min	+16,6%
1.3.5	Creación de productos y categorías por el administrador	7h	8h	+14,28%
1.3.6	Edición y borrado de productos y categorías por el administrador	3h	3h	0%
1.3.7	Edición y borrado de pedidos	3h	3h	0%
1.3.8	Estadísticas de productos y usuarios	2h	2h 45min	+37,5%
1.3.9	Generar un contenedor de aplicaciones con el producto final	5h	4h 35min	-8,33%
1.3.10	Elaborar instrucciones de instalación y puesta en producción	6h	8h	+33,33%
2.1	Seguimiento	7h	10h	+42,86%
2.1.1	Generar Informe de Desempeño	1h	1h	0%
2.1.2	Actualizar desempeño (final de la iteración 1)	0,5h	1h	+100%
2.1.3	Actualizar desempeño (mitad de la iteración 2)	0,5h	1h	+100%

2.1.4	Actualizar desempeño (final de la iteración 2)	0,5h	1h	+100%
2.1.5	Actualizar desempeño (mitad de la iteración 3)	0,5h	1h	+100%
2.1.6	Actualizar desempeño (final de la iteración 3)	0,5h	1h	+100%
2.1.7	Generar Informe de Seguimiento de la iteración 1	1h	1h	0%
2.1.8	Generar Informe de Seguimiento de la iteración 2	1h	1h	0%
2.1.9	Generar Informe de Seguimiento de la iteración 3	1h	1h	0%
3.1	Cierre	3h	6h	+100%
3.1.1	Revisar que los documentos a entregar cumplan con las normas de forma impuestas	1h	2h	+100%
3.1.2	Empaquetar todos los documentos a entregar	0.5h	1h	+100%
3.1.3	Comprobar que el producto implementa todos los requisitos	1h	2h	+100%
3.1.4	Empaquetar el producto final a entregar	0.5h	1h	+100%

Cuadro 3.12: Estimaciones de tiempo de la ejecución.

Además de las actividades de implementación al uso, también se han tenido en cuenta actividades durante el desarrollo relacionadas con la puesta en marcha de pruebas unitarias, de vista y de carga que están especificadas en la sección de **Pruebas** y posibles correcciones que se deban aplicar al proyecto para subsanar errores. Las estimaciones aparecen en el cuadro 3.13.

Nombre Actividad	Tiempo estimado	Tiempo real invertido	Variación (%)
Tests productos y categorías	3h	3h 15min	≈0%
Tests carrito de la compra	6h	7h	+16,67%
Tests de los pedidos	4h	4h 10 min	≈0%
Pruebas de carga	4h	6h 15min	+56,25%
Correcciones	10h	9h 30min	-5%

Cuadro 3.13: Estimación de pruebas y correcciones

Tal y como se citó en la Introducción de esta memoria, la idea de realización de este proyecto surgió de una conversación mantenida con un familiar. Por tanto, en los días posteriores a dicha conversación se comenzó a dar forma a la idea, desde un punto de vista teórico.

Esta fase de inicio comprende las distintas actividades que ayudaron a elucubrar lo que sería el trabajo final que se ha realizado y su estimación aparece más abajo (ver cuadro 3.14).

Nombre Actividad	Tiempo estimado	Tiempo real invertido	Variación (%)
Maduración de la idea	2h	2h	0%
Definición del objetivo del proyecto	1h	1h	0%
Definición del alcance	2h	2h	0%
Estudio de las tecnologías a utilizar	2h	2h	0%

Cuadro 3.14: Estimación del inicio

Posteriormente se estimó la fase de planificación del proyecto en la que se incluyen actividades de análisis de requisitos, análisis del costo de los materiales y demás elementos que participaron en la planificación del desarrollo del proyecto (véase el cuadro 3.15).

Nombre Actividad	Tiempo estimado	Tiempo real invertido	Variación (%)
Estructuración del proyecto en sus distintas fases	3h	3h	0%
Plan de Gestión de las Comunicaciones	2h	2h	0%
Plan de Gestión del Alcance	8h	12h 35min	+57,29%
Plan de Gestión de los Recursos	4h	9h	+125%
Plan de Gestión del Cronograma	8h	11h 45min	+46,875%
Estimaciones de costes	3h	3h	0%

Cuadro 3.15: Estimaciones de la planificación

El tiempo invertido en la confección de la documentación se contempla en el cuadro 3.16.

Nombre Actividad	Tiempo estimado	Tiempo real invertido	Variación (%)
Memoria	150h	170h	0%
Presentación	17h	17h	0%
Exportar documento de registro de tiempos	30min	30min	0%

Revisión y correcciones	8h	8h 20min	+4,16%
-------------------------	----	----------	--------

Cuadro 3.16: Estimaciones documentación

Finalmente el proyecto terminará cuando se defienda ante un tribunal, por lo que la fase de defensa no se contempla para la planificación temporal, siendo esta fase simplemente testimonial.

3.5.2. Cronograma

En este apartado se va a representar el diagrama temporal del proyecto, en el que se recogen todas las tareas que se van a realizar durante este (véase la figura 3.3).

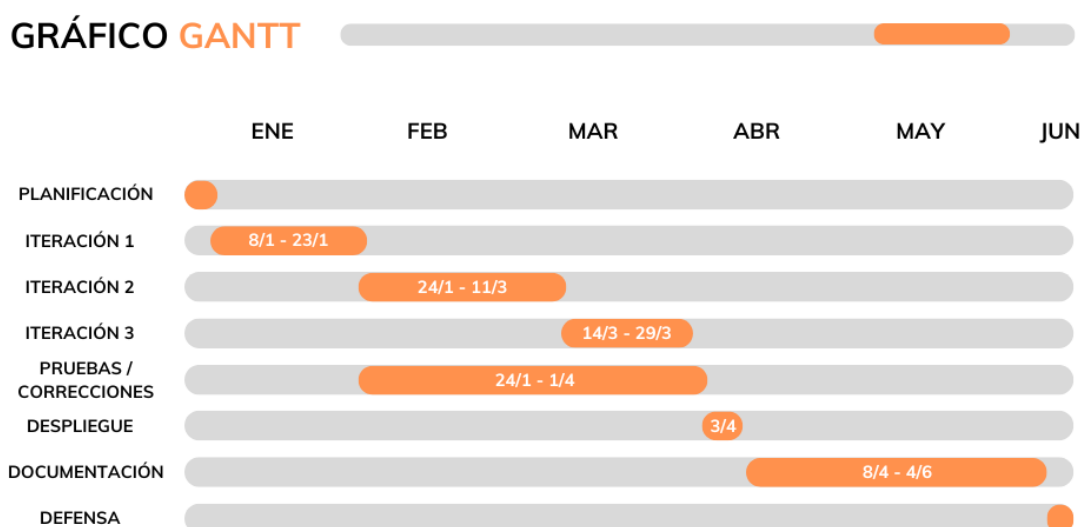


Figura 3.3: Diagrama temporal real del proyecto

Quisiera puntualizar que el diagrama que se observa corresponde a la duración real del proyecto, ya que la estimación realizada en apartados anteriores (el tiempo estimado de cada tarea) está recogido en el cronograma adjunto a este proyecto realizado con MS Project.

Las tareas relacionadas con las pruebas y las correcciones, a pesar de planificarse de cara a la finalización del código, lo que tenía más sentido común y así nos lo han inculcado durante el grado, era la implementación de las pruebas conforme se terminan las tareas de las funcionalidades previstas, para evitar arrastrar los problemas o errores que se muestren en los tests y así aumentar en lo posible la eficiencia del trabajo realizado. Es por ello que, como las pruebas se han ido realizando sobre tareas y/o requisitos implementados a partir de la segunda iteración, aparecen planificadas al mismo tiempo que esta.

El motivo por el que existen tareas que carecen de una fecha en la figura, es por mostrar el gráfico de la forma más sencilla posible y para ello aclaro en este párrafo que la primera tarea de planificación se llevó a cabo en la semana del 1 de enero y la defensa se dará en la semana del 10 de junio.

3.6. Estimación de costes

Aclarada la planificación temporal y los recursos necesarios para llevar a cabo el proyecto, es hora de poner un precio al esfuerzo que se ha realizado en estos meses. A continuación, se analizarán los costes asociados al proyecto en distintas partes, como el coste humano, el coste de los materiales y los posibles costes de servicios externos en caso de despliegue en la nube:

3.6.1. Coste Humano

Dentro de estos costes se incluyen tanto los costos de un analista, quien sería responsable de planificar el proyecto y elaborar la documentación asociada, como los costos de un desarrollador, encargado de construir la aplicación web. En el caso de este trabajo, al ser realizado de forma personal, los costos asociados a recursos humanos son de 0 €.

No obstante, si se considera que el sueldo de un desarrollador web junior en España ronda los 31,06 € por hora, teniendo en cuenta que la duración de la fase de desarrollo ha sido de 132 horas, el coste ascendería a: $132 \cdot 31,06\text{€} = 4.099,92\text{€}$. Por otro lado, la parte de planificación y análisis que le correspondería a un analista que en España de media su sueldo ronda los 40,13 € por hora, invirtiendo un total de 178 horas que es la duración que ha tenido esta fase del proyecto, el coste ascendería a: $178 \cdot 40,13\text{€} = 7.143,14\text{€}$.

Por tanto, el coste total del proyecto sería la suma de ambas cantidades: $4.099,92\text{€} + 7.143,14\text{€} = 11.243,06\text{€}$.

3.6.2. Coste Material

En la sección del Plan de Gestión de Recursos se citaron los recursos materiales que han sido empleados a lo largo del trabajo, ahora se procederá a ponerles el precio que les corresponde y estimar su amortización en cinco meses, que ha sido la duración aproximada del proyecto (véase el cuadro 3.17).

Recurso	Precio	Amortización (cinco meses)
Lenovo Ideapad 330 15-ICH	828,85 €	86,34 €
Monitor BenQ GL2780-B	232,90 €	24,26 €
Ratón inalámbrico Trust Verto (ergonómico)	21,99 €	4,58 €
Teclado Microsoft WS2-00012 Surface	109,00 €	11,35 €
Smartphone Xiaomi Redmi Note 8 Pro	249 €	25,94 €

Cuadro 3.17: Estimación de costes materiales

Si se suman las cantidades de las amortizaciones, se obtiene un coste real de las amortizaciones de 152,47 €.

3.6.3. Coste de Servicios Externos

Se ha de tener en cuenta que los tres contenedores que se especifican en la sección de **Despliegue** se tienen que alojar en servidores. Aunque en este proyecto no se ha considerado el despliegue final en un servidor, se puede realizar una estimación del coste que supondría llevarlo a cabo. Para ello, supondremos que se utilizará una sola instancia del AWS para minimizar los costes durante los primeros cinco meses de uso (si el proyecto se utilizara de forma comercial los cálculos estimados serían distintos), concretamente utilizaremos una instancia t3.medium (EC2) con dos CPUs y una memoria RAM de 4GB. El coste por el uso de esta instancia mensualmente sería de 20.44 USD, que al cambio en euros aproximadamente rondaría los 18,98 € al mes; por lo tanto el coste durante cinco meses ascendería a: $5 \cdot 18,98 \text{ €} = 94,9 \text{ €}$. Para realizar el cálculo, se ha empleado la calculadora de AWS [13] con el servidor en Alemania (Hamburgo), sistema operativo Linux, siendo una instancia compartida.

Además, durante la implementación del proyecto una de las herramientas empleadas ha sido GitHub Copilot que cuenta con licencia gratuita para estudiantes, por lo que su coste durante la realización del trabajo ha sido de 0 €. Suponiendo que se estuviera realizando un proyecto para un cliente real en una empresa real, el coste mensual de GitHub Copilot es de 10 USD, que al cambio corresponden a 9,28 € y los primeros 30 días es gratuito por la prueba de la que dispone previo a la suscripción, se tendría que considerar cuatro meses a 9,28 € por mes: $4 \cdot 9,28 \text{ €} = 37,12 \text{ €}$ a pagar en total por Copilot. Si se utiliza en lugar de unos meses, más de un año, convendría más la suscripción a su plan anual que corresponde a 100 USD que al cambio serían 92,84 €.

3.6.4. Conclusión

Para finalizar esta estimación, se procede a detallar el TCO (Total Cost of Ownership) durante un año, de este trabajo. Se calcula muy fácilmente teniendo en cuenta que ya se ha desglosado la estimación de costes en costes humanos, de recursos y de servicios externos, por lo que bastaría con organizar la información de la que ya se dispone en costes de capital (CapEx) y costes operativos (OpEx):

- Costes de capital (CapEx): se refieren a los costes de inversión inicial en activos tangibles o intangibles que se amortizan con el paso del tiempo. En nuestro caso sólo tenemos un costo de CapEx que es la amortización total de los recursos materiales, que asciende a 152,47 €.
- Costes operativos (OpEx): se refieren a los costes recurrentes de operación de un activo o servicio. En este proyecto la gran mayoría de recursos empleados, como los recursos humanos o de servicios externos, se corresponden a costes operativos; por lo que si se suman ambos costes (recursos humanos y de servicios externos) el total ascendería a 11.563,66 €.

Con el objetivo de aclarar el TCO, a continuación se organizan en un cuadro los costes de propiedad del proyecto (véase el cuadro 3.18).

Costes	Área		Precio
CapEx	Materiales		152,47 €
OpEx	Humano	Desarrollador web junior	4.099,92 €
		Analista	7.143,14 €
		Subtotal	11.243,06 €
	Servicios Externos	AWS	227,76 €
		GitHub Copilot	92,84 €
		Subtotal	320,60 €
TCO			11.716,13 €

Cuadro 3.18: Desglose del TCO para un año

En nuestro caso específicamente, el desarrollo del proyecto ha durado cinco meses, por lo que el cuadro anterior quedaría como sigue (véase el cuadro 3.19).

Costes	Área		Precio
CapEx	Materiales		152,47 €
OpEx	Humano	Desarrollador web junior	4.099,92 €
		Analista	7.143,14 €
		Subtotal	11.243,06 €
	Servicios Externos	AWS	94,9 €
		GitHub Copilot	37,12 €
		Subtotal	132,02 €
TCO			11.527,55 €

Cuadro 3.19: Desglose del TCO para los cinco meses del desarrollo

3.7. Plan de Gestión de Riesgos

En esta sección se plasmará la forma en la que se identifican, evalúan y mitigan los posibles riesgos que pueden afectar al proyecto, aunque teniendo en cuenta que el desarrollo de este trabajo no conlleva demasiados riesgos.

Con esto se quiere transmitir que el principal riesgo al que se expone el trabajo durante su realización es a experimentar un grave retraso que suponga que no se pueda cumplir con los plazos de entrega fijados. En ese caso, la única solución existente es la entrega del proyecto en el siguiente plazo de entrega, es decir, entregarlo una convocatoria más de lo esperado.

En lo que respecta a la parte técnica del proyecto, es decir, si en algún punto del desarrollo se encuentra un problema complejo que provoque un estancamiento, la solución más lógica será la de buscar información, ayudarme de algún tutorial o foro, o consultando al tutor o a compañeros. En cualquier caso, si el problema persiste durante varios días, se intentará buscar una vía de escape para que no suponga un enorme retraso en el desarrollo, en el que el tiempo al igual que en la planificación es un bien valioso y no se debe desperdiciar.

Los riesgos expuestos anteriormente serán debidamente monitorizados y controlados mediante el **Informe de Desempeño** que se incluirá en los **Anexos**.

3.8. Plan de Gestión de la Calidad

Finalmente, se abordará la manera en la que se planificará, implementará y controlará la calidad del proyecto. Este plan define los estándares de calidad, procesos de control, métricas y actividades necesarias para asegurarse de que el proyecto cumpla con los requisitos de calidad mínimos y las expectativas con respecto al producto desarrollado, garantizando así la entrega de un producto o servicio de alta calidad.

En el siguiente cuadro se recogen las acciones planificadas para la verificación de la calidad de las iteraciones contempladas en la EDT (véase el cuadro 3.20).

EDT	NOMBRE PAQUETE DE TRABAJO	ENTREGABLE	CRITERIO DE ACEPTACIÓN	MÉTODO DE VERIFICACIÓN
1.1	Iteración 1	1	Requisitos de alto nivel	Pruebas de Aceptación del Patrocinador
1.2	Iteración 2	2	Requisitos de alto nivel	Pruebas de Aceptación del Patrocinador
1.3	Iteración 3	3	Requisitos de alto nivel	Pruebas de Aceptación del Patrocinador
2.1	Seguimiento	1, 2 y 3	Documentación	Pruebas de Aceptación del Patrocinador
3.1	Cierre	Final	Documentación	Pruebas de Aceptación del Patrocinador

Cuadro 3.20: Acciones del plan de calidad

La planificación de la calidad se basará en la definición de métricas y actividades para el control de la calidad, además del cumplimiento de los requisitos específicos del proyecto.

Para el aseguramiento de la calidad se utilizarán reuniones de control de calidad como el método principal para asegurar el cumplimiento de los estándares de calidad. Los responsables del control de calidad serán el Patrocinador o tutor del proyecto y el redactor de la memoria. Ambas partes se centrarán en la revisión de los entregables previamente a su entrega. Además de dichas reuniones, se adjuntará en el Informe de Desempeño una métrica de calidad enfocada a la memoria y otra enfocada al código. Estas métricas se recogen en el esquema que sigue:

- Documentación:
 - Exactitud y coherencia de la memoria: la frecuencia de errores o discrepancias encontradas durante estas reuniones. Se usará como umbral un índice de precisión del 90% o superior, lo que significa que menos del 10% de la documentación contiene errores o inconsistencias.
 - Mínimos establecidos cumplidos: en la memoria se recogen los puntos y/o documentos mínimos (planes de gestión, análisis, desarrollo e implementación) exigidos por el coordinador de la asignatura y por el tutor, además del número mínimo de páginas que esta contiene.
- Código:
 - Comentarios: el código está debidamente comentado para que pasado cualquier período de tiempo, se pueda comprender la función que realizan las características más importantes del proyecto.
 - Pruebas sobre frontend: se implementan algunas pruebas unitarias de componentes del frontend y pruebas de vista dinámicas para asegurar el correcto funcionamiento y montaje de este.
 - Pruebas sobre backend: se implementan varios tipos de pruebas sobre el backend, como pruebas unitarias, de API y de carga para comprobar y asegurar el funcionamiento de este.
 - Estructuración correcta del proyecto: se estructurará el proyecto de manera que pueda ser mantenible a largo plazo sin dificultades por parte de un tercero teniendo esta memoria como referencia.
 - Organización de los tiempos de forma clara: para desarrollar adecuadamente el proyecto se seguirá un orden en la implementación de las funciones para evitar posibles retrasos en el desarrollo.

El control de estas métricas o indicadores se llevará a cabo en cada una de las iteraciones establecidas previamente, realizando pruebas de aceptación de documentación y del código por parte del patrocinador/tutor. Los resultados de las inspecciones y pruebas se registrarán en el **Informe de Seguimiento** de cada iteración.

Para la mejora de la calidad se aplicarán correcciones en la medida de lo posible a los cambios o sugerencias que surjan de las reuniones anteriormente mencionadas y quedarán recogidas en los informes de seguimiento correspondientes. El proceso de acción correctiva será gestionado por el patrocinador/tutor y el redactor de la memoria.

Segunda Parte

Desarrollo

Sistema Desarrollado

En este apartado se describirá con precisión cómo han sido desarrolladas las distintas partes que comprenden la aplicación, en concreto, el backend y el frontend.

4.1. Modelo de datos

Para comenzar con el desarrollo se debe tener un esquema que defina la manera en la que los componentes interactuarán entre ellos una vez creados en la base de datos. Es por ello, que la creación de un modelo de datos es fundamental para vislumbrar con exactitud esta interacción (véase la figura 4.1).

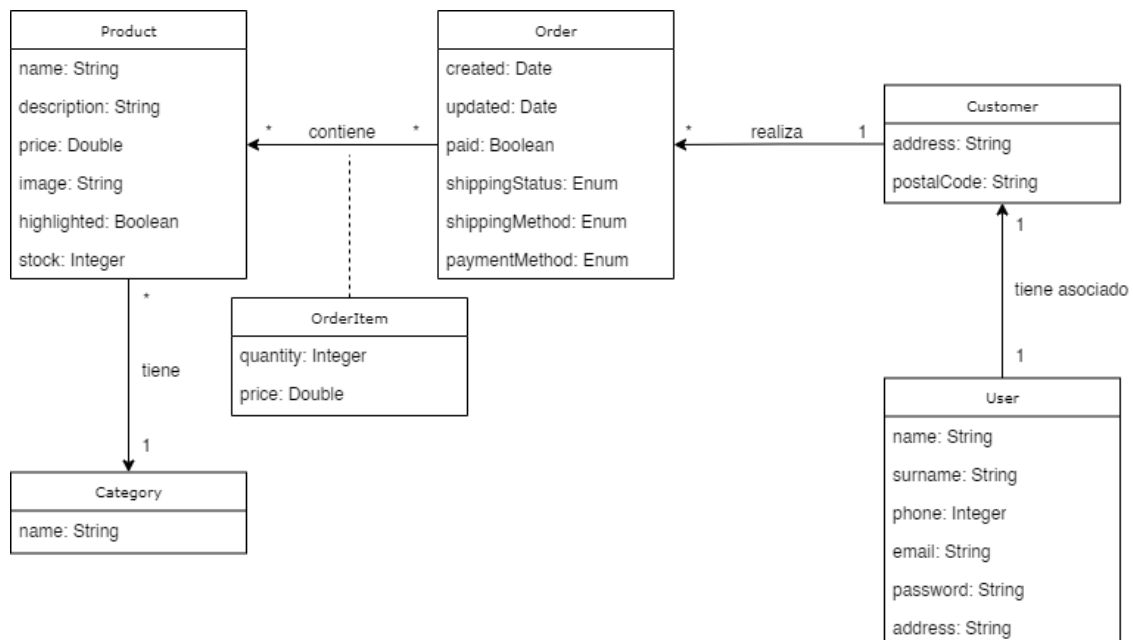


Figura 4.1: Diagrama UML de clases del backend

En el diagrama de clases del sistema se pueden distinguir cinco entidades que se relacionan entre sí, de manera que, para satisfacer la necesidad de diferenciar los productos por tipo de producto, se ha utilizado una relación múltiple para asociar a todos los productos de la base de datos una categoría. En cuanto a los clientes y usuarios, en el sistema aprovechamos la clase User de Django para heredar sus atributos y añadir algunos más en otra clase denominada Customer que tiene asociado siempre un usuario. Esto nos hace la tarea de autenticación y registro de usuario más sencilla de gestionar. Por último, los pedidos se ven reflejados en la clase Order, que tiene una tabla de asociación con producto para facilitar la tarea de relacionar un producto con su pedido.

4.2. Backend

El backend de la aplicación está formado por una API que interactúa con la base de datos proporcionada por PostgreSQL y un proyecto que se ha integrado con este que aporta la seguridad en la API. El mencionado proyecto es un ejemplo de la manera en que se podría implementar la herramienta django-two-factor-auth en cualquier proyecto Django, cuya licencia de uso se recoge en el apartado de **Licencia**. El doble factor de autenticación es una tecnología que se viene aplicando en cada vez más sitios web y permite generar un “token” de autenticación a cada usuario que lo tenga habilitado, y así se pueden proteger las peticiones a la API con autenticación 2.0, evitando que un usuario ajeno al sistema o que no tenga habilitado ese doble factor pueda acceder a ciertas áreas protegidas y usurpar información que no le corresponde o realizar ciertas acciones dentro de la página web (como por ejemplo añadir un producto al carrito).

Para gestionar los datos de la base de datos y enviarlos a través de la API al frontend de una manera más sencilla, conservando la lógica de modelos que nos proporciona Django, en cada aplicación de Django implementada se ha añadido un “serializador” y un “ViewSet”. Un “serializador” (traducción literal de la palabra “serializer”) es una clase que nos permite transformar datos de modelos Django a un formato como JSON o XML [15] y un “ViewSet” es una clase que proporciona una interfaz simple para realizar operaciones CRUD sobre un modelo en concreto, combinando la lógica de las vistas y estas operaciones en una sola clase, lo que simplifica en gran medida la creación de APIs. Normalmente, los “ViewSets” se utilizan junto con los “Routers” para vincular URLs a las acciones definidas en estos y exponer la API a través de los endpoints HTTP; que es lo que se ha hecho en el proyecto para facilitar toda la gestión de la API.

Tanto los “serializadores” como los “ViewSets” son clases que forman parte del conjunto de herramientas de Django REST Framework [14][15] para crear APIs REST de forma rápida y eficiente, que como se mencionó en el apartado de **Materias Relacionadas**, se ha utilizado para elaborar el backend de este trabajo. En las figuras que siguen se pueden analizar ejemplos de ambas clases del proyecto (véase las figuras 4.2 y 4.3).

```
from .models import Customer
from rest_framework import serializers

class CustomerSerializer(serializers.ModelSerializer):
    class Meta:
        model = Customer
        fields = ['address', 'postal_code', 'user']
```

Figura 4.2: “Serializador” de la clase Customer

```

from rest_framework import viewsets
from .models import Customer
from .serializer import CustomerSerializer
from two_factor.views.mixins import OTPRequiredMixin
from rest_framework.permissions import IsAuthenticated
from rest_framework.response import Response

class CustomerViewSet(OTPRequiredMixin, viewsets.ModelViewSet):
    permission_classes = [IsAuthenticated]
    queryset = Customer.objects.all()
    serializer_class = CustomerSerializer

    def list(self, request):
        if request.user.is_staff:
            queryset = Customer.objects.all()
            serializer = CustomerSerializer(queryset, many=True)
            return Response(serializer.data)
        else:
            return Response({"detail": "No tiene permisos para ver esta información."}, status=403)

```

Figura 4.3: “ViewSet” de la clase Customer

En las siguientes subsecciones, se procederá a detallar las partes o aplicaciones en las que se encuentra dividida el backend. Si el lector no está familiarizado con el framework Django, cabe mencionar que los proyectos creados con este se tiene un proyecto y una o varias aplicaciones, es decir, cada una de las carpetas que contiene información relativa a cada módulo (llámese productos o autenticación como en el caso de este trabajo) es una aplicación en Django, y el conjunto de esas aplicaciones y la carpeta que lleva el mismo nombre que el proyecto, que contiene los archivos de configuración, conforman lo que se denomina proyecto Django. La estructura de carpetas y organización del proyecto se encuentra descrita en la sección de **Implementación**.

4.2.1. Aplicación de autenticación

Esta aplicación gestiona todo lo relacionado con el registro y autenticación de los usuarios en la aplicación. Técnicamente es la única aplicación cuya interacción con la API se limita a la lectura de información de la base de datos, ya que la creación de nuevos usuarios y el cambio de contraseña se gestiona mediante una función recogida en el fichero views.py. Concretamente la única petición que interactúa con la API es un GET con la información de todos los usuarios del sistema y de la que se puede extraer también la información de un usuario en concreto. Esto se utilizará en frontend para mostrar los usuarios que han realizado pedidos en el panel de administración, es por esto, que se necesita tener permisos de administrador y tener activado el doble factor para tener acceso a esta información sumamente sensible del sistema.

Tal y como se expresa en el **Modelo de datos** la información que se recoge de los nuevos usuarios comprende campos como el nombre de usuario, correo electrónico, dirección y código postal. Como necesitamos asociar a un “User” con un “Customer” se necesita definir un método “save” que primero guarde en la base de datos, la información correspondiente al usuario (nombre, apellidos, nombre de usuario y contraseña); y una vez se guarda en la base de datos, se procede a almacenar la dirección del usuario y su código postal. En este caso la dirección es vital que se guarde ya que la entrega a domicilio de los pedidos se tiene que realizar en una dirección concreta y el código postal ayuda a distinguir en qué ciudad o pueblo se encuentra el cliente para posteriormente en el reparto, llevar a cabo la tarea de entrega de una forma más organizada y eficiente (aunque la aplicación se

limita a guardar esta información, no a la optimización de las rutas de entrega en función de los pedidos).

Como se utiliza la herramienta `django-two-factor-auth`, parte de la implementación de las plantillas en HTML se ha desarrollado haciendo uso de las recomendaciones de la documentación, por tanto, es necesario crear una carpeta en la que se tengan recogidas las plantillas base que se necesitan para dar formato y estilo a las páginas de registro y cambio de contraseña, cuyo nombre sea `"two_factor"`. De hecho, la razón por la que se ha escogido el desarrollo del módulo de autenticación puramente en Django es precisamente por la integración del doble factor en el proyecto; además que el desarrollo de la autenticación haciendo uso exclusivamente de Django y sus herramientas propiciaron un avance más rápido en el proyecto, evitando un incremento en las horas invertidas en esta parte en concreto de la aplicación, que cabe mencionar que se tardó más tiempo del esperado debido a la falta de conocimientos en un principio de la integración de varias herramientas en un proyecto Django. Se debe resaltar que al usar las plantillas de Django y no poder usar los estilos definidos en `Vue.js`, se ha realizado un esfuerzo añadido para que las páginas de registro, inicio de sesión (que depende de la aplicación `two_factor` que será explicada a continuación) y cambio de contraseña luzcan iguales que las implementadas con `Vue.js`.

Para finalizar con la explicación del desarrollo quisiera declarar que para que la experiencia de usuario fuera lo mejor posible en cada paso que se diera con la aplicación, se han implementado algunas páginas HTML adicionales, como la de `"signup_complete"` a la que se le redirige al usuario tras su registro exitoso en la aplicación y que insta al inicio de sesión en la página web para posteriormente activar el doble factor de autenticación y que se pueda tener acceso a todas las funcionalidades que ofrece la aplicación web a los clientes. Como último apunte a lo que se ha expresado con anterioridad, la aplicación de autenticación cuenta con un directorio en el que se recogen múltiples traducciones de las plantillas HTML para facilitar la internacionalización de estas en caso de que se necesite implementar esta casuística en futuras implementaciones. No obstante, para este proyecto sólo se ha hecho uso de la traducción al español para permitir que los clientes para los que inicialmente se ha pensado la aplicación puedan usarla cómodamente.

4.2.2. Aplicación `two_factor`

Esta aplicación sirve de apoyo a la descrita anteriormente, ya que aporta un grado de seguridad mayor al activarse la doble verificación de la identidad del usuario. Al ser un proyecto de uso público bajo licencia y no una implementación propia desde cero, sólo describiré los cambios más importantes que he realizado en él y daré una breve descripción de algunas de sus funcionalidades más destacables.

Primeramente se empezará describiendo en qué consiste esta tecnología y cómo funciona brevemente. Este proyecto utiliza la autenticación de doble factor que provee Django, que a su vez está construido sobre el framework de contraseña de un solo uso de Django, `django-otp` y el framework de autenticación integrado de Django, `django.contrib.auth`. Su funcionamiento consiste, en el caso de la generación de `"tokens"` de autenticación, en la asociación de un dispositivo que generará un código temporal de acceso a la aplicación al usuario que desee habilitar el doble factor de autenticación. Para poder habilitarlo se deberá escanear un código QR que contiene el primer código de verificación y que una vez escaneado mediante la aplicación Google Authenticator o similares, ya será esta aplicación la

que se encargue de generar códigos aleatorios cada cierto tiempo que escritos en el campo correspondiente al iniciar sesión, te facilitará el acceso a la aplicación. Si se desea eliminar el dispositivo que contiene el generador de códigos, es necesario que un desarrollador acceda a la base de datos de PostgreSQL y elimine el usuario que solicita su desactivación de la tabla “totp_devices”. Se podrían añadir correos electrónicos para la verificación o números de teléfono para recibir SMS si se desea ya que django-two-factor-auth es un framework con variedad de posibilidades, sin embargo, dado el alcance de este trabajo, se prefirió limitarlo al uso de códigos generados mediante QR.

Seguidamente, se detallarán las modificaciones que han tenido lugar en el proyecto para integrarlo con el trabajo.

En primer lugar, he modificado la descripción de la licencia de uso, ya que en el proyecto original se especificaba en cada uno de los archivos de traducción de idiomas que se pusiera el año en el que se integraba el proyecto de two_factor con el nuevo proyecto desarrollado, por tanto, se han modificado los 29 archivos correspondientes con el año 2024; además, algunos de ellos mostraban un error en la consola al ejecutarse cuyo origen tiene que ver con la configuración de Twilio en la aplicación, pero dado que no se ha utilizado finalmente Twilio para agregar un teléfono de contacto en el que poder enviar el código del doble factor vía SMS, se prefirió eliminar las líneas de código correspondientes a esa configuración en los archivos en los que había problemas de ejecución para facilitar el uso del doble factor.

Por otro lado, se ha personalizado la URL a la que se redirigía al usuario si se pulsaba sobre el logotipo de la empresa en el HTML del inicio de sesión y en la página de activación del doble factor, para mantener la consistencia de la aplicación en todo momento. El inicio de sesión y el panel de activación/desactivación del doble factor pertenecen a este proyecto y no son de implementación propia salvo los cambios anteriormente mencionados, de hecho, las plantillas base como “base_focus.html” se referencian directamente desde la aplicación “authentication” desarrollada en el subapartado anterior, por lo que cualquier cambio en las plantillas pertenecientes a la aplicación two_factor como tal, no afectan a las páginas que se visualizan en la aplicación web.

4.2.3. Aplicación “base”

Esta podría decirse que es la aplicación que menos funcionalidades recoge pero sigue siendo importante para ciertas partes del proyecto, sobre todo la parte de las **Pruebas**.

Entre sus componentes más importantes se encuentra mods.py y tests.py, el primer fichero proporciona una interfaz para realizar consultas HTTP a otros módulos del proyecto y también proporciona una manera de simular esas consultas para pruebas unitarias; el otro fichero provee de una clase base para implementar tests de una forma más sencilla, recogiendo funciones para configurar y desmontar los tests, login de prueba y logout de prueba. Cabe destacar que estos ficheros han sido extraídos de un proyecto que finalizó en el primer cuatrimestre del cuarto curso y se ha estado trabajando con ellos para la realización de pruebas unitarias en el backend de dicha aplicación; no obstante, en el apartado de **Licencia** se incluye la licencia de uso de este y como se especifica en la mencionada sección, se puede hacer uso del mismo en cualquier proyecto y si no se modifican los ficheros (el caso de este trabajo) no es necesario aportar cambios al proyecto original.

4.2.4. Aplicación de pedidos

Esta aplicación recoge todas las funcionalidades relacionadas con la gestión de los pedidos del proyecto, exceptuando el pago que se ha implementado en una aplicación a parte denominada “payment” (se describirá en la siguiente subsección).

Como se ha descrito en la sección de **Modelo de datos** un pedido necesita estos campos para poder crearse correctamente: fecha de realización, fecha de modificación, si está pagado o no, el estado del envío, tipo de envío y método de pago. Además cada pedido tiene asociados los productos que se tienen que enviar a través del objeto `OrderItem` que contiene cantidad y precio del producto solicitado; aunque prácticamente, estos objetos se tratan a través de una lista de objetos JSON que contiene dicha información y a continuación se explica el porqué de esta situación.

Dado que el carrito como se explicará más adelante se optó por implementarlo puramente en JavaScript para aumentar la eficiencia del sistema y repartir la carga de datos entre backend y frontend, una vez que el cliente solicitaba sus productos a través del carrito y formalizaba su pedido seleccionando el método de pago y de entrega, los datos del carrito que en frontend se encuentran en formato JSON debían trasladarse al backend a través de la API para guardar la información del pedido en la base de datos, previa conversión de los datos mediante los “serializadores” correspondientes.

Aclarado esto, las peticiones HTTP desarrolladas para interactuar con el backend son:

- GET a todos los pedidos: solamente accesible por el o los administradores del sistema. Devuelve una lista de todos los pedidos registrados por el sistema.
- GET a los pedidos de un usuario: solamente accesible por el usuario que lo solicita y sólo puede acceder si este tiene activado el doble factor de autenticación. Devuelve los pedidos que ha realizado en el sistema. Ningún usuario ajeno a otro puede acceder a esta información.
- POST de un pedido: esta petición solo está permitida para aquellos usuarios que tengan activado el doble factor y deseen realizar un pedido. Tras enviar la petición, se registrará un nuevo pedido en el sistema. Los datos que se requieren para completar el pedido serán:
 - Lista de productos solicitados
 - Cantidad de dichos productos
 - Tipo de entrega
 - Método de pago

Dependiendo del método de pago escogido se procederá al cobro del pedido mediante Stripe [16] o dicho pago deberá ser abonado al efectuarse la entrega.

Cuando se crea un pedido en el sistema, primero se crea el pedido y luego el objeto `OrderItem` mencionado anteriormente. Además, se crea una sesión de Stripe para procesar el pago si el cliente ha escogido la opción del pago con tarjeta.

En cualquiera de los dos casos de pago, se le envía al cliente un correo informando sobre la confirmación de su pedido.

- PUT de un pedido: esta acción está solo permitida a usuarios administradores que tengan el doble factor activado. Está orientada a la actualización de dos de los campos de un pedido, en el que se incluye el estado de entrega/envío del pedido y si se ha pagado o no, este último dato tiene mucho sentido contemplarlo dado que es necesario saber si un cliente al pagar en efectivo el pedido ha abonado el importe finalmente o no, permitiendo conocer si el usuario tiene impagos o no (cabría pensar en la posibilidad de que el administrador no registrara adecuadamente esta información en el sistema, cosa de la que el autor de la memoria no se hace cargo, ya que conllevaría problemas entre clientes y empresario).
- DELETE de un pedido: esta petición nuevamente no está permitida a los clientes, sólo a los administradores que tengan el doble factor activado. Se contempla el borrado de los pedidos siempre y cuando el administrador lo considere necesario, ya que idealmente los grandes pedidos de pan que un restaurante o bar quiera hacer, a la panadería no lo compensaría económicamente tener en tienda material que no pueda vender a un cliente más pequeño; por lo que queda en las manos del administrador si desea eliminar o no un pedido. No obstante, a pesar de que se pueda eliminar un pedido, el importe abonado con tarjeta debería ser devuelto manualmente, es decir, no está automatizada la devolución del importe, ya que esta característica no ha sido contemplada durante la implementación (el patrocinador no la ha solicitado en ningún momento).

4.2.5. Aplicación de pagos

Esta aplicación del proyecto permite el procesamiento de los pagos que realiza el cliente a través de la plataforma Stripe. La función esencial de los pagos está implementada en el archivo webhook.py que a través de la URL payment/webhook se activa para guardar la información del pedido restante, esto es, si se ha pagado correctamente el importe y se actualiza el campo del estado del envío a “Pendiente”.

4.2.6. Aplicación de productos

La última aplicación del proyecto Django corresponde a los productos. En esta se gestiona la otra parte esencial de la aplicación web que tiene que ver con la creación, edición y listado de productos de la base de datos. Para cada producto, tal y como se indica en el **Modelo de datos**, se guarda su nombre, descripción (que pueden ser por ejemplo los ingredientes con los que se haya hecho el producto), precio por unidad, una imagen (para que se pueda apreciar qué forma tiene la pieza de pan o el producto en general), si está destacado (el cliente nos exigió que en la página principal hubiera una sección de productos destacados) y el stock o la cantidad de productos que hay en existencias en la tienda.

Todos los productos tienen asociado una categoría que permite distinguir de qué tipo son, normalmente las piezas de pan se clasifican en pan normal (hecho con harina de trigo refinada) e integral, aunque también puede haber productos como regañá y picos que pertenecen a la categoría de “Pico”. Estos últimos tienen un registro más exhaustivo del stock que hay en la tienda, debido a que se venden por bolsas y estas dependen en gran medida de los pedidos que realice el panadero a su proveedor en caso de no trabajarlas en su obrador. El resto de los productos

pueden depender de la cantidad de pan que se haya pedido el día anterior para repartir, por lo que el stock en las piezas de pan no es tan importante (aunque se registre en la base de datos).

Las peticiones de la API que han sido desarrolladas para los productos son estas:

- GET a todos los productos: es la única petición que no requiere autenticación de ningún tipo para ser llamada, ya que cualquier usuario que desee consultar los productos de la tienda, debe poder hacerlo sin necesidad de darse de alta en la aplicación. Se devuelve una lista con todos los productos (incluyendo sus respectivas categorías). Esta petición devuelve por se todos los datos de los productos, por lo que no se ha contemplado una página de detalles como tal de los productos.
- POST de un producto: sólo es accesible para los administradores que tengan activado el doble factor. Esta petición añade un nuevo producto al sistema con algunos matices que se explicarán en breve. Los datos que se requieren para realizar esta petición con éxito son:
 - Nombre
 - Descripción
 - Precio
 - Imagen
 - Producto destacado
 - Stock
 - Categoría

Hay una serie de matices o restricciones aplicadas a los tipos de datos para evitar que el administrador del sistema introduzca valores en los campos que no sean válidos y puedan provocar un error en la base de datos o que no tengan mucho sentido. Estas restricciones son las siguientes:

- El precio del producto no puede ser negativo ni cero.
- La categoría que se escoge para el producto debe existir previamente en el sistema.
- No se puede crear un producto si el stock es negativo y su categoría es "Pico".
- Para evitar que se introduzca un valor en el stock que sea negativo si la categoría no es "Pico" se ha implementado una funcionalidad que permite asociarle el valor máximo de un entero a dicho valor de manera que se evite un "underflow" de datos (en el caso de las piezas de pan que, como tal no tienen un stock definido). Dicho valor máximo se limita a un entero de 32 bits que es el tamaño máximo que soporta el sistema sobre el que se está desarrollando la aplicación.
- Las imágenes que se suben a la base de datos a través de la aplicación se guardan directamente en una carpeta del backend, es por eso por lo que para evitar las imágenes repetidas se ha implementado una función denominada "check_image" que evita la repetición de imágenes de manera que se pueden las imágenes que se quieran sin llegar a sobrepasar el límite de capacidad de la carpeta y tener un pequeño

orden establecido en la misma. Django por defecto guarda una imagen repetida con un nombre distinto del archivo original generando aleatoriamente una cadena de caracteres alfanuméricos, sin importar la cantidad de imágenes que se quieran añadir; y con esta funcionalidad evitamos que se repitan las imágenes ocupando espacio innecesario en el backend. También se contempla el tamaño máximo de una imagen y el formato que debe tener para acotar el tipo de archivo que se va a almacenar posteriormente.

- PUT de un producto: al igual que los pedidos, los productos también se pueden editar de manera que solo los administradores que tengan dicho doble factor van a poder acceder a él. Todos los campos mencionados anteriormente son editables, sin embargo si un administrador elimina una categoría y había productos asociados a dicha categoría el producto no se podrá guardar hasta que tenga asociado nuevamente una categoría.
- DELETE de un producto: del mismo modo en que se pueden eliminar pedidos, también se pueden eliminar productos del sistema. Sólo es accesible por administradores que tengan el doble factor de autenticación activado.
- GET de las categorías: accesible a todos los usuarios de la aplicación, independientemente de si están registrados o no, ya que es necesario que todos los productos de la tienda se muestren con su categoría correspondiente. Devuelve una lista con todas las categorías de la tienda.
- POST de una categoría: sólo accesible para los administradores que tengan el doble factor habilitado. Para poder crear una categoría se necesita asignar un nombre a la nueva categoría.
- PUT de una categoría: a nivel de permisos funciona igual que el POST y permite modificar el nombre de cualquier categoría.
- DELETE de una categoría: al igual que el PUT, sólo accesible a administradores que tengan el doble factor activado. Tal y como se ha mencionado anteriormente en el PUT de un producto, si se elimina una categoría y se quiere actualizar aquel o aquellos productos que tenían asociado la categoría eliminada, se tiene que asociar una nueva categoría a esos productos.

4.2.7. Dificultades encontradas

En este subapartado se citarán aquellas barreras que han debido ser superadas para completar el backend con éxito. Desde el comienzo del desarrollo, casi todo lo que se ha conseguido implementar no ha sido más que una consecución de tareas y dificultades que han intentado embarrar el progreso del proyecto, pero que forman parte del proceso de aprendizaje por el que se ha debido transitar para alcanzar los objetivos de este trabajo.

Al principio del proyecto, la gran dificultad encontrada fue aprender a utilizar Django, ya que se intentó avanzar una pequeña parte en septiembre; sin embargo, las demás asignaturas supusieron un esfuerzo que a posteriori sirvieron como ejemplo para poder llevar a la práctica los conocimientos que he adquirido a lo largo de este cuarto curso. Por tanto, los primeros recursos consultados para superar estas dificultades de principiante con una tecnología hasta el momento desconocida

fueron videotutoriales que se encuentran referenciados en estos enlaces [16][18][19][20]. Dos de ellos se corresponden con tutoriales acerca de los primeros pasos que se deben dar para construir un proyecto Django y los otros contemplan la configuración del login y logout de una página web con Django. Esta última referencia fue especialmente útil, ya que al haber trabajado con Spring Boot durante el tercer curso, framework que facilitaba la gestión de la autenticación de los usuarios en un proyecto base, me costó bastante trabajo adaptarme a la nueva forma de configurar la autenticación con Django.

Durante el proceso de integración del doble factor en el proyecto se consultaron varias fuentes, entre la que se encuentra la habilitación del doble factor [21], hasta que se dio con la verdaderamente útil, que vendrá más adelante referenciada en la sección de **Licencia**.

En el resto del proyecto a nivel de backend se han ido sucediendo pequeños problemas que se han ido solventando con un poco de paciencia e investigación de las posibilidades; aunque la mayoría de las situaciones ocurrían por la falta de atención a los pequeños detalles del código.

4.3. Frontend

Descrito el backend, se aprovechará esta sección para detallar el desarrollo del frontend, es decir, la parte de la aplicación con la que interactúa el usuario. En sí, el frontend que se ha desarrollado, como ya se ha citado en otras ocasiones, ha sido fruto de la adaptación del proyecto Django base del que se ha partido para la realización del trabajo a Vue.js. No obstante, se hará hincapié en los detalles más importantes en los que se ha realizado modificaciones en la medida de lo posible.

4.3.1. Página de inicio

Este puede ser uno de los componentes que más tiempo y esfuerzo ha supuesto desarrollar en la parte del frontend. Esto es porque la página de inicio del proyecto base de Django no tenía configurado los estilos del HTML y convertir dicha página base a una página que pudiera interpretar Vue.js correctamente fue todo un reto.

Está compuesto por varios elementos que son fundamentales en toda la aplicación, una barra de navegación que está implementada en el componente “Navbar.vue” y un pie de página que cuya implementación se recoge en el componente “Footer.vue”. El primero de ellos es el que más problemas ha generado debido a los estilos utilizados, es decir, anteriormente a la puesta en marcha del proyecto, los conocimientos en CSS del redactor de la memoria no eran demasiado extensos; es más, la última vez que se tuvieron que poner en práctica fue en la asignatura IISI 2, por lo que para hacer frente a este tipo de contratiempos se ha recurrido en numerosas ocasiones a la documentación de CSS disponible en Internet, aunque las inteligencias artificiales han ayudado en gran medida a ahorrar una inmensa cantidad de tiempo en lo que ha estilos respecta. Para que quede constancia en esta memoria, se destaca la gran dificultad que supuso la colocación de los botones de inicio de sesión, cierre de sesión y registro, justo a la derecha de la barra de navegación y el resto de los elementos, justo a la izquierda; para que fuera lo más parecida posible a la barra de navegación de la página de inicio del proyecto base de Django del que se partió. Una librería imprescindible para dar forma a la página en todos los aspectos ha sido Bootstrap 5 con la que se tuvieron

algunos problemas de integración que están referenciados en este enlace [22]. Un detalle importante en el que se invirtió bastante tiempo fue la obtención del correo electrónico del usuario que ha iniciado sesión para colocarlo en la esquina superior derecha, ya que se necesitó configurar una variable de configuración en el archivo `settings.py`, concretamente `CORS_ALLOW_CREDENTIALS` y Axios se configuró para enviar cookies de la sesión en cada solicitud.

La forma en la que está organizada la página de inicio, con una imagen de fondo que concuerda con la temática de este proyecto, se debe las ideas que se extrajeron de un videotutorial en el que se explicaba cómo implementar una página de inicio o “landing page” con Vue.js [23].

En la barra de navegación se encuentran distintos menús, como el acceso al catálogo de productos, la página “Sobre nosotros” que cuenta la historia de la panadería, y un desplegable en la parte derecha que nos permitiría acceder a “Mis pedidos”, activar el doble factor si no se ha activado durante el proceso de registro del usuario y “Cambiar contraseña”. Exceptuando la parte de “Mis pedidos”, el resto de las opciones se manejan con páginas en HTML desde el backend.

Luego, en cuanto a la manera en la que se contrae la barra de navegación, se pensó desde un principio en la implementación de un frontend que fuera responsive como se ha indicado a lo largo de esta memoria, por lo que era fundamental programar con el pensamiento de que la página podría ser vista desde un móvil. Para poder visualizar el contenido de la página en el formato en el que se vería en cualquier dispositivo móvil, se ha tomado como referencia el tamaño de la pantalla de un iPhone 14 Pro Max, y la razón de su elección no es otra sino la de escoger alguna pantalla de entre las disponibles en el modo de Inspección del navegador. Teniendo esto en cuenta, la barra de navegación se contrae para que en un dispositivo móvil se pueda acceder a cada uno de los menús sin que la barra de navegación exceda el tamaño permitido por el dispositivo que se está usando y así poder distinguir más fácilmente cada uno de los elementos de esta. También se ha tomado como referencia la barra de navegación empleada en el proyecto original de Django.

En la página de inicio es posible añadir productos destacados al carrito, así como iniciar sesión en la aplicación o navegar a otras partes de esta usando la barra de navegación.

4.3.2. Catálogo de productos

Siguiendo con la línea temporal del desarrollo, el catálogo de productos fue la siguiente tarea a realizar. Para ello, se han seguido varios videotutoriales que se encuentran referenciados en el siguiente enlace [24]. Se crearon un componente y una vista para el catálogo, el componente es “ListProduct.vue” y la vista “ProductsView.vue”. En el componente reside la parte más importante de la lógica del catálogo, ya que en este se recoge la petición a la API de backend para la obtención de los productos y su disposición en forma de tarjetas. Las peticiones a la API se han realizado haciendo uso de la librería Axios [25] de JavaScript, con la que se pueden obtener fácilmente datos en formato JSON y gestionar los datos de una manera fácil e intuitiva con las promesas.

Tanto en el catálogo como en la página de inicio es necesario realizar dos peticiones a la API de backend para poder traernos la información relativa a los productos y sus categorías. También se implementa la lógica con JavaScript necesaria para poder añadir productos al carrito y aumentarle la cantidad que se

desea, aunque se ha contemplado una serie de restricciones en el campo en el que se introduce el valor de la cantidad, para evitar que se puedan agregar cantidades negativas o nulas (valor cero).

Como último detalle a destacar de este componente se ha contemplado con condicionales de Vue.js (conocidos en su framework como `<template v-if>`) el stock de los productos más sensibles como lo son los de la categoría “Pico”, es decir, que si el producto que se va a mostrar en la tarjeta pertenece a dicha categoría, se indicará tanto la cantidad de productos que quedan en stock como su disponibilidad mediante un símbolo de Disponible o Agotado; intentando en todo momento que todos los usuarios puedan comprenderlo sin esfuerzo, teniendo en cuenta la accesibilidad del sistema ante el que se encontrarán los clientes.

Si se observa atentamente la vista de productos, es decir, en el componente “ProductsView.vue”, se podrá deducir que es un componente con una lógica muy simple, en el que sólo se realiza la importación de otros tres componentes, el “navbar”, el “footer” y el componente de la lista de productos mencionado anteriormente. Esto es porque en el videotutorial se sugería implementar como componentes independientes cada una de las partes de las vistas y además es bien sabido por los ingenieros que una buena práctica para el buen mantenimiento del software que se construye es la de modularizar en la medida de lo posible los elementos que forman parte del resultado final de un proyecto. Es por ello por lo que se ha priorizado a lo largo de todo el desarrollo del frontend, si fuera posible, que se implementen en componentes separados las vistas de la aplicación; aunque, como se podrá leer más adelante en esta sección, no siempre ha sido posible dicha separación por la forma en la que trabaja Vue.js cierto tipo de componente.

4.3.3. Carrito de la compra

En otras secciones se ha mencionado la especial importancia que ha tenido la implementación del carrito de la compra en este proyecto. Concretamente porque en lugar de desarrollarlo en Django se ha implementado en JavaScript desde cero para acelerar el procesamiento de la información obtenida de los productos solicitados por el cliente.

De entre las acciones que se pueden realizar se pueden destacar el aumento o reducción de la cantidad deseada de un producto y eliminar un producto del carrito. Para añadir más complejidad al componente y que visualmente fuera más familiar para el cliente, las opciones de pago y envío se muestran en la misma pantalla, pero con el fondo más oscuro. Esto es así, por la configuración de los estilos y contenedores que residen en el componente “PaymentOptions.vue”, que además permite confirmar o cancelar el pedido.

Sin embargo, si el cliente no se ha registrado o no ha iniciado sesión en la aplicación, se le mostrará un mensaje en el que se le incitará a realizar tal acción en la página y no se le mostrarán las opciones de pago y envío mencionadas. Una vez se haya dado de alta en la página, el carrito seguirá teniendo los mismos productos que hubieran sido seleccionados previamente, ya que los datos de los productos solicitados se guardan en las cookies del navegador y por tanto no desaparecen, a no ser que la caché del navegador la borre el cliente de manera voluntaria.

Uno de los grandes problemas a los que se tuvieron que hacer frente durante el desarrollo del carrito de la compra fue la transmisión o envío de los datos del carrito

al backend para poder crear un nuevo pedido. El error ocurrió por la falta de conocimiento respecto a la protección contra ataques CSRF que Django tiene activada por defecto y es que no cualquiera podía realizar un pedido si se recuerdan las restricciones aplicadas a la petición POST de un pedido. El cliente debía haber iniciado sesión previamente a la realización de un pedido, sin embargo, aunque se iniciara sesión el backend no reconocía al usuario autenticado. Tras varios días de investigación [26][27][28], se concluyó que el problema radicaba en una opción al enviar una petición POST con Axios, en la que se adjunta el token de autenticación generado por Django al iniciar sesión en el backend. También era necesario añadir una opción de configuración en el archivo settings.py de Django, concretamente la opción `CSRF_COOKIE_NAME = "csrftoken"`. Solo así, Django reconocería al usuario que está realizando la petición y podría aceptar el envío de los datos del pedido para posteriormente crearlo. No obstante, al ser una cookie con datos de sesión del cliente, era de vital importancia eliminar la información sensible al cerrar sesión; por lo que se contempló esa funcionalidad al cerrar sesión y evitar posibles filtrados de información por un tercero que acceda a las cookies del navegador.

El “footer” jugó un papel importante en este componente dado que se tenía que colocar pegado al pie de página y al no saber hacerlo, me dispuse a investigar por Internet en busca de una respuesta [29]. Al final se trataba de una simple configuración en CSS para que estuviera siempre en el pie de la página, aunque no siempre se deseaba colocarlo en el pie, esto depende en gran medida del espacio que ocupen el resto de los elementos de los componentes en general, por lo que se ha recurrido en más de una ocasión a condicionales de plantilla de Vue.js para gestionar su colocación.

Uno de los ficheros más importantes del proyecto que afecta mayormente en las funcionalidades que tiene el carrito es el store.js. En él se encuentran todos los métodos relacionados con las posibles acciones que se pueden realizar en el carrito, como lo son añadir un producto, eliminarlo, aumentar o disminuir la cantidad de productos, enviar la orden de creación del pedido al backend, una vez creado el pedido guardar su id en las cookies para luego mostrarla en la pantalla de pedido completado y mostrar las opciones de pago cuando el cliente haga click en el botón de “Realizar pedido”. Pero no sólo gestiona todo lo que se ha mencionado acerca del pedido, también se obtiene la información del usuario que ha iniciado sesión en el sistema, carga las categorías desde el backend para posteriormente mostrarlas en cualquier componente y carga el carrito para que se pueda trabajar con él. En definitiva, gracias a este componente la mayoría de las funcionalidades del frontend realizan su función correctamente.

Finalmente, destacaría que este componente ha sido uno de los pocos sobre los que se han realizado pruebas unitarias para comprobar que efectivamente, funciona tal y como se espera que lo haga. Sin embargo, el testing de este componente no ha sido sencillo [33][34], se ha necesitado consultar varios aspectos [35], como la forma en la que se pueden implementar las funciones de prueba del store.js para poder hacer las llamadas a la API sin necesidad de que esté operativa realmente y también comprobar que se realizan las llamadas correctamente a las funcionalidades internas sin que se tengan que ejecutar directamente o ejecutando una función de prueba que realice su misma función sin necesidad de utilizar las funciones originales. Todas las pruebas implementadas se explican con más detalle en la sección de **Pruebas**.

4.3.4. Mis pedidos

Otro de los requisitos solicitados por el cliente fue la posibilidad de acceder a los pedidos que realizaba en la aplicación y así se ha recogido en una sección dentro de la barra de navegación que corresponde al usuario. En la esquina superior derecha donde aparece el correo del usuario que se ha autenticado en la aplicación web puede observarse que se despliegan una serie de opciones cuando se deja el cursor sobre el correo y en el que aparece entre ellas la opción de “Mis pedidos”.

En esta página se muestran los pedidos y los productos asociados a los pedidos del usuario que ha iniciado sesión y para poder acceder a ella como ya se ha explicado anteriormente en otras secciones se debe activar el doble factor de autenticación. Una de las características que tiene esta página es la capacidad de filtrar por identificador del pedido, de manera que se tiene un acceso más sencillo a todos los pedidos realizados con anterioridad. También cuenta con paginación para que se muestren en cada página cinco pedidos y que no sea una página con scroll infinito en el que los clientes se pierdan buscando sus pedidos. Como es evidente, cualquier modificación que el administrador lleve a cabo sobre los pedidos de los clientes, aparecerá en esta página, informando a los usuarios de los nuevos cambios que se han producido y así poder realizar un seguimiento de los pedidos.

Por último, se puede mencionar que se ha implementado un círculo de carga que puede resultar familiar al usuario de la aplicación para mostrar que se están cargando los pedidos y que no se muestre directamente un mensaje de “No hay pedidos” cuando en realidad sí hay pedidos en la base de datos.

4.3.5. Administración de productos

Aunque Django facilita la administración de los objetos que están registrados en la base de datos y también en sus ficheros admin.py, para hacer aún más simple la tediosa tarea de gestionar la página desde el punto de vista de un administrador, se ha implementado todo un sistema para poder actualizar y crear nuevos productos y categorías en la tienda.

Para ello, se han creado varios componentes como “ListProductAdmin.vue”, “AdminCreateProduct.vue”, “AdminEditProduct.vue”, “ListCategoriesAdmin”, “AdminCreateCategory.vue” y “AdminEditCategory.vue”. El primero de todos, como su propio nombre indica, lista todos los productos de la base de datos pero los muestra de una manera en particular. A diferencia del catálogo de productos, en el que los productos se muestran en tarjetas con un número y un botón de “Añadir”, en este componente por el contrario en las tarjetas se muestran dos botones distintos, uno en el que se accede a la edición del producto y otro que tiene la capacidad de eliminarlo de la base de datos. Si el administrador quiere eliminar un producto, se le muestra un aviso en el navegador para que confirme la eliminación del producto si está seguro de ello, y una vez haga click en “Aceptar” se eliminará el producto y se le mostrará un mensaje indicando que el producto ha sido eliminado correctamente. Tras unos segundos el mensaje desaparecerá. Además en la parte superior de la página existe un botón para “Crear Producto” en el que se accede a un formulario para crear un nuevo producto en el catálogo. Ninguna de estas opciones es accesible si el administrador no tiene activado el doble factor, por lo tanto, se le avisa con un mensaje en la pantalla para que lo active si no lo ha hecho previamente.

El segundo componente que se cita contiene un formulario para crear un nuevo producto, en el que se solicita el nombre del producto, su precio y una descripción, entre otros campos. El más interesante y que llevó más tiempo implementar fue la gestión de la subida de la imagen que se desea agregar al producto. Esta gestión comprende dos funciones principales, la primera de ellas es una llamada “handleFileUpload” en la que se gestiona completamente la subida de la imagen se envía al backend para comprobar si efectivamente la imagen cumple con los requisitos que se solicitan y/o existe en la base de datos, en cuyo caso se le informará al administrador de que existe en la base de datos y se propondrá al administrador seleccionar una imagen distinta. La otra función o configuración se encuentra integrada dentro del método “createProduct” en el que se utiliza un “FormData” para poder agregar la imagen al formulario de creación del producto y crearlo correctamente con la imagen agregada. Cuando se añade el producto con éxito a la base de datos, se redirige al administrador nuevamente a la lista de productos mencionada anteriormente y se muestra un mensaje de éxito en dicha página.

El tercer componente, “AdminEditProduct” funciona de una manera similar al anterior, con la única diferencia de que en este componente se cargan los datos del producto a modificar. De igual manera, también se pueden subir imágenes y se gestiona de la misma forma; aunque si ya tiene asociada una imagen el producto que se desea modificar y no se quiere eliminar la imagen, se deja tal y como está y no se guardaría de nuevo en la base de datos.

Pasando al componente de listado de categorías, se limita a mostrar en una tabla todas las categorías existentes en el sistema. Para poder crear, editar o eliminar una categoría es necesario que el administrador tenga activada la doble verificación. Si no es así, se le mostrará un mensaje indicándole que debe activarla para acceder a las funciones mencionadas.

La creación de categorías es un componente bien simple en el que lo único que se pretende es que se rellene un campo con el nombre de la categoría y ya se podría crear. Con la edición ocurre algo similar, basta con teclear el nombre que se quiere poner en lugar del antiguo y ya estaría hecho.

La razón por la que existen vistas que integran el listado de productos y de categorías, es decir, hay un componente “AdminProductsView.vue” y “AdminCategoriesView.vue” y no hay como tal vistas de la edición y creación de ambos objetos es por aquello que se citó al final de la subsección del **Catálogo de productos**, es decir, estos componentes de edición y creación no soportaban la creación de vistas para integrarlos, ya que los estilos definidos no se ajustaban correctamente y por tanto, ninguno de los componentes funcionaba como se esperaba. Así que se decidió separarlos en distintos componentes que tuvieran su “Navbar” y su “Footer” integrados en ellos.

4.3.6. Administración de pedidos

La gestión de la administración de los pedidos la conforman dos componentes del frontend denominados “ListOrderAdmin.vue” y “AdminEditOrder.vue”. Como su propio nombre indica el primero está diseñado para listar los pedidos de toda la base de datos y el otro sirve para editarlos.

“ListOrderAdmin” actúa de una manera muy similar a “MyOrders” ya que ambos en realidad están pensados para listar los pedidos y sus productos asociados, la diferencia radica en el acceso que se tiene a cada una de estas páginas en la

aplicación y las funcionalidades que ofrece una respecto a la otra. En “MyOrders” la característica más destacada es el listado y la búsqueda de pedidos, pero no ofrece la posibilidad de interactuar con los pedidos más allá de mover las tablas que aparecen para distinguir todos los campos que recogen; sin embargo, “ListOrderAdmin” considera dos botones en cada pedido, uno para poder editarlo y otro para eliminarlo, aunque como ya se ha detallado en otra sección de esta memoria, la eliminación no abarca la devolución del importe abonado del pedido, por lo que se limita a eliminar el pedido de la base de datos. Si se desea eliminar, se mostrará un aviso, al igual que ocurría con los productos o las categorías, preguntando al administrador si está seguro de dicha eliminación y tras darle al botón de “Aceptar” se borraría del sistema. Como el listado de todos los pedidos de la aplicación puede convertirse en una tarea engorrosa dado que podrían existir multitud de clientes que realicen pedidos diariamente a la panadería, se ha considerado la paginación de los pedidos, mostrándose de cinco en cinco y se ha agregado una barra de búsqueda que permite filtrar por nombre y apellidos de cliente y número de pedido para facilitar la tarea de encontrar pedidos en concreto. Además, la colocación del nombre del cliente que ha realizado el pedido justo encima de este permite distinguir mejor los pedidos entre cada cliente.

Cuando se edita un pedido, entra en juego el componente “AdminEditOrder.vue” en el que se encuentran cuatro campos modificables, el estado del pedido, si está o no pagado, el método de envío y el método de pago.

4.3.7. Estadísticas

Las estadísticas se han desarrollado usando un componente principal de vista y otro auxiliar para poder implementar correctamente las gráficas.

El componente auxiliar, denominado “BarChart.vue” contiene las instrucciones necesarias para generar un gráfico de barras configurando los parámetros que luego hará falta rellenar con datos y con opciones para que se muestre la información que el administrador solicitó en los requisitos.

La vista principal, “AdminStatsView.vue” contiene la lógica que trae los datos del backend para realizar las consultas estadísticas de filtrado necesarias para mostrar los productos más vendidos y los usuarios que más pedidos realizan en el sistema. Los productos más vendidos se obtienen siguiendo estos pasos:

1. Mediante una solicitud GET se obtienen todos los pedidos del servidor.
2. Se recorre cada pedido y cada producto solicitado para contar la cantidad total de cada artículo vendido.
3. Una vez se cuentan todos los productos, se realizan solicitudes adicionales para poder obtener la información asociada a esos productos.
4. Se ordenan los productos por la cantidad vendida y se almacena la información en una variable que luego será consultada por el gráfico.
5. Se preparan los datos para que se visualicen en el gráfico y se configuran las etiquetas que aparecerán en él para que correspondan a los productos más vendidos y sus cantidades.

De una manera similar se calculan los usuarios que más pedidos realizan:

1. Inicia una solicitud GET para obtener todos los clientes del servidor.
2. Una vez que se obtienen los clientes, se inicializa un objeto al que se ha llamado “userCounts” para llevar la cuenta de los pedidos de cada usuario.
3. Luego, se realiza una solicitud GET para obtener todos los pedidos del servidor.
4. Una vez que se obtienen los pedidos, se recorre cada pedido y se aumenta el contador del cliente correspondiente en el objeto “userCounts”.
5. Después de contar todos los pedidos, se realizan varias solicitudes adicionales para obtener la información de cada usuario.
6. Una vez que se obtiene toda la información del usuario, se ordena los usuarios por la cantidad de pedidos y se almacenan los usuarios con más pedidos en una variable que posteriormente se utilizará para mostrar los usuarios en una tabla.

También se muestra el total de productos existente en el catálogo y el número total de pedidos por si fuera de interés para los administradores.

4.3.8. Página “Sobre Nosotros”

Dado que el cliente nos solicitó que mostráramos una página con información relativa a la tienda, así se ha plasmado en el componente “AboutView.vue”, uno de los más simples que se pueden encontrar en la parte del frontend, ya que se limita a mostrar un párrafo con la información y una imagen de un panadero amasando pan en su obrador.

Aún así, esta es una página que ha mostrado bastante resistencia a la colocación del pie de página en la parte inferior de esta, teniendo que realizar una operación bastante compleja respecto al tamaño de la pantalla para evitar que el “footer” tapara el texto en algunas pantallas. El problema se solucionó trabajando con varios “footer” que se muestran en función del tamaño de la pantalla del invitado para que se adapte mejor a las distintas circunstancias.

4.4. Seguridad

Respecto a la seguridad de la aplicación web desarrollada, se ha tenido en cuenta varios aspectos sobre todo de configuración para evitar un posible ataque a esta. En el backend, el archivo settings.py está configurado para que a nivel de CORS las únicas peticiones que se aceptan son de parte del frontend, es decir, el único origen permitido mediante la política CORS es el “localhost:3000” aunque como posteriormente se explicará en el apartado de **Despliegue** se utilizó primero la dirección “localhost:5173” que es la que aparece por defecto cuando se levanta el frontend utilizando el comando “npm run dev” y por eso también se permite en el CORS dicha dirección. Además los únicos orígenes permitidos en la política CSRF son esas direcciones.

Para aumentar la seguridad de la configuración del backend se ha creado un archivo con las variables de entorno necesarias para que el backend funcione correctamente y así evitemos que un tercero ajeno al sistema pueda modificar cierto tipo de configuración si accede al servidor. Entre esas variables se encuentran por ejemplo la clave secreta de Django que se utiliza para encriptar las contraseñas, la URL a la que tienen que ir dirigidas las peticiones de frontend y backend así como la información

sensible de la configuración de la plataforma de pago con Stripe, la configuración de los correos electrónicos que se mandan con la cuenta de la panadería y la información relativa a la base de datos con la que se trabaja en el proyecto. Como aspecto relevante a destacar de este fichero con variables de entorno se recoge un archivo “example.env” un ejemplo para que se puedan personalizar estas variables en caso de querer utilizarlas en otro sistema distinto al sistema en el que se ha desarrollado la aplicación web. Sin embargo, a lo largo del grado se nos ha insistido mucho en evitar subir al repositorio ficheros de configuración que pudieran alterar el comportamiento de las páginas desarrolladas en función del dispositivo en el que se ejecute, ya que no todos los compañeros hemos trabajado con las mismas herramientas. Por tanto, el archivo .env que hace que funcione correctamente el backend no se sube al repositorio y esto evita que la clave secreta de Django quede expuesta al igual que las demás variables que forman parte de este archivo.

Por otro lado, aunque ya se ha citado anteriormente durante esta memoria, se recalca la importancia de la eliminación de las cookies de sesión del cliente que está autenticado en el sistema tras cerrar sesión, de manera que se evite una filtración de la información sensible del usuario y que pueda ser extorsionado de alguna forma por un tercero, al tener acceso si esta cookie estuviera presente y realizar pedidos sin su consentimiento, por poner un ejemplo.

En el frontend ocurre algo similar al backend, pero de una forma más liviana, es decir, también hay un archivo con variables de entorno y que solamente contiene dos URLs, una con la dirección a la API de backend y otra con la dirección del backend “acortada”. Esto se ha diseñado así porque para realizar las peticiones a la API de backend resultaba muy útil tener la primera de ellas pudiendo realizar las peticiones que estén implementadas en esta, y la segunda URL es útil en la medida en la que se puede redirigir al usuario a la sección de autenticación implementada bajo la URL “/localhost:8000/account”.

Paralelamente se ha decidido aumentar la robustez con la que se trabaja con las URLs del frontend mediante la configuración de las rutas por las que se pueden navegar en la aplicación. Esto quiere decir, que en el archivo “index.js” se contempla la posibilidad de que un usuario con rol de cliente, sin permisos de administración, quiera acceder de una forma más forzada a las secciones del administrador. Esta técnica se conoce como pruebas de “hacking” en las que se busca acceder a apartados no permitidos por los permisos e intentar burlar la seguridad de la aplicación utilizando ciertas URLs en el navegador. Si un cliente intentara acceder a estas secciones, se le redirigiría directamente a la página de inicio sin posibilidad de acceder a los lugares en los que estaría intentando acceder.

Implementación

En este apartado se explicarán las principales partes del código desarrollado y se facilitará el acceso al repositorio en el que está contenido el código completo de esas partes. Quisiera destacar que la implementación del proyecto en un principio se iba a contemplar en dos grandes repositorios, pero para simplificar la gestión del backend y frontend, se ha decidido trabajar sobre un único repositorio que contenga dos grandes carpetas, una con el backend y otra con el frontend.

Acceso al repositorio: <https://github.com/josgarber6/panaderia>

5.1. Implementación del backend

Como se lleva describiendo durante esta memoria, el backend se ha desarrollado utilizando el framework Django y como base de datos se ha utilizado PostgreSQL. Para conectar el backend con el frontend se ha utilizado una API que se ha creado haciendo uso del Django REST Framework.

La estructura del backend se encuentra organizada por aplicaciones, de manera que cada funcionalidad destacable dentro de él está dentro de una carpeta que contiene por lo general los siguientes ficheros:

- `__init__.py`: indica a Django que la carpeta se trata de un módulo del proyecto.
- `admin.py`: permite agregar al módulo de administración de Django los modelos de la base de datos y así poder modificar, crear o eliminar objetos desde el panel de administración por defecto de Django.
- `apps.py`: se utiliza para configurar la aplicación.
- `models.py`: en este se registran las clases de Python que formarán la base de datos una vez realicemos las migraciones correspondientes.
- `serializer.py`: se ha explicado en la sección de desarrollo del **Backend**. Es una clase que nos permite transformar datos de modelos Django a un formato como JSON o XML.
- `urls.py`: aquí se gestionan las acciones a realizar de las URLs que se definan en este archivo.
- `viewsets.py`: se ha explicado en la sección de desarrollo del **Backend**. Es una clase que proporciona una interfaz simple para realizar operaciones CRUD sobre un modelo en concreto.

Además todas las aplicaciones tienen una carpeta llamada “migrations” que en algunas aplicaciones aparece con varios ficheros y otras que sólo tienen el `__init__.py`, esto es porque las aplicaciones que contienen modelos que migrar a la base de datos recogen en esta carpeta las migraciones con la información que contiene la base de datos sobre sus modelos.

Por tanto, las aplicaciones de autenticación, base, pedidos y productos contienen los ficheros mencionados previamente. La del doble factor de autenticación, al provenir de un proyecto externo también contiene esos ficheros, pero con algunas excepciones que no se detallarán.

La aplicación de autenticación y el doble factor tienen un directorio llamado “locale” en el que se recogen todas las traducciones de las plantillas HTML del doble factor de autenticación, necesarias para poder mostrar las páginas de autenticación en español. Además, la de autenticación es la única aplicación que tiene un directorio llamado “templates” porque es la única en la que las vistas asociadas a la aplicación se han implementado en el backend.

La aplicación “payment” contiene menos ficheros que el resto ya que sólo se han conservado los necesarios para que funcione la aplicación correctamente, sin necesidad de ocupar más espacio con ficheros vacíos.

Por último, la aplicación “product” contiene un directorio añadido que las otras no tienen, denominado “fixtures” en el que se incluye un fichero en formato JSON con los datos que se utilizarán para el despliegue de la aplicación posteriormente con Docker.

El resto de las carpetas, es decir, “backend”, “htmlcov”, “loadtest”, “media” y “static” no son como tal aplicaciones, pero son fundamentales para el proyecto. En “backend” se encuentran los archivos de configuración del proyecto y un archivo urls.py donde se gestionan las URLs a nivel de proyecto. En “htmlcov” se encuentran los resultados obtenidos al realizar un análisis de cobertura en el proyecto, previo a la implementación de pruebas unitarias en el backend para conocer qué archivos necesitaban más tests. En el directorio “loadtest” se recogen los ficheros relacionados con las **Pruebas de carga**. En “media” se encuentran las imágenes de los productos que forman parte de las subidas de imágenes que se han ido realizando durante el proceso de desarrollo. Ahí es donde van a parar las imágenes que se suben desde el frontend. Por último, en “static” se localizan las imágenes que permiten mostrar el logo de la panadería en las páginas de autenticación del usuario.

5.2. Implementación del frontend

El frontend se ha desarrollado como se viene describiendo en la sección de desarrollo del **Frontend**, utilizando Vue.js 3 con Vite, aunque casi todo el desarrollo ha implicado mayor esfuerzo en la comprensión de Vue.js más que en Vite, que se ha limitado a ficheros de configuración y otros detalles.

En términos generales, el frontend lo forman todos los componentes que se han creado para la página, además del router y las “views” implementadas. En Vue.js se trabaja con componentes que se dividen en tres partes, “template”, “script” y “style”. La parte del “template” contempla todo el código en HTML de la página, en el que se pueden incluir tablas, gráficos y todo tipo de elementos HTML como contenedores, botones, barras de desplazamiento, entre otros. La parte del “script”, comprende el código en JavaScript necesario para traer los datos del backend al frontend y mostrarlo en la página que se ha renderizado, también se pueden importar librerías como Bootstrap que se ha utilizado para dar algo de estilo a ciertos elementos de la página. Por último, la sección de “style” como su propio nombre indica, será donde se localicen los fragmentos de código correspondientes a los estilos de la página renderizada, esto puede incluir código en CSS como en el caso de este trabajo. Además, una de las ventajas fundamentales que tiene Vue.js es la posibilidad de utilizar los estilos que se definen en un componente en otros si se desea manejar la sección de estilos de manera que afecte a todo el proyecto. Esto se puede conseguir si en la etiqueta “<style>” no se

agrega la palabra clave “scoped” que limitaría el uso de los estilos definidos en esta sección sólo y exclusivamente al componente en el que son definidos.

La estructura de carpetas está organizada en tres partes, una carpeta denominada “public” que contiene el icono que aparece en la esquina de la pestaña del navegador, otra llamada “src” donde se encuentra el código fuente de todo el frontend y otra “tests” que a su vez tiene otro directorio “unit” en el que se localizan los tests unitarios realizados sobre los componentes más importantes de la aplicación.

La carpeta “src” está dividida en varias subcarpetas, “assets”, “components”, “router” y “views”. “assets” contiene algunos elementos gráficos empleados en la visualización de la aplicación, como el logo de la panadería, la imagen del fondo de la página, la “hamburguesa” que se muestra cuando la barra de navegación se oculta en las pantallas pequeñas, el icono del carrito de la compra y la lupa, la imagen de un panadero llamado Sánchez generado con inteligencia artificial para dar estilo a la página “Sobre Nosotros” y por último la estrella que aparece en los productos destacados.

“components” es la carpeta más importante de todas, porque en esta reside todo el código fuente de los componentes como su propio nombre indica en inglés. Dentro de la carpeta se encuentran varias subcarpetas organizadas por la función que realizan según las partes de la aplicación, por eso se denominan “Cart”, “Order”, “Payment”, y “Product”. Tanto los componentes que tienen que ver con las funcionalidades de los clientes como la de los administradores se encuentran en la misma carpeta para reducir el número de subcarpetas que contendría el proyecto ya que no es necesario llevar a cabo una estructuración tan estricta en el trabajo por el tamaño que este tiene. Eso sí, para la nomenclatura de los componentes del cliente y del administrador se han seguido dos patrones distintos para poder distinguirlos de una forma sencilla aunque haya pocos componentes, conteniendo “Admin” todos los que tenían que ver de alguna manera con la administración de la página. Fuera de la carpeta “components” se sitúan tres componentes, siendo dos de ellos esenciales en los demás componentes ya que en todas las páginas hay una barra de navegación y un pie de página, por eso se han colocado a ese nivel en la carpeta, siendo estos componentes “Footer.vue”, “Navbar.vue” y “BarChart.vue”, aunque este último sólo se utiliza en las estadísticas.

En el directorio “router” sólo hay un archivo llamado “index.js” que contiene las rutas de la aplicación web. Este funciona de una manera similar al archivo “urls.py” de Django y en él se han implementado las restricciones de acceso para los clientes que quisieran acceder a la parte de la administración.

Por último en la carpeta “views” están las vistas principales de la aplicación, formada por otros componentes, como lo son la página principal, el catálogo de productos, la página “Sobre Nosotros” y las estadísticas, además de las vistas de administración de productos y categorías.

Cabe señalar que en la carpeta “src” se encuentran algunos archivos sueltos que si no estuvieran en esta carpeta, la aplicación web directamente no funcionaría correctamente; ya que el archivo App.vue se podría decir que es el componente principal de la página que mediante el uso de “RouterView” permite navegar allá donde indique la ruta de la URL del navegador y este componente necesita montarse como todos en otro componente, pero en realidad se monta en una aplicación importada directamente desde Vue.js en el archivo “main.js”. Para poder usar las rutas definidas en “router” se declara su uso en este archivo, al igual que el módulo “store.js” que como se ha descrito en profundidad en el apartado de desarrollo de **Frontend** contiene las acciones que se pueden llevar a cabo en la página, como añadir/eliminar productos del carrito. En este fichero también se importan la librería Bootstrap para dar estilo a las páginas.

Gestión del despliegue

Para asegurar un despliegue eficiente y fácil de gestionar de nuestro proyecto, se ha optado por utilizar contenedores Docker [30] ya que la “dockerización” es una de las prácticas más comunes para autocontener una aplicación web. En este caso, se han configurado archivos Dockerfile tanto para el backend como para el frontend y también es necesario disponer de un archivo .env con las variables de entorno del correo electrónico de la empresa, la clave secreta de Django, las claves de Stripe, y la configuración del superusuario de Django.

Para simplificar aún más el proceso de despliegue, se ha desarrollado un archivo docker-compose.yml que contiene las instrucciones necesarias para poner en funcionamiento un contenedor Docker con la base de datos, concretamente PostgreSQL en su versión 16.2, otro contenedor con el backend, que se configura a través del archivo launch.sh; y un último contenedor con el frontend, que apunta al puerto 3000 del terminal donde se ejecuta.

Como el contenedor se ejecuta en el puerto 3000 y el puerto por defecto de ejecución de Vue.js es el 5173, se tuvo que consultar en Internet cómo cambiar el puerto por defecto para que el desarrollo en las últimas etapas fuera más sencillo [32]. También cabe señalar que para la “dockerización” de esta aplicación se ha utilizado como referencia la teoría y práctica impartida en EGC [31] para el despliegue de aplicaciones y la gestión de contenedores Docker.

Pruebas

Cuando se pone en marcha el desarrollo de cualquier sistema de información, automáticamente se debe pensar en la puesta en funcionamiento de un conjunto de pruebas, ya que como decía Edgar W. Dijkstra: *“Program testing can be used to show the presence of bugs, but never to show their absence!”* (*“Las pruebas de programa pueden ser usadas para mostrar la presencia de errores, ¡pero nunca para mostrar su ausencia!”*); por tanto, la implementación de pruebas en este trabajo demuestra que los errores existen, pero como en todo proyecto siempre se pueden realizar más pruebas. Sin embargo, eso no quiere decir, que no se hayan contemplado varios tipos de prueba, con el objetivo de mostrar la experiencia adquirida en el desarrollo de pruebas a lo largo del grado.

Concretamente el plan de pruebas contemplado para este trabajo ha sido el siguiente.

7.1. Pruebas de backend

7.1.1. Pruebas de Modelo

Para las pruebas de modelo, se han considerado la implementación de pruebas tanto con persistencia en la base de datos, para comprobar que los objetos se guardan correctamente en la misma; como sin persistencia, para mejorar la cobertura de algunas líneas de código.

Concretamente, las que se han llevado a cabo, han sido las de los modelos de “Product” y “Category”, que fueron los primeros modelos en implementarse durante el desarrollo y que también eran más interesantes de probar. Estas pruebas están recogidas en el fichero `/backend/products/tests.py` del proyecto.

Como apunte a estas pruebas, conforme se fueron implementando se realizaron varios análisis de cobertura para comprobar la cantidad de código que cubrían estas pruebas. Para ello se empleó la herramienta “coverage” de Python que genera un informe detallado con el porcentaje de cobertura que se obtuvo (véase la figura 7.1). Aunque se debe destacar que el porcentaje global de cobertura no se tuvo en cuenta en las conclusiones extraídas de este informe, ya que el objetivo era comprobar que tanto “Product” como “Category” tenían un alto porcentaje; y además, se implementarían pruebas de varios tipos que limitarían el tiempo disponible para abarcar todo el proyecto en cuanto a cobertura se refiere. El informe completo se puede analizar en visualizando el archivo “index.html” que se localiza en la carpeta “htmlcov” utilizando cualquier visor de archivos HTML (en el caso del trabajo se ha visualizado usando Google Chrome).

Coverage report: 44%

coverage.py v7.4.3, created at 2024-03-11 13:50 +0100

Module	statements	missing	excluded	coverage
authentication__init__.py	0	0	0	100%
authentication\admin.py	1	0	0	100%
authentication\apps.py	4	0	0	100%
authentication\forms.py	21	0	0	100%
authentication\migrations\0001_initial.py	7	0	0	100%
authentication\migrations__init__.py	0	0	0	100%
authentication\models.py	7	1	0	86%
authentication\tests.py	1	0	0	100%
authentication\urls.py	3	0	0	100%
authentication\views.py	45	27	0	40%
backend__init__.py	0	0	0	100%
backend\asgi.py	4	4	0	0%
backend\settings.py	37	0	0	100%
backend\urls.py	8	0	0	100%
backend\wsgi.py	4	4	0	0%
base__init__.py	0	0	0	100%
base\admin.py	1	0	0	100%
base\apps.py	4	0	0	100%
base\migrations__init__.py	0	0	0	100%
base\models.py	1	0	0	100%
base\tests.py	1	0	0	100%
base\views.py	1	0	0	100%
manage.py	12	2	0	83%
product__init__.py	0	0	0	100%

Figura 7.1: Informe de cobertura de backend

7.1.2. Pruebas de vista con API

Como este trabajo se ha basado fundamentalmente en el consumo de una API en Django por parte de un framework de frontend, las pruebas de la API son esenciales para comprobar el correcto funcionamiento de esta. Cabe destacar que la integración del doble factor de autenticación en el proyecto ha impedido en gran medida que algunas de estas pruebas estáticas devuelvan un estado “200 OK”, debido a la necesidad de adjuntar tokens de autenticación en cada petición y con el tiempo que se le ha dedicado a la realización de pruebas, no ha sido posible contemplar esta casuística más compleja a la hora de implementar estas pruebas.

Como es evidente se ha considerado la gestión de errores en estos casos, para que se devuelva un código de estado “403 Forbidden” para impedir que los usuarios que no tengan activado el doble factor y/o que no dispongan de los permisos necesarios, pudieran realizar la acción que solicitaban. No obstante, si se desactivan las comprobaciones de permisos de los viewsets de productos y categorías, sí se devuelve el código HTTP mencionado previamente.

Las pruebas estáticas que presentan estos “problemas” para la devolución de un estado “200 OK” serían estas:

- test_create_category_from_api
- test_create_product_from_api

7.1.3. Pruebas de carga

Otra de las pruebas que se han llevado a cabo en el proyecto han sido las pruebas de carga, para analizar brevemente cuál es el rendimiento de la aplicación web bajo cierto estrés. Como estas pruebas también dependen de la API de Django y de los permisos que esta contiene para proteger ciertos endpoints, han causado ciertos estragos a la hora de probar dichos endpoints (devolviendo el error 403 mencionado en la sección anterior).

Para realizar estas pruebas se ha hecho uso del software Locust con el que se pueden efectuar pruebas de carga en proyectos Django de forma sencilla y rápida. Quiero mencionar que estas pruebas se han ejecutado utilizando usuarios de prueba durante el desarrollo, por lo que si se desean ejecutar en un dispositivo distinto al original donde fue desarrollada la aplicación, deben cambiar algunos parámetros en el locustfile.py que se encuentra en el directorio /backend/loadtest

Se recomienda encarecidamente al revisor del código de este proyecto que cree un usuario administrador de la forma: "username": "admin", "email": "admin@admin.com", "password": "admin", y otro usuario sin permisos de administrador de la forma: "username": "drkenobi", "password": "contra123" para poder ejecutar estas pruebas de carga asegurándose el buen funcionamiento de las mismas.

En la siguiente figura se pueden observar los resultados obtenidos en estas pruebas y es destacable que el sistema es capaz de rendir hasta llegar a un límite de 100 usuarios recurrentes, accediendo 10 usuarios por segundo al sistema.

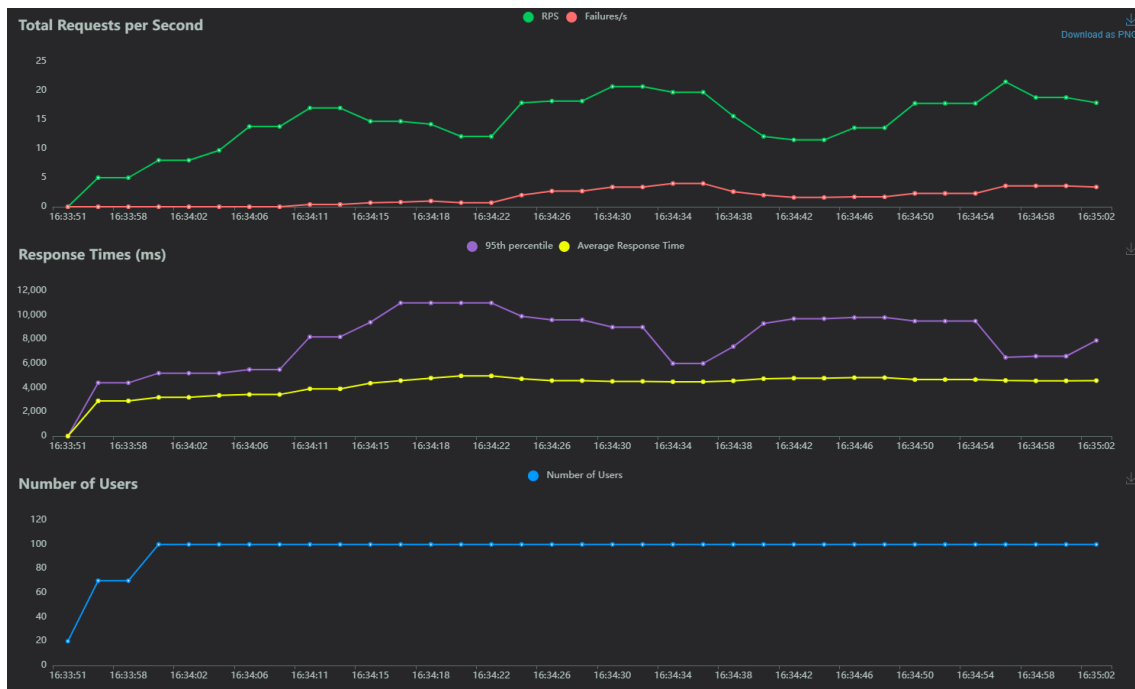


Figura 7.2: Gráficos que muestran el número total de peticiones por segundo al sistema y los tiempos de respuesta de este.

Estas pruebas de carga han sido desarrolladas utilizando varios recursos de Internet como este videotutorial en el que se introducían [36] y la wiki oficial de la asignatura EGC [31].

7.2. Pruebas de frontend

7.2.1. Pruebas unitarias

Para los componentes de Vue.js más significativos como el carrito de la compra, la página de inicio o el listado de productos se han implementado pruebas unitarias de Vue, las cuales en su mayoría corresponden a pruebas de montaje de componentes una vez renderizado el frontend, o a la aparición de determinados elementos en el DOM.

En el carrito por ejemplo se prueba que se pueda añadir un producto, que se pueda eliminar, que se muestre vacío, que se aumente la cantidad de un producto, entre otros aspectos.

7.2.2. Pruebas de vista dinámicas

Una parte imprescindible de la aplicación es la realización de pedidos por parte de los usuarios, lo que conlleva un testing en profundidad, es decir, llevar a cabo pruebas end-to-end. Se han probado las compras con y sin autenticación, para comprobar que la aplicación evita que un usuario pueda realizar un pedido si no está registrado en el sistema; también se ha probado sin el doble factor de autenticación habilitado para comprobar que un usuario que no lo tenga activado no puede realizar un pedido; y, finalmente se ha probado a realizar varios pedidos, con y sin pasarela de pago (ya que la aplicación permite realizar un pedido pagando en efectivo por si el cliente no se siente cómodo trabajando con una tarjeta y/o porque prefiere hacerlo contra reembolso) y comprobando que en ambos casos, se le redirige a la sección “Mis Pedidos” en la que aparece su último pedido, incluyendo los productos y cantidades que deseaba. Para llevarlas a cabo, se ha utilizado Selenium, y como he descrito anteriormente, en algunas se necesita el doble factor de autenticación habilitado por parte del usuario que inicia sesión en la aplicación, lo que implica que para la correcta ejecución de estos tests.

Es recomendable dar de alta a un usuario y luego activarle el doble factor de autenticación, de manera que en el archivo `test_selenium.py` que se encuentra en el directorio `/backend/order`, se deben modificar los datos de inicio de sesión para que usuario y contraseña coincidan con los del usuario que se ha dado de alta al comienzo de la prueba. Esto se tiene que modificar en las funciones “login” (previamente dar de alta a un usuario y no activarle el doble factor) y “login_and_2fa”. Una vez se ha iniciado sesión en el navegador controlado por Selenium, la ejecución de la prueba se para hasta que el usuario que está probando la aplicación introduce el código del doble factor en el recuadro “Token”. Acto seguido se introducen los productos que se desean pedir supuestamente y se realiza el pedido.

En el caso de que se escoja el pago con tarjeta de crédito, debe de tenerse en cuenta que para que un pedido aparezca como pagado en la sección “Mis Pedidos” es necesario ejecutar estos comandos en la consola: “stripe login” (acceso a la plataforma de pago que procesa las transacciones); “stripe listen --forward-to localhost:8000/payment/webhook” (se tendría que configurar una cuenta con Stripe para ejecutarlo todo correctamente).

Manual de usuario

En este apartado se mostrarán todas las funcionalidades que contiene la página web desarrollada. Esta parte de la memoria está destinada a enseñar a utilizar el sistema a todos los nuevos usuarios de esta aplicación web o a modo de recordatorio para aquellos que olviden cómo funciona.

Este capítulo está dividido en dos partes, una para los nuevos clientes que deseen empezar a utilizar la página y otra para los nuevos administradores para que aprovechen al máximo todas las funcionalidades de gestión que provee la página.

8.1. Página principal

Es la página que todos los usuarios de la aplicación podrían ver independientemente del registro o no en la aplicación. En ella aparecen varios botones con los que pueden acceder a varias partes del sistema, entre las que aparecen en la figura 8.1 de izquierda a derecha serían el botón que lleva a la página de inicio estando en cualquier parte de la aplicación (correspondería al logo, que se puede pulsar), el botón de Productos que lleva directamente al catálogo de la tienda, Sobre Nosotros que dirige al usuario a una página con información sobre la panadería, luego se divisan dos botones, uno de inicio de sesión que más adelante se explicará el formulario al que dirige y otro de registro en el que el usuario que accede a la página se puede dar de alta para efectuar pedidos. Por último, se tiene el carrito de la compra con un número que aumentará o disminuirá en función del número de productos que se añadan al carrito.



Figura 8.1: Vista de la página principal desde un ordenador

Más abajo se encuentra la sección de productos destacados, un botón para acceder al catálogo de productos (Ver más) y un pie de página en el que se recoge la información de contacto de la panadería (véase la figura 8.2 y 8.3).

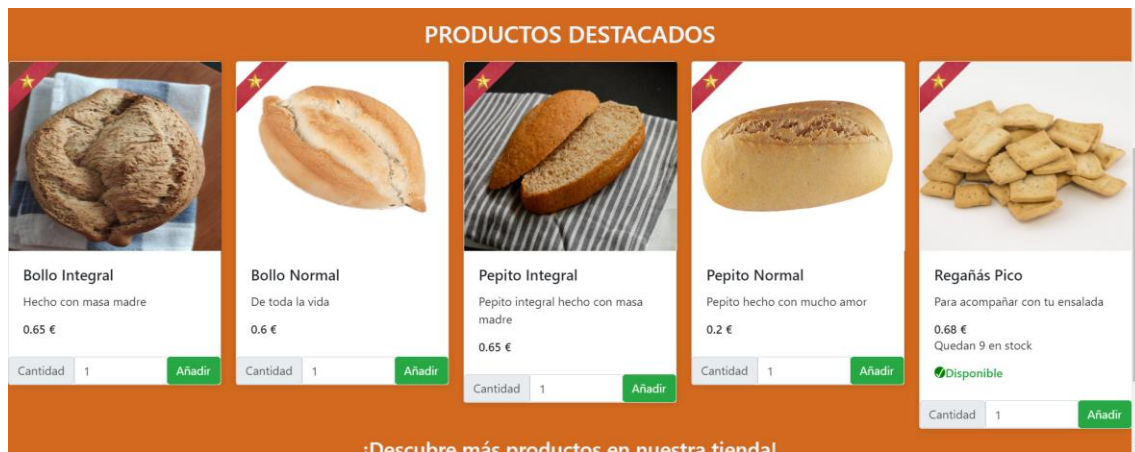


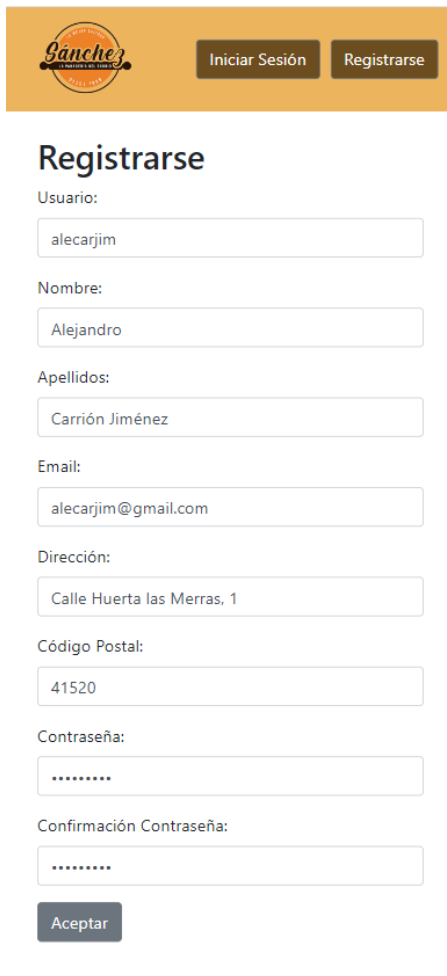
Figura 8.2: Sección de productos destacados de la página principal



Figura 8.3: Sección de productos destacados y pie de página

8.2. Registro y acceso

Si pulsamos en el botón de Registrarse la página nos llevará a un formulario como el que aparece en la figura 8.4. En él tendremos que rellenar los campos de nombre de usuario, nombre y apellidos, dirección de correo electrónico (donde nos llegarán los correos de confirmación de los pedidos que hagamos), dirección del domicilio donde queramos que nos hagan entrega de los pedidos (es muy importante que esta dirección se introduzca correctamente, ya que no será posible cambiarla a posteriori una vez nos registremos), código postal y una contraseña.



Registrarse

Usuario:
alecarjim

Nombre:
Alejandro

Apellidos:
Carrión Jiménez

Email:
alecarjim@gmail.com

Dirección:
Calle Huerta las Merras, 1

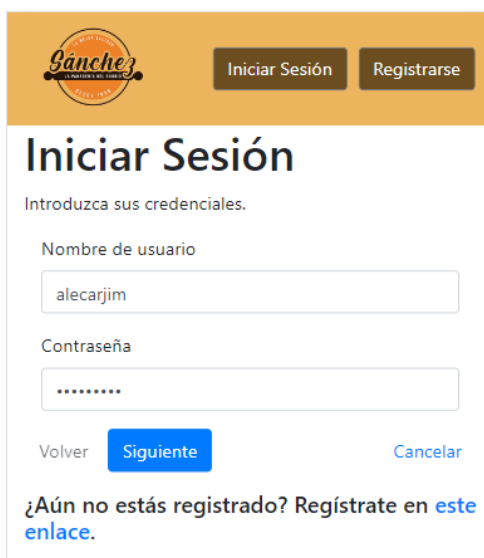
Código Postal:
41520

Contraseña:

Confirmación Contraseña:

Aceptar

Figura 8.4: Formulario de registro visto desde el móvil



Iniciar Sesión

Introduzca sus credenciales.

Nombre de usuario
alecarjim

Contraseña

Volver **Siguiete** Cancelar

¿Aún no estás registrado? Regístrate en [este enlace](#).

Figura 8.6: Formulario de inicio de sesión (móvil)

Una vez que se han rellenado estos datos, tendremos que activar el doble factor de autenticación. Primero la página nos mostraría un mensaje de Registro Completo (véase la figura 8.5) y nos obligaría a iniciar sesión (véase la figura 8.6) para activar el doble factor de autenticación (véase la figura 8.7).



Registro Completo

Enhorabuena, te has registrado con éxito.

Para continuar, por favor activa el doble factor de autenticación

Inicia sesión para activarlo

Figura 8.5: Registro Completo visto desde el móvil



Activar Autenticación de Dos Factores

Está a punto de llevar la seguridad de su cuenta al siguiente nivel. Siga los pasos de este asistente para habilitar la autenticación de dos factores.

Volver **Siguiete** Cancelar

Figura 8.7: Menú de activación del doble factor (móvil)

Ahora haríamos click en el botón Siguiente y nos daría a escoger entre las opciones de autenticación disponibles (véase la figura 8.8). En nuestro caso se ha escogido el generador de códigos porque es la opción más simple de integrar con cualquier dispositivo móvil, ya que existen aplicaciones como Google Authenticator o extensiones en Google Chrome como Authenticator (véase la figura 8.9) que permiten introducir el QR que nos aparece en la página de generación de códigos y posteriormente copiar y pegar los códigos de una forma más sencilla en la aplicación de la panadería.



Figura 8.8: Selección del método de autenticación (móvil)



Figura 8.9: Código QR del generador de códigos (visto desde un ordenador)

Así que cuando se escanee el QR en cualquiera de las aplicaciones mencionadas, se deberá insertar en el recuadro que aparece más abajo con la palabra “Token” encima y darle a siguiente para activarlo con éxito. Tras hacer esto, se le muestra al cliente una pantalla en la que tiene la opción de dirigirse a la tienda o de acceder a las múltiples opciones de configuración que ofrece el doble factor (véase la figura 8.10).



Figura 8.10: Éxito en la activación del doble factor de autenticación

Si en lugar de registrarnos queremos iniciar sesión el proceso es ligeramente similar. Primero debemos pulsar en el botón de Iniciar Sesión de la barra de navegación y rellenar el formulario que nos aparece (véase la figura 8.11).

Figura 8.11: Formulario de inicio de sesión

Luego tendríamos que poner el código que nos aparece en Google Authenticator o Authenticator de Chrome en el campo “Token” (véase la figura 8.12) y darle a Siguiente para que nos lleve de nuevo a la página de inicio con nuestra sesión iniciada correctamente.

Figura 8.12: Código del doble factor en el campo “Token”

Si quisiéramos cerrar sesión, bastaría con hacer click sobre el botón Cerrar Sesión de la barra de navegación para poder hacerlo adecuadamente.

También tenemos la opción de cambiar la contraseña si la que tenemos no nos parece segura o simplemente porque queramos cambiarla. En el menú del usuario (véase la figura 8.13) haremos click en la opción Cambiar contraseña que nos llevará a un formulario (véase la figura 8.14) en el que tendremos que escribir la antigua contraseña, y la nueva contraseña dos veces (por motivos de seguridad) y tras esto, podremos cambiarla. Si queremos ver las contraseñas, basta con hacer click sobre la opción Mostrar Contraseñas.



Figura 8.13: Menú del usuario en la barra de navegación

A screenshot of a form titled 'Cambiar Contraseña'. It contains three input fields: 'Contraseña Antigua:', 'Nueva Contraseña:', and 'Confirmación Nueva Contraseña:'. Each field has a password mask (dots). Below the fields is a checkbox labeled 'Mostrar Contraseñas'. At the bottom are two buttons: 'Aceptar' (blue) and 'Cancelar' (red).

Figura 8.14: Formulario de cambio de contraseña

Si por alguna razón no se ha activado el doble factor de autenticación durante la fase de registro tal y como se indicaba al comienzo de esta sección, existe la posibilidad de activarlo a posteriori (véanse las figuras 8.15 y 8.16), ya que como se ha mencionado en multitud de ocasiones durante este documento, sin el doble factor activado no se puede realizar un pedido y consultar el estado de estos en la página de Mis pedidos (que aparece más adelante en este manual).



Figura 8.15: Opción de activación del doble factor

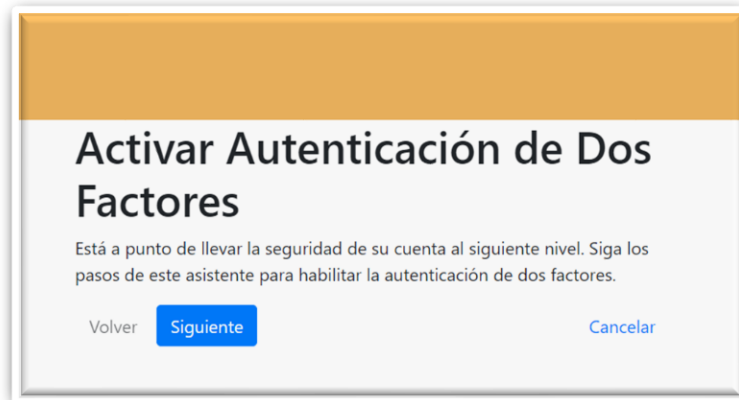


Figura 8.16: Activación de la autenticación en dos pasos

Una vez en esta página habría que seguir los pasos que se indicaron previamente en las figuras 8.7, 8.8, 8.9 y 8.10 para activarlo.

8.3. Manual del cliente

Las próximas subsecciones estarán dedicadas a detallar el uso de la aplicación a los futuros clientes de esta. Comprende el catálogo de productos, el carrito de la compra, las instrucciones para efectuar los pagos con tarjeta y la sección de los pedidos del cliente.

8.3.1. Catálogo de productos

Si pulsamos sobre el botón de Productos en la barra de navegación, la aplicación nos dirigirá al Catálogo de Productos (véase la figura 8.17) en el que se encuentran todos los productos disponibles en la tienda. Este catálogo puede variar en contenido en función de los productos que añada o elimine el administrador de la tienda desde sus funcionalidades.

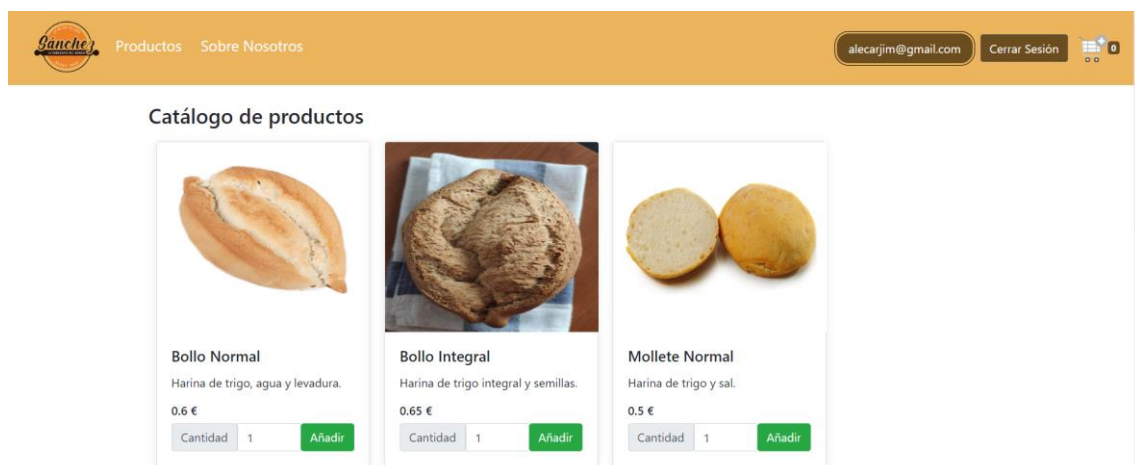


Figura 8.17: Catálogo de productos

Desde el catálogo se pueden añadir productos a la cesta pulsando en el botón añadir en el producto que queramos. Además es posible cambiar la cantidad de producto que deseamos añadir, bien escribiendo la cantidad o pulsando sobre las flechas que aparecen justo al lado de la casilla del número (ver figura 8.18). La página mostrará una alerta (véase la figura 8.19) en caso de que se introduzca una

cantidad negativa y se pulse acto seguido en Añadir, que se puede quitar haciendo click en el botón Aceptar.



Figura 8.18: Flechas para aumentar o disminuir la cantidad de un producto (indicadas con la flecha roja)

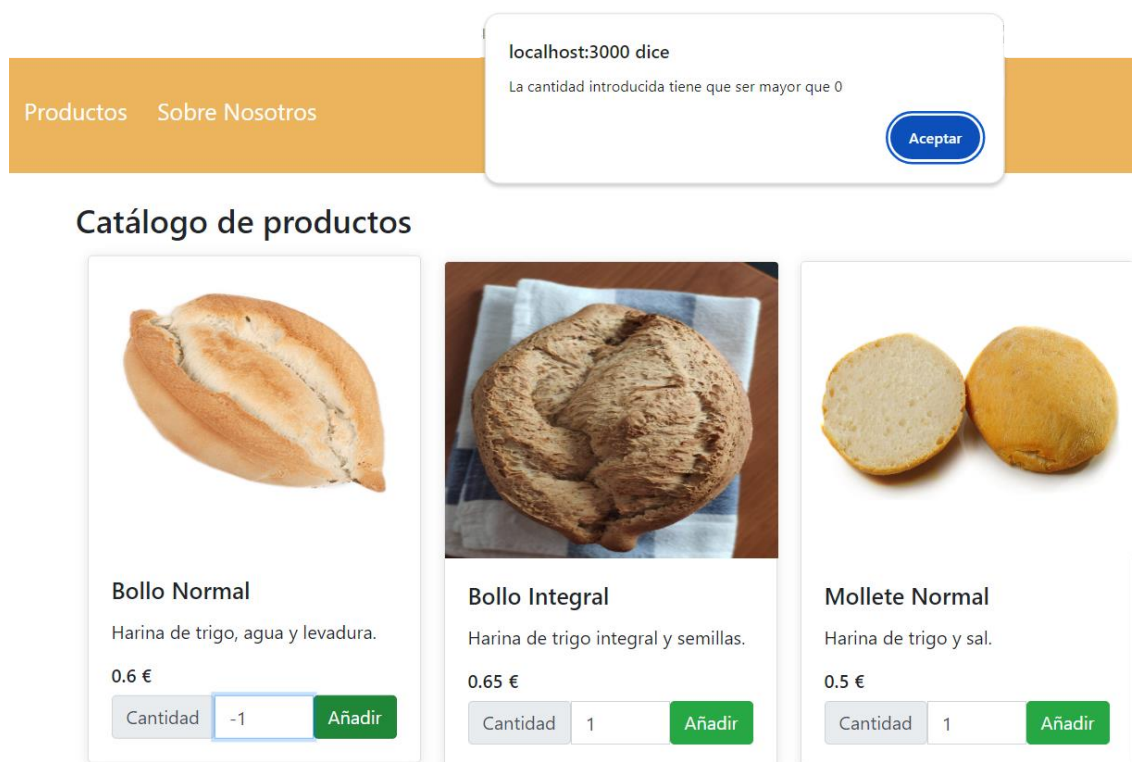


Figura 8.19: Mensaje de alerta mostrado con cantidad negativa

Como se podrá comprobar, si se añade un producto a la cesta con la cantidad 1, el número que aparece justo al lado del icono del carrito de la compra (véase la figura 8.20).

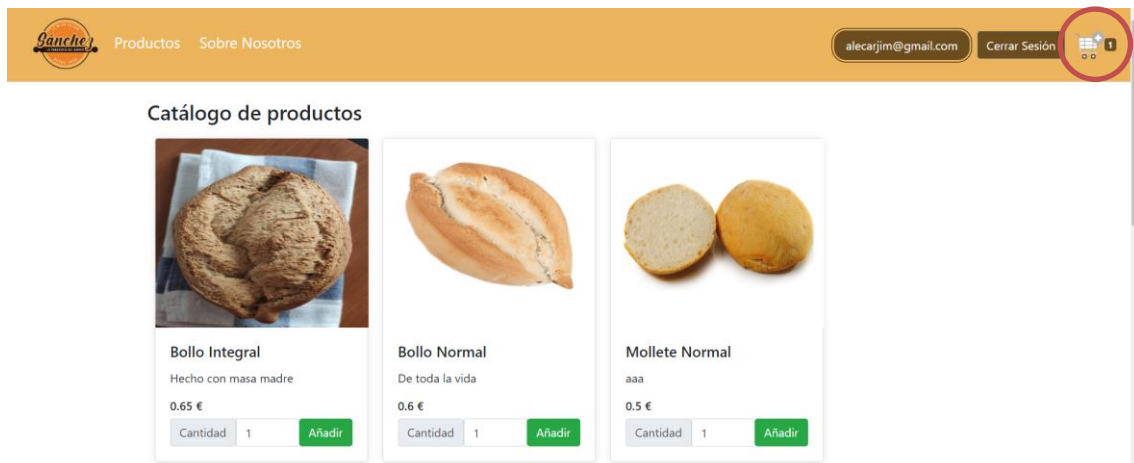


Figura 8.20: Cambio en el número de ítems del carrito (indicado en el círculo rojo)

8.3.2. Carrito de la compra

Si queremos ver los productos que hemos añadido al carrito, cambiar la cantidad que necesitamos de estos o eliminarlos del carrito, pulsaremos sobre el icono del carrito que nos llevará a la página del carrito de la compra (véase la figura 8.21). Para disminuir o aumentar la cantidad de los productos que ya tenemos en el carrito, bastaría con pulsar en el botón “-” ó “+” respectivamente para efectuar el cambio de cantidad (en la figura 8.21 se señalan con unas flechas). Si lo que se desea es eliminarlo de la cesta, se deberá pulsar en el botón Eliminar (señalado con un círculo rojo). También es posible volver a la tienda, de hecho hay dos formas de volver a la tienda, una consiste en pulsar el botón Continuar comprando y la otra es pulsando el botón Productos, que como anteriormente se ha citado sirve para ir a la tienda desde cualquier parte de la aplicación, ya que la barra de navegación siempre aparecerá en la parte superior de la página web.

Finalmente si se quiere realizar el pedido, se debe pulsar en Realizar Pedido y justo después de haberlo pulsado nos aparecerá un recuadro (véase la figura 8.22) en el que debemos seleccionar la forma de pago y envío del pedido que estamos realizando.

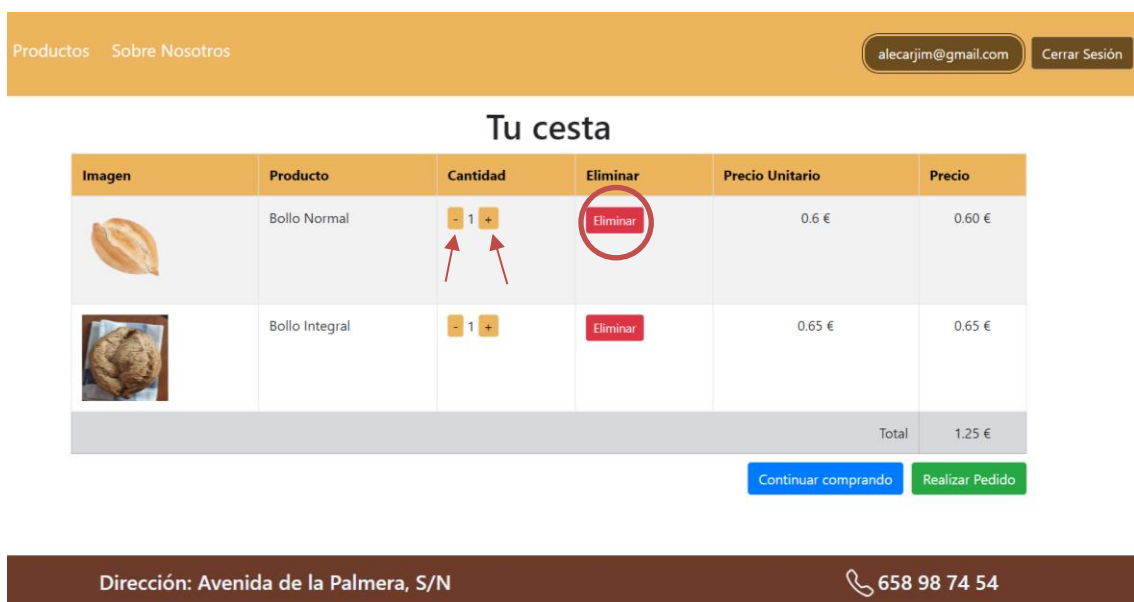


Figura 8.21: Vista del carrito de la compra (fragmento)

The screenshot shows a shopping cart interface titled "Tu cesta". It contains a table with columns: Imagen, Producto, Cantidad, Eliminar, Precio Unitario, and Precio. Two items are listed: "Bollo Muesli" and "Bollo". A modal dialog titled "Selecciona el tipo de pago y entrega" is overlaid on the cart. The modal contains two dropdown menus: "Tipo de Pago:" with "Tarjeta de Crédito" selected, and "Tipo de Entrega:" with "Entrega estándar" selected. Below the dropdowns are two buttons: "Confirmar Pedido" (green) and "Cancelar" (red). At the bottom of the cart, there is a "Total" row showing "1.25 €". Below the cart table are two buttons: "Continuar comprando" (blue) and "Realizar Pedido" (green).

Imagen	Producto	Cantidad	Eliminar	Precio Unitario	Precio
	Bollo Muesli	1		0.6 €	0.60 €
	Bollo	1		0.65 €	0.65 €
Total					1.25 €

Continuar comprando Realizar Pedido

Figura 8.22: Selección de tipo de pago y entrega (fragmento)

Si escogemos Pago en Efectivo y pulsamos en Confirmar Pedido, tras unos segundos nos aparecerá una pantalla (véase la figura 8.23) en la que se nos dice que el pedido ha sido creado correctamente y que podemos consultar la información del pedido en un enlace. Si pulsamos en el enlace nos llevará a la página de Mis pedidos (véase la figura 8.24) en la que podremos realizar un seguimiento de todos los pedidos que hayamos hecho en la página y buscar pedidos por el número de pedido en la barra de búsqueda que aparece señalada en la figura.

The screenshot shows a confirmation page titled "¡Pedido realizado! El ID para el seguimiento de tu pedido es 18". Below the title, there is a link "este enlace" for consulting the order status. Below that, there is a message "Gracias por tu compra." and another message "Recibirás un correo electrónico de confirmación pronto." At the bottom, there is a blue button labeled "Volver a la tienda".

¡Pedido realizado! El ID para el seguimiento de tu pedido es 18

Puedes consultar el estado de tu pedido en [este enlace](#).

Gracias por tu compra.

Recibirás un correo electrónico de confirmación pronto.

Volver a la tienda

Figura 8.23: Pedido realizado con éxito (independientemente del método de pago)



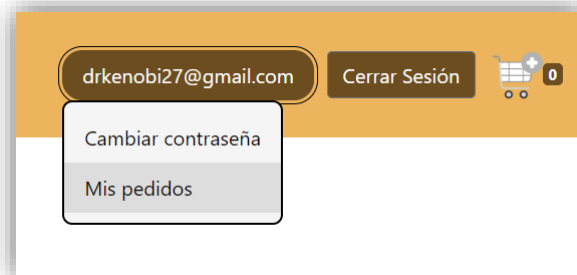
Mis pedidos

Dirección	Código Postal	Fecha de creación	Estado	Pagado	Método de envío	Método de pago	Total
Calle Huerta las Merras, 1	41520	15/05/2024 15:57	Enviado	No	Entrega estándar	Contra-reembolso	1.25 €

Producto	Categoría	Cantidad	Precio	Subtotal
Bollo	Normal	1	0.60	0.60 €
Bollo	Integral	1	0.65	0.65 €

Figura 8.24: Mis pedidos

Si queremos consultar nuestros pedidos en cualquier momento podemos hacerlo accediendo al menú de usuario y haciendo click en la opción Mis pedidos (véase la figura 8.25).



Cerrar Sesión



Cambiar contraseña

Mis pedidos

Figura 8.25: Menú del usuario

← **Panadería Kenobi** TEST MODE

Pagar a Panadería Kenobi

1,75 €

Bollo	0,65 €
Bollo	0,60 €
Mollete	0,50 €

Información de contacto

Correo electrónico [Continuar con link](#)
alecarjim@gmail.com

Método de pago

Tarjeta iDEAL Bancontact Giropay

Información de la tarjeta

4242 4242 4242 4242 VISA

06 / 24 242

Nombre del titular de la tarjeta

Alejandro

País o región

España

Pagar

Figura 8.26: Pago con Stripe

En cualquiera de los dos casos, nos llegaría un correo electrónico de confirmación del pedido como el que se puede ver en la figura 8.27.

Panadería Sánchez - Confirmación de pedido Recibidos x

panaderia.sanchez.viso@gmail.com
para mí ▼

¡Hola, Alejandro!

Tu pedido con número de seguimiento 19 ha sido recibido. ¡Gracias por comprar con nosotros!

Detalles del pedido:

Producto	Categoría	Precio unitario	Cantidad	Total
Bollo	Integral	0,65 €	1	0,65 €
Bollo	Normal	0,60 €	1	0,60 €
Mollete	Normal	0,50 €	1	0,50 €

Importe total: 1,75 €

Dirección de entrega: Calle Huerta las Merras, 1, 41520

Método de pago: Tarjeta

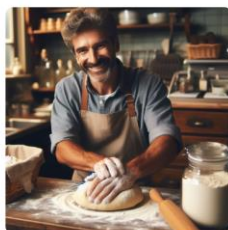
Gracias nuevamente por tu compra.

Figura 8.27: Correo electrónico de confirmación

8.3.3. Sobre Nosotros

Por último, si desea consultar más información acerca de la panadería puede pulsar en el botón Sobre Nosotros y le aparecerá una página como la que se representa en la figura 8.28.

Acerca de Panadería Sánchez



La empresa familiar Panadería Sánchez, fundada en la tradición panadera, que tiene una historia que se remonta a generaciones, ha sido un pilar en la comunidad desde hace mucho tiempo, ofreciendo productos horneados frescos y deliciosos con un toque casero inconfundible.

Panadería Sánchez inicia una nueva etapa de expansión y digitalización con la intención de atraer a más personas y compartir su amor por el arte de la panadería. La decisión tomada por la familia Sánchez muestra su compromiso con su legado y su anhelo de llegar a un público más amplio con sus productos. Cada pieza de pan que sale del obrador se evalúa minuciosamente en términos de calidad y sabor, para que la experiencia gastronómica sea inolvidable.

Figura 8.28: Sobre Nosotros.

8.4. Manual del administrador

A continuación se recogen todas las funcionalidades que un administrador puede llevar a cabo en la página web. Recorreremos la administración de productos, de categorías, de pedidos y las estadísticas disponibles.

8.4.1. Administración de productos

Si accedemos con permisos de administración a la página web (deberíamos tener las credenciales de un administrador, es decir, su usuario y contraseña, además del código del doble factor para ese administrador), podremos realizar todas las funcionalidades que se han explicado en el apartado de **Sistema Desarrollado** para los administradores. En esta sección se explicarán en concreto las distintas acciones que podemos realizar en el catálogo de productos.

Para comenzar, haremos click en el menú Administración (véase la figura 8.29) donde nos aparecerán varias opciones, Productos, Categorías, Pedidos y Estadísticas. En esta ocasión pulsaremos en la opción Productos donde nos aparecerá un listado con todos los productos de una manera similar a como se veían los productos en el catálogo de los clientes pero con unos botones más interesantes (véase la figura 8.30).



Figura 8.29: Menú de Administración

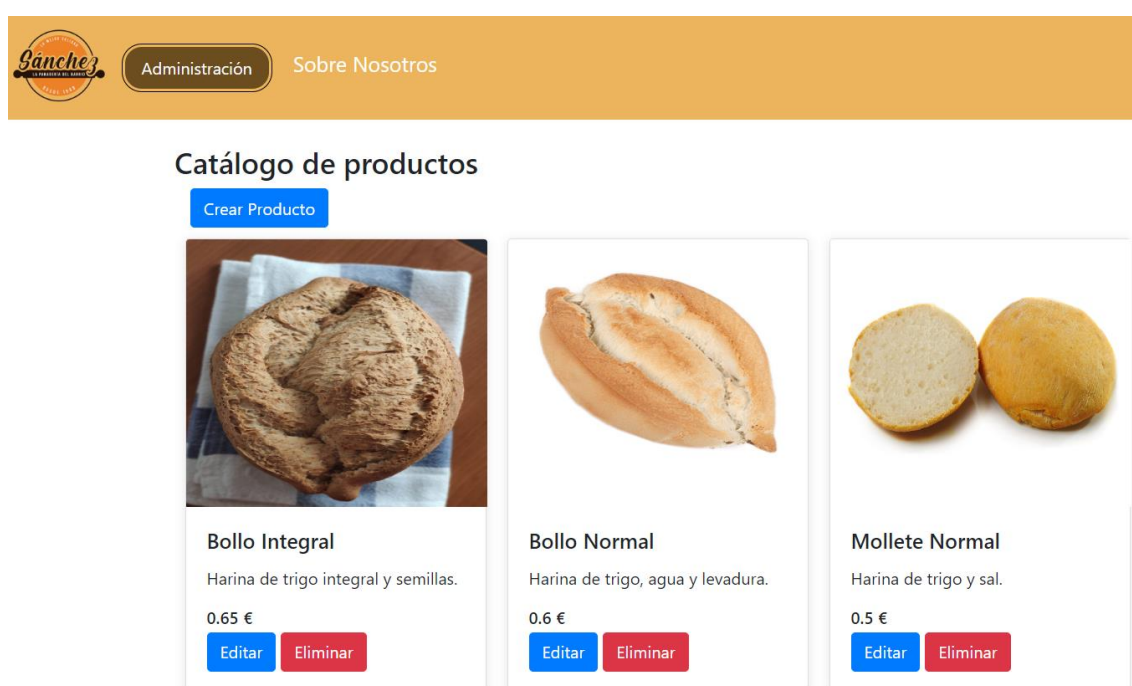
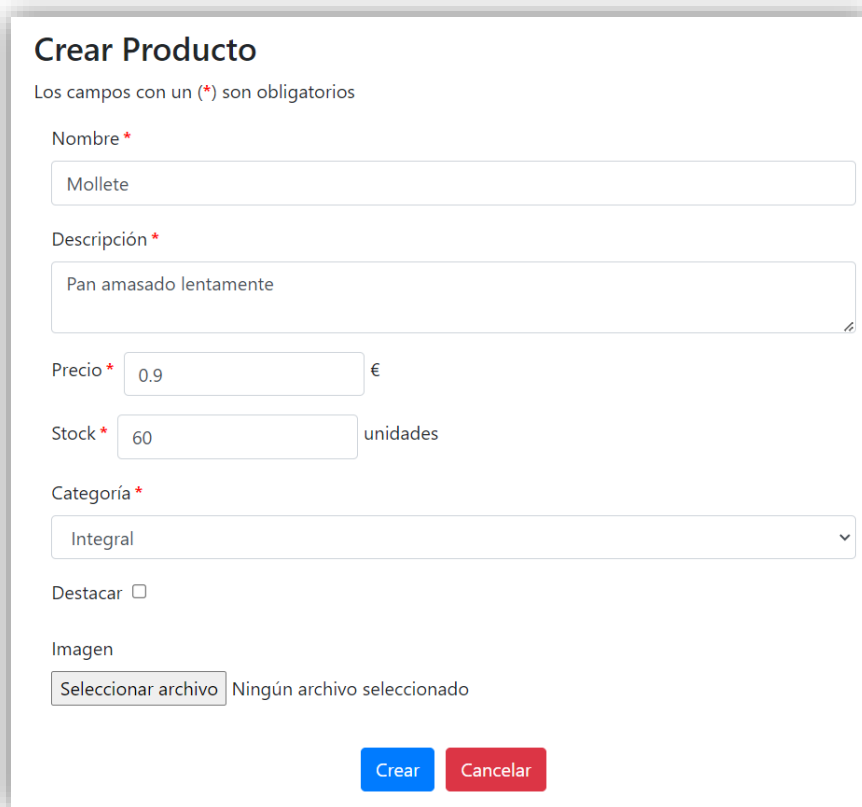


Figura 8.30: Catálogo de productos del administrador

Como se puede observar tenemos tres botones distintos para interactuar con el catálogo, uno para crear nuevos productos y otros dos para editar y eliminar cualquier producto de la página web.

Si queremos crear un producto, se pulsará en el botón Crear Producto y nos aparecerá un formulario (véase la figura 8.31) que tendremos que rellenar para completar la creación del producto. Una vez relleno, le daremos al botón de Crear y si todo ha sido rellenado correctamente nos aparecerá un mensaje de Producto creado correctamente (ver figura 8.33) que se quitará tras cinco segundos. Si hay cualquier error en alguno de los campos en los que puedan introducirse datos no válidos (precio y stock, aunque este último depende de la categoría) se informará mediante un mensaje de error (véase la figura 8.32) de cuál es dicho campo que está incorrectamente rellenado para que se pueda corregir la entrada de datos con facilidad (el mensaje de error aparece tras hacer click en Crear). Si no se quiere añadir ninguna imagen al producto, basta con no seleccionar ningún archivo; y si

queremos destacarlo, se pulsará el recuadro que acompaña a la opción Destacar para marcar un tick azul.



Crear Producto

Los campos con un (*) son obligatorios

Nombre *
Mollete

Descripción *
Pan amasado lentamente

Precio * 0.9 €

Stock * 60 unidades

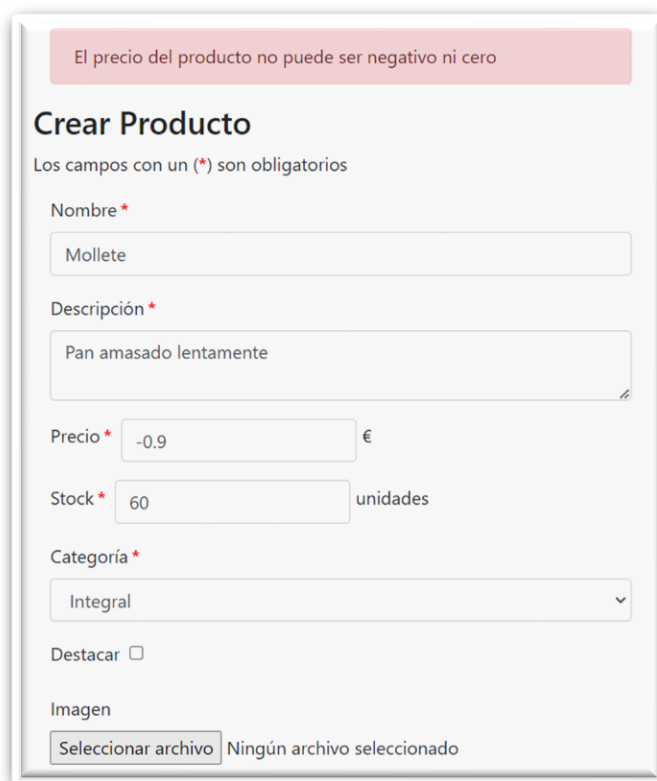
Categoría *
Integral

Destacar ☐

Imagen
Seleccionar archivo Ningún archivo seleccionado

Crear Cancelar

Figura 8.31: Formulario de creación de productos



El precio del producto no puede ser negativo ni cero

Crear Producto

Los campos con un (*) son obligatorios

Nombre *
Mollete

Descripción *
Pan amasado lentamente

Precio * -0.9 €

Stock * 60 unidades

Categoría *
Integral

Destacar ☐

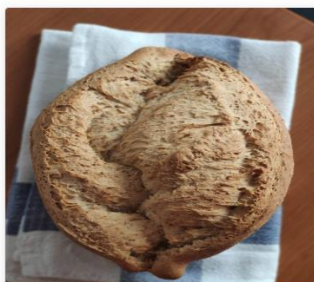
Imagen
Seleccionar archivo Ningún archivo seleccionado

Figura 8.32: Error en la introducción del precio de un producto

Producto creado correctamente

Catálogo de productos

Crear Producto



Bollo Integral

Harina de trigo integral y semillas.

0.65 €

Editar

Eliminar



Bollo Normal

Harina de trigo, agua y levadura.

0.6 €

Editar

Eliminar

Figura 8.33: Mensaje mostrado al crear un producto con éxito (visto desde una pantalla con tamaño inferior a 992 píxeles, por eso aparece la hamburguesa, que está señalada en la figura con un círculo rojo, que contiene el menú recogido)

Si lo que queremos es actualizar un producto, pulsamos en el botón Editar del producto que queramos modificar y nos aparecerá exactamente el mismo formulario que se mostró en la figura 8.31 con los datos actuales que tiene el producto. Una vez que terminemos de modificar los datos que sean necesarios, para confirmar los cambios pulsamos el botón de Actualizar y nos mostrará un mensaje informando de que el producto se ha actualizado correctamente (véase la figura 8.34).

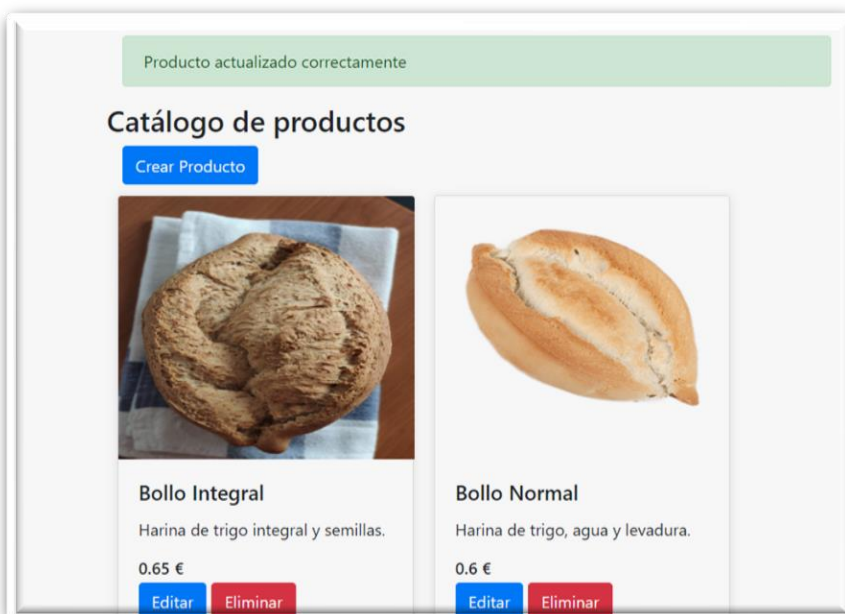


Figura 8.34: Mensaje que informa de la actualización con éxito

Para eliminar un producto, tendremos que dirigirnos al producto que queramos eliminar y hacer click en el botón Eliminar. Después nos aparecerá una alerta en el navegador (véase la figura 8.35) para que confirmemos el borrado del producto. Es suficiente con pulsar en Aceptar para terminar el proceso de eliminación y que nos aparezca un mensaje informándonos de que el producto ha sido correctamente eliminado (véase la figura 8.36).

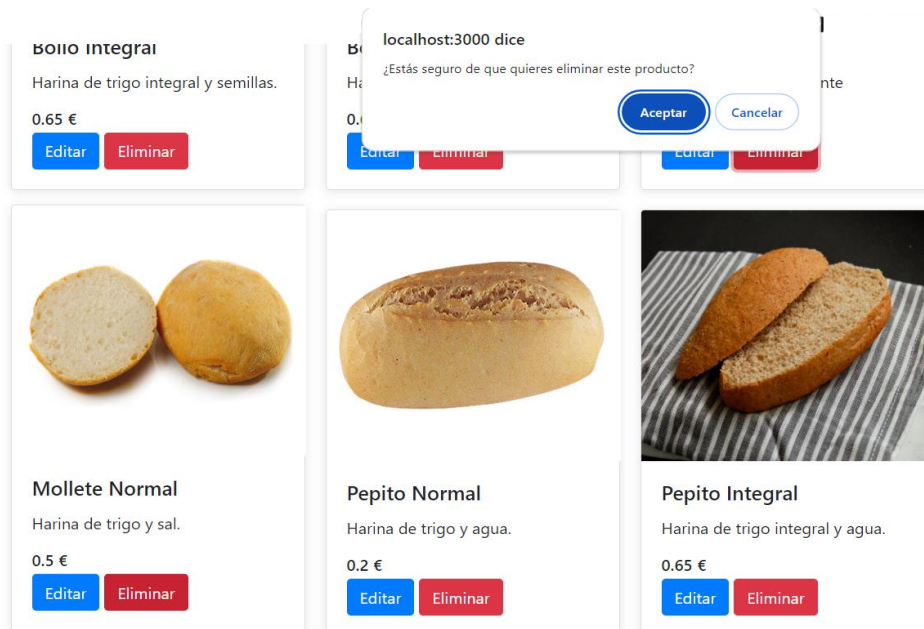


Figura 8.35: Alerta de confirmación de eliminación del producto

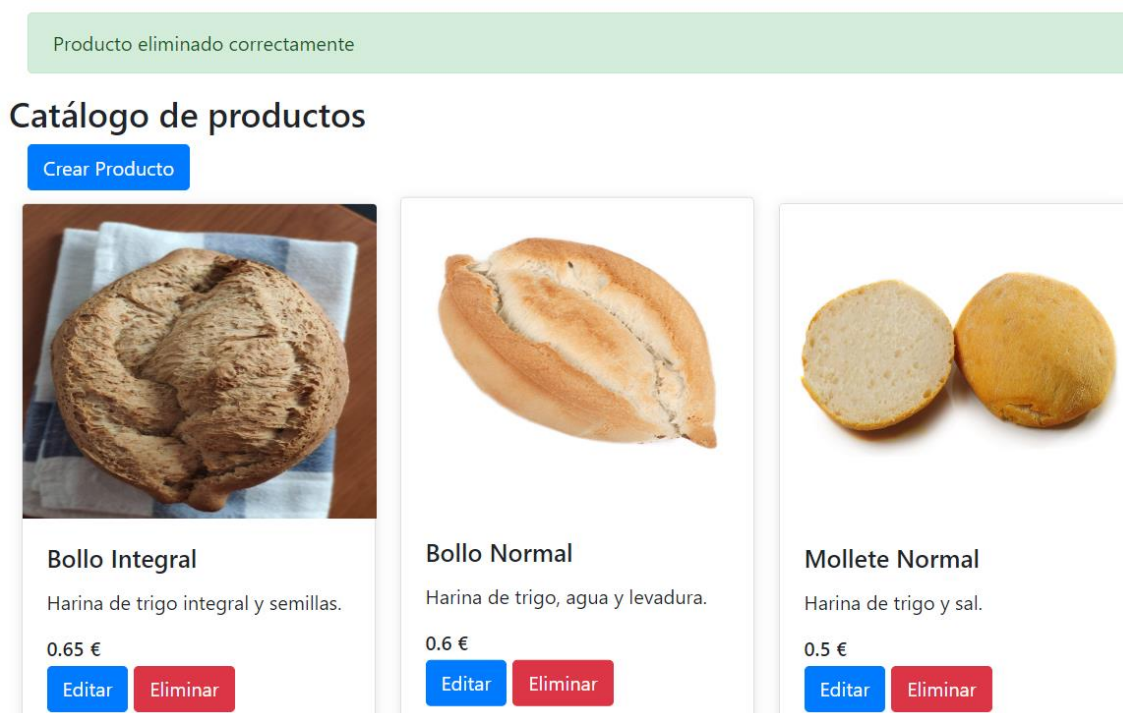


Figura 8.36: Mensaje que informa de la correcta eliminación del producto

8.4.2. Administración de categorías

En el caso de que queramos listar todas las categorías que existen en la página web y crear una categoría, editar una que ya exista o eliminarlas podremos hacerlo a través de la opción Categorías que se encuentra en el menú desplegable de Administración (véase la figura 8.37). Cuando pulsemos la opción la página nos llevará a un listado con todas las categorías existentes y podremos ver que hay varias acciones disponibles a realizar en esta parte de la página (véase la figura 8.38), Crear Categoría, Editar y Eliminar.



Figura 8.37: Menú de Administración (Categorías seleccionado)

Categorías	
Crear Categoría	
Nombre	Acciones
Normal	Editar Eliminar
Pico	Editar Eliminar
Integral	Editar Eliminar

Figura 8.38: Listado de categorías

Si queremos crear una categoría pulsaremos sobre el botón Crear Categoría y nos llevaría a un formulario simple (véase la figura 8.39) en el que sólo con rellenar el nombre que queramos que tenga la categoría y hacer click en el botón de Crear, ya nos añadiría una categoría a la lista de las ya existentes.

Crear Categoría

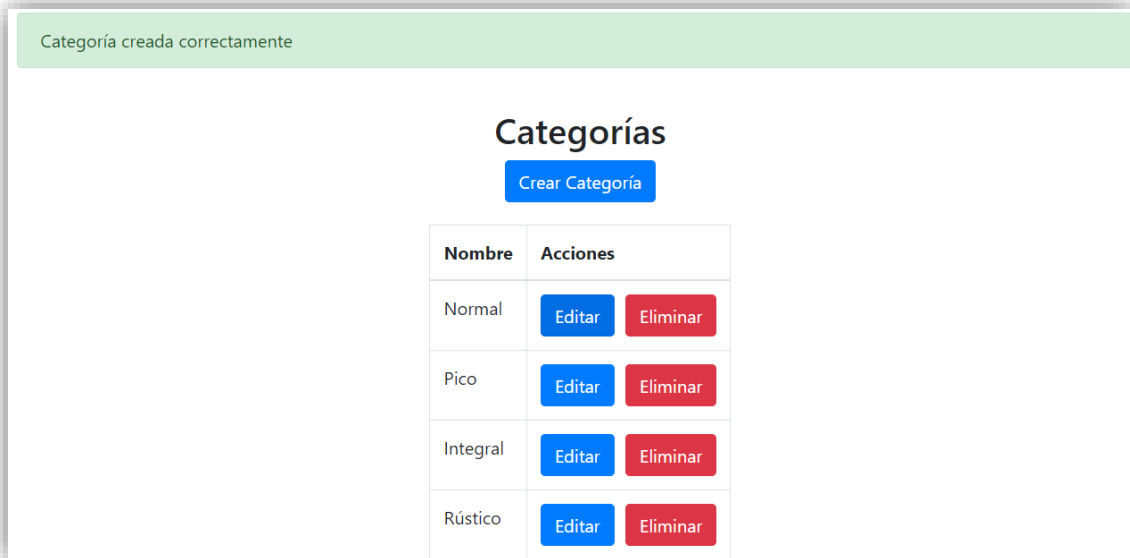
Los campos con un (*) son obligatorios

Nombre*

[Crear](#) [Cancelar](#)

Figura 8.39: Formulario de creación de categorías

Al igual que con los productos, una vez creada correctamente, la página nos mostrará un mensaje informándonos de la creación exitosa de la categoría (véase la figura 8.40).



Categoría creada correctamente

Categorías

Crear Categoría

Nombre	Acciones
Normal	<button>Editar</button> <button>Eliminar</button>
Pico	<button>Editar</button> <button>Eliminar</button>
Integral	<button>Editar</button> <button>Eliminar</button>
Rústico	<button>Editar</button> <button>Eliminar</button>

Figura 8.40: Mensaje de creación exitosa de categoría

Si le damos al botón de Editar a la categoría que se ha creado como muestra para este manual, podremos cambiarle el nombre y en lugar de “Rústico” podemos denominarla “Multicereales” (véase la figura 8.41). De igual manera, al modificarlo se nos mostraría un mensaje indicando que la operación se ha realizado de manera satisfactoria (véase la figura 8.42).



Editar Categoría

Los campos con un (*) son obligatorios

Nombre *

Actualizar Cancelar

Figura 8.41: Formulario de edición de una categoría

Categoría actualizada correctamente

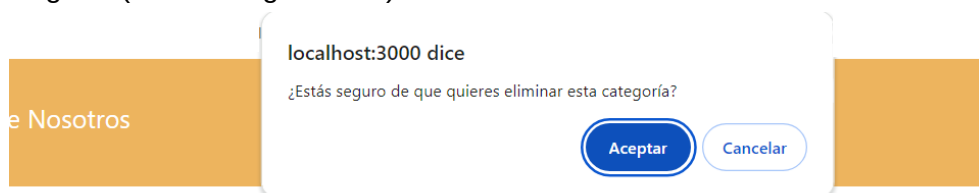
Categorías

Crear Categoría

Nombre	Acciones
Normal	<button>Editar</button> <button>Eliminar</button>
Pico	<button>Editar</button> <button>Eliminar</button>
Integral	<button>Editar</button> <button>Eliminar</button>
Multicereales	<button>Editar</button> <button>Eliminar</button>

Figura 8.42: Mensaje mostrado al actualizar satisfactoriamente una categoría

Si lo que queremos en un momento dado es eliminar una categoría, pulsamos en el botón Eliminar y tendremos que confirmar la eliminación haciendo click en Aceptar en la alerta que se nos muestra para evitar el borrado no intencionado de una categoría, al igual que ocurría con los productos (véase la figura 8.43). Finalmente se nos muestra un mensaje que confirma la correcta eliminación de la categoría (véase la figura 8.44).



Categorías

Crear Categoría

Nombre	Acciones
Normal	<button>Editar</button> <button>Eliminar</button>
Pico	<button>Editar</button> <button>Eliminar</button>
Integral	<button>Editar</button> <button>Eliminar</button>
Multicereales	<button>Editar</button> <button>Eliminar</button>

Figura 8.43: Alerta de confirmación de eliminación de una categoría

Categoría eliminada correctamente

Categorías

Crear Categoría

Nombre	Acciones
Normal	Editar Eliminar
Pico	Editar Eliminar
Integral	Editar Eliminar

Figura 8.44: Mensaje que informa de la correcta eliminación de una categoría

8.4.3. Administración de pedidos

Desde el panel de administración de pedidos (véase la figura 8.46) se pueden monitorizar todos los pedidos que han realizado los clientes de la página. Para acceder a este panel debemos pulsar en Pedidos, en el menú de Administración (véase la figura 8.45).



Figura 8.45: Menú de Administración (Pedidos seleccionado)

Sobre Nosotros

admin@admin.comCerrar Sesión

Buscar por nombre, apellido o ID de pedido

Pedidos

José María García

Pedido 19

EditarEliminar

Dirección	Código Postal	Fecha de creación	Estado	Pagado	Método de envío	Método de pago	Total
Calle Huerta las Merras, 1	41520	15/05/2024 16:16	Enviado	Sí	Entrega estándar	Tarjeta	1.75 €

Producto	Categoría	Cantidad	Precio	Subtotal
Bollo	Integral	1	0.65	0.65 €
Bollo	Normal	1	0.60	0.60 €
Mollete	Normal	1	0.50	0.50 €

Alejandro Carrión Jiménez

Pedido 18

EditarEliminar

Dirección	Código Postal	Fecha de creación	Estado	Pagado	Método de envío	Método de pago	Total
-----------	---------------	-------------------	--------	--------	-----------------	----------------	-------

Figura 8.46: Panel de administración de pedidos

Como se puede observar en las figuras 8.47 y 8.48 respectivamente se puede hacer una búsqueda de pedidos, tanto por nombre y apellidos del usuario como por identificador del pedido (que sería el número que tiene asociado el pedido). También se pueden acceder a los pedidos de una forma más ordenada gracias a la paginación, pudiendo ver hasta cinco pedidos a la vez (véase la figura 8.49).

7

Pedidos

José María García

Pedido 17

EditarEliminar

Dirección	Código Postal	Fecha de creación	Estado	Pagado	Método de envío	Método de pago	Total
Calle Huerta las Merras, 1	41520	11/05/2024 17:26	Enviado	No	Entrega estándar	Contra-reembolso	0.68 €

Producto	Categoría	Cantidad	Precio	Subtotal
Regañás	Pico	1	0.68	0.68 €

José María García

Pedido 7

EditarEliminar

Dirección	Código Postal	Fecha de creación	Estado	Pagado	Método de envío	Método de pago	Total
Calle Huerta las Merras, 1	41520	29/03/2024 21:54	Enviado	No	Entrega estándar	Contra-reembolso	0.65 €

Figura 8.47: Búsqueda por ID del pedido

Pedidos

José María García

Pedido 19

Editar

Eliminar

Dirección	Código Postal	Fecha de creación	Estado	Pagado	Método de envío	Método de pago	Total
Calle Huerta las Merras, 1	41520	15/05/2024 16:16	Enviado	Sí	Entrega estándar	Tarjeta	1.75 €

Producto	Categoría	Cantidad	Precio	Subtotal
Bollo	Integral	1	0.65	0.65 €
Bollo	Normal	1	0.60	0.60 €
Mollete	Normal	1	0.50	0.50 €

José María García

Pedido 17

Editar

Eliminar

Dirección	Código Postal	Fecha de creación	Estado	Pagado	Método de envío	Método de pago	Total
Calle Huerta las Merras, 1	41520	11/05/2024 17:26	Enviado	No	Entrega estándar	Contra-reembolso	0.68 €

Producto	Categoría	Cantidad	Precio	Subtotal
----------	-----------	----------	--------	----------

Figura 8.48: Búsqueda por nombre del cliente

José María García

Pedido 15

Editar

Eliminar

Dirección	Código Postal	Fecha de creación	Estado	Pagado	Método de envío	Método de pago	Total
Calle Huerta las Merras, 1	41520	03/04/2024 23:12	Enviado	No	Entrega estándar	Contra-reembolso	6.50 €

Producto	Categoría	Cantidad	Precio	Subtotal
Bollo	Integral	10	0.65	6.50 €

Anterior

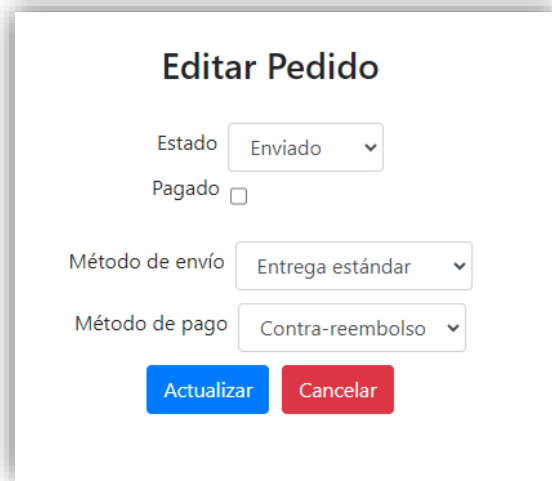
Siguiente

Dirección: Avenida de la Palmera, S/N

658 98 74 54

Figura 8.49: Botones para ir a la página siguiente o anterior en el listado de pedidos

Si queremos editar un pedido, pulsamos en el botón Editar y nos aparecerá un formulario (véase la figura 8.50) en el que podremos modificar el estado del pedido, si está pagado o no, el método de envío y el método de pago.



Editar Pedido

Estado Enviado ▼

Pagado ☐

Método de envío Entrega estándar ▼

Método de pago Contra-reembolso ▼

Actualizar Cancelar

Figura 8.50: Formulario de edición de un pedido

Una vez que lo actualicemos y pulsemos en el botón Actualizar nos aparecerá un mensaje muy similar a los expuestos anteriormente para los productos y categorías, indicándonos que el pedido ha sido actualizado correctamente.

Si lo que queremos es eliminarlo, el proceso es igual que el explicado anteriormente con los productos y las categorías, cuando pulsemos en Eliminar el navegador nos preguntará si queremos eliminarlo (véase la figura 8.51) y si hacemos click en Aceptar se eliminará y se nos mostrará un mensaje indicándonos que el pedido ha sido eliminado correctamente.

José María García

Pedido 17 Editar Eliminar

localhost:3000 dice

¿Estás seguro de que quieres eliminar este pedido?

Aceptar Cancelar

Dirección	Código Postal	Fecha de creación	Estado	Pagado	Método de envío	Método de pago	Total
Calle Huerta las Merras, 1	41520	11/05/2024 17:26	Enviado	No	Entrega estándar	Contra-reembolso	0.68 €

Producto	Categoría	Cantidad	Precio	Subtotal
Regañás	Pico	1	0.68	0.68 €

Figura 8.51: Alerta de confirmación de eliminación de un pedido

8.4.4. Estadísticas

Para finalizar esta guía se introducirá a los administradores al panel de estadísticas en el que pueden realizar un estudio de los productos más vendidos en la panadería a través de la página web y de los clientes que más pedidos efectúan para tener un control más exhaustivo de lo que ocurre en la aplicación (véase la figura 8.52).

Productos Totales: 7
Pedidos Totales: 13



Figura 8.52: Vista de las estadísticas de la página web

Licencia

En el software existen múltiples tipos de licencia en función del servicio para el que se desea utilizar. Las hay más restrictivas, como las licencias de secreto comercial o las licencias privativas, que buscan garantizar más derechos legales a los autores del proyecto; y las hay más permisivas, como el copyleft o las de dominio público, que ofrecen más derechos sobre el software a los usuarios. En nuestro caso, al tratarse de un proyecto para una institución pública, no se va a utilizar una licencia restrictiva, porque se persigue fomentar la colaboración y el intercambio de conocimientos en la comunidad de desarrollo de software. De hecho, el objetivo final de este proyecto es, por un lado, a modo didáctico como ejemplo de proyecto de integración de Django con la tecnología del doble factor de autenticación y Vue.js; y por otro lado, usarlo como referencia para realizar otras páginas web de comercio electrónico para cualquier negocio que necesite expandirse por Internet y tener un escaparate online en el que puedan captar nuevos clientes.

Teniendo en cuenta estos propósitos, la licencia que mejor se adapta a este proyecto es la LGPL (GNU Lesser General Public License) [37] la razón por la que se va a utilizar esa y no la GPL (GNU General Public License) es la siguiente. La licencia LGPL permite a los usuarios utilizar, modificar y distribuir el software, tanto para proyectos comerciales como no comerciales (como es nuestro caso), mientras garantiza que los cambios en el código fuente estén disponibles para la comunidad de desarrollo, asegurando que el software es libre para todos sus usuarios. Sin embargo, la principal diferencia respecto de la GPL es que la elegida para este proyecto puede enlazarse a un proyecto no-GPL, que puede ser software libre o privativo.

Como se ha expuesto anteriormente, este proyecto puede considerarse como referencia para crear páginas web de una forma bastante sencilla que posteriormente pueden utilizarse con fines lucrativos (por ejemplo, vender productos online). Lo que no estaría permitido es guardar los cambios que se hayan realizado en el software que esté bajo esta licencia, por lo que todo nuevo código que no modifique aquello que formaba parte del código original puede ser considerado por el propietario de la empresa que ha contratado los servicios digitales como privativo. Por tanto, si se desea utilizar este proyecto con fines lucrativos, se puede utilizar siempre y cuando se cumpla con los términos recogidos en la licencia LGPL.

Otro aspecto importante que destacar de este proyecto es la integración que se ha realizado con el sistema de autenticación de múltiples factores (más conocido como MFA [38]) de Django, concretamente el doble factor de autenticación (2FA ó Two Factor Authentication). Para facilitar esta integración, se ha utilizado un proyecto [39] de código abierto bajo la licencia del MIT, y cuya licencia incluye lo siguiente (véase la figura 9.1):

Code Blame 19 lines (16 loc) · 1.03 KB

```
1 Copyright (C) 2014 Bouke Haarsma
2
3 Permission is hereby granted, free of charge, to any person obtaining a copy of
4 this software and associated documentation files (the "Software"), to deal in
5 the Software without restriction, including without limitation the rights to
6 use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
7 of the Software, and to permit persons to whom the Software is furnished to do
8 so, subject to the following conditions:
9
10 The above copyright notice and this permission notice shall be included in all
11 copies or substantial portions of the Software.
12
13 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
14 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
15 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
16 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
17 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
18 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
19 SOFTWARE.
```

Figura 9.1: Derechos de autor sobre el proyecto django-two-factor-auth

En ella se puede apreciar claramente que se tiene permiso para tratar el proyecto sin restricciones, incluidos los derechos a usar, copiar y modificar entre otros aspectos dicho proyecto. Por lo tanto, se ha integrado parte de ese proyecto en este trabajo para implementar con mayor facilidad el doble factor de autenticación y conseguir un mayor nivel de seguridad en la aplicación web.

Por último, se debe aclarar el empleo de ciertos archivos que han sido extraídos de otro proyecto desarrollado a lo largo del primer cuatrimestre del cuarto curso, concretamente el `mods.py` y el `tests.py` de backend, para la realización de pruebas unitarias. Estos ficheros contienen tanto funciones como comentarios relacionados con este proyecto y que se pueden utilizar bajo licencia AGPLv3.0 [40] siendo solamente necesario aportar los cambios realizados a los archivos si estos han sido modificados. En el caso de este trabajo, como se puede comprobar en los repositorios del apartado **Implementación**, no han sido modificados en ningún aspecto.

Tercera Parte

Cierre

Conclusión

Para finalizar esta memoria, debo destacar que el resultado final del proyecto ha sido una aplicación que no sólo corresponde a un trabajo fin de grado, sino que podría ser implementada en cualquier empresa real como en la que se ha basado este trabajo.

El desarrollo de esta aplicación ha sido una experiencia muy gratificante ya que no sólo me ha servido para aprender una nueva tecnología y ampliar mis conocimientos técnicos, sino también ha sido una etapa en la que he podido crecer como persona. Siento que me ha ayudado a fortalecer habilidades como la paciencia, la resiliencia y sobre todo la perseverancia, cualidades clave para poder superar con éxito cualquier tipo de circunstancia o proyecto en el que me embarque en mi vida. Además, me ha servido para medir mi capacidad de autoexigencia, debido a que soy una persona muy exigente conmigo mismo y en numerosas ocasiones me lleva a intentar mejorar la calidad de los productos que desarrollo al extremo; teniendo que valorar si realmente merece o no la pena la inversión de tiempo y esfuerzo para los resultados que se obtendrán, sabiendo que una parte fundamental de este trabajo era que fuera de la mejor calidad posible y trabajar día a día con ese objetivo en mente me ha ayudado a mantenerme motivado durante todo el proyecto.

El trabajo y el esfuerzo que ha implicado este proyecto me ha aportado un gran crecimiento profesional y considero que ha supuesto un enorme reto para mí. Trabajar con Vue.js ha sido una experiencia magnífica, sobre todo porque la curva de aprendizaje de este framework sabiendo previamente manejar me con plantillas de HTML me ha ayudado muchísimo a avanzar más rápido en el proyecto. Los conocimientos que he ido adquiriendo me resultarán muy útiles tanto a nivel profesional como para el desarrollo de nuevos proyectos personales.

En numerosas ocasiones a lo largo del desarrollo del proyecto me he preguntado si realmente sería capaz de llevar a cabo el trabajo. Los problemas y obstáculos con los que me he encontrado me han hecho dudar del avance de este, ya fuera porque algo no estaba saliendo tal y como lo planteé desde un principio o porque estaba estancado en la programación de alguna funcionalidad. Tras horas de inversión estudiando alternativas y posibles soluciones, el trabajo realizado ha dado sus frutos, sobreponiéndome ante todas las eventualidades y finalizar el proyecto.

Planes futuros

En este apartado se recogen todas aquellas nuevas funcionalidades o acciones que, en un futuro, podrían formar parte de la aplicación desarrollada. Estas funcionalidades se podrían considerar como mejoras propuestas para la aplicación:

- **Implementar un sistema de notificaciones:** esto permitiría a los administradores enviar mensajes a los clientes cuando no repartan pan por motivos diversos (por ejemplo, períodos vacacionales de la empresa, fiestas a nivel nacional o algún imprevisto ocurrido en la panadería que obligue a no poder repartir pan ese día). También puede servir para enviar mensajes promocionales que inciten a la realización de pedidos de nuevos productos en la tienda.
- **Envío de correo mensual a los clientes:** se podría implementar una funcionalidad que permitiera a los clientes saber cuántos pedidos han realizado durante un mes y el coste de los pedidos. Esto permitiría que los clientes puedan realizar un análisis de subida de precios o estadísticas similares.
- **Suscripciones para usuarios premium:** para las grandes empresas que necesitan realizar pedidos de grandes cantidades de pan, como los restaurantes o los bares, que tuvieran la posibilidad de pagar una suscripción para acceder a ofertas o descuentos cada cierto tiempo.
- **Dirección de entrega de los pedidos:** se podría facilitar un formulario para cambiar la dirección de entrega de los pedidos, ya que actualmente sólo se permite introducir la dirección de entrega durante el registro.
- **Implementar paginación y un buscador en el catálogo de productos:** así se haría más fácil también la búsqueda de productos si el catálogo es demasiado grande.
- **Buscar empresas interesadas en esta página web.** Una vez finalizado el desarrollo, se mostrará el proyecto a diferentes panaderías con el fin de encontrar una que pueda estar interesada en integrarla en ella.

Bibliografía

A continuación, se expone el listado de fuentes consultadas para el desarrollo de esta aplicación. Cabe mencionar que parte de la documentación consultada se incluye en alguno de estos enlaces, como el de la Wiki de EGC, ya que las consultas corresponden a apuntes de asignaturas cursadas, por tanto, para evitar redundancias, se ha puesto un enlace de acceso genérico. El listado de fuentes se ordena por orden de aparición en el documento.

- [1] PostgreSQL. (2024, 11 mayo). Recuperado de <https://www.postgresql.org/>
- [2] Django. (s. f.-b). Recuperado de <https://www.djangoproject.com/>
- [3] JSON. (s. f.). Recuperado de <https://www.json.org/json-es.html>
- [4] Django Two-Factor Authentication Documentation — Django Two-Factor Authentication 1.12.1 documentation. (s. f.). Recuperado de <https://django-two-factor-auth.readthedocs.io/en/1.12.1/>
- [5] Kinsta. (2023, 15 junio). Qué es Node.js y por qué deberías usarlo. Recuperado de <https://kinsta.com/es/base-de-conocimiento/que-es-node-js/#:~:text=escrito%20en%20JavaScript,-.Node.,del%20servidor%20r%C3%A1pidas%20y%20escalables.>
- [6] Vue.js. (s. f.). Recuperado de <https://vuejs.org/guide/introduction.html>
- [7] Vite. (s. f.). Recuperado de <https://vitejs.dev/guide/why.html>
- [8] Rolfo, M. (2023, 2 junio). React vs Angular vs Vue: ¿Qué framework elegir? Recuperado de <https://codigoencasa.com/react-vs-angular-vs-vue-que-framework-elegir/>
- [9] Gonçalves, M. J. (2021, 24 noviembre). Angular vs React vs Vue. Recuperado de <https://www.hiberus.com/crecemos-contigo/angular-vs-react-vs-vue/>
- [10] Locust.io. (s. f.). Recuperado de <https://locust.io/>
- [11] Selenium. (s. f.). Recuperado de <https://www.selenium.dev/>
- [12] PMBOK® Guide (2021). (2021). Recuperado de <https://www.pmi.org/pmbok-guide-standards/foundational/pmbok>
- [13] AWS Pricing Calculator. (s. f.). Recuperado de <https://calculator.aws/#/createCalculator/ec2-enhancement>
- [14] Christie, T. (s. f.-b). Home - Django REST framework. Recuperado de <https://www.django-rest-framework.org/>
- [15] Arenas, D. B. (2014, 25 junio). Introducción a Django REST Framework - Enrique Oriol. Recuperado de <https://blog.enriqueoriol.com/2014/06/introduccion-django-rest-framework.html>

- [16] Stripe | Una infraestructura financiera para internet. (s. f.). Recuperado de <https://stripe.com/es>
- [17] Programming with Mosh. (2021, 28 junio). *Python Django Tutorial for Beginners* [Archivo de vídeo]. Recuperado de <https://www.youtube.com/watch?v=rHux0gMZ3Eg>
- [18] Django, L. (s. f.-b). Django Login and Logout tutorial. Recuperado de <https://learndjango.com/tutorials/django-login-and-logout-tutorial>
- [19] Installation — Django Debug Toolbar 4.2.0 Documentation. (s. f.). Recuperado de <https://django-debug-toolbar.readthedocs.io/en/latest/installation.html>
- [20] Fazt. (2022, 9 septiembre). *Django, curso de django para principiantes* [Archivo de vídeo]. Recuperado de <https://www.youtube.com/watch?v=T1intZyhXDU>
- [21] Saikumar. (2023, 18 septiembre). How to enable Two-Factor Authentication (2FA) in Django - StudyGyaan. Recuperado de <https://studygyaan.com/django/how-to-enable-2fa-in-django>
- [22] Using Bootstrap 5 with Vue 3. (s. f.). Recuperado de <https://stackoverflow.com/questions/65547199/using-bootstrap-5-with-vue-3>
- [23] 1ManStartup. (2020, 9 agosto). *COMPLETE GUIDE - How to build a landing page with Vue.js* [Archivo de vídeo]. Recuperado de <https://www.youtube.com/watch?v=l1M790tSs1w>
- [24] codigofacilito. (2019, 6 febrero). *Presentación del taller - Taller de Django REST y Vue* [Archivo de vídeo]. Recuperado de <https://www.youtube.com/watch?v=dDjAfK1XzAM>
- [25] Empezando | Axios Docs. (s. f.). Recuperado de <https://axios-http.com/es/docs/intro>
- [26] Django. (s. f.-a). Recuperado de <https://docs.djangoproject.com/en/5.0/howto/csrf/#using-csrf-protection-with-ajax>
- [27] Christie, T. (s. f.). Authentication - Django REST framework. Recuperado de <https://www.django-rest-framework.org/api-guide/authentication/#sessionauthentication>
- [28] CSRF Failed: CSRF token missing or incorrect. (s. f.). Recuperado de <https://stackoverflow.com/questions/26639169/csrf-failed-csrf-token-missing-or-incorrect>
- [29] Posicionar footer SIEMPRE pegado al pie de pagina. (s. f.). Recuperado de <https://es.stackoverflow.com/questions/10757/posicionar-footer-siempre-pegado-al-pie-de-pagina>
- [30] Docker: Accelerated Container Application Development. (2024, 23 enero). Recuperado de <https://www.docker.com/>

- [31] 2023/2024 - Wiki de EGC. (s. f.). Recuperado de <https://1984.lsi.us.es/wiki-egc/index.php/2023/2024>
- [32] How to Change vite application port. (s. f.). Recuperado de <https://stackoverflow.com/questions/71763113/how-to-change-vite-application-port>
- [33] How do I use Vue3 with vue-jest? (s. f.). Recuperado de <https://stackoverflow.com/questions/68274961/how-do-i-use-vue3-with-vue-jest>
- [34] WebDevWithArtisan. (2022, 6 diciembre). *Unit Test in Vue 3 | Automated testing | Vitest* [Archivo de vídeo]. Recuperado de <https://www.youtube.com/watch?v=mnX85rnlqAM>
- [35] Developero. (2023, 18 octubre). ⚡ *Cómo usar VITEST MOCK AXIOS con TYPESCRIPT Fácil y Rápido! | Vitest Mock 2024* [Archivo de vídeo]. Recuperado de <https://www.youtube.com/watch?v=cXcXXZ0y4MA>
- [36] Coderos. (2022, 6 agosto). *¿Cómo hacer pruebas de carga de nuestra app ó página web? | Probando Locust* [Archivo de vídeo]. Recuperado de <https://www.youtube.com/watch?v=Tn7CV5XIUW8>
- [37] GNU Lesser General Public License v3.0 - GNU Project - Free Software Foundation. (s. f.). Recuperado de <https://www.gnu.org/licenses/lgpl-3.0.html>
- [38] ¿Qué es la autenticación multifactor? - Explicación de la autenticación multifactor - AWS. (s. f.). Recuperado de <https://aws.amazon.com/es/what-is/mfa/>
- [39] Jazzband. (s. f.). GitHub - jazzband/django-two-factor-auth: Complete Two-Factor Authentication for Django providing the easiest integration into most Django projects. Recuperado de <https://github.com/jazzband/django-two-factor-auth>
- [40] Egc. (s. f.). decide-part-lorca/LICENSE at main · EGC-23-24/decide-part-lorca. Recuperado de <https://github.com/EGC-23-24/decide-part-lorca/blob/main/LICENSE>
- [41] Meijomil, S. (2024, 9 enero). ▷ Qué es un framework, para qué sirve y ejemplos | InboundCycle. Recuperado de <https://www.inboundcycle.com/diccionario-marketing-online/framework#:~:text=Un%20framework%20es%20un%20conjunto,construir%20y%20personalizar%20sus%20aplicaciones.>

Para finalizar este apartado cabe mencionar la utilización de herramientas de inteligencia artificial como ChatGPT o GitHub Copilot para el apoyo a la realización de las tareas de código y/o documentación del proyecto.

Glosario de Términos

API: herramienta que permite el intercambio de información y funcionalidades entre aplicaciones.

Backend: parte de una aplicación no visible por sus usuarios que contiene el esquema de la base de datos que se utiliza en el servicio ofrecido y funciones que generalmente operan sobre los datos obtenidos de dicha base de datos.

Frontend: parte de una aplicación que muestra los datos que el cliente solicita a través de una interfaz gráfica. Estos datos son mostrados en formato digital usando HTML, CSS y un lenguaje de programación que codifique su lógica.

Responsive: es un término en inglés que en español se traduciría literalmente como responsividad. La responsividad de un sistema hace referencia a la capacidad de su frontend para adaptarse a distintos dispositivos móviles, concretamente a sus pantallas, permitiendo una interacción fluida con los elementos que conforman la página web.

Framework: conjunto de herramientas, guías y estructuras predefinidas que se utilizan para desarrollar y organizar software de manera eficiente. [41]

HTTP: protocolo de Transferencia de Hipertexto.

Issue: objeto que puedes crear en un proyecto de GitHub en el que se puede planificar, discutir y realizar un seguimiento del trabajo.

Commit: instantánea que se deposita en la línea de tiempo de un proyecto de Git. Contiene los cambios del proyecto que se han deseado incluir en dicha instantánea.

Anexos

En este apartado se incluyen los informes de seguimiento y desempeño, resultado de las reuniones con el tutor de este TFG. Cabe destacar que la realización de la documentación se ha llevado a cabo a posteriori del desarrollo de la aplicación, por lo que las revisiones de la memoria se han realizado tras las iteraciones que se tenían planificadas.

13.1. Informe de Seguimiento de la Iteración 1

13.1.1. Objetivo de la Reunión

Realizar una revisión de los avances del proyecto hasta la finalización de la primera iteración. Esto incluye la implementación del módulo de autenticación, la integración del doble factor de verificación y la construcción del modelo de datos.

13.1.2. Evaluación del Código

- Comentarios

¿El código está debidamente comentado? (Sí/No)

En esta primera iteración no se han comentado a fondo las funcionalidades implementadas, aunque esto es debido a que gran parte de la funcionalidad reside en la integración del doble factor de autenticación.

- Pruebas sobre frontend:

¿Se han implementado pruebas unitarias? (Sí/No)

En esta iteración no se han considerado aún pruebas unitarias, teniendo en cuenta que sólo se ha implementado la página principal de la aplicación.

¿Se han implementado pruebas de vista dinámicas? (Sí/No)

En esta iteración no.

- Pruebas sobre backend:

¿Se han implementado pruebas unitarias? (Sí/No)

En esta iteración no se han implementado este tipo de pruebas.

¿Se han implementado pruebas de API? (Sí/No)

En esta iteración no se han implementado este tipo de pruebas.

¿Se han implementado pruebas de carga? (Sí/No)

En esta iteración no se han implementado este tipo de pruebas.

- Estructuración del proyecto:

¿El proyecto está estructurado de manera mantenible? (Sí/No)

Sí, se está organizando el proyecto de manera que pueda ser fácilmente mantenible.

- Organización de los tiempos:
¿Se sigue un orden claro en la implementación de las funcionalidades? (Sí/No)

En esta iteración, en la que se asume que el estudiante está adaptándose a las circunstancias en las que se desenvuelve el proyecto, se trata de corregir en la medida de lo posible el desfase temporal producido por la integración del doble factor.

13.1.3. Acciones Correctivas y Sugerencias

Se propone la resolución de los problemas a los que se pueda enfrentar el alumno de la manera más intuitiva que considere. La tecnología de frontend que se debe adoptar es la última versión disponible de Vue.js.

13.1.4. Plan de Acción

Tareas a realizar: durante la siguiente iteración, las actividades correspondientes al segundo entregable de la **EDT** deben estar correctamente resueltas.

Fecha de entrega: estimada para el 10 de marzo.

13.1.5. Aprobación

Firma del Tutor/Patrocinador: *JA Galindo*

Firma del Redactor del Proyecto: *José María GB*

13.2. Informe de Seguimiento de la Iteración 2

13.2.1. Objetivo de la Reunión

Realizar una revisión de los avances del proyecto hasta la finalización de la segunda iteración. Esto incluye la implementación del catálogo de productos, el carrito de la compra y la visualización de la marca corporativa, la marca de productos agotados y los productos destacados en la página principal.

13.2.2. Evaluación del Código

- Comentarios
¿El código está debidamente comentado? (Sí/No)

En esta iteración sí se han comentado alguna de las funcionalidades desarrolladas, sobre todo de la parte del carrito de la compra.

- Pruebas sobre frontend:
¿Se han implementado pruebas unitarias? (Sí/No)

Sí, el carrito, el catálogo de productos y la página principal tienen pruebas unitarias para asegurar el correcto funcionamiento de los componentes.

¿Se han implementado pruebas de vista dinámicas? (Sí/No)
En esta iteración no.

- Pruebas sobre backend:

¿Se han implementado pruebas unitarias? (Sí/No)

Sí, los productos y las categorías tienen este tipo de pruebas implementadas.

¿Se han implementado pruebas de API? (Sí/No)

Sí, se han realizado estas pruebas tanto a los productos como a las categorías.

¿Se han implementado pruebas de carga? (Sí/No)

En esta iteración no se han implementado este tipo de pruebas.

- Estructuración del proyecto:

¿El proyecto está estructurado de manera mantenible? (Sí/No)

Sí, se sigue organizando el proyecto de manera que pueda ser fácilmente mantenible.

- Organización de los tiempos:

¿Se sigue un orden claro en la implementación de las funcionalidades? (Sí/No)

Sí, primero se están centrando los esfuerzos en aquellas funcionalidades que por lógica deben implementarse con anterioridad a otras como la pasarela de pago, que será implementada durante la siguiente iteración.

13.2.3. Acciones Correctivas y Sugerencias

En esta reunión no se ha propuesto ningún tipo de acción correctiva ni sugerencia, ya que el tutor ha evaluado positivamente la evolución del proyecto.

13.2.4. Plan de Acción

Tareas a realizar: durante la siguiente iteración, las actividades correspondientes al tercer entregable de la **EDT** deben estar correctamente resueltas.

Fecha de entrega: estimada para el 3 de abril.

13.2.5. Aprobación

Firma del Tutor/Patrocinador:

Firma del Redactor del Proyecto:

13.3. Informe de Seguimiento de la Iteración 3

13.3.1. Objetivo de la Reunión

Realizar una revisión de los avances del proyecto hasta la finalización de la tercera iteración. Esto incluye la integración de la pasarela de pago, el seguimiento de los pedidos, la generación de un contenedor Docker para autocontener la aplicación desarrollada y otras actividades que se recogen en el **Diccionario de la EDT**.

13.3.2. Evaluación del Código

- Comentarios

¿El código está debidamente comentado? (Sí/No)

En esta iteración se han comentado todas las funcionalidades que por su complejidad necesitaban comentarse, para un análisis posterior más comprensible si es necesario.

- Pruebas sobre frontend:

¿Se han implementado pruebas unitarias? (Sí/No)

Sólo se han implementado pruebas unitarias de los componentes y funcionalidades mencionados en el anterior informe de seguimiento.

¿Se han implementado pruebas de vista dinámicas? (Sí/No)

Sí, en esta iteración se han contemplado pruebas dinámicas de la realización de pedidos haciendo uso de Stripe.

- Pruebas sobre backend:

¿Se han implementado pruebas unitarias? (Sí/No)

Sólo se han implementado pruebas unitarias en backend durante la iteración dos.

¿Se han implementado pruebas de API? (Sí/No)

Sólo se han implementado pruebas de API en backend durante la iteración dos.

¿Se han implementado pruebas de carga? (Sí/No)

Sí.

- Estructuración del proyecto:

¿El proyecto está estructurado de manera mantenible? (Sí/No)

Sí, se sigue organizando el proyecto de manera que pueda ser fácilmente mantenible.

- Organización de los tiempos:

¿Se sigue un orden claro en la implementación de las funcionalidades? (Sí/No)

Sí, en esta iteración, al igual que en la anterior se están implementando las funcionalidades de manera que sigan una secuencia

lógica para que todo lo que se termine de construir pueda funcionar correctamente.

13.3.3. Acciones Correctivas y Sugerencias

Por parte del tutor se ha aconsejado la corrección de algunos errores que aparecen en la consola del navegador al realizar cierto tipo de acciones, como el inicio de sesión. Por tanto, se ha sugerido que se corrijan estos errores y si es necesario que se depure la página lo máximo posible para evitar que haya algún tipo de disonancia entre lo que la página debe hacer y lo que hace (errores menores en general).

13.3.4. Plan de Acción

Para la próxima “iteración” se comenzaría la fase de memoria del proyecto. La fecha estimada para su entrega sería el 27 de mayo.

13.3.5. Aprobación

Firma del Tutor/Patrocinador: *JA Galindo*

Firma del Redactor del Proyecto: *José María GB*

13.4. Informe de desempeño

13.4.1. Actualización a mitad de la iteración 1 (15/1/2024)

Véase el cuadro 13.1.

ID PAQUETE DE TRABAJO	PAQUETE DE TRABAJO	ID DE ACTIVIDAD	ACTIVIDAD	% COMPLETADO
1.1	Iteración 1	1.1.1	Realizar modelo de datos	100
1.1	Iteración 1	1.1.2	Inicialización	100
1.1	Iteración 1	1.1.3	Gestión de usuarios	100
1.1	Iteración 1	1.1.4	Integración del doble factor	40
2.1	Seguimiento	2.1.1	Generar Informe de Desempeño	100
2.1	Seguimiento	2.1.2	Actualizar desempeño (mitad de la iteración 1)	100
2.1	Seguimiento	2.1.3	Actualizar desempeño (final de la iteración 1)	0
2.1	Seguimiento	2.1.8	Generar Informe de Seguimiento de la iteración 1	0

Cuadro 13.1: Actividades completadas en la mitad de la iteración 1

A continuación en el cuadro 13.2 se recogen los problemas encontrados.

PROBLEMA	PROPUESTA DE SOLUCIÓN	DECISIÓN
Integración del doble factor mediante la documentación oficial.	1. No se integra el doble factor (poco recomendable). 2. Integración de un proyecto de ejemplo encontrado en GitHub de la forma más sencilla posible.	La segunda solución se adapta mejor a las necesidades de seguridad del cliente y a las exigencias técnicas del tutor.

Cuadro 13.2: Problemas encontrados en la mitad de la iteración 1

Seguidamente aparecen los riesgos que han aparecido en esta mitad (véase el cuadro 13.3).

RIESGO	OBSERVACIÓN
Desperdicio de tiempo con alguna dificultad.	Se han analizado las distintas alternativas que se han encontrado en Internet y se va a optar por la integración de dicha alternativa de manera que se consuma el menor tiempo posible.

Cuadro 13.3: Seguimiento de riesgos en la mitad de la iteración 1

El rendimiento en esta parte de la iteración 1 aparece en el cuadro 13.4.

REQUISITOS REALIZADOS	REQUISITOS TOTALES (PRODUCTO)
• PD06 • PD13 • PD14 • PD19	El número total de requisitos de producto asciende a 34, de los cuales hay 4 llevados a cabo correctamente. Para más detalles véase la Recopilación de Requisitos .

Cuadro 13.4: Rendimiento en la mitad de la iteración 1

Por último, en el cuadro 13.5 se recoge el seguimiento de la calidad del proyecto hasta la primera mitad de la iteración 1.

DOCUMENTACIÓN		CÓDIGO	
UMBRAL	ACTUAL	UMBRAL	ACTUAL
10% de inconsistencias o errores	No se han detectado errores	Todas las funcionalidades de gran envergadura tienen comentarios	No hay comentarios

Cuadro 13.5: Seguimiento de la calidad en la mitad de la iteración 1

13.4.2. Actualización al final de la iteración 1 (23/1/2024)

Véase el cuadro 13.6.

ID PAQUETE DE TRABAJO	PAQUETE DE TRABAJO	ID DE ACTIVIDAD	ACTIVIDAD	% COMPLETADO
1.1	Iteración 1	1.1.1	Realizar modelo de datos	100
1.1	Iteración 1	1.1.2	Inicialización	100

1.1	Iteración 1	1.1.3	Gestión de usuarios	100
1.1	Iteración 1	1.1.4	Integración del doble factor	100
2.1	Seguimiento	2.1.1	Generar Informe de Desempeño	100
2.1	Seguimiento	2.1.2	Actualizar desempeño (mitad de la iteración 1)	100
2.1	Seguimiento	2.1.3	Actualizar desempeño (final de la iteración 1)	100
2.1	Seguimiento	2.1.8	Generar Informe de Seguimiento de la iteración 1	100

Cuadro 13.6: Actividades completadas al final de la iteración 1

A continuación en el cuadro 13.7 se recogen los problemas encontrados.

PROBLEMA	PROPUESTA DE SOLUCIÓN	DECISIÓN
En esta fase no se ha encontrado ningún problema.	-	-

Cuadro 13.7: Problemas encontrados al final de la iteración 1

Seguidamente aparecen los riesgos que han aparecido en esta fase (véase el cuadro 13.8).

RIESGO	OBSERVACIÓN
No se han manifestado riesgos en esta fase de la iteración.	-

Cuadro 13.8: Seguimiento de riesgos al final de la iteración 1

El rendimiento al final de la iteración 1 aparece en el cuadro 13.9.

REQUISITOS REALIZADOS	REQUISITOS TOTALES (PRODUCTO)
- PD04 - PD28	El número total de requisitos de producto asciende a 34, de los cuales hay 6 llevados a cabo correctamente. Para más detalles véase la Recopilación de Requisitos .

Cuadro 13.9: Rendimiento al final de la iteración 1

Por último, en el cuadro 13.10 se recoge el seguimiento de la calidad del proyecto hasta el final de la iteración 1.

DOCUMENTACIÓN		CÓDIGO	
UMBRAL	ACTUAL	UMBRAL	ACTUAL
10% de inconsistencias o errores	No se han detectado errores	Todas las funcionalidades de gran envergadura tienen comentarios	No hay comentarios

Cuadro 13.10: Seguimiento de la calidad al final de la iteración 1

13.4.3. Actualización a mitad de la iteración 2 (15/2/2024)

Véase el cuadro 13.11.

ID PAQUETE DE TRABAJO	PAQUETE DE TRABAJO	ID DE ACTIVIDAD	ACTIVIDAD	% COMPLETADO
1.2	Iteración 2	1.2.1	Mostrar marca corporativa	100
1.2	Iteración 2	1.2.2	Catálogo de productos	70
1.2	Iteración 2	1.2.3	Marca de productos agotados	0
1.2	Iteración 2	1.2.4	Mostrar productos destacados en la página principal	0
1.2	Iteración 2	1.2.5	Carrito de la compra	5
2.1	Seguimiento	2.1.4	Actualizar desempeño (mitad de la iteración 2)	100
2.1	Seguimiento	2.1.5	Actualizar desempeño (final de la iteración 2)	0
2.1	Seguimiento	2.1.9	Generar Informe de Seguimiento de la iteración 2	0

Cuadro 13.11: Actividades completadas en la mitad de la iteración 2

A continuación en el cuadro 13.12 se recogen los problemas encontrados.

PROBLEMA	PROPUESTA DE SOLUCIÓN	DECISIÓN
Conversión de plantillas HTML de Django de la página original a la interpretación de plantillas de Vue.js.	1. Buscar documentación en Internet acerca del uso de CSS. 2. Utilizar herramientas de inteligencia artificial que conviertan las plantillas.	Se ha optado por una conjunción de ambas propuestas, ya que los estilos de la página, aunque son importantes, no dejan de ser requisitos de calidad del proyecto, que tienen un menor impacto en las funcionalidades. Aún así, también se necesita una personalización a fondo de las vistas, por lo que ambas soluciones aliviarían la carga de trabajo que supone la personalización.

Cuadro 13.12: Problemas encontrados en la mitad de la iteración 2

Seguidamente aparecen los riesgos que han aparecido en esta mitad (véase el cuadro 13.13).

RIESGO	OBSERVACIÓN
Desperdicio de tiempo con alguna dificultad.	Se van a aplicar distintas herramientas y soluciones haciendo uso de inteligencia artificial y documentación para aliviar la carga de trabajo y llevarlo a cabo de una manera más eficiente.

Cuadro 13.13: Seguimiento de riesgos en la mitad de la iteración 2

El rendimiento en esta parte de la iteración 2 aparece en el cuadro 13.14.

REQUISITOS REALIZADOS	REQUISITOS TOTALES (PRODUCTO)
• PD01 • PD05 • PD18 • PD33	El número total de requisitos de producto asciende a 34, de los cuales hay 10 llevados a cabo correctamente. Para más detalles véase la Recopilación de Requisitos .

Cuadro 13.14: Rendimiento en la mitad de la iteración 2

Por último, en el cuadro 13.15 se recoge el seguimiento de la calidad del proyecto hasta la primera mitad de la iteración 2.

DOCUMENTACIÓN		CÓDIGO	
UMBRAL	ACTUAL	UMBRAL	ACTUAL
10% de inconsistencias o errores	No se han detectado errores	Todas las funcionalidades de gran envergadura tienen comentarios	No hay comentarios

Cuadro 13.15: Seguimiento de la calidad en la mitad de la iteración 2

13.4.4. Actualización al final de la iteración 2 (10/3/2024)

Véase el cuadro 13.16.

ID PAQUETE DE TRABAJO	PAQUETE DE TRABAJO	ID DE ACTIVIDAD	ACTIVIDAD	% COMPLETADO
1.2	Iteración 2	1.2.1	Mostrar marca corporativa	100
1.2	Iteración 2	1.2.2	Catálogo de productos	100
1.2	Iteración 2	1.2.3	Marca de productos agotados	100
1.2	Iteración 2	1.2.4	Mostrar productos destacados en la página principal	100
1.2	Iteración 2	1.2.5	Carrito de la compra	100
2.1	Seguimiento	2.1.4	Actualizar desempeño (mitad de la iteración 2)	100
2.1	Seguimiento	2.1.5	Actualizar desempeño (final de la iteración 2)	100
2.1	Seguimiento	2.1.9	Generar Informe de Seguimiento de la iteración 2	100

Cuadro 13.16: Actividades completadas al final de la iteración 2

A continuación en el cuadro 13.17 se recogen los problemas encontrados.

PROBLEMA	PROPUESTA DE SOLUCIÓN	DECISIÓN
En esta fase no se ha encontrado ningún problema.	-	-

Cuadro 13.17: Problemas encontrados al final de la iteración 2

Seguidamente aparecen los riesgos que han aparecido en esta fase (véase el cuadro 13.18).

RIESGO	OBSERVACIÓN
No se han manifestado riesgos en esta fase de la iteración.	-

Cuadro 13.18: Seguimiento de riesgos al final de la iteración 2

El rendimiento al final de la iteración 2 aparece en el cuadro 13.19.

REQUISITOS REALIZADOS	REQUISITOS TOTALES (PRODUCTO)
• PD02 • PD03 • PD08 • PD09 • PD10 • PD11 • PD12 • PD27	El número total de requisitos de producto asciende a 34, de los cuales hay 18 llevados a cabo correctamente. Para más detalles véase la Recopilación de Requisitos .

Cuadro 13.19: Rendimiento al final de la iteración 2

Por último, en el cuadro 13.20 se recoge el seguimiento de la calidad del proyecto hasta el final de la iteración 2.

DOCUMENTACIÓN		CÓDIGO	
UMBRAL	ACTUAL	UMBRAL	ACTUAL
10% de inconsistencias o errores	No se han detectado errores	Todas las funcionalidades de gran envergadura tienen comentarios	Se han comentado varias funcionalidades del carrito para facilitar su comprensión

Cuadro 13.20: Seguimiento de la calidad al final de la iteración 2

13.4.5. Actualización a mitad de la iteración 3 (25/3/2024)

Véase el cuadro 13.21.

ID PAQUETE DE TRABAJO	PAQUETE DE TRABAJO	ID DE ACTIVIDAD	ACTIVIDAD	% COMPLETADO
1.3	Iteración 3	1.3.1	Integrar la pasarela de pago	100
1.3	Iteración 3	1.3.2	Seguimiento de pedidos por usuarios	100

1.3	Iteración 3	1.3.3	Confirmación de compra por correo	100
1.3	Iteración 3	1.3.4	Seguimiento de pedidos por el administrador	0
1.3	Iteración 3	1.3.5	Creación de productos y categorías por el administrador	0
1.3	Iteración 3	1.3.6	Edición y borrado de productos y categorías por el administrador	0
1.3	Iteración 3	1.3.7	Edición y borrado de pedidos por el administrador	0
1.3	Iteración 3	1.3.8	Estadísticas de productos y usuarios	0
1.3	Iteración 3	1.3.9	Generar un contenedor de aplicaciones con el producto final	0
1.3	Iteración 3	1.3.10	Elaborar instrucciones de instalación y puesta en producción	0
2.1	Seguimiento	2.1.6	Actualizar desempeño (mitad de la iteración 3)	100
2.1	Seguimiento	2.1.7	Actualizar desempeño (final de la iteración 3)	0
2.1	Seguimiento	2.1.10	Generar Informe de Seguimiento de la iteración 3	0

Cuadro 13.21: Actividades completadas en la mitad de la iteración 3

A continuación en el cuadro 13.22 se recogen los problemas encontrados.

PROBLEMA	PROPUESTA DE SOLUCIÓN	DECISIÓN
En esta fase no se ha encontrado ningún problema.	-	-

Cuadro 13.22: Problemas encontrados en la mitad de la iteración 3

Seguidamente aparecen los riesgos que han aparecido en esta mitad (véase el cuadro 13.23).

RIESGO	OBSERVACIÓN
No se han manifestado riesgos en esta fase de la iteración.	-

Cuadro 13.23: Seguimiento de riesgos en la mitad de la iteración 3

El rendimiento en esta parte de la iteración 3 aparece en el cuadro 13.24.

REQUISITOS REALIZADOS	REQUISITOS TOTALES (PRODUCTO)
• PD07 • PD15 • PD16 • PD17 • PD20	El número total de requisitos de producto asciende a 34, de los cuales hay 29 llevados a cabo correctamente. Para más detalles véase la Recopilación de Requisitos .

• PD21 • PD22 • PD29 • PD30 • PD31 • PD34	
--	--

Cuadro 13.24: Rendimiento en la mitad de la iteración 3

Por último, en el cuadro 13.25 se recoge el seguimiento de la calidad del proyecto hasta la primera mitad de la iteración 3.

DOCUMENTACIÓN		CÓDIGO	
UMBRAL	ACTUAL	UMBRAL	ACTUAL
10% de inconsistencias o errores	No se han detectado errores	Todas las funcionalidades de gran envergadura tienen comentarios	Se han comentado varias funcionalidades de los ViewSet de los productos y los pedidos para facilitar su comprensión a posteriori

Cuadro 13.25: Seguimiento de la calidad en la mitad de la iteración 3

13.4.6. Actualización al final de la iteración 3 (3/4/2024)

Véase el cuadro 13.26.

ID PAQUETE DE TRABAJO	PAQUETE DE TRABAJO	ID DE ACTIVIDAD	ACTIVIDAD	% COMPLETADO
1.3	Iteración 3	1.3.1	Integrar la pasarela de pago	100
1.3	Iteración 3	1.3.2	Seguimiento de pedidos por usuarios	100
1.3	Iteración 3	1.3.3	Confirmación de compra por correo	100
1.3	Iteración 3	1.3.4	Seguimiento de pedidos por el administrador	100
1.3	Iteración 3	1.3.5	Creación de productos y categorías por el administrador	100
1.3	Iteración 3	1.3.6	Edición y borrado de productos y categorías por el administrador	100
1.3	Iteración 3	1.3.7	Edición y borrado de pedidos por el administrador	100
1.3	Iteración 3	1.3.8	Estadísticas de productos y usuarios	100
1.3	Iteración 3	1.3.9	Generar un contenedor de aplicaciones con el producto final	100
1.3	Iteración 3	1.3.10	Elaborar instrucciones de instalación y puesta en producción	100

2.1	Seguimiento	2.1.6	Actualizar desempeño (mitad de la iteración 3)	100
2.1	Seguimiento	2.1.7	Actualizar desempeño (final de la iteración 3)	100
2.1	Seguimiento	2.1.10	Generar Informe de Seguimiento de la iteración 3	100

Cuadro 13.26: Actividades completadas al final de la iteración 3

A continuación en el cuadro 13.27 se recogen los problemas encontrados.

PROBLEMA	PROPUESTA DE SOLUCIÓN	DECISIÓN
Pequeños errores en el código.	Revisar el proyecto a fondo, en busca de posibles errores menores que puedan provocar una mala imagen en la aplicación final y darle una solución acorde a su importancia.	Realizar una revisión exhaustiva de los pequeños detalles y darles una solución apropiada.

Cuadro 13.27: Problemas encontrados al final de la iteración 3

Seguidamente aparecen los riesgos que han aparecido en esta última fase (véase el cuadro 13.28).

RIESGO	OBSERVACIÓN
Retraso en el entregable.	Se ha podido retrasar el entregable debido a que podría haber tenido problemas al configurar Docker, pero al final ha sido posible entregar el proyecto a tiempo para la demostración que se quería realizar al tutor.

Cuadro 13.28: Seguimiento de riesgos en la mitad de la iteración 3

El rendimiento en esta parte de la iteración 3 aparece en el cuadro 13.29.

REQUISITOS REALIZADOS	REQUISITOS TOTALES (PRODUCTO)
Se han realizado todos los requisitos restantes.	El número total de requisitos de producto asciende a 34, y en este punto del proyecto se han cumplido todos y cada uno de ellos. Para más detalles véase la Recopilación de Requisitos .

Cuadro 13.29: Rendimiento en la mitad de la iteración 3

Por último, en el cuadro 13.30 se recoge el seguimiento de la calidad del proyecto hasta el final de la iteración 3.

DOCUMENTACIÓN		CÓDIGO	
UMBRAL	ACTUAL	UMBRAL	ACTUAL
10% de inconsistencias o errores	No se han detectado errores	Todas las funcionalidades de gran envergadura tienen comentarios	Se han comentado todas las funcionalidades más importantes del proyecto para que sean más comprensibles en un futuro

Cuadro 13.30: Seguimiento de la calidad en la mitad de la iteración 3

13.5. Seguimiento de la Documentación

Para dar fin al apartado de Anexos se describe el seguimiento que se ha realizado a esta memoria durante su redacción.

13.5.1. Borrador de la memoria (24/4/2024)

Durante la primera parte de la redacción de este documento se solicitó por parte del tutor el desarrollo de los casos de uso, las pruebas llevadas a cabo, la gestión del despliegue y la licencia del proyecto. Una vez fueron consultados los documentos de referencia a tener en cuenta para la elaboración de la memoria a través de la plataforma oficial de la escuela, se procedió a la realización de las primeras etapas de esta, que comprende la **Introducción**.

- Exactitud y Coherencia

En el cuadro 13.31 se recoge la frecuencia de errores y el índice de precisión de la documentación redactada.

FRECUENCIA DE ERRORES/DISCREPANCIAS	ÍNDICE DE PRECISIÓN (%)
No se han encontrado errores en los documentos revisados	100%

Cuadro 13.31: Frecuencia de errores e índice de precisión de la documentación

- Cumplimiento de Mínimos Establecidos

En el cuadro 13.32 se recogen los apartados mínimos exigidos por el tutor a tener desarrollados y el número de páginas que contienen a fecha del seguimiento.

APARTADOS MÍNIMOS EXIGIDOS	NÚMERO DE PÁGINAS
Análisis/recopilación de requisitos	20
Pruebas desarrolladas	4
Licencia	2
Despliegue	1
Introducción	4

Cuadro 13.32: Apartados mínimos exigidos y número de páginas que contienen

13.5.2. Avances hasta la primera mitad de la memoria (8/5/2024)

Tras finalizar los apartados mencionados anteriormente, se solicitó el desarrollo de las materias relacionadas y la planificación completa del proyecto.

- Exactitud y Coherencia

En el cuadro 13.33 se recoge la frecuencia de errores y el índice de precisión de la documentación redactada.

FRECUENCIA DE ERRORES/DISCREPANCIAS	ÍNDICE DE PRECISIÓN (%)
No se han encontrado errores en los documentos revisados	100%

Cuadro 13.33: Frecuencia de errores e índice de precisión de la documentación

- Cumplimiento de Mínimos Establecidos

En el cuadro 13.34 se recogen los apartados mínimos exigidos por el tutor a tener desarrollados y el número de páginas que contienen a fecha del seguimiento.

APARTADOS MÍNIMOS EXIGIDOS	NÚMERO DE PÁGINAS
Plan de Gestión de las Comunicaciones	1
Definición del alcance	8
Plan de Gestión de los Recursos	1
Plan de Gestión del Cronograma	5
Estimación de costes	3
Plan de Gestión de Riesgos	1
Plan de Gestión de la Calidad	2

Cuadro 13.34: Apartados mínimos exigidos y número de páginas que contienen

13.5.3. Avances hasta la segunda mitad de la memoria (13/5/2024)

Tras finalizar los apartados mencionados anteriormente, se solicitó la redacción del sistema desarrollado y su implementación, además del manual de usuario completo.

- Exactitud y Coherencia

En el cuadro 13.35 se recoge la frecuencia de errores y el índice de precisión de la documentación redactada.

FRECUENCIA DE ERRORES/DISCREPANCIAS	ÍNDICE DE PRECISIÓN (%)
Se han encontrado errores menores y/o discrepancias en estas secciones: - Solapamiento entre introducción y motivación. - Justificación adecuada de la utilización del proyecto externo del doble factor y sus ventajas frente a una implementación propia. - Colocar mejor la secuenciación de las actividades (la figura no se puede leer correctamente). - Los márgenes y el sangrado deben ajustarse al contenido y no sobrepasar el espacio adecuado para ello.	90% (se puede realizar una mejora significativa en las secciones afectadas para corregir los errores)

Cuadro 13.35: Frecuencia de errores e índice de precisión de la documentación

- Cumplimiento de Mínimos Establecidos

En el cuadro 13.36 se recogen los apartados mínimos exigidos por el tutor a tener desarrollados y el número de páginas que contienen a fecha del seguimiento.

APARTADOS MÍNIMOS EXIGIDOS	NÚMERO DE PÁGINAS
Sistema Desarrollado	17
Implementación	3
Manual de usuario completo	24

Cuadro 13.36: Apartados mínimos exigidos y número de páginas que contienen

13.5.4. Finalización de la memoria (24/5/2024)

Tras finalizar los apartados mencionados anteriormente, se solicitó la finalización de la redacción de la memoria, plasmando los agradecimientos, la conclusión, los planes futuros y los anexos.

- Exactitud y Coherencia

En el cuadro 13.37 se recoge la frecuencia de errores y el índice de precisión de la documentación redactada.

FRECUENCIA DE ERRORES/DISCREPANCIAS	ÍNDICE DE PRECISIÓN (%)
No se han encontrado errores en los documentos revisados	100%

Cuadro 13.37: Frecuencia de errores e índice de precisión de la documentación

- Cumplimiento de Mínimos Establecidos

En el cuadro 13.38 se recogen los apartados mínimos exigidos por el tutor a tener desarrollados y el número de páginas que contienen a fecha del seguimiento.

APARTADOS MÍNIMOS EXIGIDOS	NÚMERO DE PÁGINAS
Agradecimientos	1
Conclusión	1
Planes futuros	1
Anexos	15

Cuadro 13.38: Apartados mínimos exigidos y número de páginas que contienen

13.6. Documentación adicional

En esta sección se encuentran los enlaces al informe generado por Clockify durante la realización del proyecto, el cronograma de MS Project 2021 y la secuenciación de las actividades también con MS Project 2021.

Enlace al Informe de Clockify: [Clockify Informe De Tiempo Detallado 11 09 2023-27 05 2024](#)

Enlace al cronograma: [3.4-Cronograma-v1.0](#)

Enlace a la secuenciación de actividades: [3.3-Secuenciación de actividades-v1.0](#)