



**Universidad de Valladolid**

**ESCUELA DE INGENIERÍA INFORMÁTICA  
DE SEGOVIA**

**Grado en Ingeniería Informática  
de Servicios y Aplicaciones**

---

**Sopin: Web de comercio accesible mediante lenguaje natural**

---

**Alumno: Carlos San Frutos Gómez  
Tutor: Francisco J. González Cabrera**





---

# Agradecimientos

Quiero dar las gracias a mi familia y mis amigos que me han apoyado en todo momento durante este camino que ahora llega a su fin. En especial quiero agradecer a mis padres y mi hermana por no perder nunca la fe en mí.

Por otra parte me gustaría dar las gracias a Nacho y todos mis compañeros de clase por haberme acompañado durante este tiempo académico, a Sofía por todo el apoyo y la ayuda brindados durante el desarrollo de este Trabajo y a Sergio por el apoyo incondicional en los momentos difíciles.

También dar las gracias a todos los profesores que me han enseñado todo lo que sé, en especial a mi tutor Francisco J. González Cabrera. Sin todas estas personas no habría podido llegar hasta aquí.



# Resumen

A lo largo de este tiempo, en el que los avances gracias a internet han sido enormes, también ha evolucionado el desarrollo de las páginas web, modificando su diseño, creando nueva funcionalidad, mejorando su adaptación a los distintos dispositivos, la personalización para cada grupo social o país, etc. En los últimos años una de las grandes novedades ha sido la aparición de los asistentes virtuales. Alexa (Amazon), Siri(Apple) o Cortana(Windows) son algunos de los asistentes virtuales que más sobrenombre tienen en la actualidad.

Los comercios tradicionales han tenido que digitalizarse para poder seguir luchando en este mundo tecnológico y así seguir siendo competitivos en el mercado. De esta forma muchas empresas han tomado la decisión de incluir asistentes virtuales en sus páginas web para que las compras online sean cada vez más cercanas y personales, como solían ser las tiendas de antes. Uno de los problemas de la venta online era lo distante y fría que resultaba a veces, es decir, se trataba de una venta impersonal. Por este motivo el auge de los asistentes de lenguaje natural han ido en aumento a lo largo de los últimos años.

Los comercios online viven actualmente de un modelo de autoservicio en el que los clientes son independientes. Una plataforma usable es de gran ayuda, ya que el objetivo de ésta es ser intuitiva y hacer que el cliente pueda comprar de una forma sencilla por si mismo. El uso de las nuevas tecnologías para la mayoría de usuarios no es ningún problema, esto no quita que existan otros usuarios minoritarios, con el mismo derecho a poder disfrutar de los nuevos avances, que necesitan ayuda u orientación a la hora de realizar compras por internet.

Una persona mayor que no se maneja bien con un periférico mecánico (mouse), porque durante su vida no lo ha usado nunca, una persona con vista reducida o con problemas motrices, etc. En el caso en que un usuario con este perfil quiera realizar compras online, pero sus condiciones no se lo permitan, un asistente virtual que permita el íntegro funcionamiento de la web con la voz puede mejorar mucho su experiencia de usuario. De esta forma tendrían la posibilidad de realizar las tareas que antes no podían o eran muy difíciles de llevar a cabo de una manera mucho más sencilla, intuitiva y natural.

Este Proyecto busca conseguir que cualquier persona pueda utilizar una página web de compras online utilizando su voz, como si fuera un puntero, de una forma sencilla e intuitiva, intentando en todo momento que el usuario reciba el feedback necesario para que le resulte lo más natural y sencillo posible y siempre sepa en que punto del proceso de compra está, o en que página se encuentra. Este proyecto no solo está enfocado al manejo por voz de la aplicación, sino que también está desarrollado para que pueda ser utilizado plenamente con los periféricos punteros tradicionales.

**Palabras clave: Asistente Virtual, Ecommerce, Diseño Web, Node JS, Compras online, lenguaje natural**

---

# Abstract

During this period of time, in which advances due to the internet have been enormous, the development of web pages has also evolved, modifying their design, creating new functionality, improving their adaptability to different devices, customising them for each social group or country, etc. In recent years, one of the great innovations has been the appearance of virtual assistants. Alexa (Amazon), Siri (Apple) or Cortana (Windows) are some of the most popular virtual assistants nowadays.

Traditional shops have had to digitalise in order to be able to continue fighting in this technological world and therefore remain competitive in the market. Thus, many companies have decided to include virtual assistants on their websites to make online shopping more and more personal, as it used to be in the shops of the past. One of the problems with online shopping was that it was sometimes distant and cold, i.e. it was impersonal. This is why the rise of natural language assistants has been increasing over the last few years.

Online shops nowadays live on a self-service model in which customers are independent. A usable platform is a great help, as the aim of the platform is to be intuitive and to make it easy for the customer to buy on their own. The use of new technologies for the most users is not a problem, but this does not mean that there are other minority users, with the same right to benefit from new developments, who need help or orientation when shopping on the Internet.

An elderly person who is not able to use a mechanical device (mouse) because he/she has never used it during his/her life, a person with reduced eyesight or with motor problems, etc. In the case that a user with this profile wants to make online purchases, but their conditions do not allow it, a virtual assistant that allows the entire operation of the web with the voice can greatly improve their user experience. In this way, they would be able to perform tasks that were previously impossible or very difficult to carry out in a much simpler, more intuitive and natural way.

This project aims to ensure that anyone can use an online shopping website using their voice, as if it were a mouse pointer, in a simple and intuitive way, trying at all times to ensure that the user receives the necessary feedback so that it is as natural and simple as possible and always knows where they are in the purchasing process, or which page they are on. This project is not only focused on the voice operation of the application, but it is also developed so that it can be fully used with traditional pointing peripherals.

**Keywords:** Virtual Assistant, Ecommerce, Web Design, Node JS, online shopping, natural language

---

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	3
1.2. Objetivos . . . . .	4
1.3. Organización de la memoria . . . . .	4
<b>2. Estado del Arte</b>	<b>7</b>
<b>3. Análisis</b>	<b>9</b>
3.1. Descripción de los actores . . . . .	9
3.2. Reglas de negocio . . . . .	10
3.3. Requisitos de Usuario . . . . .	11
3.3.1. Requisitos de Usuario Manual . . . . .	11
3.3.2. Requisitos de Usuario Voz . . . . .	12
3.4. Casos de Uso . . . . .	13
3.4.1. Especificación de los casos de uso . . . . .	20
3.5. Requisitos Funcionales . . . . .	38
3.6. Requisitos de Información . . . . .	45
3.7. Requisitos No Funcionales . . . . .	47
<b>4. Planificación y Presupuesto</b>	<b>49</b>
4.1. Metodología . . . . .	49
4.1.1. Herramientas para la organización . . . . .	50
4.2. Planificación . . . . .	52
4.2.1. Estimación . . . . .	52
4.2.2. Coste Real . . . . .	53
4.3. Presupuesto . . . . .	54
4.3.1. Presupuesto estimado . . . . .	54
4.3.2. Coste . . . . .	56
<b>5. Diseño</b>	<b>59</b>
5.1. Arquitectura . . . . .	59
5.2. Diagrama de flujo . . . . .	61
5.3. Interfaz de Usuario . . . . .	62
5.3.1. Tienda Catálogo . . . . .	62
5.3.2. Tienda Ofertas . . . . .	63
5.3.3. Carrito de la Compra . . . . .	64

5.3.4. Historial de Pedidos. . . . .	65
5.3.5. Tutorial con comandos. . . . .	66
<b>6. Implementación</b>	<b>67</b>
6.1. Herramientas y tecnologías utilizadas . . . . .	67
6.2. Herramientas de Soporte . . . . .	68
6.3. Tecnologías Utilizadas . . . . .	68
6.3.1. Node JS . . . . .	69
6.3.2. Mongoose . . . . .	70
6.4. Implementación de la base de datos . . . . .	71
6.5. Estructura del Proyecto . . . . .	73
6.5.1. Front-End . . . . .	73
6.5.2. Back-End . . . . .	75
<b>7. Pruebas</b>	<b>83</b>
7.1. Pruebas de Caja Negra . . . . .	83
7.2. Pruebas conexiones Back-End . . . . .	85
<b>8. Mejoras</b>	<b>91</b>
<b>9. Conclusiones</b>	<b>93</b>
<b>Bibliografía</b>	<b>93</b>
<b>Apéndices</b>	<b>97</b>
<b>A. Despliegue</b>	<b>99</b>
<b>B. Manual de Usuario</b>	<b>107</b>
B.1. Tienda Catálogo . . . . .	107
B.2. Tienda Ofertas . . . . .	108
B.3. Carrito de la Compra . . . . .	109
B.4. Historial de Pedidos. . . . .	110
B.5. Tutorial con comandos. . . . .	111

# Índice de figuras

3.1. Diagrama de Casos de Uso del Usuario Manual. . . . .	14
3.2. Diagrama de Casos de Uso del Usuario Voz (General). . . . .	16
3.3. Diagrama de Casos de Uso del Usuario Voz (Principal). . . . .	17
3.4. Diagrama de Casos de Uso del Usuario Voz (Carrito). . . . .	18
3.5. Diagrama de Casos de Uso del Usuario Voz (Historial). . . . .	18
3.6. Diagrama de Casos de Uso del Usuario Voz (Ofertas). . . . .	19
3.7. Diagrama de Casos de Uso del Usuario Voz (Tutorial). . . . .	19
3.8. Diagrama BBDD No SQL . . . . .	46
4.1. Tablero Trello del proyecto. . . . .	50
4.2. Toggl Track. . . . .	51
4.3. Diagrama de Gantt Inicial. . . . .	53
4.4. Diagrama de Gantt Real. . . . .	53
5.1. Diseño de la aplicación. . . . .	60
5.2. Diagrama de flujo. . . . .	61
5.3. Tienda Catálogo. . . . .	62
5.4. Tienda Ofertas. . . . .	63
5.5. Carrito de la compra. . . . .	64
5.6. Historial de Pedidos. . . . .	65
5.7. Comandos disponibles. . . . .	66
6.1. Esquema de la arquitectura Mongoose, Mongo DB Y Node JS. . . . .	70
6.2. Planes para la base de datos Mongo DB. . . . .	71
6.3. Vista de la información mostrada de un Cluster en Mongo DB. . . . .	72
6.4. Árbol de directorios Front-end . . . . .	73
6.5. Árbol de directorios Back-end . . . . .	75
7.1. 404 Not Found message. . . . .	88
7.2. 400 Bad Request message. . . . .	88
7.3. 200 OK message. . . . .	89
A.1. Página de descarga de Visual Studio Code . . . . .	99
A.2. Primer paso en la instalación de Visual Studio Code . . . . .	100
A.3. Último paso en la instalación de Visual Studio Code . . . . .	100
A.4. Visualización de las extensiones necesarias 1 . . . . .	101

A.5. Visualización de las extensiones necesarias 1 . . . . .	102
A.6. Visualización de los dos proyectos abiertos (Back-end y Fron-end). . . . .	103
A.7. Instalación de Node JS. . . . .	104
A.8. Inicio del servidor mediante el comando 'npm start' . . . . .	104
A.9. Inicio de la aplicación a través del proyecto Front-end. . . . .	105
A.10. Aplicación desplegada correctamente. . . . .	105
B.1. Tienda Catálogo. . . . .	107
B.2. Tienda Ofertas. . . . .	108
B.3. Carrito de la compra. . . . .	109
B.4. Historial de Pedidos. . . . .	110
B.5. Comandos disponibles. . . . .	111
B.6. Comandos disponibles 2. . . . .	112

# Índice de tablas

3.1. Descripción del Actor-01: Usuario Manual. . . . .	9
3.2. Descripción del Actor-02: Usuario Voz. . . . .	9
3.3. Reglas de negocio . . . . .	10
3.4. Requisitos de Usuario del sistema. . . . .	11
3.5. Requisitos de Usuario del sistema. . . . .	12
3.6. Casos de Uso del Usuario Manual. . . . .	13
3.7. Casos de Uso del Usuario Voz. . . . .	15
3.8. Especificación CU-01: Añadir un producto al carrito. . . . .	20
3.9. Especificación CU-02: Eliminar un producto al carrito. . . . .	20
3.10. Especificación CU-03: Vaciar carrito. . . . .	21
3.11. Especificación CU-04: Comenzar proceso de compra. . . . .	21
3.12. Especificación CU-05: Mostrar los productos en oferta del catálogo. . . . .	22
3.13. Especificación CU-06: Mostrar todo el catálogo de productos. . . . .	22
3.14. Especificación CU-07: Ir a la Tienda. . . . .	23
3.15. Especificación CU-08: Ir al Historial de Pedidos. . . . .	23
3.16. Especificación CU-09: Mostrar Tutorial. . . . .	24
3.17. Especificación CU-10: Cerrar tutorial. . . . .	24
3.18. Especificación CU-11: Saber qué usuario está conectado. . . . .	25
3.19. Especificación CU-12: Saber los productos del catálogo. . . . .	25
3.20. Especificación CU-13: Saber el precio de un producto. . . . .	26
3.21. Especificación CU-14: Añadir un producto al carrito. . . . .	26
3.22. Especificación CU-15: Saber el dinero acumulado en el carrito. . . . .	27
3.23. Especificación CU-16: Saber si un producto se encuentra en el catálogo. . . . .	27
3.24. Especificación CU-17: Saber el número de productos del carrito. . . . .	28
3.25. Especificación CU-18: Saber el nombre de los productos que hay en el carrito. . . . .	28
3.26. Especificación CU-19: Eliminar un producto concreto del carrito. . . . .	29
3.27. Especificación CU-20: Vaciar el carrito. . . . .	29
3.28. Especificación CU-21: Comenzar proceso de compra. . . . .	30
3.29. Especificación CU-22: Saber los productos en oferta del catálogo. . . . .	31
3.30. Especificación CU-23: Mostrar los productos en oferta del catálogo. . . . .	31
3.31. Especificación CU-24: Mostrar todo el catálogo de productos. . . . .	32
3.32. Especificación CU-25: Moverse a la Tienda. . . . .	32
3.33. Especificación CU-26: Moverse al Historial de Pedidos. . . . .	33
3.34. Especificación CU-27: Saber el nombre de los pedidos del historial. . . . .	33
3.35. Especificación CU-28: Saber los datos de un pedido en concreto. . . . .	34

3.36. Especificación CU-29: Saber los productos de un pedido concreto. . . . .	34
3.37. Especificación CU-30: Saber la página en la que se está. . . . .	35
3.38. Especificación CU-31: Ver modal tutorial. . . . .	35
3.39. Especificación CU-32: Cerrar modal. . . . .	36
3.40. Especificación CU-33: Dictar comandos. . . . .	36
3.41. Especificación CU-34: Saludar. . . . .	37
3.42. Requisitos Funcionales asociados al Caso de Uso CU-01: Añadir un producto al carrito. . . . .	38
3.43. Requisitos Funcionales asociados al Caso de Uso CU-02: Eliminar un producto al carrito. . . . .	38
3.44. Requisitos Funcionales asociados al Caso de Uso CU-03: Vaciar Carrito. . . . .	38
3.45. Requisitos Funcionales asociados al Caso de Uso CU-04: Comenzar proceso de compra. . . . .	38
3.46. Requisitos Funcionales asociados al Caso de Uso CU-05: Mostrar los productos en oferta del catálogo. . . . .	39
3.47. Requisitos Funcionales asociados al Caso de Uso CU-06: Mostrar todo el catálogo de productos. . . . .	39
3.48. Requisitos Funcionales asociados al Caso de Uso CU-07: Ir a la Tienda. . . . .	39
3.49. Requisitos Funcionales asociados al Caso de Uso CU-08: Ir al Historial de productos. . . . .	39
3.50. Requisitos Funcionales asociados al Caso de Uso CU-09: Mostrar Tutorial. . . . .	39
3.51. Requisitos Funcionales asociados al Caso de Uso CU-10: Cerrar tutorial. . . . .	39
3.52. Requisitos Funcionales asociados al Caso de Uso CU-11: Saber qué usuario está conectado. . . . .	40
3.53. Requisitos Funcionales asociados al Caso de Uso CU-12: Saber los productos del catálogo. . . . .	40
3.54. Requisitos Funcionales asociados al Caso de Uso CU-13: Saber el precio de un producto. . . . .	40
3.55. Requisitos Funcionales asociados al Caso de Uso CU-14 : Añadir un producto al carrito. . . . .	40
3.56. Requisitos Funcionales asociados al Caso de Uso CU-15 : Saber el dinero acumulado en el carrito. . . . .	40
3.57. Requisitos Funcionales asociados al Caso de Uso CU-16 : Saber si un producto se encuentra en el catálogo. . . . .	41
3.58. Requisitos Funcionales asociados al Caso de Uso CU-17 : Saber el número de productos del carrito. . . . .	41
3.59. Requisitos Funcionales asociados al Caso de Uso CU-18 : Saber el nombre de los productos que hay en el carrito. . . . .	41
3.60. Requisitos Funcionales asociados al Caso de Uso CU-19 : Eliminar un producto concreto del carrito. . . . .	41
3.61. Requisitos Funcionales asociados al Caso de Uso CU-20 : Vaciar carrito. . . . .	41
3.62. Requisitos Funcionales asociados al Caso de Uso CU-21 : Comenzar proceso de compra. . . . .	42
3.63. Requisitos Funcionales asociados al Caso de Uso CU-22 : Saber los productos en oferta del catálogo. . . . .	42
3.64. Requisitos Funcionales asociados al Caso de Uso CU-23 : Mostrar los productos en oferta del catálogo. . . . .	42

3.65. Requisitos Funcionales asociados al Caso de Uso CU-24 : Mostrar todo el catálogo de productos. . . . .	42
3.66. Requisitos Funcionales asociados al Caso de Uso CU-25 : Moverse a la Tienda. . .	43
3.67. Requisitos Funcionales asociados al Caso de Uso CU-26 : Moverse al historial de pedidos. . . . .	43
3.68. Requisitos Funcionales asociados al Caso de Uso CU-27 : Saber el nombre de los pedidos del historial. . . . .	43
3.69. Requisitos Funcionales asociados al Caso de Uso CU-28 : Saber los datos de un pedido concreto. . . . .	43
3.70. Requisitos Funcionales asociados al Caso de Uso CU-29 : Saber los productos de un pedido. . . . .	43
3.71. Requisitos Funcionales asociados al Caso de Uso CU-30 : Saber la página en la que se está. . . . .	44
3.72. Requisitos Funcionales asociados al Caso de Uso CU-31 : Ver modal tutorial. . .	44
3.73. Requisitos Funcionales asociados al Caso de Uso CU-32 : Cerrar modal tutorial. .	44
3.74. Requisitos Funcionales asociados al Caso de Uso CU-33 : Dictar comandos. . . .	44
3.75. Requisitos Funcionales asociados al Caso de Uso CU-34 : Saludar. . . . .	44
3.76. Requisitos No Funcionales: Seguridad . . . . .	47
3.77. Requisitos No Funcionales: Usabilidad . . . . .	47
3.78. Requisitos No Funcionales: Robustez . . . . .	47
3.79. Requisitos No Funcionales: Disponibilidad . . . . .	47
4.1. Costes de herramientas y recursos hardware. . . . .	54
4.2. Costes asociados a recursos humanos. . . . .	55
4.3. Costes Totales. . . . .	55
4.4. Actualización costes de herramientas y recursos hardware. . . . .	56
4.5. Actualización costes asociados a recursos humanos. . . . .	56
4.6. Costes Totales. . . . .	57
5.1. Requisitos No Funcionales: Seguridad . . . . .	60
7.1. Prueba de Caja Negra CN-01: Realización del proceso de compra completo utilizando comandos vocales con objetivo satisfactorio. . . . .	83
7.2. Prueba de Caja Negra CN-02: Realización del proceso de compra completo utilizando periféricos tradicionales con objetivo satisfactorio. . . . .	84
7.3. Prueba de Caja Negra CN-03: Realización del proceso de compra completo utilizando comandos vocales con objetivo de fallo. . . . .	84
7.4. Prueba de Caja Negra CN-04: Realización del proceso de compra completo utilizando periféricos tradicionales con objetivo de fallo. . . . .	85



# Capítulo 1

## Introducción

[4] Antes de que existiera internet y las compras online, se compraba en negocios y tiendas físicas de forma personal. Allí se podía ver, tocar y probar cada producto justo antes de comprarlo y llevárselo a su casa instantáneamente, incluso los compradores tenían dependientes a su disposición para resolver cualquier duda o recibir cualquier descripción o anotación requerida del producto. Algunos de los problemas que esto conllevaba era el transporte obligatorio hasta la tienda física, la limitación en cuanto a número de tiendas y vendedores que se podían visitar en un determinado tiempo, o incluso la variedad o competitividad entre diferentes vendedores. Después de esto empezó a aparecer una nueva forma de comercio revolucionaria llamada venta por catálogo. Esta nueva forma de comprar y vender se basaba en la distribución de catálogos directamente a los compradores, fue la primera forma de comprar sin antes ver físicamente el producto; funcionaba mediante fotos ilustrativas. [8]

No fue hasta los años 90 cuando se desencadenaría el boom de internet, con la aparición de la World Wide Web ("WWW") se despuntó el comercio electrónico, que creció como nunca antes. En 1995 nacieron Amazon e eBay, que en la actualidad se mantienen como 2 de las empresas de venta electrónica más grandes del mundo. Hemos visto como el crecimiento del comercio en la última década ha sido exponencial y ha hecho que la forma de comprar cambie para siempre. Ahora tenemos la posibilidad de comprar cientos de productos sin movernos de nuestra casa a un solo click. [8]

A lo largo de este tiempo, en el que los avances gracias a internet han sido enormes, también ha evolucionado el desarrollo de las páginas web, modificando su diseño, creando nueva funcionalidad, mejorando su adaptación a los distintos dispositivos, la personalización para cada grupo social o país, etc.

En los últimos años una de las grandes novedades ha sido la aparición de los asistentes virtuales. Alexa (Amazon), Siri (Apple) o Cortana (Windows) son algunos de los asistentes virtuales que más sobrenombre tienen en la actualidad. Según la Wikipedia, "Un asistente virtual es un agente software que ayuda a usuarios de sistemas computacionales, automatizando y realizando tareas con la mínima interacción hombre-máquina. La interacción que se da entre un asistente virtual y una persona, debe ser natural, una persona se comunica usando la voz y el asistente virtual lo procesa, interpreta y responde de la misma manera." [15]

Los comercios tradicionales han tenido que digitalizarse para poder seguir luchando en este mundo tecnológico y así seguir siendo competitivos en el mercado. De esta forma muchas empresas han tomado la decisión de incluir asistentes virtuales en sus páginas web para que las compras online sean cada vez más cercanas y personales, como solían ser las tiendas de antes. Uno de los problemas de la venta online era lo distante y fría que resultaba a veces, es decir, se trataba de una venta impersonal. Por este motivo el auge de los asistentes de lenguaje natural han ido en aumento a lo largo de los últimos años. [12]

Los comercios online viven actualmente de un modelo de autoservicio en el que los clientes son independientes. Una plataforma usable es de gran ayuda, ya que el objetivo de ésta es ser intuitiva y hacer que el cliente pueda comprar de una forma sencilla por si mismo. El uso de las nuevas tecnologías para la mayoría de usuarios no es ningún problema, esto no quita que existan otros usuarios minoritarios, con el mismo derecho a poder disfrutar de los nuevos avances, que necesitan ayuda u orientación a la hora de realizar compras por internet. Esto puede ser debido a diferentes motivos. Una persona mayor que no se maneja bien con un periférico mecánico (mouse), porque durante su vida no lo ha usado nunca, una persona con vista reducida o con problemas motrices, etc. En el caso en que un usuario con este perfil quiera realizar compras online, pero sus condiciones no se lo permitan, un asistente virtual que permita el íntegro funcionamiento de la web con la voz puede mejorar mucho su experiencia de usuario. De esta forma tendrían la posibilidad de realizar las tareas que antes no podían o eran muy difíciles de llevar a cabo de una manera mucho más sencilla, intuitiva y natural. [11]

Los asistentes virtuales como ayuda para las personas dependientes comienza a postular como una solución cada vez más real, todo gracias a este gran desarrollo tecnológico que venimos experimentando durante todo este tiempo. Estos pequeños compañeros que ni si quieren se pueden ver, ya sea porque están dentro de un altavoz, una pantalla o cualquier objeto pueden ayudar mucho más ya que su verdadero potencial está aun por descubrirse. Uno de los sectores de la sociedad en el que se puede comprobar todo este potencial es en el mencionado anteriormente, las personas dependientes. Estas personas necesitan toda la ayuda posible y no siempre es viable satisfacer sus necesidades. Existen electrodomésticos y objetos domóticos, por ejemplo, que solventan muchos de los problemas que surgen debido a la dependencia de estas personas. Podrían bajar y subir persianas, pedir citas en diferentes instituciones, apagar y encender luces, controlar la TV, regular la calefacción, o incluso hacer la compra. Todo esto sin ni siquiera moverse del sitio. No se puede olvidar una de las cosas que más sufren las personas dependientes, la soledad. Estos sistemas virtuales ayudan en muchos casos a acercar las posibilidades tecnológicas que construyen este mundo y hacerlas un poco más cercanas, amenas y familiares. [7]

## 1.1. Motivación

Me gustaría comenzar la motivación con un caso cercano y un tanto anómalo con el que me encontré durante mi período de estudios en Alemania, donde cursaba una beca Erasmus+. Un día estaba en la calle con los amigos que allí hice, cuando se nos acercó una persona que hablaba castellano en busca de ayuda. Nos empezó a decir que necesitaba alguna persona que pudiera manejarse de forma fluida con cualquier dispositivo tecnológico con acceso a internet ya que disponía de una discapacidad en la vista que le hacía imposible mirar directamente cualquier pantalla de cualquier dispositivo. El proceso que teníamos que llevar a cabo para ayudarlo era el de intermediario entre la aplicación web que quería utilizar y él mismo. Nos iría dando instrucciones de lo que teníamos que hacer y los pasos que tendríamos que seguir con la información pertinente para poder realizar el trámite. Al final nosotros en ese momento no hicimos otra cosa más que tener el rol que tendría cualquier asistente virtual. El usuario iba diciendo los pasos que había que seguir, nosotros lo realizábamos y le íbamos devolviendo el feedback necesario para que supiera en que punto del proceso estaba y cual sería el paso siguiente. Si esta persona hubiera tenido a su disposición algún asistente virtual embebido en la página que quería utilizar, sus problemas hubieran disminuido y habría perdido su condición de dependencia.

Existen este tipo de usuarios de los que se ha hablado en el párrafo inmediatamente superior y en el apartado introductorio de este documento. Para ellos, realizar un proceso tan específico y cotidiano como comprar online, o realizar cualquier tipo de trámite, resulte una tarea complicada, o que les resulta prácticamente imposible realizarlo de forma habitual. Es importante intentar satisfacer las necesidades de todos los tipos de usuarios y de incluirlos en nuestras aplicaciones. Por este motivo, la idea principal para el desarrollo de esta aplicación nace de poder adaptar a una página web de comercio ya establecida, un asistente virtual de lenguaje natural para el manejo de ésta. De esta forma todos los usuarios referidos anteriormente podrían, sin ningún problema, hacer uso íntegro de la aplicación desarrollada en este proyecto: Sopin.

Este Proyecto busca conseguir que cualquier persona pueda utilizar una página web de compras online utilizando su voz, como si fuera un puntero, de una forma sencilla e intuitiva, intentando en todo momento que el usuario reciba el feedback necesario para que le resulte lo más natural y sencillo posible y siempre sepa en que punto del proceso de compra está, o en que página se encuentra. Este proyecto no solo está enfocado al manejo por voz de la aplicación, sino que también está desarrollado para que pueda ser utilizado plenamente con los periféricos punteros tradicionales.

En conclusión, “SOPIN” es una aplicación orientada a mejorar el manejo de una página web mediante el uso del lenguaje natural. Está enfocada a facilitar el uso de los recursos ofrecidos por una web de comercio por parte de usuarios que no puedan utilizar punteros tradicionales, como un ratón o un teclado.

## 1.2. Objetivos

Este proyecto afronta los problemas explicados en el apartado anterior realizando el desarrollo de una aplicación que cumpla los objetivos que se describen a continuación:

OBJ-1 Correcto procesamiento de la información introducida en la aplicación a través de la voz

OBJ-1.1 Crear funcionalidad que responda ante un estímulo vocal.

OBJ-1.2 Realizar un feedback constante por medio de dicciones de voz por parte de la aplicación.

OBJ-2 Correcto procesamiento de la información introducida en la aplicación a través de los periféricos mecánicos.

OBJ-2.1 Crear funcionalidad que responda ante un estímulo mecánico.

OBJ-2.2 Realizar un feedback constante por medio de recursos visuales en la aplicación.

OBJ-3 Permitir un flujo conversacional fluido y constante entre el usuario y el sistema.

OBJ-3.1 El usuario podrá utilizar todas las funciones que ofrece la aplicación con el uso de su voz.

OBJ-4 Facilitar el uso de herramientas y aplicaciones relacionadas con el comercio online para personas dependientes.

OBJ-5 Acercar la tecnología y hacerla más familiar para personas que no se familiaricen con su uso.

## 1.3. Organización de la memoria

El documento actual se estructura en capítulos en los que se tratan diferentes aspectos del proyecto. Los capítulos que componen este documento son los siguientes:

**Capítulo 1. Introducción:** A lo largo del Capítulo 1 se realizará una introducción sobre el contexto en el que se desarrolla este proyecto, así como la motivación de este y los objetivos que se pretenden conseguir.

**Capítulo 2. Estado del Arte:** A lo largo del Capítulo 2 se analizan brevemente algunas aplicaciones con puntos en común al proyecto desarrollado en este documento.

**Capítulo 3. Análisis:** A lo largo del Capítulo 2 se realizará un análisis del problema. En este capítulo se encuentran detallados los actores del sistema, requisitos de usuario, requisitos funcionales y no funcionales, etc.

**Capítulo 4. Planificación y Presupuesto:** A lo largo del Capítulo 4 se detalla una estimación del coste en tiempo y recursos del proyecto y, posteriormente, se realiza un análisis del coste real del proyecto.

**Capítulo 5. Diseño:** A lo largo del Capítulo 5 se detallan las distintas arquitecturas del proyecto, complementadas con sus respectivos diagramas explicativos. Además se especificará el diseño de la base de datos y las interfaces de usuario.

**Capítulo 6. Implementación:** A lo largo del Capítulo 6 se describen detalles sobre la implementación y desarrollo del proyecto relacionados principalmente con las tecnologías empleadas para el desarrollo.

**Capítulo 7. Pruebas:** A lo largo del Capítulo 7 se pueden apreciar las distintas pruebas realizadas para asegurar que la aplicación cumple los requisitos y funciona correctamente.

**Capítulo 8. Mejoras:** A lo largo del Capítulo 8 se exponen algunas mejoras para el futuro que podrían añadirse al proyecto.

**Capítulo 9. Conclusiones:** A lo largo del Capítulo 9 se exponen las conclusiones que se han extraído del desarrollo del proyecto.

**Apéndices:** En este apartado del documento se encuentra información adicional asociada a la entrega del proyecto, así como manuales y detalles de la aplicación desarrollada.



## Capítulo 2

# Estado del Arte

Como ya se ha comentado en el apartado introductorio de este documento los asistentes virtuales forman parte, cada vez más, de nuestro día a día. Gracias a los continuos avances que se producen en el campo tecnológico y sobre todo en la inteligencia artificial, el procesamiento natural de las máquinas cada vez es más sofisticado haciendo que parezcan reales.

Nos encontramos en una etapa de constante cambio y avance gracias al desarrollo que ha provocado internet. Los avances en inteligencia artificial, el uso de los datos para la toma de decisiones o el uso del Cloud, forman parte de nuestra vida cotidiana. Derivado de este contexto nace la necesidad de intentar facilitar nuestra relación directa con la tecnología, y encontrar nuevas funcionalidades que ofrezcan valor a las personas. Una de las formas de lograr esto es mediante la humanización de la tecnología con el uso de la interfaz vocal. [14]

Todo este camino comenzó en los años 20 cuando se consiguió que un perro de juguete Radio Rex” se asomase al pronunciar su nombre. Desde entonces ha pasado mucho tiempo y la precisión en el reconocimiento de la voz ha sido muy notable. A partir de 2010, gracias a la aparición del Deep Learning fue cuando de verdad se empezaron a conseguir precisiones muy altas, convirtiéndolo en algo real y haciendo que dejara de ser algo ficticio. Como ocurre en el campo de la informática, el idioma inglés es, sin duda, el más predominante. Es por eso que la precisión que encontramos en con este idioma no es comparable con el de el resto. Esto no quita que el resto de idiomas vayan aumentando cada vez más su precisión hasta que consigan alcanzar la misma fluidez. [14]

Aunque la voz no siempre es recomendable para cualquier tipo de interacciones, estamos entrando en un nuevo paradigma en cómo interactuaremos con la tecnología. La domótica también será un gran beneficiado de este nuevo paradigma ya que permitirá que la tecnología más cotidiana, como es la del hogar, sea cada vez más cercana a nosotros hasta el punto en que podamos controlar todos los aspectos de nuestra vida únicamente con el uso de nuestra voz. [14]

[14]De todas formas, como ya se ha dicho en el parrafo anterior, la interfaz de la voz no siempre es adecuada y hay ocasiones en las que la voz no resuelve de forma más rápida que la pantalla un problema de un usuario.



## Capítulo 3

# Análisis

En esta sección del documento se especificará todo lo relacionado con el Análisis que se ha llevado a cabo para el desarrollo del proyecto.

### 3.1. Descripción de los actores

En esta primera sección se abordará la descripción de los distintos actores que interactuarán con nuestra aplicación software.3.1:

<b>A-01</b>	<b>Usuario Manual</b>
<b>Versión</b>	1.0 (15/04/2021).
<b>Descripción</b>	Este usuario utilizará toda la funcionalidad de la aplicación y podrá moverse por las páginas que conforman dicha aplicación utilizando métodos de manejo tradicionales, como un ratón.
<b>Comentarios</b>	No necesita autenticación para utilizar la herramienta.

Tabla 3.1: Descripción del Actor-01: Usuario Manual.

<b>A-02</b>	<b>Usuario Voz</b>
<b>Versión</b>	1.0 (15/04/2021).
<b>Descripción</b>	Este usuario utilizará toda la funcionalidad de la aplicación y podrá moverse por las páginas que conforman dicha aplicación mediante el uso exclusivo de la voz.
<b>Comentarios</b>	No necesita autenticación para utilizar la herramienta.

Tabla 3.2: Descripción del Actor-02: Usuario Voz.

## 3.2. Reglas de negocio

Las reglas de negocio son condiciones que se debe satisfacer cuando se realiza una actividad de negocio. Una regla puede imponer una política de negocio, tomar una decisión o inferir nuevos datos de datos existentes.3.3:

ID	Descripción
RN-01	Se podrá utilizar la aplicación íntegramente con la voz.
RN-02	Se podrá utilizar la aplicación mediante el uso de periféricos punteros tradicionales.
RN-03	El asistente podrá acoplarse a las necesidades de la página de compras.
RN-04	Correcto procesamiento de la información introducida en la aplicación a través de la voz.
RN-05	Realizar un feedback constante por medio de dicciones vocales por parte de la aplicación.
RN-06	La aplicación permitirá un flujo conversacional fluido entre el usuario y el sistema.
RN-07	Facilitar las compras online a personas dependientes.
RN-08	Acercar los avances tecnológicos a personas que no están familiarizadas con el uso de estos.

Tabla 3.3: Reglas de negocio

### 3.3. Requisitos de Usuario

En esta sección se presentarán los distintos requisitos de usuario que han de verse satisfechos por la aplicación software, así como las características y peculiaridades de estos. En la Tabla 3.5 se muestran los distintos requisitos de usuario asociados al actor *Usuario*:

#### 3.3.1. Requisitos de Usuario Manual

ID	Nombre
RU-01	Un usuario puede añadir un producto al catálogo mediante el uso de un periférico físico.
RU-02	Un usuario puede eliminar un producto del carrito mediante el uso de un periférico físico.
RU-03	Un usuario puede vaciar el carrito mediante el uso de un periférico físico
RU-04	Un usuario puede comenzar el proceso de compra mediante el uso de un periférico físico
RU-05	Un usuario puede mostrar los productos en oferta del catálogo mediante el uso de un periférico físico
RU-06	Un usuario puede mostrar todo el catálogo de productos mediante el uso de un periférico físico
RU-07	Un usuario puede ir a la Tienda mediante el uso de un periférico físico
RU-08	Un usuario puede ir al Historial de Pedidos mediante el uso de un periférico físico
RU-09	Un usuario puede mostrar tutorial mediante el uso de un periférico físico
RU-10	Un usuario puede cerrar el tutorial mediante el uso de un periférico físico

Tabla 3.4: Requisitos de Usuario del sistema.

### 3.3.2. Requisitos de Usuario Voz

ID	Nombre
RU-11	Un usuario puede saber su nombre de usuario cuando está conectado en la aplicación mediante el uso de su voz.
RU-12	Un usuario puede saber los productos del catálogo mediante el uso de la voz
RU-13	Un usuario puede saber el precio de un producto mediante el uso de la voz
RU-14	Un usuario puede añadir un producto al catálogo mediante el uso de la voz
RU-15	Un usuario puede saber el dinero acumulado en el carrito mediante el uso de la voz
RU-16	Un usuario puede saber si un producto se encuentra en el carrito mediante el uso de la voz
RU-17	Un usuario puede saber el número de productos del carrito mediante el uso de la voz
RU-18	Un usuario puede saber el nombre de los productos que hay en el carrito mediante el uso de la voz
RU-19	Un usuario puede eliminar un producto concreto del carrito mediante el uso de la voz
RU-20	Un usuario puede vaciar el carrito mediante el uso de la voz
RU-21	Un usuario puede comenzar el proceso de compra mediante el uso de la voz
RU-22	Un usuario puede saber los productos en oferta del catálogo mediante el uso de la voz
RU-23	Un usuario puede mostrar los productos en oferta del catálogo mediante el uso de la voz
RU-24	Un usuario puede mostrar todo el catálogo de productos mediante el uso de la voz
RU-25	Un usuario puede moverse a la Tienda mediante el uso de la voz
RU-26	Un usuario puede moverse al Historial de Pedidos mediante el uso de la voz
RU-27	Un usuario puede saber el nombre de los pedidos del historial mediante el uso de la voz
RU-28	Un usuario puede saber los datos de un pedido en concreto mediante el uso de la voz
RU-29	Un usuario puede saber los productos de un pedido concreto mediante el uso de la voz
RU-30	Un usuario puede saber la página en la que se está mediante el uso de la voz
RU-31	Un usuario puede ver el modal del tutorial mediante el uso de la voz
RU-32	Un usuario puede cerrar el modal del tutorial mediante el uso de la voz
RU-33	Un usuario puede escuchar los posibles comandos del tutorial mediante el uso de la voz.
RU-34	Un usuario puede saludar al sistema mediante el uso de la voz

Tabla 3.5: Requisitos de Usuario del sistema.

### 3.4. Casos de Uso

A lo largo de esta sección se presentan los distintos casos de uso asociados a los requisitos de usuario anteriormente descritos, así como la especificación de alguno de ellos y el diagrama de casos de uso ligado a cada usuario. En la Tabla 3.6 se muestran los distintos casos de uso obtenidos a partir de los requisitos del Usuario Manual y en la Tabla 3.7 se muestran los distintos casos de uso obtenidos a partir de los requisitos del Usuario Voz.

<b>ID</b>	<b>RU Asociado</b>	<b>Nombre</b>
<b>CU-01</b>	RU-01	Añadir un producto al carrito
<b>CU-02</b>	RU-02	Eliminar un producto del carrito
<b>CU-03</b>	RU-03	Vaciar carrito
<b>CU-04</b>	RU-04	Comenzar proceso de compra
<b>CU-05</b>	RU-05	Mostrar los productos en oferta del catálogo
<b>CU-06</b>	RU-06	Mostrar todo el catálogo de productos
<b>CU-07</b>	RU-07	Ir a la Tienda
<b>CU-08</b>	RU-08	Ir al Historial de Pedidos
<b>CU-09</b>	RU-09	Mostrar tutorial
<b>CU-10</b>	RU-10	Cerrar tutorial

Tabla 3.6: Casos de Uso del Usuario Manual.

Una vez obtenidos los casos de uso asociados a los requisitos de usuario, se ha generado el Diagrama de Casos de Uso asociado a cada uno de los actores que interactúan con el sistema, obteniendo una versión esquemática y simplificada de la funcionalidad que ha de satisfacer el sistema. En la Figura 3.1 se muestra el Diagrama de Casos de Uso de la aplicación para el Usuario Manual. En la Figura :3.2 se puede ver el diagrama de casos de uso del usuario Voz, al ser tan grande este diagrama ha sido dividido en diagramas más pequeños que se pueden apreciar en las figuras: 3.3, 3.4, 3.5, 3.6 y 3.7.

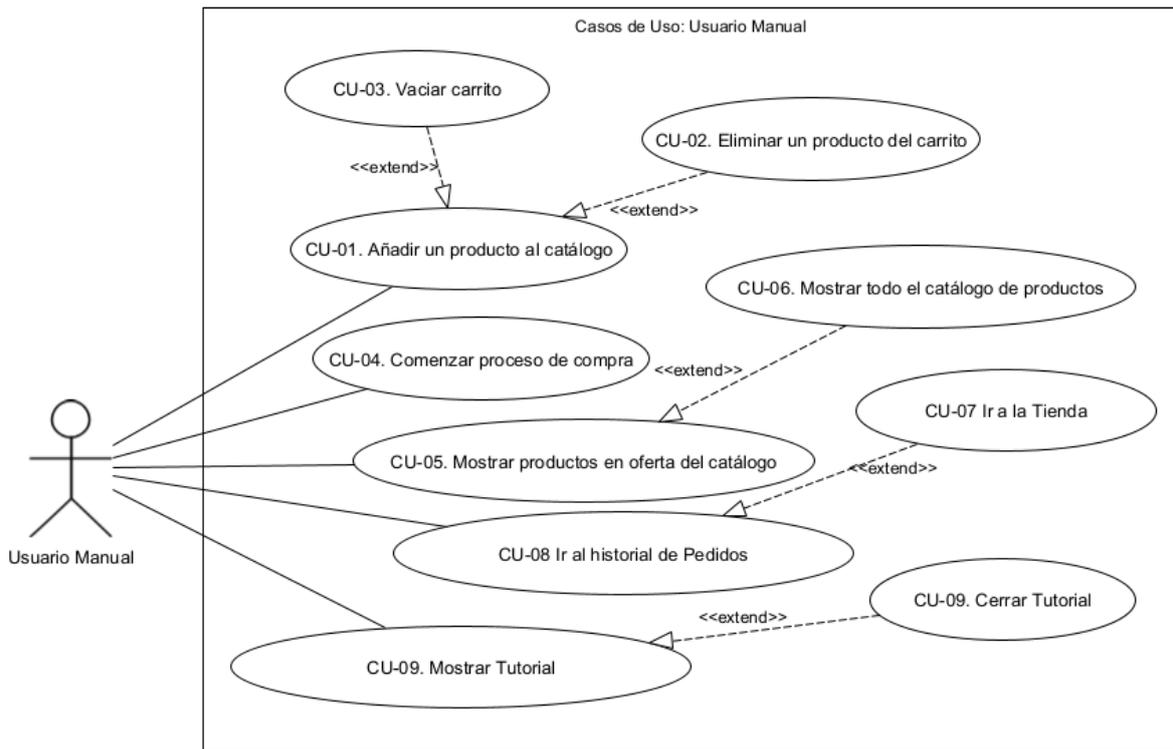


Figura 3.1: Diagrama de Casos de Uso del Usuario Manual.

<b>ID</b>	<b>RU Asociado</b>	<b>Nombre</b>
<b>CU-11</b>	RU-11	Saber qué usuario está conectado
<b>CU-12</b>	RU-12	Saber los productos del catálogo
<b>CU-13</b>	RU-13	Saber el precio de un producto
<b>CU-14</b>	RU-14	Añadir un producto al carrito
<b>CU-15</b>	RU-15	Saber el dinero acumulado en el carrito
<b>CU-16</b>	RU-16	Saber si un producto se encuentra en el catálogo
<b>CU-17</b>	RU-17	Saber el número de productos del carrito
<b>CU-18</b>	RU-18	Saber el nombre de los productos que hay en el carrito
<b>CU-19</b>	RU-19	Eliminar un producto concreto del carrito
<b>CU-20</b>	RU-20	Vaciar el carrito
<b>CU-21</b>	RU-21	Comenzar proceso de compra
<b>CU-22</b>	RU-22	Saber los productos en oferta del catálogo
<b>CU-23</b>	RU-23	Mostrar los productos en oferta del catálogo
<b>CU-24</b>	RU-24	Mostrar todo el catálogo de productos
<b>CU-25</b>	RU-25	Moverse a la Tienda
<b>CU-26</b>	RU-26	Moverse al Historial de Pedidos
<b>CU-27</b>	RU-27	Saber el nombre de los pedidos del historial
<b>CU-28</b>	RU-28	Saber los datos de un pedido en concreto
<b>CU-29</b>	RU-29	Saber los productos de un pedido concreto
<b>CU-30</b>	RU-30	Saber la página en la que se está
<b>CU-31</b>	RU-31	Ver modal tutorial
<b>CU-32</b>	RU-32	Cerrar modal
<b>CU-33</b>	RU-33	Dictar comandos
<b>CU-34</b>	RU-34	Saludar

Tabla 3.7: Casos de Uso del Usuario Voz.



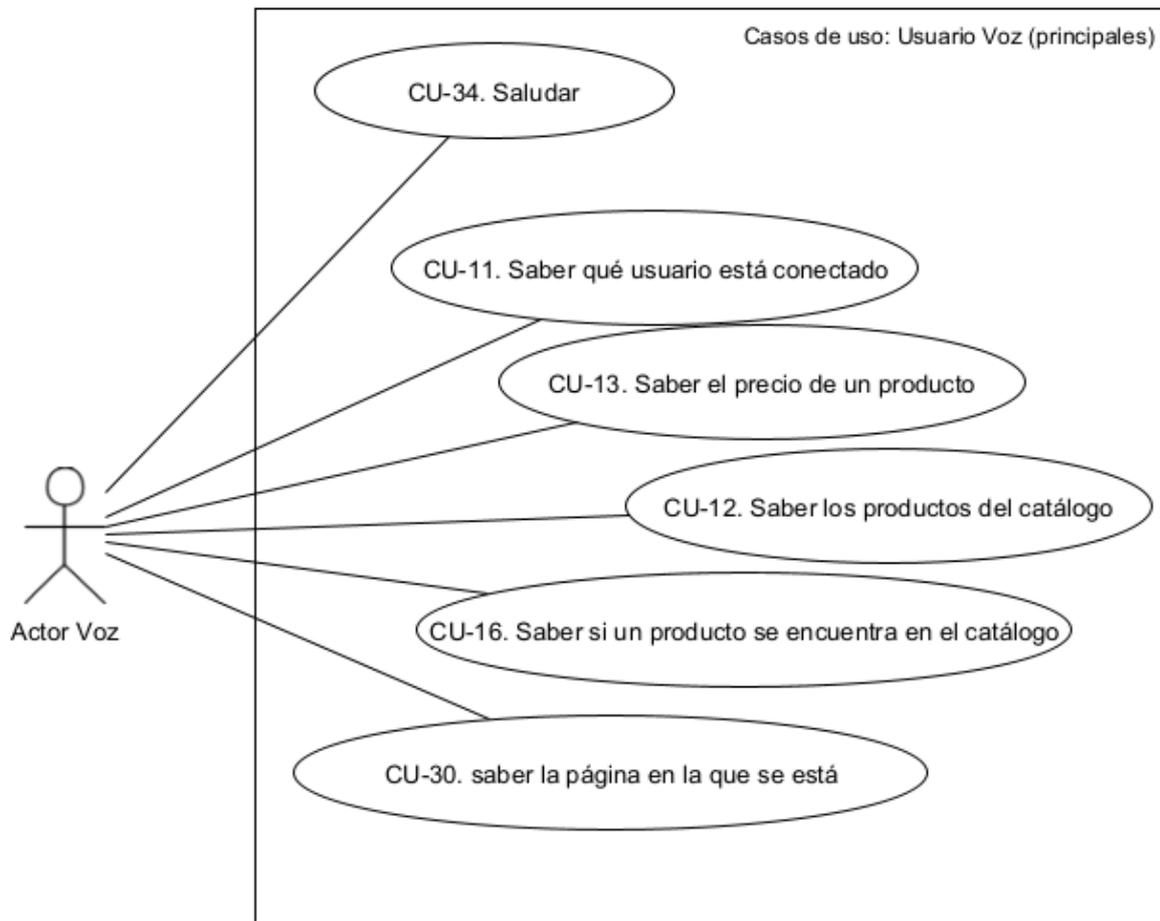


Figura 3.3: Diagrama de Casos de Uso del Usuario Voz (Principal).

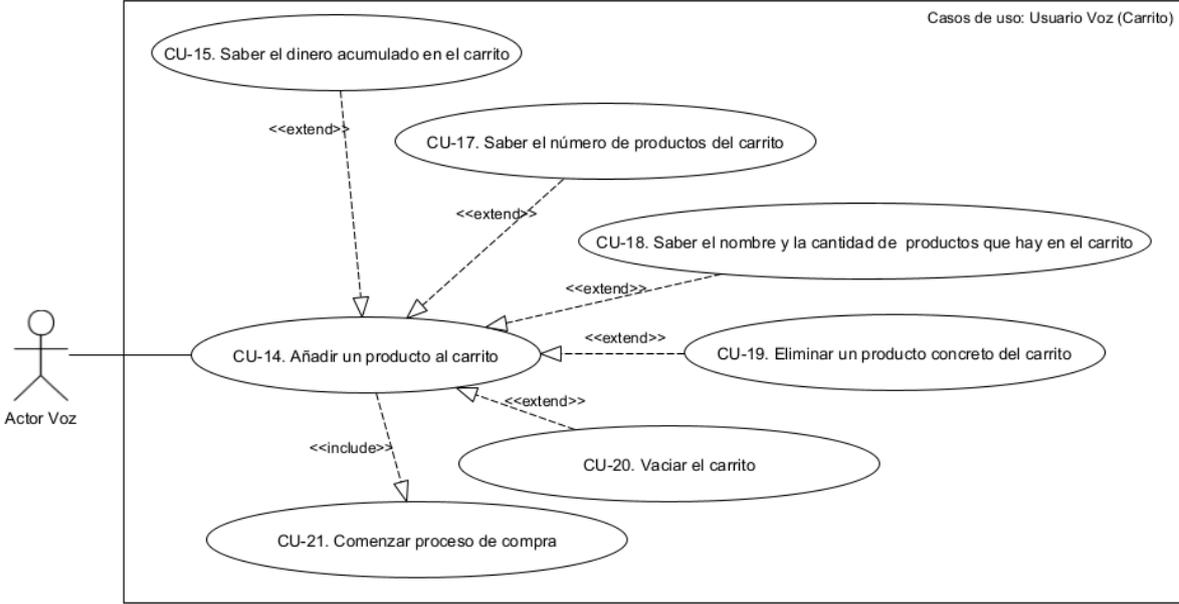


Figura 3.4: Diagrama de Casos de Uso del Usuario Voz (Carrito).

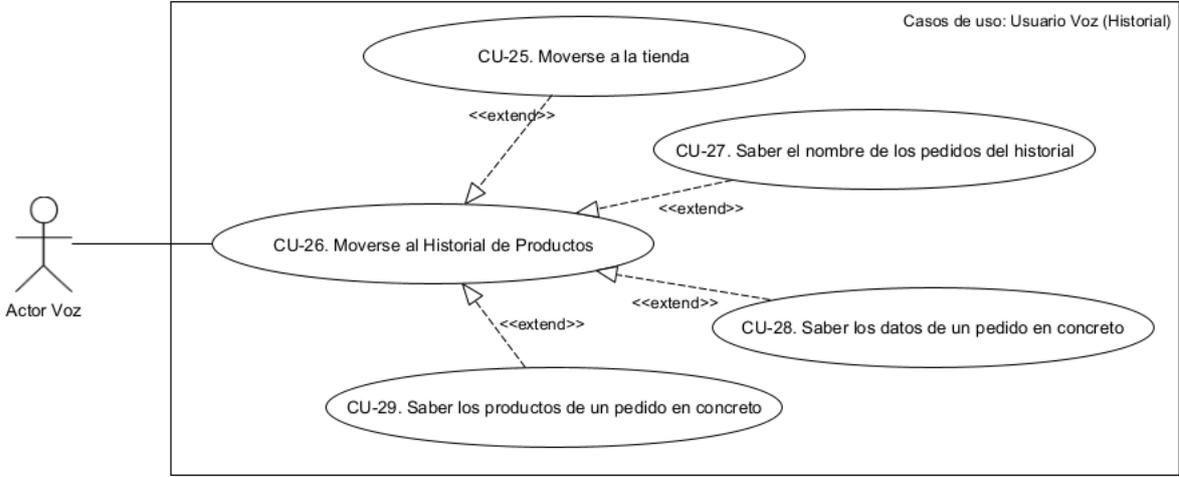


Figura 3.5: Diagrama de Casos de Uso del Usuario Voz (Historial).

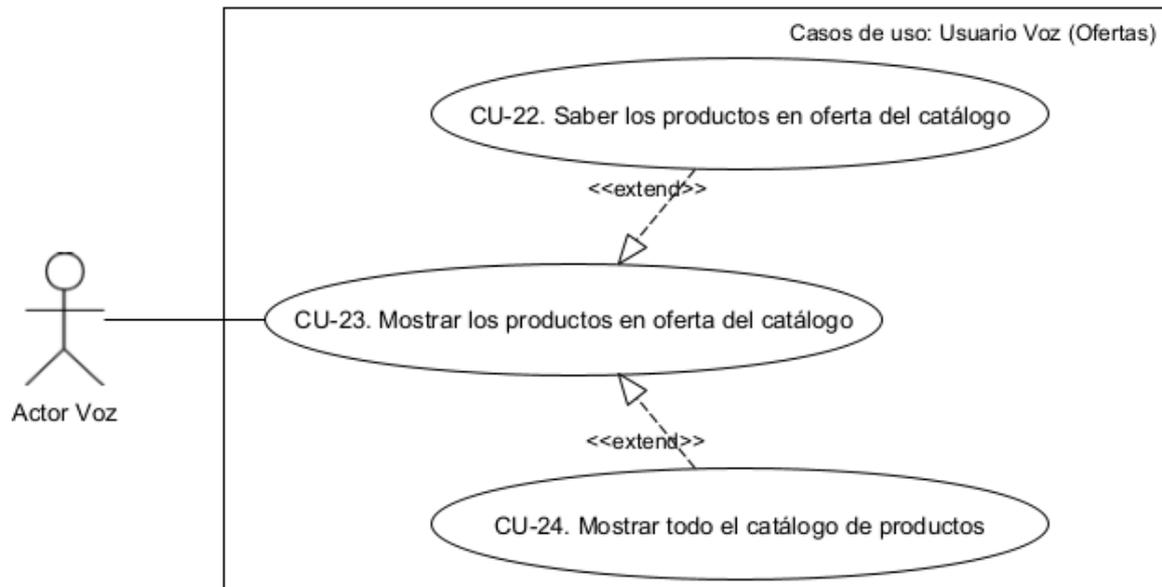


Figura 3.6: Diagrama de Casos de Uso del Usuario Voz (Ofertas).

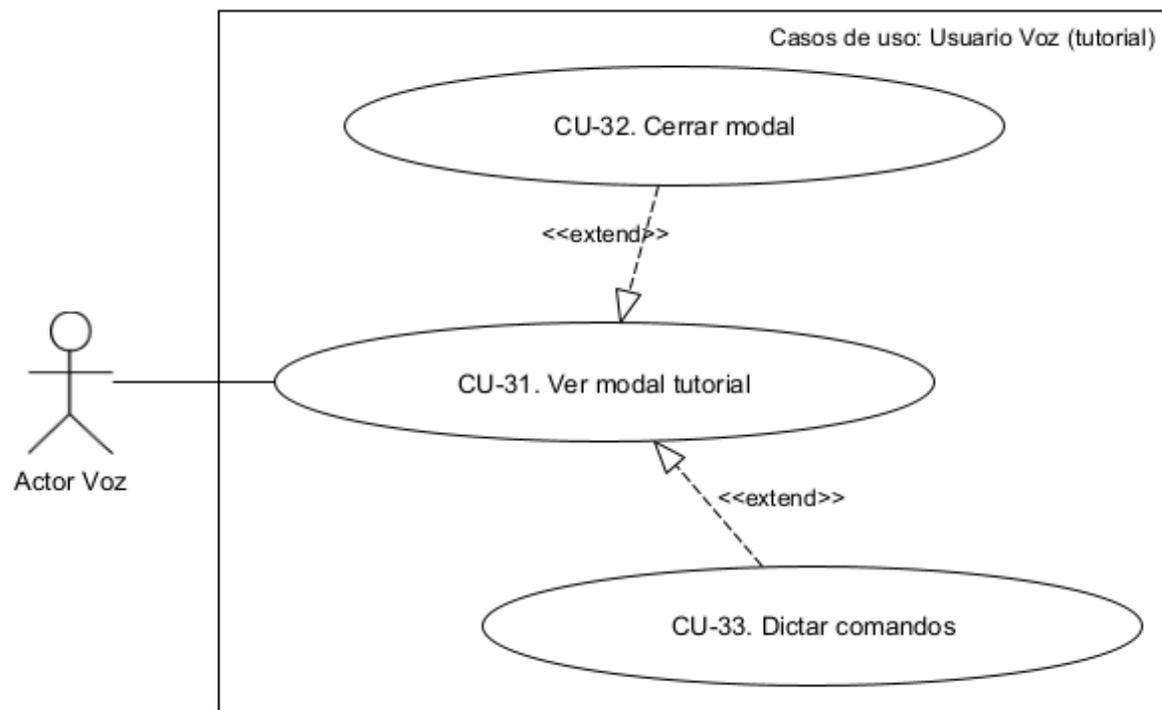


Figura 3.7: Diagrama de Casos de Uso del Usuario Voz (Tutorial).

### 3.4.1. Especificación de los casos de uso

En esta subsección se realizará una especificación de los casos de uso obtenidos a partir de los requisitos de usuario.

<b>CU-01</b>	<b>Añadir un producto al carrito</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Manual.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El sistema añade un producto seleccionado al carrito de la compra.
<b>Disparador</b>	El usuario pulsa el botón para añadir productos al carrito.
<b>Precondiciones</b>	Los datos de los productos han de haberse cargado correctamente desde la base de datos al inicio de la aplicación.
<b>Postcondiciones</b>	En el carrito aparece añadido el producto seleccionado.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón 'Añadir al carrito'.</li> <li>2. El sistema muestra en el apartado Carrito de la compra el producto seleccionado con alguna información de éste.</li> <li>3. El sistema activa el botón Comprar y el botón vaciar el carrito.</li> <li>4. Caso de Uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.8: Especificación CU-01: Añadir un producto al carrito.

<b>CU-02</b>	<b>Eliminar un producto al carrito.</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Manual.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El sistema elimina uno de los productos que se encuentran dentro del carrito.
<b>Disparador</b>	El usuario pulsa el botón rojo con una X para eliminar el producto deseado .
<b>Precondiciones</b>	Debe existir un producto añadido al carrito para poder ser eliminado.
<b>Postcondiciones</b>	El producto ha sido eliminado del carrito.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón rojo 'X'.</li> <li>2. El sistema elimina satisfactoriamente el producto seleccionado del carrito.</li> <li>3. Si es el último producto que quedaba en el carrito, se desactivan los botones Comprar y Vaciar carrito.</li> <li>4. Caso de Uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.9: Especificación CU-02: Eliminar un producto al carrito.

<b>CU-03</b>	<b>Vaciar carrito</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Manual.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El sistema elimina todos los productos añadidos al catálogo.
<b>Disparador</b>	El usuario pulsa el botón Vaciar carrito.
<b>Precondiciones</b>	Para poder vaciar el carrito antes deben existir productos añadidos al mismo.
<b>Postcondiciones</b>	El carrito queda vacío.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1.El usuario pulsa el botón 'Vaciar carrito'.</li> <li>2.El sistema elimina del carrito todos los productos que se encontraban almacenados.</li> <li>3.El botón Vaciar carrito se desactiva.</li> <li>4.Caso de Uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.10: Especificación CU-03: Vaciar carrito.

<b>CU-04</b>	<b>Comenzar proceso de compra</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Manual.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El sistema comienza el proceso de compra para registrar el pedido en la base de datos y realizar el pago.
<b>Disparador</b>	El usuario pulsa el botón de compra
<b>Precondiciones</b>	Deben existir productos añadidos al carrito
<b>Postcondiciones</b>	El pedido ha sido registrado en la BBDD y el pedido realizado correctamente.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón comprar.</li> <li>2. El sistema pide al usuario los datos necesarios.</li> <li>3. El usuario completa el formulario aportando los datos necesarios (Nombre Pedido, Tarjeta de crédito o débito, email).</li> <li>4. El sistema comprueba la validez de los datos introducidos.</li> <li>5. El usuario compra el pedido.</li> <li>6. El sistema muestra un mensaje de éxito y registra el pedido en la base de datos.</li> <li>7. Caso de Uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	7a. El sistema muestra un mensaje de error y cancela el proceso de compra.
<b>Excepciones</b>	4. Los datos introducidos son incorrectos. El sistema muestra un mensaje de error y vuelve al paso 2.

Tabla 3.11: Especificación CU-04: Comenzar proceso de compra.

<b>CU-05</b>	<b>Mostrar los productos en oferta del catálogo.</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Manual.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El sistema muestra los productos en oferta del catálogo
<b>Disparador</b>	El usuario pulsa el botón switch de ofertas
<b>Precondiciones</b>	El botón debe estar desactivado
<b>Postcondiciones</b>	El sistema muestra solo las ofertas y activa el botón switch.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón switch de ofertas.</li> <li>2. El sistema muestra únicamente las ofertas.</li> <li>3. El sistema activa el botón switch.</li> <li>4. Caso de Uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.12: Especificación CU-05: Mostrar los productos en oferta del catálogo.

<b>CU-06</b>	<b>Mostrar todo el catálogo de productos</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Manual.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El sistema muestra el catálogo de productos y desactiva el botón switch de ofertas.
<b>Disparador</b>	El usuario pulsa el botón switch de ofertas
<b>Precondiciones</b>	El botón switch de ofertas debe estar activado.
<b>Postcondiciones</b>	El sistema muestra todo el catálogo de productos y desactiva el botón switch de ofertas.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón switch de ofertas y.</li> <li>2. El sistema desactiva el botón switch de ofertas.</li> <li>3. El sistema muestra todos los productos del catálogo.</li> <li>4. Caso de Uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.13: Especificación CU-06: Mostrar todo el catálogo de productos.

<b>CU-07</b>	<b>Ir a la Tienda</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Manual.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El sistema redirige al usuario a la página de la Tienda.
<b>Disparador</b>	El usuario pulsa el botón Tienda
<b>Precondiciones</b>	El usuario debe encontrarse en la página historial de pedidos.
<b>Postcondiciones</b>	El usuario se encontrará en la Tienda.
<b>Flujo Normal</b>	1. El usuario pulsa el botón Tienda. 2. El sistema redirige al usuario a la Tienda. 3. Caso de Uso finalizado correctamente.
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.14: Especificación CU-07: Ir a la Tienda.

<b>CU-08</b>	<b>Ir al Historial de Pedidos</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Manual.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El sistema redirige al usuario a la página de historial de pedidos.
<b>Disparador</b>	El usuario pulsa el botón Historial de Pedidos.
<b>Precondiciones</b>	El usuario debe encontrarse en la Tienda.
<b>Postcondiciones</b>	El usuario se encontrará en el Historial de Pedidos.
<b>Flujo Normal</b>	1. El usuario pulsa el botón Historial de Pedidos. 2. El sistema redirige al usuario al Historial de pedidos. 3. Caso de Uso finalizado satisfactoriamente.
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.15: Especificación CU-08: Ir al Historial de Pedidos.

<b>CU-09</b>	<b>Mostrar Tutorial.</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Manual.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El sistema muestra la ventana con los comandos posibles.
<b>Disparador</b>	El usuario pulsa el botón ver comandos
<b>Precondiciones</b>	Ninguna.
<b>Postcondiciones</b>	El sistema muestra una ventana con los comandos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón Ver comandos.</li> <li>2. El sistema muestra la ventana de comandos.</li> <li>3. Caso de Uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.16: Especificación CU-09: Mostrar Tutorial.

<b>CU-10</b>	<b>Cerrar tutorial</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Manual.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El sistema cierra la ventana de comandos.
<b>Disparador</b>	El usuario pulsa en el botón X o fuera de la ventana de comandos.
<b>Precondiciones</b>	Ninguna.
<b>Postcondiciones</b>	El sistema cierra la ventana de comandos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón X o pulsa fuera de la ventana de comandos.</li> <li>2. El sistema cierra la ventana de comandos.</li> <li>3. Caso de Uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.17: Especificación CU-10: Cerrar tutorial.

<b>CU-11</b>	<b>Saber qué usuario está conectado.</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pregunta al sistema qué usuario está conectado y el sistema le responde con su nombre de usuario.
<b>Disparador</b>	El usuario pregunta al sistema por el usuario conectado.
<b>Precondiciones</b>	El usuario debe haber iniciado sesión previamente.
<b>Postcondiciones</b>	El sistema dicta en voz alta el nombre con el que el usuario ha iniciado sesión.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema responde con el nombre de usuario.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.18: Especificación CU-11: Saber qué usuario está conectado.

<b>CU-12</b>	<b>Saber los productos del catálogo.</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pregunta por los productos que se encuentran en el catálogo y el sistema responde.
<b>Disparador</b>	El usuario pregunta por los productos del catálogo.
<b>Precondiciones</b>	El usuario debe encontrarse en el catálogo de productos.
<b>Postcondiciones</b>	El sistema dicta en voz alta los productos del catálogo.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema responde con la lista de productos del catálogo.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.19: Especificación CU-12: Saber los productos del catálogo.

<b>CU-13</b>	<b>Saber el precio de un producto</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pregunta por el precio de un producto y el sistema le responde
<b>Disparador</b>	El usuario pregunta al sistema por el precio de un producto concreto
<b>Precondiciones</b>	El usuario debe estar en la Tienda.
<b>Postcondiciones</b>	El sistema dicta en voz alto el precio del producto dicho por el usuario.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema responde con el precio del producto.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	2a. El producto no se encuentra en el catálogo.
<b>Excepciones</b>	–

Tabla 3.20: Especificación CU-13: Saber el precio de un producto.

<b>CU-14</b>	<b>Añadir un producto al carrito.</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario dice el producto que quiere añadir y la cantidad y el sistema lo añade al carrito.
<b>Disparador</b>	El usuario dice que quiere añadir una cantidad de productos y el nombre de un producto.
<b>Precondiciones</b>	El usuario debe encontrarse en la Tienda.
<b>Postcondiciones</b>	El producto ha sido añadido al carrito.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema añade la cantidad de productos al carrito.</li> <li>4. El sistema responde diciendo que ha sido añadido correctamente.</li> <li>5. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	2a. El producto no se encuentra en el catálogo.
<b>Excepciones</b>	–

Tabla 3.21: Especificación CU-14: Añadir un producto al carrito.

<b>CU-15</b>	<b>Saber el dinero acumulado en el carrito.</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pregunta por el dinero acumulado en el carrito y el sistema le responde.
<b>Disparador</b>	El usuario pregunta por el dinero acumulado en el carrito.
<b>Precondiciones</b>	El usuario debe encontrarse en la Tienda.
<b>Postcondiciones</b>	El sistema responde la cantidad de dinero acumulada en el carrito.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema responde con el precio acumulado en el carrito.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.22: Especificación CU-15: Saber el dinero acumulado en el carrito.

<b>CU-16</b>	<b>Saber si un producto se encuentra en el catálogo.</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pregunta por la existencia en el catálogo de un producto y el sistema le responde si está o no está.
<b>Disparador</b>	El usuario pregunta por la existencia de un producto en el catálogo.
<b>Precondiciones</b>	El usuario debe encontrarse en el catálogo de productos.
<b>Postcondiciones</b>	El sistema responde si existe o no dicho producto.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema responde que el producto se encuentra en el catálogo.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	3a. El sistema responde que el producto no se encuentra en el catálogo.

Tabla 3.23: Especificación CU-16: Saber si un producto se encuentra en el catálogo.

<b>17</b>	<b>Saber el número de productos del carrito</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pregunta por el número de productos del carrito y el sistema le responde.
<b>Disparador</b>	El usuario pregunta por el número de productos del carrito.
<b>Precondiciones</b>	El usuario debe encontrarse en la Tienda.
<b>Postcondiciones</b>	El sistema le responde con el número de productos del carrito.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema responde con el número de productos del carrito.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	3a. El sistema responde que el carrito está vacío si no se encuentra ningún producto añadido.
<b>Excepciones</b>	–

Tabla 3.24: Especificación CU-17: Saber el número de productos del carrito.

<b>CU-18</b>	<b>Saber el nombre de los productos que hay en el carrito</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pregunta por los productos que hay en el carrito y el sistema le responde.
<b>Disparador</b>	El usuario pregunta por los productos que hay en el carrito
<b>Precondiciones</b>	El usuario debe encontrarse en la Tienda.
<b>Postcondiciones</b>	El sistema responde con una lista de los nombres de los productos que hay en el carrito.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema responde con la lista de los nombres de los productos que hay en el carrito.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	3a. El sistema responde que el carrito está vacío, si no se encuentra ningún producto añadido.
<b>Excepciones</b>	–

Tabla 3.25: Especificación CU-18: Saber el nombre de los productos que hay en el carrito.

<b>CU-19</b>	<b>Eliminar un producto concreto del carrito.</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario dice que quiere eliminar un producto del carrito y el sistema lo elimina.
<b>Disparador</b>	El usuario dice que quiere eliminar un producto en concreto.
<b>Precondiciones</b>	El usuario debe encontrarse en la Tienda.
<b>Postcondiciones</b>	El sistema elimina el producto dicho por el usuario del carrito.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema elimina el producto del carrito.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	3a. El sistema no encuentra el producto en el carrito y dice que no se ha encontrado.
<b>Excepciones</b>	–

Tabla 3.26: Especificación CU-19: Eliminar un producto concreto del carrito.

<b>CU-20</b>	<b>Vaciar el carrito</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario dice que quiere vaciar el carrito y el sistema lo vacía
<b>Disparador</b>	El usuario dice que quiere vaciar el carrito.
<b>Precondiciones</b>	El usuario debe encontrarse en la Tienda.
<b>Postcondiciones</b>	El sistema vacía el carrito.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema Vacía el carrito.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	3a. El sistema dice que el carrito ya está vacío.
<b>Excepciones</b>	–

Tabla 3.27: Especificación CU-20: Vaciar el carrito.

<b>CU-21</b>	<b>Comenzar proceso de compra.</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario dice que quiere comenzar el proceso de compra y el sistema comienza el proceso de compra.
<b>Disparador</b>	El usuario pide pagar la compra.
<b>Precondiciones</b>	El usuario debe encontrarse en la Tienda.
<b>Postcondiciones</b>	El sistema realiza el pedido y lo guarda en la base de datos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema pregunta por los datos correspondientes del pedido (email, nombre del pedido).</li> <li>4. El sistema registra las palabras del usuario y recopila la información.</li> <li>5. El sistema pregunta al usuario si los datos son correctos.</li> <li>6. El usuario responde Con SI o NO.</li> <li>7. El sistema registra las palabras del usuario y recopila la información.</li> <li>8. El sistema registra el pedido en la base de datos.</li> <li>9. El sistema Vacía el carrito.</li> <li>10. El sistema dicta en voz alta que el proceso de compra ha finalizado correctamente.</li> <li>11. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	<ol style="list-style-type: none"> <li>2a. El carrito está vacío por lo que no puede comenzar la compra</li> <li>2b. El sistema dice que el carrito está vacío y no puede completarse la compra.</li> <li>7a. El sistema recibe un NO del usuario y pregunta si desea repetir los datos. <ol style="list-style-type: none"> <li>7a.1. El sistema repite el proceso desde el paso 3</li> <li>7a.2. El sistema finaliza el proceso de compra.</li> </ol> </li> </ol>
<b>Excepciones</b>	–

Tabla 3.28: Especificación CU-21: Comenzar proceso de compra.

<b>CU-22</b>	<b>Saber los productos en oferta del catálogo</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario dice que quiere escuchar el nombre de los productos en oferta del catálogo y el sistema se los dice.
<b>Disparador</b>	El usuario pregunta cuales son los productos en oferta.
<b>Precondiciones</b>	El usuario debe encontrarse en el catálogo de ofertas.
<b>Postcondiciones</b>	El sistema dice una lista con los nombres de los productos en oferta.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema dicta el nombre de los productos en oferta del catálogo.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.29: Especificación CU-22: Saber los productos en oferta del catálogo.

<b>CU-23</b>	<b>Mostrar los productos en oferta del catálogo</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario dice que quiere ver los productos en oferta del catálogo y el sistema los muestra.
<b>Disparador</b>	El usuario dice que quiere ver los productos en oferta.
<b>Precondiciones</b>	El usuario debe encontrarse en el Catálogo de productos.
<b>Postcondiciones</b>	El sistema muestra los productos en oferta
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema muestra los productos en oferta del catálogo.</li> <li>4. El sistema activa el botón switch de ofertas.</li> <li>5. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.30: Especificación CU-23: Mostrar los productos en oferta del catálogo.

<b>CU-24</b>	<b>Mostrar todo el catálogo de productos</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pide ver el catálogo de productos y el sistema lo muestra.
<b>Disparador</b>	El usuario pide ver el catálogo de productos.
<b>Precondiciones</b>	El usuario debe encontrarse en el catálogo de ofertas.
<b>Postcondiciones</b>	El sistema mostrará el catálogo de productos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema muestra los todos los productos del catálogo.</li> <li>4. El sistema desactiva el botón switch de ofertas.</li> <li>5. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.31: Especificación CU-24: Mostrar todo el catálogo de productos.

<b>CU-25</b>	<b>Moverse a la Tienda</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pide moverse a la página de la Tienda y el sistema le redirige a la página.
<b>Disparador</b>	El usuario dice que quiere ir a la Tienda
<b>Precondiciones</b>	El usuario debe encontrarse en el Historial de pedidos.
<b>Postcondiciones</b>	El sistema redirige al usuario a la Tienda
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema redirige al usuario a la Tienda y se lo comunica en voz alta.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.32: Especificación CU-25: Moverse a la Tienda.

<b>CU-26</b>	<b>Moverse al Historial de Pedidos</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pide ir al Historial de Pedidos y el sistema le redirige.
<b>Disparador</b>	El usuario pide ir al Historial de Pedidos.
<b>Precondiciones</b>	El usuario debe encontrarse en la Tienda.
<b>Postcondiciones</b>	El sistema redirige al usuario al Historial de Pedidos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema redirige al usuario al Historial de productos y se lo comunica en voz alta.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.33: Especificación CU-26: Moverse al Historial de Pedidos.

<b>CU-27</b>	<b>Saber el nombre de los pedidos del historial</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario quiere saber el nombre de los pedidos del historial y el sistema se lo dice.
<b>Disparador</b>	El usuario pide saber el nombre de los pedidos del historial.
<b>Precondiciones</b>	El usuario debe encontrarse en el historial de productos.
<b>Postcondiciones</b>	El sistema dicta en voz alta la lista con el nombre de los pedidos que hay en el historial.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema dicta en voz alta la lista de nombres de los pedidos que hay en el historial de productos.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	3a. El sistema dicta en voz alta que no se encuentra ningún pedido en el historial.
<b>Excepciones</b>	–

Tabla 3.34: Especificación CU-27: Saber el nombre de los pedidos del historial.

<b>CU-28</b>	<b>Saber los datos de un pedido en concreto</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario quiere saber los datos de un pedido concreto y el sistema se lo dice.
<b>Disparador</b>	El usuario quiere saber los datos de un pedido concreto.
<b>Precondiciones</b>	El usuario debe encontrarse en el historial de producto.
<b>Postcondiciones</b>	El sistema dicta al usuario los datos del pedido.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema dicta en voz alta los datos del pedido.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	3a. El sistema dicta en voz alta que el pedido dicho no existe en el historial.
<b>Excepciones</b>	–

Tabla 3.35: Especificación CU-28: Saber los datos de un pedido en concreto.

<b>CU-29</b>	<b>Saber los productos de un pedido concreto</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pregunta por los productos de un pedido concreto y el sistema se lo dice.
<b>Disparador</b>	El usuario pregunta por los productos de un pedido concreto.
<b>Precondiciones</b>	El usuario debe encontrarse en el historial de productos.
<b>Postcondiciones</b>	El sistema dicta los productos del pedido.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema dicta en voz alta los productos del pedido.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	3a. El sistema dicta en voz alta que el pedido dicho no existe en el historial.
<b>Excepciones</b>	–

Tabla 3.36: Especificación CU-29: Saber los productos de un pedido concreto.

<b>CU-30</b>	<b>Saber la página en la que se está</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pide saber la página en la que se encuentra actualmente y el sistema se lo dice.
<b>Disparador</b>	El usuario pregunta en qué página está.
<b>Precondiciones</b>	Ninguna.
<b>Postcondiciones</b>	El sistema responde en qué página se encuentra.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema dicta en voz alta la página en la que se encuentra el usuario.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.37: Especificación CU-30: Saber la página en la que se está.

<b>CU-31</b>	<b>Ver modal tutorial</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pide ver la ventana de los comandos y el sistema la muestra.
<b>Disparador</b>	El usuario pide ver la ventana de comandos.
<b>Precondiciones</b>	Ninguna.
<b>Postcondiciones</b>	El sistema muestra la ventana de comandos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema muestra la ventana de comandos.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.38: Especificación CU-31: Ver modal tutorial.

<b>CU-32</b>	<b>Cerrar modal</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pide cerrar la ventana de comandos y el sistema la cierra.
<b>Disparador</b>	El usuario pide cerra la ventana de comandos.
<b>Precondiciones</b>	La ventana de comandos debe estar abierta.
<b>Postcondiciones</b>	El sistema cierra la ventana de comandos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema cierra la ventana de comandos.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.39: Especificación CU-32: Cerrar modal.

<b>CU-33</b>	<b>Dictar comandos</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario pide saber los comandos y el sistema se los dicta.
<b>Disparador</b>	El usuario pide saber los comandos.
<b>Precondiciones</b>	Ninguna.
<b>Postcondiciones</b>	El sistema dicta todos los comandos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema dicta todos los comandos.</li> <li>4. Caso de uso finalizado satisfactoriamente.</li> </ol>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.40: Especificación CU-33: Dictar comandos.

<b>CU-34</b>	<b>Saludar</b>
<b>Versión</b>	1.0 (16/09/2021).
<b>Actor Principal</b>	Usuario Voz.
<b>Actor Secundario</b>	–
<b>Descripción</b>	El usuario dice Hola y el sistema le saluda.
<b>Disparador</b>	El usuario dice Hola.
<b>Precondiciones</b>	Ninguna.
<b>Postcondiciones</b>	El sistema saluda al usuario.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. El usuario dice el comando en voz alta.</li> <li>2. El sistema registra las palabras del usuario y recopila la información.</li> <li>3. El sistema saluda y pregunta por el nombre del usuario.</li> <li>4. El sistema registra las palabras del usuario y recopila la información.</li> <li>5. El sistema da la bienvenida al usuario hablándole por su nombre.</li> </ol> <p>4. Caso de uso finalizado satisfactoriamente.</p>
<b>Flujo Alternativo</b>	–
<b>Excepciones</b>	–

Tabla 3.41: Especificación CU-34: Saludar.

### 3.5. Requisitos Funcionales

En esta sección se van a tratar los diferentes requisitos funcionales asociados a los requisitos de usuario mencionados anteriormente. Un requisito funcional sirve para definir una función del sistema bien sea software o de alguno de sus componentes. Por este motivo, los requisitos funcionales es un apartado a considerar ya que nos indican cómo se ha de comportarse el sistema acorde a los requisitos descritos. A continuación se mencionan nuevamente los requisitos de usuario del sistema añadiendo a estos sus requisitos funcionales asociados.

ID	Nombre
<b>RF-01</b>	El sistema comprueba si un producto ha sido seleccionado por el usuario
<b>RF-02</b>	El sistema muestra al usuario el producto seleccionado en la lista del carrito.
<b>RF-03</b>	El sistema actualiza el precio total del carrito

Tabla 3.42: Requisitos Funcionales asociados al Caso de Uso CU-01: Añadir un producto al carrito.

ID	Nombre
<b>RF-04</b>	El sistema comprueba si el botón para eliminar un producto ha sido seleccionado.
<b>RF-05</b>	El sistema elimina el producto del carrito.
<b>RF-03</b>	El sistema actualiza el precio total del carrito.

Tabla 3.43: Requisitos Funcionales asociados al Caso de Uso CU-02: Eliminar un producto al carrito.

ID	Nombre
<b>RF-06</b>	El sistema comprueba si se ha pulsado el botón para vaciar el carrito.
<b>RF-07</b>	El sistema elimina todos los productos de la lista del carrito.
<b>RF-03</b>	El sistema actualiza el precio total del carrito.

Tabla 3.44: Requisitos Funcionales asociados al Caso de Uso CU-03: Vaciar Carrito.

ID	Nombre
<b>RF-08</b>	El sistema comprueba si se ha pulsado el botón para comprar.
<b>RF-09</b>	El sistema guarda la información del carrito en Local Storage
<b>RF-10</b>	El sistema guarda los datos correspondientes con el pedido (Número de tarjeta, Email, Nombre de pedido).
<b>RF-11</b>	El sistema crea en la base de datos un nuevo pedido.
<b>RF-12</b>	El sistema muestra un mensaje al usuario mostrándole el resultado de la operación.
<b>RF-13</b>	El sistema elimina del Local Storage la información.
<b>RF-14</b>	El sistema elimina todos los productos del carrito.

Tabla 3.45: Requisitos Funcionales asociados al Caso de Uso CU-04: Comenzar proceso de compra.

ID	Nombre
<b>RF-15</b>	El sistema comprueba si se ha pulsado el botón para mostrar ofertas, activándolo.
<b>RF-16</b>	El sistema muestra solo las ofertas.
<b>RF-17</b>	El sistema cambia el estado del botón switch de las ofertas a activado.

Tabla 3.46: Requisitos Funcionales asociados al Caso de Uso CU-05: Mostrar los productos en oferta del catálogo.

ID	Nombre
<b>RF-18</b>	El sistema comprueba si se ha pulsado el botón para mostrar ofertas, desactivándolo
<b>RF-19</b>	El sistema muestra el catálogo de productos
<b>RF-20</b>	El sistema cambia el estado del botón switch de las ofertas a desactivado

Tabla 3.47: Requisitos Funcionales asociados al Caso de Uso CU-06: Mostrar todo el catálogo de productos.

ID	Nombre
<b>RF-21</b>	El sistema comprueba si se ha pulsado el botón para ir a la tienda
<b>RF-22</b>	El sistema redirige al usuario a la página Tienda
<b>RF-23</b>	El sistema muestra la página Tienda

Tabla 3.48: Requisitos Funcionales asociados al Caso de Uso CU-07: Ir a la Tienda.

ID	Nombre
<b>RF-24</b>	El sistema comprueba si se ha pulsado el botón Historial de Pedidos
<b>RF-25</b>	El sistema redirige al usuario a la página Historial de Pedidos
<b>RF-26</b>	El sistema muestra la página Historial de Pedidos.

Tabla 3.49: Requisitos Funcionales asociados al Caso de Uso CU-08: Ir al Historial de productos.

ID	Nombre
<b>RF-27</b>	El sistema comprueba si se ha pulsado el botón ver comandos
<b>RF-28</b>	El sistema muestra la pantalla donde se encuentran todos los comandos

Tabla 3.50: Requisitos Funcionales asociados al Caso de Uso CU-09: Mostrar Tutorial.

ID	Nombre
<b>RF-29</b>	El sistema comprueba si se ha pulsado fuera de la ventana de comandos o ha pulsado el botón X
<b>RF-30</b>	El sistema cierra la ventana de comandos

Tabla 3.51: Requisitos Funcionales asociados al Caso de Uso CU-10: Cerrar tutorial.

ID	Nombre
<b>RF-31</b>	El sistema escucha el comando '¿Qué usuario está conectado?'
<b>RF-32</b>	El sistema registra el nombre del usuario conectado
<b>RF-33</b>	El sistema dicta en voz alta el nombre del usuario conectado

Tabla 3.52: Requisitos Funcionales asociados al Caso de Uso CU-11: Saber qué usuario está conectado.

ID	Nombre
<b>RF-34</b>	El sistema escucha el comando '¿En qué página estoy?'
<b>RF-35</b>	El sistema registra la página actual en la que se encuentra el usuario
<b>RF-36</b>	El sistema dicta en voz alta la página actual

Tabla 3.53: Requisitos Funcionales asociados al Caso de Uso CU-12: Saber los productos del catálogo.

ID	Nombre
<b>RF-37</b>	El sistema escucha el comando '¿Cuánto cuesta ¡articulo¿¡producto¿?'
<b>RF-38</b>	El sistema registra el producto dictado por el usuario y busca el precio del producto
<b>RF-39</b>	El sistema dicta en voz alta el precio del producto

Tabla 3.54: Requisitos Funcionales asociados al Caso de Uso CU-13: Saber el precio de un producto.

ID	Nombre
<b>RF-40</b>	El sistema escucha el comando 'Quiero añadir ¡numero¿¡producto¿al carrito'
<b>RF-02</b>	El sistema muestra al usuario el producto seleccionado en la lista del carrito.
<b>RF-03</b>	El sistema actualiza el precio total del carrito
<b>RF-41</b>	El sistema dicta por voz si el producto ha sido añadido correctamente

Tabla 3.55: Requisitos Funcionales asociados al Caso de Uso CU-14 : Añadir un producto al carrito.

ID	Nombre
<b>RF-42</b>	El sistema escucha el comando '¿Cuánto dinero tengo en el carrito?'
<b>RF-43</b>	El sistema registra el dinero acumulado en el carrito
<b>RF-44</b>	El sistema dicta en voz alta el dinero acumulado en el carrito

Tabla 3.56: Requisitos Funcionales asociados al Caso de Uso CU-15 : Saber el dinero acumulado en el carrito.

ID	Nombre
<b>RF-45</b>	El sistema escucha el comando '¿Hay ¡producto¿en el catálogo?'
<b>RF-46</b>	El sistema busca el producto en el catálogo
<b>RF-47</b>	El sistema dicta en voz alta si se ha encontrado el producto o no

Tabla 3.57: Requisitos Funcionales asociados al Caso de Uso CU-16 : Saber si un producto se encuentra en el catálogo.

ID	Nombre
<b>RF-48</b>	El sistema escucha el comando '¿Cuántos productos hay en el carrito?'
<b>RF-49</b>	El sistema calcula el número de productos que hay en el carrito.
<b>RF-50</b>	El sistema dicta en voz alta el número de productos del carrito.

Tabla 3.58: Requisitos Funcionales asociados al Caso de Uso CU-17 : Saber el número de productos del carrito.

ID	Nombre
<b>RF-51</b>	El sistema escucha el comando '¿Qué productos hay en el carrito?'
<b>RF-52</b>	El sistema registra todos los productos que hay en el carrito
<b>RF-53</b>	El sistema dicta en voz alta el nombre de los productos del carrito

Tabla 3.59: Requisitos Funcionales asociados al Caso de Uso CU-18 : Saber el nombre de los productos que hay en el carrito.

ID	Nombre
<b>RF-54</b>	El sistema escucha el comando 'Quiero eliminar el producto ¡producto¿del carrito.'
<b>RF-05</b>	El sistema elimina el producto del carrito
<b>RF-03</b>	El sistema actualiza el precio total del carrito.
<b>RF-55</b>	El sistema dicta por voz que el producto ha sido eliminado correctamente.

Tabla 3.60: Requisitos Funcionales asociados al Caso de Uso CU-19 : Eliminar un producto concreto del carrito.

ID	Nombre
<b>RF-56</b>	El sistema escucha el comando 'Quiero vaciar el carrito.'
<b>RF-08</b>	El sistema elimina todos los productos de la lista del carrito
<b>RF-03</b>	El sistema actualiza el precio total del carrito.
<b>RF-57</b>	El sistema dicta en voz alta que se ha vaciado correctamente el carrito.

Tabla 3.61: Requisitos Funcionales asociados al Caso de Uso CU-20 : Vaciar carrito.

ID	Nombre
<b>RF-58</b>	El sistema escucha el comando 'Quiero pagar mi compra'
<b>RF-11</b>	El sistema guarda la información del carrito en Local Storage
<b>RF-59</b>	El sistema escucha los datos del pedido (Email, Nombre de pedido)
<b>RF-13</b>	El sistema crea en la base de datos un nuevo pedido.
<b>RF-15</b>	El sistema elimina del Local Storage la información
<b>RF-60</b>	El sistema dicta en voz alta que el producto ha sido realizado correctamente.
<b>RF-16</b>	El sistema elimina todos los productos del carrito.

Tabla 3.62: Requisitos Funcionales asociados al Caso de Uso CU-21 : Comenzar proceso de compra.

ID	Nombre
<b>RF-61</b>	El sistema escucha el comando '¿Cuáles son los productos en oferta del catálogo?'
<b>RF-62</b>	El sistema registra los nombres de todos los productos en oferta.
<b>RF-63</b>	El sistema dicta en voz alta los nombres de todos los productos en oferta.

Tabla 3.63: Requisitos Funcionales asociados al Caso de Uso CU-22 : Saber los productos en oferta del catálogo.

ID	Nombre
<b>RF-64</b>	El sistema escucha el comando 'Quiero ver los productos en oferta del catálogo'
<b>RF-18</b>	El sistema muestra solo las ofertas
<b>RF-19</b>	El sistema cambia el estado del botón switch de las ofertas a activado.
<b>RF-65</b>	El sistema dicta en voz alta que se están mostrando correctamente las ofertas.

Tabla 3.64: Requisitos Funcionales asociados al Caso de Uso CU-23 : Mostrar los productos en oferta del catálogo.

ID	Nombre
<b>RF-66</b>	El sistema escucha el comando 'Quiero ver todos los productos del catálogo.'
<b>RF-21</b>	El sistema muestra el catálogo de productos.
<b>RF-22</b>	El sistema muestra el botón switch de las ofertas a desactivado.
<b>RF-67</b>	El sistema dicta en voz alta que se muestra el catálogo de productos.

Tabla 3.65: Requisitos Funcionales asociados al Caso de Uso CU-24 : Mostrar todo el catálogo de productos.

ID	Nombre
<b>RF-68</b>	El sistema escucha el comando 'Quiero ir a la tienda.'
<b>RF-24</b>	El sistema redirige al usuario a la página Tienda
<b>RF-25</b>	El sistema muestra la página Tienda
<b>RF-69</b>	El sistema dicta en voz alta que se ha redireccionado correctamente a la Tienda

Tabla 3.66: Requisitos Funcionales asociados al Caso de Uso CU-25 : Moverse a la Tienda.

ID	Nombre
<b>RF-70</b>	El sistema escucha el comando 'Quiero ir al historial de pedidos.'
<b>RF-24</b>	El sistema redirige al usuario a la página Historial de Pedidos
<b>RF-25</b>	El sistema muestra la página Historial de Pedidos
<b>RF-71</b>	El sistema dicta en voz alta que se ha redireccionado correctamente al historial de pedidos.

Tabla 3.67: Requisitos Funcionales asociados al Caso de Uso CU-26 : Moverse al historial de pedidos.

ID	Nombre
<b>RF-72</b>	El sistema escucha el comando '¿Cuáles son los pedidos del historial?.'
<b>RF-73</b>	El sistema registra el nombre de todos los pedidos del historial
<b>RF-74</b>	El sistema dicta en voz alta los nombres de los pedidos del historial

Tabla 3.68: Requisitos Funcionales asociados al Caso de Uso CU-27 : Saber el nombre de los pedidos del historial.

ID	Nombre
<b>RF-75</b>	El sistema escucha el comando '¿Cuáles son los datos del pedido Nombre Pedido?.'
<b>RF-76</b>	El sistema busca y registra los datos del pedido.
<b>RF-77</b>	El sistema dicta en voz alta los datos del pedido.

Tabla 3.69: Requisitos Funcionales asociados al Caso de Uso CU-28 : Saber los datos de un pedido concreto.

ID	Nombre
<b>RF-78</b>	El sistema escucha el comando '¿Cuáles son los productos del pedido Nombre Pedido?.'
<b>RF-79</b>	El sistema busca y registra los datos de los productos que hay en el pedido.
<b>RF-80</b>	El sistema dicta en voz alta los datos de los productos que hay en el pedido.

Tabla 3.70: Requisitos Funcionales asociados al Caso de Uso CU-29 : Saber los productos de un pedido.

ID	Nombre
<b>RF-81</b>	El sistema escucha el comando '¿En qué página estoy?.'
<b>RF-82</b>	El sistema registra la página en la que está el usuario.
<b>RF-83</b>	El sistema dicta en voz alta la página en la que está el usuario.

Tabla 3.71: Requisitos Funcionales asociados al Caso de Uso CU-30 : Saber la página en la que se está.

ID	Nombre
<b>RF-84</b>	El sistema escucha el comando 'Quiero escuchar los comandos.'
<b>RF-85</b>	El sistema muestra la pantalla donde se encuentran todos los comandos
<b>RF-86</b>	El sistema dicta en voz alta que se están viendo los comandos disponibles.

Tabla 3.72: Requisitos Funcionales asociados al Caso de Uso CU-31 : Ver modal tutorial.

ID	Nombre
<b>RF-87</b>	El sistema escucha el comando 'Quiero cerrar la ventana de comandos.'
<b>RF-88</b>	El sistema cierra la ventana de comandos
<b>RF-89</b>	El sistema dicta en voz alta que la ventana de productos ha sido cerrada.

Tabla 3.73: Requisitos Funcionales asociados al Caso de Uso CU-32 : Cerrar modal tutorial.

ID	Nombre
<b>RF-90</b>	El sistema escucha el comando '¿Cuáles son los comandos posibles?'
<b>RF-91</b>	El sistema registra todos los comandos posibles.
<b>RF-92</b>	El sistema dicta en voz alta todos los comandos posibles.

Tabla 3.74: Requisitos Funcionales asociados al Caso de Uso CU-33 : Dictar comandos.

ID	Nombre
<b>RF-93</b>	El sistema escucha el comando 'Hola.'
<b>RF-94</b>	El sistema responde en voz alta saludando y pregunta por su nombre
<b>RF-95</b>	El sistema da la bienvenida en voz alta.

Tabla 3.75: Requisitos Funcionales asociados al Caso de Uso CU-34 : Saludar.

## 3.6. Requisitos de Información

[2] [9] En este apartado se detallará el análisis de los datos utilizados en la aplicación. En este caso la elicitación de requisitos de información es un poco diferente debido a que la información utilizada y generada por la aplicación no es guardada en una estructura de base de datos habitual Relacional. En este caso los datos se guardan en una base de datos no relacional o no SQL. Las bases de datos no relacionales son un sistema de almacenamiento de información que se caracteriza por no usar el lenguaje SQL para las consultas. Esto no quiere decir que no se utilice el lenguaje SQL, pero no es utilizado para realizar consultas sino como ayuda. La otra gran diferencia es que no almacenan los datos en tablas, registros ni campos. El motivo para utilizar este tipo de bases de datos es su gran escalabilidad y flexibilidad ante los cambios en la definición de los datos almacenados.

A continuación se puede ver los objetos que forman la BBDD en formato JSON, lenguaje usualmente utilizado en este tipo de bases de datos para insertar y extraer datos.

```
items{
  "_id": "Id",
  "title": "string",
  "price": "Number",
  "image": "string",
  "offer": "boolean"
}

orders{
  "_id": "Id",
  "nombre": "string",
  "email": "string",
  "fecha": "Date",
  "cantProductosTotal": "Number",
  "priceTotal": "Number",
  "items": [
    "id": "ref(items{_id})": "Id",
    "qty": "Number"
  ]
}
```

[9] Si queremos ilustrar visualmente la base de datos podemos utilizar un grafo dirigido etiquetado en el que tenemos 2 tipos de nodos. Los nodos ovalados y circulares hacen referencia a objetos dentro de la base de datos, mientras que los nodos rectangulares hacen referencia a los atributos que forman los objetos. El nodo circular hace referencia a un Array que almacena 2 atributos: un dato numérico y un id de los items, haciendo referencia al objeto items. En las etiquetas se describe el tipo de dato que se almacena. Todas estas particularidades pueden verse en la Figura 3.8

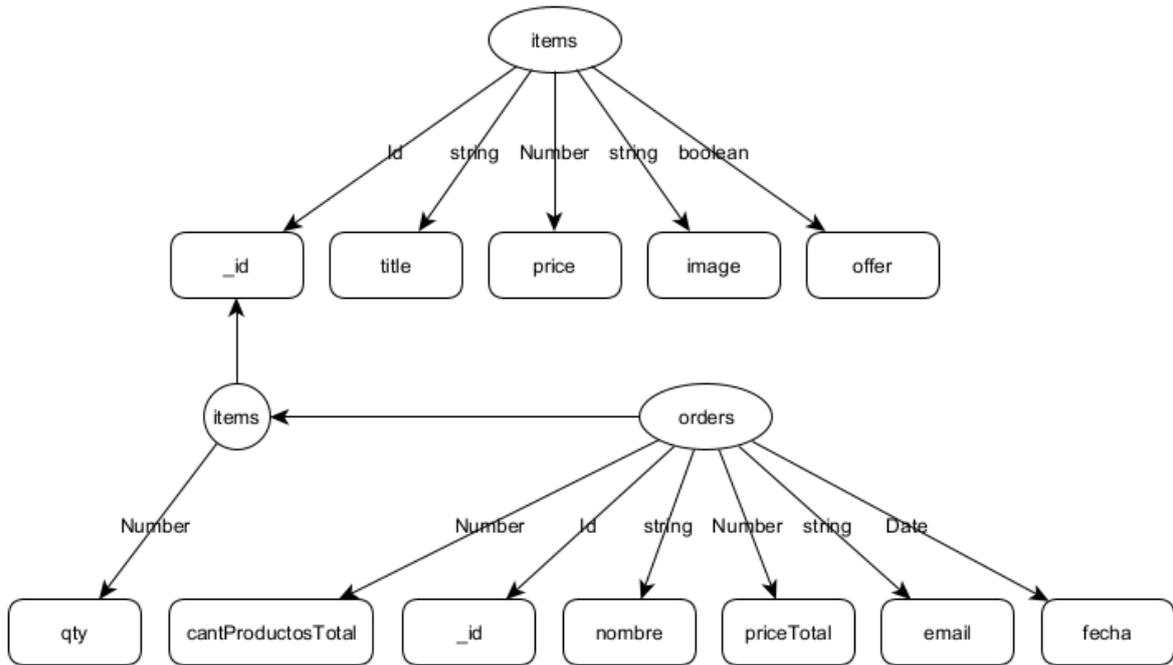


Figura 3.8: Diagrama BBDD No SQL

### 3.7. Requisitos No Funcionales

Los requisitos no funcionales son aquellos que definen ciertas restricciones del sistema que tienen que ser cumplidas, relativas a la seguridad, robustez, etc. Por lo tanto especifican los criterios que se deben cumplir para que sea adecuado el uso de la aplicación.

<b>Seguridad</b>	
<b>ID</b>	<b>Descripción</b>
<b>RNF-01</b>	Los datos pertenecientes a la tarjeta de crédito o débito no deberá guardarse en la base de datos.
<b>RNF-02</b>	Los permisos del micrófono únicamente tendrán el propósito de poder manejar la aplicación y la información delicada no será guardado ni registrado.

Tabla 3.76: Requisitos No Funcionales: Seguridad

<b>Usabilidad</b>	
<b>ID</b>	<b>Descripción</b>
<b>RNF-03</b>	La aplicación deberá ser intuitiva y fácil de usar.
<b>RNF-04</b>	La aplicación deberá poder ser usada íntegramente con punteros manuales.
<b>RNF-05</b>	La aplicación deberá poder ser usada íntegramente con la voz.
<b>RNF-06</b>	La aplicación deberá estar disponible en castellano, al igual que el asistente debe comprender ese idioma.
<b>RNF-07</b>	La aplicación únicamente trabajará con el Euro como unidad monetaria.

Tabla 3.77: Requisitos No Funcionales: Usabilidad

<b>Robustez</b>	
<b>ID</b>	<b>Descripción</b>
<b>RNF-08</b>	El sistema deberá estar preparado para afrontar anomalías.
<b>RNF-09</b>	El sistema mostrará el feedback necesario al usuario durante su experiencia utilizando la aplicación.

Tabla 3.78: Requisitos No Funcionales: Robustez

<b>Disponibilidad</b>	
<b>ID</b>	<b>Descripción</b>
<b>RNF-10</b>	La aplicación deberá estar disponible continuamente, durante 24 horas al día y 7 días a la semana.

Tabla 3.79: Requisitos No Funcionales: Disponibilidad



## Capítulo 4

# Planificación y Presupuesto

### 4.1. Metodología

En esta sección del documento se procederá a realizar una explicación de la metodología llevada a cabo para el desarrollo de este Trabajo.

Al tratarse de un proyecto de un tamaño comedido y ser yo el único individuo que ha trabajado en él, he llevado a cabo una metodología de trabajo cercana al Agile, aunque mucho más simplificada. Este tipo de metodología requiere una gran autogestión y organización por parte del alumno, siendo éste el único encargado de limitar el tiempo que tarda en realizar las tareas que él mismo ha fijado.

[18] Agile es una metodología de gestión de proyectos caracterizada por la creación de productos que los clientes realmente desean, utilizando ciclos cortos de trabajo que permiten una producción rápida y una revisión constante, si procede. Dentro de Agile existen diferentes marcos de trabajo que añaden un conjunto de estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular. Dentro de los marcos de trabajo que existen en la metodología Agile, el utilizado para este proyecto ha sido Kanban. Kanban ofrece un enfoque visual para la gestión de proyectos en el que los equipos de desarrollo crean representaciones físicas de sus tareas, a menudo utilizando tableros reales o en aplicaciones. Las tareas se desplazan a través de etapas predeterminadas con el fin de hacer un seguimiento del progreso e identificar obstáculos habituales. [6]

El principal motivo para la elección de este tipo de metodología fue lo desconocido del proyecto y el poco tiempo real de desarrollo que hubo, ya que nunca había realizado una aplicación basada en el uso de la voz como entrada de datos. Debido a esto era posible que los requisitos se fueran modificando a lo largo del avance del proyecto, por este motivo era necesaria la versatilidad que brinda Agile.

### 4.1.1. Herramientas para la organización

Como se ha comentado en el apartado anterior, para poder utilizar esta metodología correctamente hacen falta algunas herramientas para la organización de las tareas que conforman el proyecto. Las herramientas utilizadas para la gestión y coordinación del proyecto son las siguientes:

- **Trello:** es una de las aplicaciones de gestión de proyectos más utilizadas tanto a nivel profesional como personal. Es una herramienta basada en el método Kanban que sirve para gestionar tareas permitiendo la organización y gestión de proyectos.

Trello tendrá un uso organizativo de todas las tareas que componen este proyecto además de controlar el avance de este. Realiza una división por columnas en las que se incluirán tarjetas donde se detallarán las tareas del proyecto (Figura 4.1). La división de columnas se ha delimitado de la siguiente forma:

- **TO DO:** En esta columna se incluirán todas aquellas tareas que serán abordadas durante el desarrollo del proyecto.
- **DOING:** En esta columna se incluirán todas aquellas tareas que se están realizando actualmente.
- **DONE:** En esta columna se incluirán todas aquellas tareas que se hayan completado a falta de validarlas. A pesar de que la tarea se encuentre en esta columna, no se considerará completada hasta que no sea revisada y validada.
- **FINISHED:** En esta columna se incluirán aquellas tareas que han sido completadas y validadas.

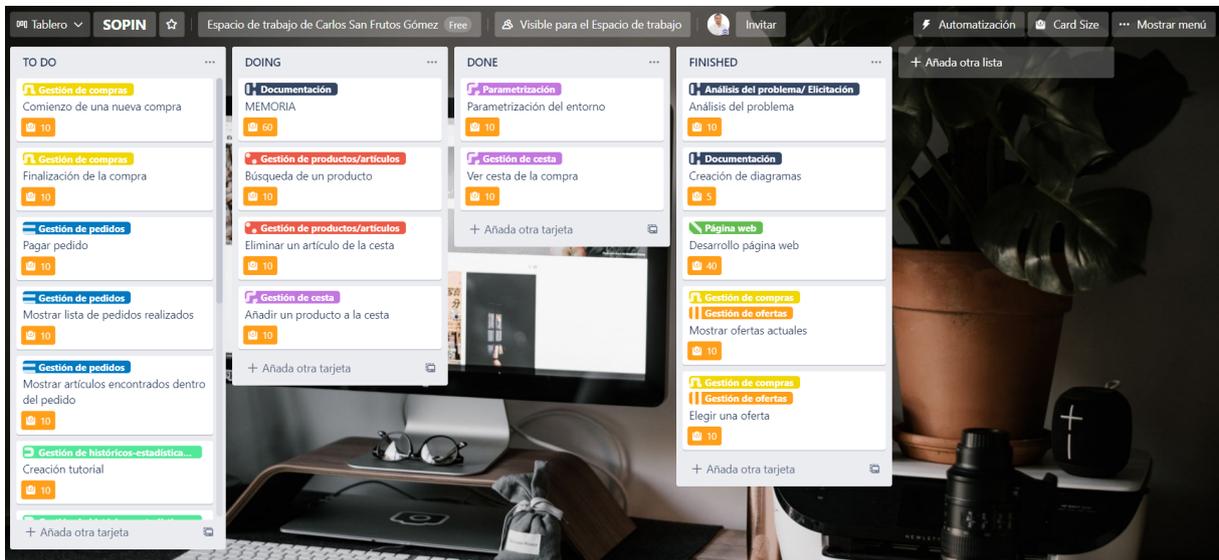


Figura 4.1: Tablero Trello del proyecto.

- **Toggl Track:** Es una herramienta que permite registrar el tiempo que inviertes en realizar un proyecto. Te ofrece la posibilidad de crear entradas que puedes utilizar para separar el tiempo que inviertes en realizar diferentes tareas. Además, te ofrece la posibilidad de crear etiquetas para que puedas separar los tiempos de las tareas en diferentes proyectos. En mi caso he creado una única etiqueta denominada TFG a la que he ido añadiendo los tiempos de las diferentes tareas. Cada tiempo tiene el nombre de la tarea principal a la que pertenecen con una pequeña descripción aclaratoria entre paréntesis. Esto se puede ver en la Figura 4.2

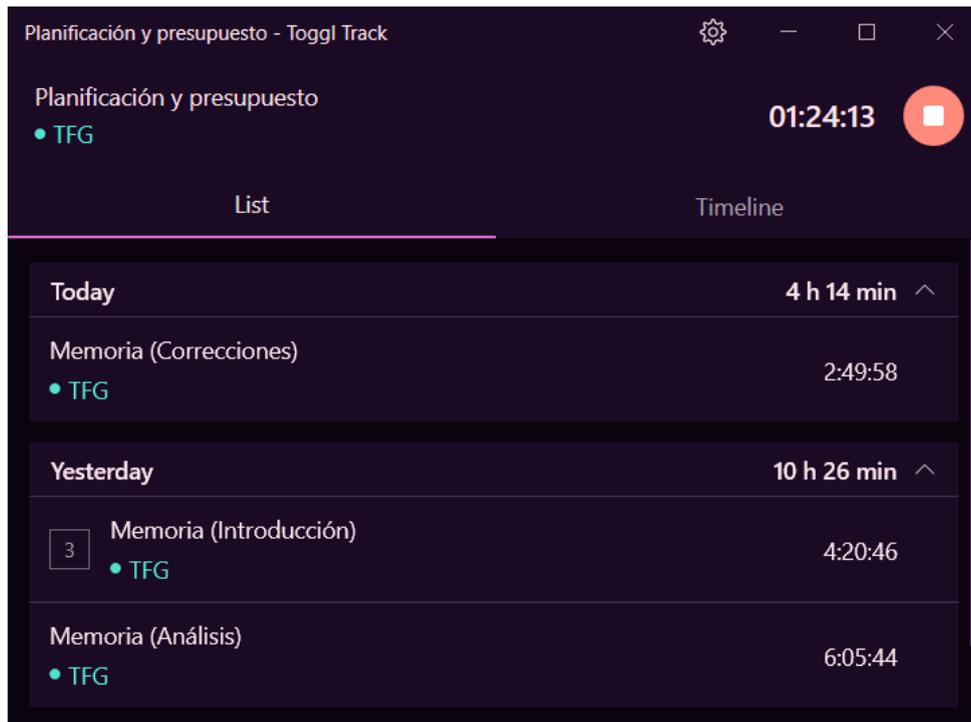


Figura 4.2: Toggl Track.

## 4.2. Planificación

En esta sección del documento se van a detallar todos los aspectos relacionados con la planificación del proyecto. Para ello vamos a apoyarnos de dos Diagramas de Gantt, un diagrama del inicio del proyecto (Figura 4.3) y otro del final del proyecto (Figura 4.4). Para realizar los diagramas de Gantt se han organizado las tareas del proyecto en diferentes grupos: Aprendizaje y Comienzo, Análisis, Diseño, Desarrollo, Pruebas y Documentación.

### 4.2.1. Estimación

El Trabajo de Fin de grado está compuesto por 12 Créditos ECTS, lo que equivale a unas 300 horas de trabajo. Por este motivo se intentó dividir el trabajo de uno de los grupos repartiendo de esta forma las horas que se invertirían a cada uno de los grupos. A continuación se detallará en que consiste cada grupo y que tareas alberga, además de las horas estimadas de esfuerzo de cada grupo.

- **Aprendizaje y Comienzo:** Este grupo de tareas está compuesto por todo lo relacionado al comienzo del proyecto, primera lectura de la propuesta, discusiones y puesta en común de primeros objetivos y del tema del Proyecto. Además también incluye toda la parte de aprendizaje de tecnologías desconocidas que fueran a utilizarse para el proyecto. El tiempo estimado de duración de este grupo de tareas fueron 2 semanas, trabajando una media de 10 horas por semana hacen un total de **20 horas**.
- **Análisis:** Este grupo de tareas está compuesto por todo lo relacionado con la elicitación de requisitos, creación de diagramas, estructuras de datos, etc. El tiempo estimado de duración de este grupo de tareas fueron 2 semanas, trabajando una media de 20 horas por semana hacen un total de **40 horas**.
- **Diseño:** Este grupo de tareas está compuesto por todo lo relacionado con el diseño visual de la aplicación (estado del arte), estructura y arquitectura de la aplicación, etc. El tiempo estimado de duración de este grupo de tareas fueron 2 semanas, trabajando una media de 10 horas por semana hacen un total de **20 horas**.
- **Desarrollo:** Este grupo de tareas está compuesto por todo lo relacionado con el desarrollo e implementación de la aplicación que seguirá las pautas marcadas por los 2 grupos anteriores. El tiempo estimado de duración de este grupo de tareas fueron 7 semanas, trabajando una media de 25 horas por semana hacen un total de **175 horas**.
- **Pruebas:** Este grupo de tareas está compuesto por todo lo relacionado con la prueba del software creado y corrección de fallos para su correcto funcionamiento. El tiempo estimado de duración de este grupo de tareas fueron 1 semanas, trabajando una media de 25 horas por semana hacen un total de **25 horas**.
- **Documentación:** Este grupo de tareas está compuesto por todo lo relacionado con la creación de la memoria que acompaña al proyecto. El tiempo estimado de duración de este grupo de tareas fueron 2 semanas, trabajando una media de 25 horas por semana hacen un total de **50 horas**.

La suma de las horas de todos los grupos de tareas da un total de 330 horas, lo que me pareció justo, teniendo en cuenta que en este tipo de proyectos siempre se plantean inconvenientes, por lo que aportaba algo de margen al desarrollo del proyecto.

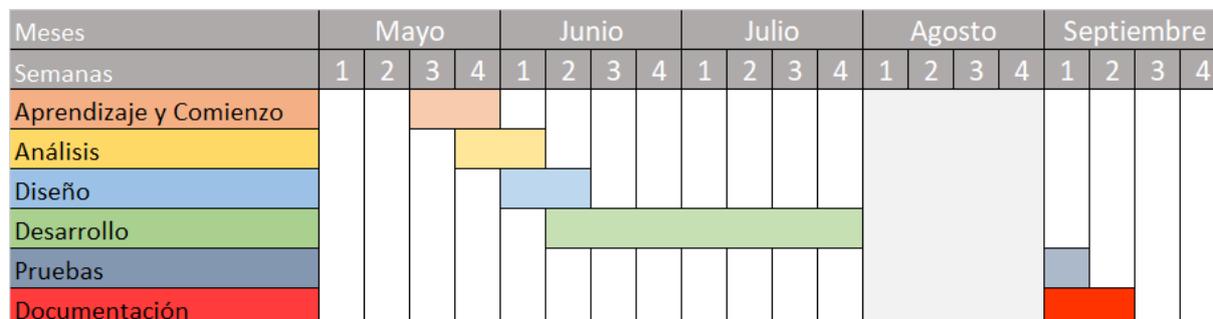


Figura 4.3: Diagrama de Gantt Inicial.

#### 4.2.2. Coste Real

La planificación inicial del proyecto no se ha podido cumplir por diversas razones. Durante los meses de Junio, Julio y Agosto compaginé el desarrollo de este proyecto junto con las prácticas en empresa. Esto produjo que el tiempo que disponía durante los días entre semana para dedicarle a este proyecto fuera menor que el esperado. Además, en el mes de Agosto, por motivos personales y debido a la incorporación en un equipo de desarrollo dentro la empresa en la que realicé las prácticas, el tiempo restante que podía invertir en este proyecto se redujo drásticamente. Por estos motivos la fecha de finalización tuvo que retrasarse.

Además de esto y debido a que mi fecha límite de entrega es el 20 de septiembre del año 2021 para que este proyecto pudiera entrar en el curso académico 20-21, durante el mes de Septiembre las horas de trabajo semanales que se han dedicado han aumentado drásticamente. Por estos motivos el proyecto ha tenido que sufrir recortes en cuanto a funcionalidades. Por último y debido a cambios en las primeras decisiones que se tomaron en cuanto a lenguajes y tecnologías a utilizar, el proceso de aprendizaje que debería haber durado únicamente 2 semanas aumentó, lo que derivó en el aumento de tiempo de el resto de grupos de tareas de trabajo. A continuación en la Figura 4.4 pueden verse los cambios producidos. Al final del proyecto se han invertido un total de 450 horas.

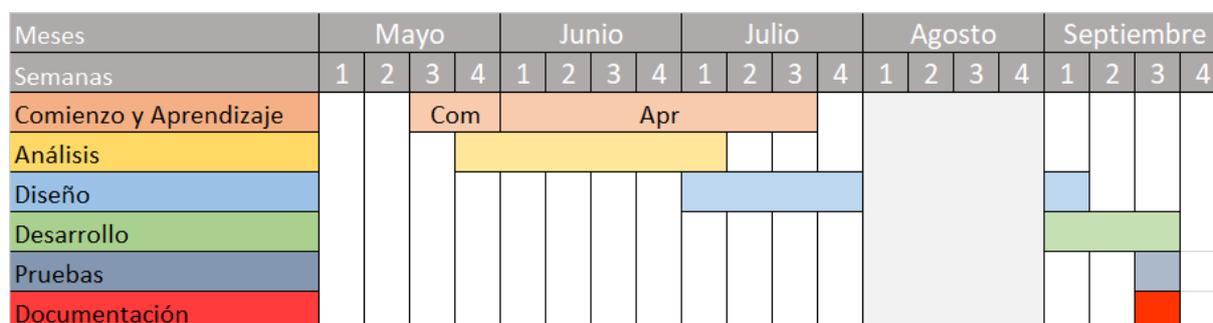


Figura 4.4: Diagrama de Gantt Real.

### 4.3. Presupuesto

En esta sección se llevará a cabo el cálculo de los presupuestos, incluyendo las herramientas que se han utilizado para el desarrollo, al igual que el hardware con el que se ha trabajado. Para el cálculo de los presupuestos, a parte del software y el hardware, también se tendrá cuenta el coste de recursos humanos.

Para la parte Hardware, todo este proyecto se ha realizado utilizando un ordenador personal, que por especificaciones puede considerarse de gama alta. La configuración de características técnicas del mismo es: Procesado Intel i7-8565U, 16GB de RAM, 500GB SSD, arquitectura de 64 bits y tarjeta gráfica dedicada GTX-1050. También se han utilizado algunas herramientas que en condiciones normales serían de pago, pero que al ser estudiante sus licencias se facilitan por la universidad gratuitamente. El servidor utilizado para desplegar la aplicación es de desarrollo por lo que no se tendrá en cuenta para el cálculo del presupuesto. A parte de todo esto es importante tener en cuenta el coste de mantenimiento, servicios utilizados y periféricos. Estos serían el coste de la luz, internet, segunda pantalla, etc.

#### 4.3.1. Presupuesto estimado

A continuación, en la Tabla 4.1 se puede ver el desglose de todo lo que se ha tenido en cuenta para el cálculo del presupuesto:

Herramienta	Coste (mes)	Vida útil	% de uso	Total
Ordenador Personal	1500€ (coste total)	5 años	15 %	225€
Monitor adicional	120€ (coste total)	10 años	10 %	12€
Internet	30€	3 meses	100 %	90€
Coste Energético	8€	3 meses	100 %	24€
GitHub	0€	–	–	0€
Visual Studio Code	0€	–	–	0€
Overleaf	0€	–	–	0€
yEd Graph Editor	0€	–	–	0€
Microsoft Office 365	0€ (licencia gratuita estudiante UVA)	–	–	0€
Sublime	0€	–	–	0€
Trello	0€	–	–	0€
Toggl Track	0€	–	–	0€
<b>TOTAL</b>				<b>351€</b>

Tabla 4.1: Costes de herramientas y recursos hardware.

Seguidamente se va a calcular el coste de recursos humanos. Al tratarse de un Trabajo de Fin de Grado, solo existe un trabajador. A pesar de ello, el trabajador ha tenido diferentes roles a lo largo del desarrollo del proyecto. Es por esto que se ha tenido en cuenta el coste de cada uno de los roles. (Ver Tabla 4.2)

<b>Rol</b>	<b>Salario (hora)</b>	<b>Horas</b>	<b>Total</b>
Analista	14€	80	1.120€
Desarrollador Web	10€	200	2.000€
<b>TOTAL</b>			<b>3.120€</b>

Tabla 4.2: Costes asociados a recursos humanos.

Si sumamos el coste de recursos humanos junto con el coste de las herramientas software y hardware y añadiendo el margen de beneficio calculado de la mitad del coste de la mano de obra, el balance total es el siguiente (4.3):

<b>Coste Total</b>	
Recursos Humanos	3.120€
Hardware y Software	351€
Margen de beneficio	1560€
<b>TOTAL</b>	<b>5.031€</b>

Tabla 4.3: Costes Totales.

### 4.3.2. Coste

Después del cambio de planificación una vez realizado el proyecto y debido al aumento de horas que han sido llevadas a cabo para su finalización el presupuesto final aumentaría respecto al estimado. Es por eso que a continuación se detallan de nuevo los costes de hardware, software y recursos humanos ajustándose a las nuevas características.

En la Tabla 4.4 encontramos el desglose de los costes de hardware y software:

Herramienta	Coste (mes)	Vida útil	% de uso	Total
Ordenador Personal	1500€ (coste total)	5 años	20 %	300€
Monitor adicional	120€ (coste total)	10 años	15 %	18€
Internet	30€	3 meses	100 %	90€
Coste Energético	8€	3 meses	100 %	24€
GitHub	0€	–	–	0€
Visual Studio Code	0€	–	–	0€
Overleaf	0€	–	–	0€
yEd Graph Editor	0€	–	–	0€
Microsoft Office 365	0€ (licencia gratuita estudiante UVA)	–	–	0€
Sublime	0€	–	–	0€
Trello	0€	–	–	0€
Toggl Track	0€	–	–	0€
<b>TOTAL</b>				<b>432€</b>

Tabla 4.4: Actualización costes de herramientas y recursos hardware.

Seguidamente calculamos el sobre coste en cuanto a recursos humanos(Ver Tabla 4.5)

Rol	Salario (hora)	Horas	Total
Analista	14€	100	1.400€
Desarrollador Web	10€	250	2.500€
<b>TOTAL</b>			<b>3.900€</b>

Tabla 4.5: Actualización costes asociados a recursos humanos.

Después de estos ajustes y añadiendo el margen de beneficio, calculado de la mitad del coste de la mano de obra, el balance total quedaría de la siguiente manera (Ver Tabla 4.6) :

<b>Coste Total</b>	
Recursos Humanos	3.900€
Hardware y Software	432€
Margen de beneficio	1560€
<b>TOTAL</b>	<b>5.892€</b>

Tabla 4.6: Costes Totales.



# Capítulo 5

## Diseño

En este capítulo del documento se procederá a explicar los aspectos más importantes del diseño de la aplicación. Se comentarán las diferentes estructuras y arquitecturas empleadas así como la interfaz de usuario.

### 5.1. Arquitectura

En esta sección se muestra el diseño de la aplicación desarrollada. El sistema contará con una parte Front-End, donde se encontrará la librería utilizada para el desarrollo del asistente virtual y la maquetación y alguna de la funcionalidad de la aplicación. Por otro lado, contará con una parte Back-End que será la encargada de utilizar los datos de la base de datos para cargarlos en la aplicación. También se encargará de hacer el recorrido contrario y recopilar los datos introducidos por el front. A su vez la parte del servidor (Back-End), tendrá comunicación directa con la pasarela de pago (Stripe) y con la Base de Datos (Mongo DB). El acceso a la aplicación será mediante el uso de un navegador web. La Figura 5.1 muestra el diseño y arquitectura de la aplicación:

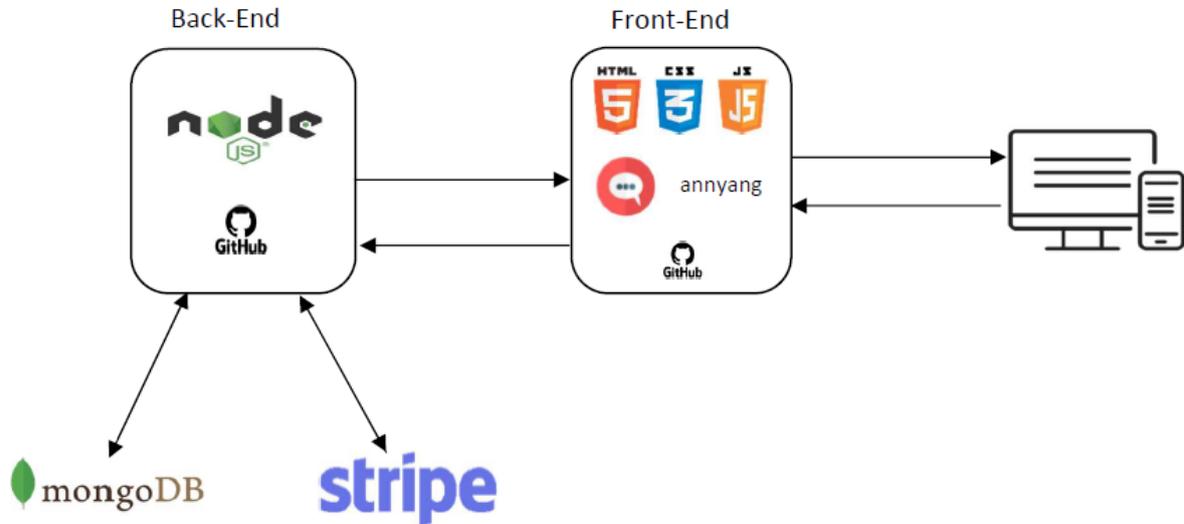


Figura 5.1: Diseño de la aplicación.

El flujo de datos e información que tendría la aplicación sería el siguiente. Al iniciar la aplicación la parte de Back-end extrae los datos de los productos y pedidos almacenados en la base de datos y los envía a través de un método GET HTTP a la parte Front-end. En la parte Front-end estos datos son recogidos e inyectados dinámicamente en el código HTML de la página.

Todas las funcionalidades que lleva a cabo el asistente virtual se encuentran en la parte de front-end, al igual que la funcionalidad que se realiza utilizando los periféricos convencionales. Desde la parte front-end se cargan los productos en el carrito, bien por métodos tradicionales, bien con el uso de la voz. Cuando ya se tiene un pedido con todos los productos añadidos al carrito, se podrá empezar con el proceso de compra. Para ello, se guarda en LocalStorage la información de los productos del carrito y cuando se insertan correctamente los datos del pedido, todos los datos son enviados mediante un método de fetching POST al back-end para que este pueda realizar otro método POST HTTP para realizar la pasarela con el sistema de pago STRIPE y con la BBDD para almacenar los datos del pedido en ésta.

Los paquetes utilizados en el Back-end por Node js para realizar todas las funciones son los que se pueden ver en la Tabla 5.1 junto con las versiones de los mismos:

Dependencias Node JS	Versión
<b>cors</b>	2.8.5
<b>express</b>	4.17.1
<b>mongoose</b>	6.0.5
<b>nodemon</b>	2.0.12
<b>path</b>	0.12.7
<b>stripe</b>	8.174.0

Tabla 5.1: Requisitos No Funcionales: Seguridad

## 5.2. Diagrama de flujo

En esta sección se puede ver un diagrama flujo conversacional entre un usuario y el sistema. En el se detallan los flujos que se pueden llevar a cabo y los caminos posibles a seguir cuando un usuario interactua con el sistema mediante la voz. Dispone de 3 caminos principales: Tutorial, Tienda e Historial de Pedidos.

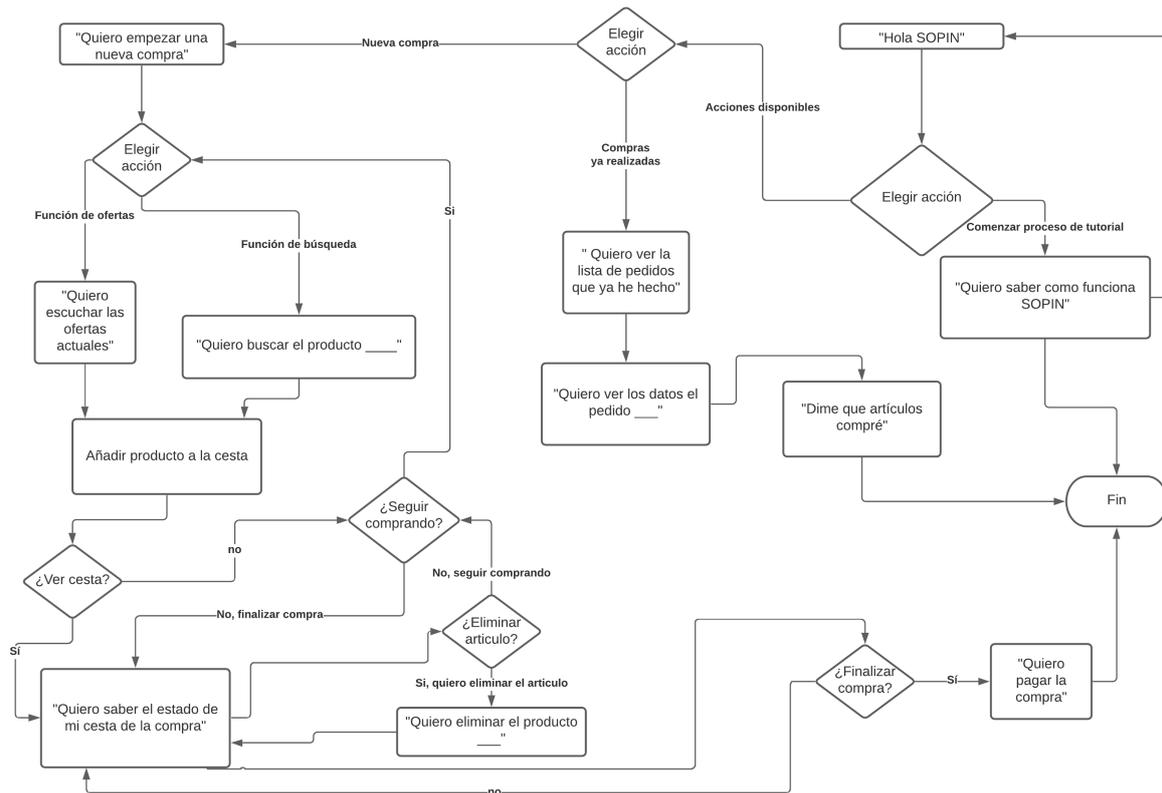


Figura 5.2: Diagrama de flujo.

### 5.3. Interfaz de Usuario

En esta sección se mostrarán las diferentes pantallas que visualizará el usuario mientras esté utilizando la aplicación.

#### 5.3.1. Tienda Catálogo

En esta pantalla el usuario podrá ver todos los productos que se ofrecen en el catálogo. Dispondrán de un nombre, un precio y la posibilidad de añadirlos al carrito. (Ver Figura B.1)

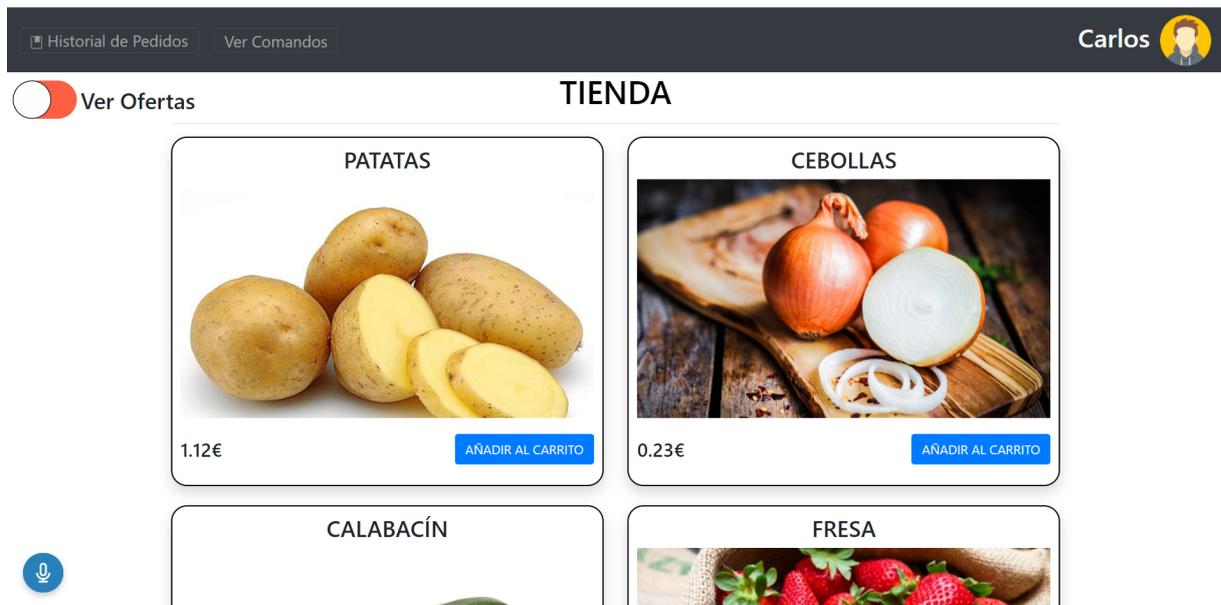


Figura 5.3: Tienda Catálogo.

### 5.3.2. Tienda Ofertas

En esta pantalla el usuario podrá ver todos los productos que se ofrecen en el catálogo de ofertas. Dispondrán de un nombre, un precio y la posibilidad de añadirlos al carrito. (Ver Figura B.2)

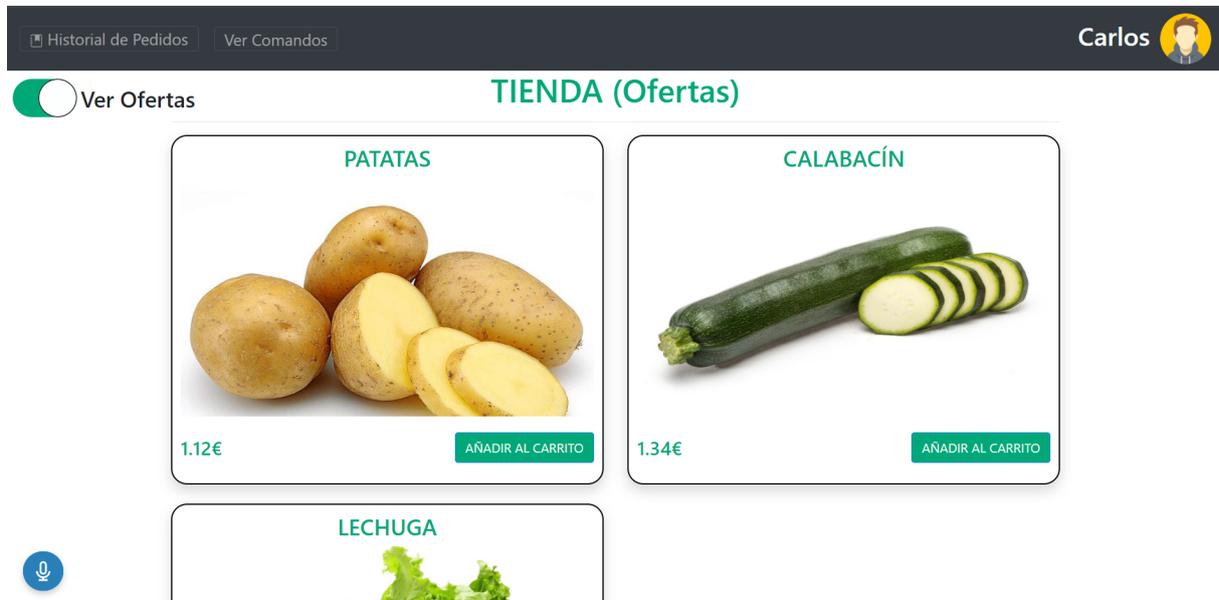


Figura 5.4: Tienda Ofertas.

### 5.3.3. Carrito de la Compra

En esta pantalla el usuario podrá ver todos los productos añadidos en el carrito. Dispondrá de la información de cada producto añadido: su foto, nombre, precio y cantidad y el precio total acumulado dentro del pedido. Además, se podrán eliminar individualmente los productos del carrito o vaciar el carrito completo pulsando vaciar. Por último también se ofrece la posibilidad de finalizar el pedido y comprar. (Ver Figura B.3)

CARRITO			
Producto	Precio	Cantidad	
 Calabacín	1.34€	<input type="text" value="1"/>	<input type="button" value="X"/>
 Fresa	3.5€	<input type="text" value="5"/>	<input type="button" value="X"/>
 Lechuga	2€	<input type="text" value="3"/>	<input type="button" value="X"/>
 Patatas	1.12€	<input type="text" value="3"/>	<input type="button" value="X"/>
Total 28.20€			<input type="button" value="Vaciar"/> <input type="button" value="Comprar"/>

Figura 5.5: Carrito de la compra.

### 5.3.4. Historial de Pedidos.

En esta pantalla el usuario podrá ver todos los pedidos que ha realizado a lo largo del tiempo. En cada pedido se registra información como el nombre del pedido, email, fecha, Número de productos, precio total del pedido y la lista con una pequeña imagen, el nombre de los productos y su cantidad. (Ver Figura B.4)

**HISTORIAL DE PEDIDOS**

Nombre de Pedido: Mi pedido 112	Nombre de Pedido: Video 1.	Nombre de Pedido: Mi pedido
Email: test@test.com Fecha: 13-9-2021 19:22:52 N° Productos: 2 Precio Total: 18.99 € Productos: 6 - Lechuga - 2€    1 - Sonic - 6.99€	Email: Carlos@carlos.com. Fecha: 13-9-2021 19:50:22 N° Productos: 2 Precio Total: 18.43 € Productos: 6 - Lechuga - 2€    1 - Sonic - 6.99€	Email: test@test.com Fecha: 13-9-2021 19:57:7 N° Productos: 4 Precio Total: 7.07 € Productos: 1 - Lechuga - 2€    1 - Calabacin - 1.34€ 1 - Fresa - 3.5€    1 - Cebollas - 0.23€
Nombre de Pedido: Compra 1.	Nombre de Pedido: Pedido 123123	Nombre de Pedido: Pedido 1.
Email: Test@test.com. Fecha: 13-9-2021 20:19 N° Productos: 6 Precio Total: 44.63 € Productos:	Email: test@test.com Fecha: 13-9-2021 20:9:14 N° Productos: 1 Precio Total: 2 € Productos:	Email: Carlos@carlos.com. Fecha: 14-9-2021 11:28:19 N° Productos: 4 Precio Total: 13.83 € Productos:

Figura 5.6: Historial de Pedidos.

### 5.3.5. Tutorial con comandos.

En esta pantalla el usuario podrá ver todos los comandos que existen en la aplicación para poder utilizarlos. Esta pantalla se trata de un modal que se abre en cualquier punto de la aplicación y sirve como tutorial o guía de uso. (Ver Figura B.5)

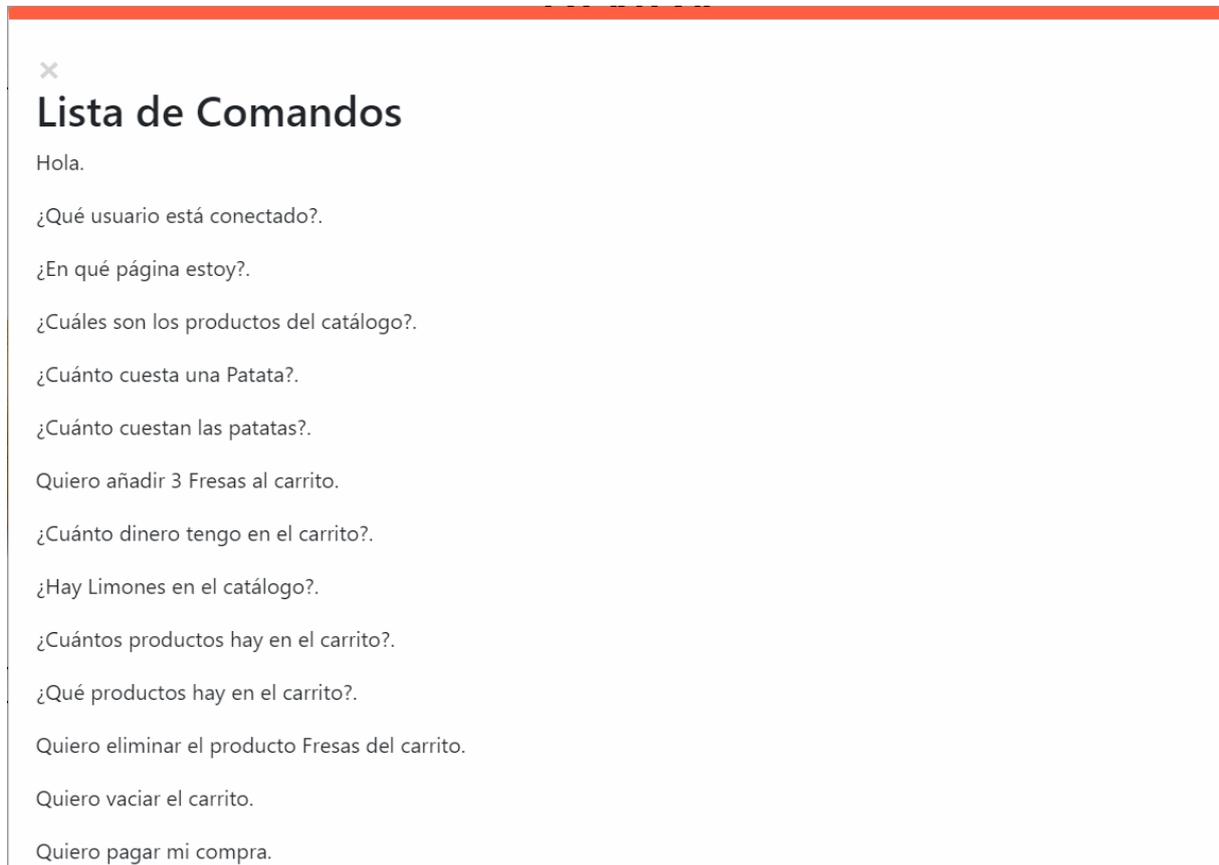


Figura 5.7: Comandos disponibles.

## Capítulo 6

# Implementación

En este capítulo del documento se detallarán aspectos relacionados con la implementación del código de la aplicación. Este código ha sido desarrollado utilizando toda la información expuesta en los apartados de Análisis y Diseño.

### 6.1. Herramientas y tecnologías utilizadas

- **Visual Studio Code:** Editor de código fuente para desarrollo web en sistemas Windows. Incluye soporte para el software de control de versiones Git.
- **Git:** Sistema de control de versiones creado por Linus Torvalds, que permite gestionar el código de una manera mucho más simple. Es el sistema de control de versiones más usado en el mundo ya que destaca en su eficiencia y fiabilidad a la hora del mantenimiento de versiones de aplicaciones. Emplea un Algoritmo Delta, el cual obtiene los bytes que han sido modificados desde la última versión del archivo.
- **GitHub:** Red social basada en el sistema de control de versiones de Git que permite gestionar proyectos software y controlar las versiones del código en la nube. Permite la creación de repositorios que, al enlazarse con los repositorios locales de Git, almacenan nuestro código en dicha nube. Estos repositorios pueden ser públicos o privados.
- **Stripe:** Stripe es un API para pagos que da impulso al comercio de las empresas que operan en internet.
- **Microsoft Edge:** Navegador utilizado para comprobar el correcto desarrollo de la aplicación.

## 6.2. Herramientas de Soporte

A continuación se muestran las herramientas utilizadas, con una pequeña descripción, para el soporte del proyecto:

- **Sublime:** Sublime text es un editor de texto y editor de código fuente.
- **Adobe Acrobat Reader:** Lector de PDF. Software gratuito.
- **yEd Graph Editor:** Aplicación que permite realizar diagramas UML de todo tipo. Software gratuito y accesible. Ha sido utilizado para realizar diagramas de casos de uso.
- **Microsoft Office:** Editor de texto desarrollado por Microsoft y utilizado para la realización algunos diagramas y documentos del proyecto.
- **Microsoft Excel:** Hoja de cálculo desarrollada por Microsoft y utilizada para la realización de diagramas de Gantt y temas administrativos del proyecto.
- **LucidChart:** Es un software de diagramación online que ayuda a individuos y equipos a visualizar procesos, sistemas y estructuras organizacionales. Ha sido utilizado para el desarrollo del diagrama de flujo.

## 6.3. Tecnologías Utilizadas

- **Node JS:** Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. [17]
- **Annyang:** Annyang es una librería de JS para reconocimiento de voz. La librería es capaz de utilizar el micrófono de nuestra computadora para reconocer nuestras palabras, escribirlas en pantalla o incluso realizar acciones basadas en el reconocimiento de las mismas. [5] Para más información visitar la documentación de Annyang: [1]
- **JavaScript:** JavaScript es un lenguaje de programación interpretado o de secuencias de comandos que te permite implementar funciones complejas en páginas web. Se define como orientado a objetos, está basado en prototipos, imperativo, débilmente tipado y dinámico. [3]
- **HTML y CSS:** HTML es el lenguaje de marcado que usamos para estructurar y dar significado a nuestro contenido web, por ejemplo, definiendo párrafos, encabezados y tablas de datos, o insertando imágenes y vídeos en la página. Por otro lado CSS es un lenguaje de reglas de estilo que usamos para aplicar estilo a nuestro contenido HTML, por ejemplo, establecer colores de fondo y tipos de letra, y distribuir nuestro contenido en múltiples columnas. [3]
- **Mongo DB:** MongoDB es un sistema de base de datos NoSQL, orientado a documentos y de código abierto. En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. [16]

- **Mongoose:** Mongoose es una biblioteca de modelado de datos de objetos (ODM) para MongoDB y Node.js. [10]

### 6.3.1. Node JS

Node JS está ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Está diseñado para crear aplicaciones network escalables. Por cada conexión creada con Node JS, se activa la devolución de llamada o callback, en caso contrario Node JS entrará en estado pasivo o dormido. [13]

Esto contrasta con el modelo de concurrencia más común de hoy en día, en el que se emplean hilos del Sistema Operativo. Las redes basadas en hilos son relativamente ineficientes y muy difíciles de usar. Casi ninguna función en Node.js realiza I/O directamente, por lo que el proceso nunca se bloquea. Por ello, es muy propicio desarrollar sistemas escalables en Node.js. [13]

HTTP es un elemento destacado en Node.js, diseñado teniendo en cuenta la transmisión de operaciones con streaming y baja latencia. Esto hace que Node.js sea muy adecuado para la base de una librería o un framework web. [13]

A continuación se puede ver un fragmento de código de como sería la creación básica de un servidor utilizando Node.js:

```
////////////////////////////////////CODE////////////////////////////////////
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hola Mundo');
});

server.listen(port, hostname, () => {
  console.log(`El servidor se está ejecutando en http://${hostname}:${port}/`);
});
////////////////////////////////////
```

### 6.3.2. Mongoose

Como se ha comentado anteriormente, Mongoose es una biblioteca de modelado de datos de objetos. Gestiona las relaciones entre los datos, proporciona la validación del esquema, y se utiliza de traductor entre los objetos en el código y la representación de esos objetos en MongoDB. En la siguiente Figura (6.1) se puede ver un esquema de como se integra Mongoose con Node js y MongoDB. [10]

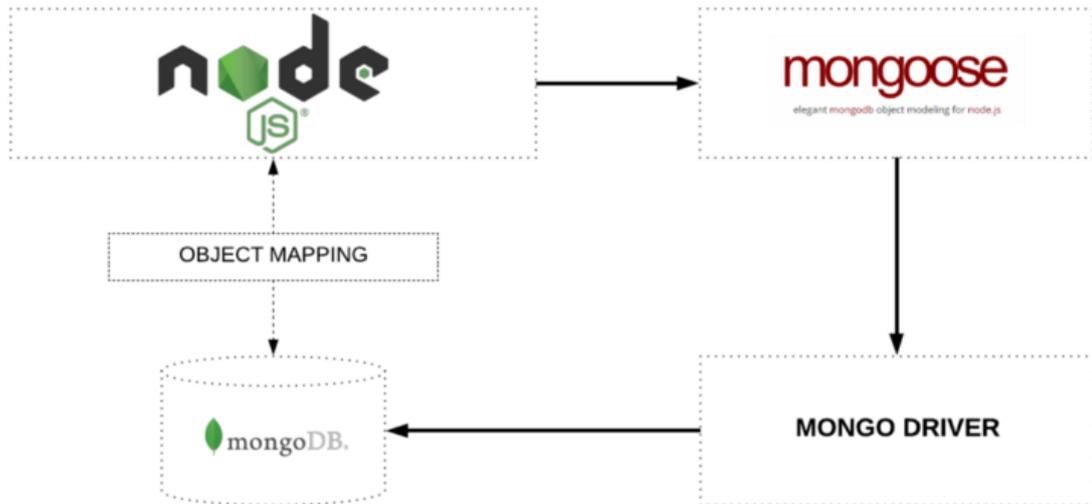


Figura 6.1: Esquema de la arquitectura Mongoose, Mongo DB Y Node JS.

## 6.4. Implementación de la base de datos

Para el guardado de todos los datos utilizados en la aplicación nos apoyaremos en una base de datos no relacional llamada Mongo DB. Mongo DB es una base de datos distribuida, basada en documentos y de uso general que ha sido diseñada para desarrolladores de aplicaciones modernas y para la era de la nube. Al estar alojada en la nube, no es necesaria una instalación en local en el equipo de desarrollo, lo que simplifica enormemente su implementación.

Para poder utilizarla lo único que hay que hacer es registrarse y crearse una cuenta, o utilizar una cuenta de Google. Una vez tengamos creada una cuenta, nos pedirá que aceptemos su política de privacidad para poder utilizar los servicios que ofrece. Después de esto nos pedirá que elijamos un plan para nuestra base de datos (??). Para esta aplicación se ha elegido el plan gratuito, ya que al tratarse de un desarrollo académico era suficiente con los recursos prestados por la versión gratuita.

The image shows three pricing plans for MongoDB:

- Serverless (PREVIEW):** For serverless applications that aren't critical with variable traffic. Minimal configuration required.
  - ✓ Pay only for the operations you run
  - ✓ Resources scale seamlessly to meet your workload
  - ✓ Always-on security and backups
 Starting at **\$0.30/1M reads**
- Dedicated:** For production applications with sophisticated workload requirements. Advanced configuration controls.
  - ✓ Network isolation and fine-grained access controls
  - ✓ On-demand performance advice
  - ✓ Multi-region and multi-cloud options available
 Starting at **\$0.08/hr\***  
\*estimated cost \$56.94/month
- Shared (FREE):** For learning and exploring MongoDB in a cloud environment. Basic configuration options.
  - ✓ No credit card required to start
  - ✓ Explore with sample datasets
  - ✓ Upgrade to dedicated clusters for full functionality
 Starting at **FREE**

Figura 6.2: Planes para la base de datos Mongo DB.

Después de elegir el plan, nos pedirá que elijamos la localización de nuestro servidor compartido. Cuanto más cerca elijamos la localización, mejor. Esto es así debido a que obtendremos una menor latencia en el envío y recepción de datos. Además nos dejará elegir el proveedor de servicios: Amazon Web Services, Google Cloud o Microsoft Azure. Después de elegir estas opciones nos creará un Cluster. Dicho Cluster será nuestra base de datos. Tardará unos minutos en crear el cluster y cuando esté creado podremos comenzar a crear las estructuras de datos de los objetos que queramos almacenar.

Una vez tenemos todo configurado podremos acceder a la creación de estructuras de datos a través del botón Browse Collections. Además dispondremos de una página principal del cluster donde podremos ver un pequeño resumen de distinta información perteneciente a nuestra base de datos, como las conexiones realizadas, la tamaño de datos almacenado o los B/s de entrada y salida. (Ver Figura 6.3)

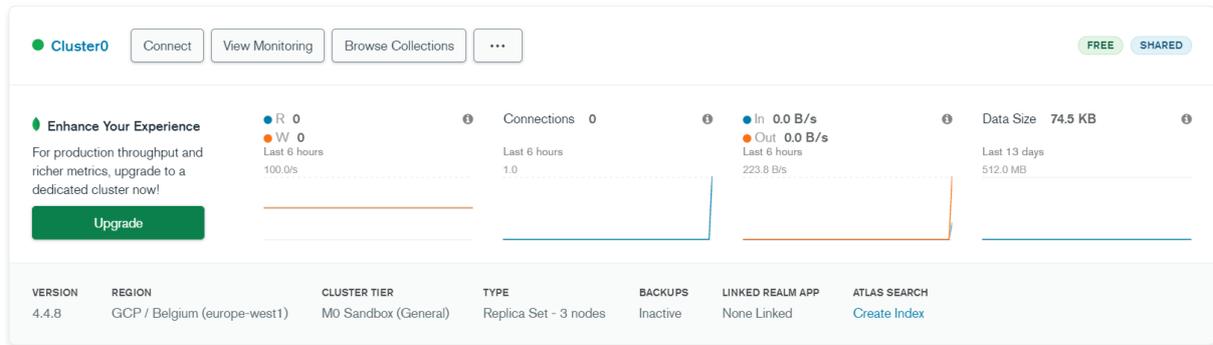


Figura 6.3: Vista de la información mostrada de un Cluster en Mongo DB.

## 6.5. Estructura del Proyecto

En esta sección se hará una breve descripción de la estructura del proyecto. El Trabajo se divide en dos partes: por un lado tenemos el Front-end y por otro lado tenemos el Back-end.

### 6.5.1. Front-End

A continuación se muestra el árbol de directorios del Front-End (Figura 6.4):

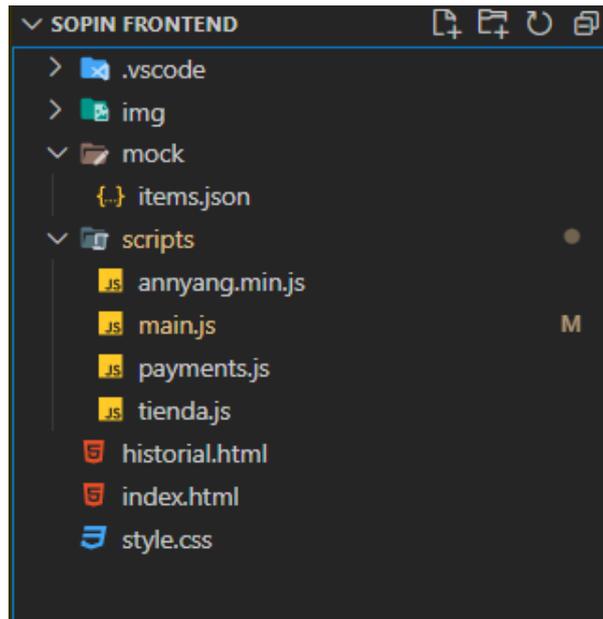


Figura 6.4: Árbol de directorios Front-end

Seguidamente se detallará el contenido de cada una de las carpetas del árbol de directorios así como algunos archivos:

- **/img:** En este directorio se guardan algunas de las imágenes utilizadas en la aplicación.

- **/mock:** En la carpeta mock se guarda un json de algunos items de prueba utilizados en el desarrollo de la aplicación. En la versión final de la aplicación no se usarán aunque fueron muy útiles en los primeros pasos de desarrollo. A continuación se puede ver una muestra del contenido del archivo items.json:

```
[
  {
    "id": "1ec58676-898f-4fa5-9fcf-1f386dda4724",
    "title": "Patatas",
    "price": 1.12,
    "image": "patata.jpg",
    "offer": true
  },
  {
    "id": "420bad3c-974e-4749-9e35-7ea303320e93",
    "title": "Cebollas",
    "price": 0.23,
    "image": "cebolla.jpg",
    "offer": false
  }
]
```

- **/scripts:** En la carpeta scripts se almacenan todos los archivos JavaScript del Front-end. En ellos es donde se encuentra toda la funcionalidad del proyecto que el usuario ve. En esta carpeta está importada la librería annyang.min.js desde donde se cargan las funciones del asistente virtual. También se encuentra el main.js, que es el archivo principal desde el que comienza la aplicación. También están el archivo payments.js, encargado de toda la funcionalidad referida a los pagos de los pedidos y la finalización de los mismos y el archivo tienda.js, encargado de toda la funcionalidad referida al carrito de la compra.
- **historial.html:** En este archivo se encuentra el desarrollo y la maquetación de la página del historial de Pedidos. Es un archivo HTML.
- **index.html:** En este archivo se encuentra el desarrollo y la maquetación de la página principal de la Tienda (Catálogo de productos y ofertas y carrito). Es un archivo HTML.
- **style.css:** En este archivo CSS es una hoja de estilos encargado del diseño de las páginas index.html e historial.html.

### 6.5.2. Back-End

A continuación se muestra el árbol de directorios del Back-End (Figura 6.5):

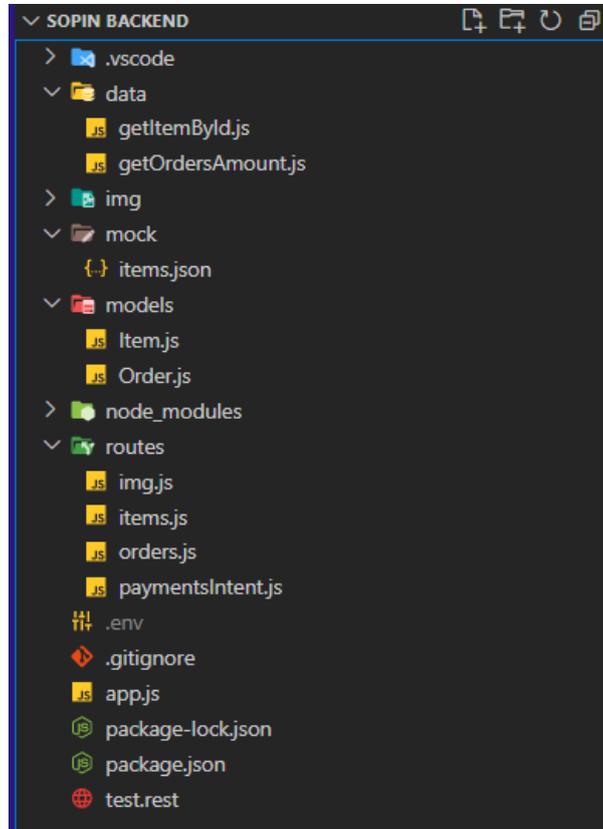


Figura 6.5: Árbol de directorios Back-end

- **/data:** En esta carpeta se encuentran contenidos dos archivos JavaScript cuyo único cometido es el de hacer la funcionalidad de getters de algunos atributos de los objetos almacenados en la base de datos.

```

////////////////////////////////////getItemById////////////////////////////////////
const Item = require('../models/Item');

module.exports = async function getItemById(id) {
  const itemFromDB = await Item.findById(id);
  return itemFromDB;
};
////////////////////////////////////

```

```

////////////////////////////////////getOrdersAmount////////////////////////////////////
const getItemById = require('./getItemById');

module.exports = getOrderAmount = async (items) => {
  let amount = 0;

  for (let index = 0; index < items.length; index++) {
    const item = items[index];
    //  petición a base de datos
    const itemDB = await getItemById(item.id);

    let operation = itemDB.price * item.qty;
    amount += operation;
  }
  const onlyTwoDecimals = amount.toFixed(2);
  const parsedAmount = parseInt(onlyTwoDecimals.replace('.', ''), 10);
  return parsedAmount;
  //////////////////////////////////////
};

```

- **/img:** En esta carpeta se almacenan las imágenes de los productos incluidos en el catálogo de productos de la aplicación. En la base de datos se guardan las rutas de estas imágenes, para que cuando los productos sean cargados, se extraigan las imágenes de esta carpeta.
- **/mock:** En la carpeta mock se guarda un json de algunos items de prueba utilizados en el desarrollo de la aplicación. En la versión final de la aplicación no se usarán aunque fueron muy útiles en los primeros pasos de desarrollo. A continuación se puede ver una muestra del contenido del archivo items.json:

```

[
  {
    "id": "1ec58676-898f-4fa5-9fcf-1f386dda4724",
    "title": "Patatas",
    "price": 1.12,
    "image": "patata.jpg",
    "offer": true
  },
  {
    "id": "420bad3c-974e-4749-9e35-7ea303320e93",
    "title": "Cebollas",
    "price": 0.23,
    "image": "cebolla.jpg",
    "offer": false
  }
]

```

- **/models:** En la carpeta models se almacenan los Modelos creados utilizando Mongoose. El objetivo es crear una validación para que cuando se cree un objeto predeterminado y se quiera comprobar la validez de la estructura del mismo (atributos obligatorios por ejemplo) tenga la obligación de ser creado con la estructura aquí descrita. Existe un modelo para los productos (Items) y para los pedidos (Orders). A continuación una muestra del código de ambos:

```
////////////////////////////////////ITEM MODEL////////////////////////////////////
const mongoose = require('mongoose');

const itemSchema = new mongoose.Schema({
  title: {
    type: String,
    required: true,
  },
  price: {
    type: Number,
    required: true,
  },
  image: {
    type: String,
    required: true,
  },
  offer: {
    type: Boolean,
    required: true,
  }
});

module.exports = mongoose.model('Item', itemSchema);
////////////////////////////////////
```

```
//////////////////////////////////ORDER MODEL//////////////////////////////////
const mongoose = require('mongoose');

const orderSchema = new mongoose.Schema({
  // id del pedido, email cliente, array objetos
  nombre: {
    type: String,
    required: true,
  },
  email: {
    type: String,
    required: true,
  },
  fecha: {
    type: String,
    required: true,
  },
  cantProductosTotal:{
    type: Number,
    required: true,
  },
  priceTotal:{
    type: Number,
    required: true,
  },
  items: [
    {
      _id: false,
      id: String,
      qty: Number,
    }
  ]
});

module.exports = mongoose.model('Order', orderSchema);
//////////////////////////////////
```

- **/node\_modules:** En esta carpeta están almacenados todos los paquetes que utiliza Node js de forma nativa o los importados por el programador para el desarrollo de la aplicación.

- **/routes:** En esta carpeta están almacenados todos los archivos encargados de definir los métodos de GET y POST. Estos métodos son los se encargan de la conexión del Back-end con Stripe, MongoDB y el Front-end. A modo de ejemplo se puede ver a continuación el código del archivo items.js. El resto de archivos incluidos en esta carpeta son muy similares en cuanto a la estructura:

```
////////////////////////////////////items.js////////////////////////////////////
const express = require('express');
const getItemById = require('../data/getItemById')
const router = express.Router();

//MODELS
const Item = require('../models/Item');
// GET ITEMS
router.get('/', async (req, res) => {
  try{
    const itemsFromDB = await Item.find();
    console.log(': itemsFromDB', itemsFromDB);
    res.json(itemsFromDB);
  }catch(err){
    res.json({message: err.message})
  }
});

// GET ITEM
router.get('/item', async (req, res) => {
  const itemId = req.body.itemId;
  try{
    const itemFromDB = await getItemById(itemId);
    res.json(itemFromDB);
  }catch(err){
    res.json({message: err.message})
  }
});
```

```

// CREATE ITEM
router.post('/', async (req, res) => {
  console.log(req.body);
  const item = new Item({
    title: req.body.title,
    price: req.body.price,
    image: req.body.image,
    offer: req.body.offer,
  });
  try {
    const newItem = await item.save();
    res.json(newItem);
  } catch (err) {
    res.json({ message: err.message });
  }
});
module.exports = router;
////////////////////////////////////

```

- **.env:** En este archivo están definidas como variables globales las claves privadas de la base de datos y de la pasarela de pago. Esto es así por un tema de seguridad. De esta forma en la implementación del código no aparece en ningún momento la definición de estas variables, solo la referencia a las mismas.
- **.gitignore:** Este archivo sirve para definir qué carpetas o archivos no se quieren subir al repositorio de git en los commits.
- **app.js:** Este es el archivo principal del proyecto de Back-end. En el se definen todas las características explicadas en los puntos anteriores para que la aplicación pueda funcionar correctamente. Además es donde se encuentra la configuración del servidor sobre el que funciona la aplicación. A continuación se muestra el código de dicho documento a modo de ejemplo:

```

const express = require('express');
const app = express();
const cors = require('cors');
const mongoose = require('mongoose');

require('dotenv').config();

// BBDD
mongoose.connect(process.env.DB_URL, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});
const db = mongoose.connection;
db.on('error', (error) => console.error(error));
db.on('open', () => console.log('Conectado a BBDD'));

```

```
// Import Routes
const itemsRoutes = require('./routes/items');
const imgRoutes = require('./routes/img');
const paymentIntentRoutes = require('./routes/paymentsIntent');
const ordersRoutes = require('./routes/orders');

// MIDDLEWARE
app.use(cors());
app.use(express.json());

app.use('/items', itemsRoutes);
app.use('/img', imgRoutes);
app.use('/create-payment-intent', paymentIntentRoutes);
app.use('/order', ordersRoutes);

// Rutas
app.get('/', (req, res) => {
  res.send('HOME!');
});

//Start
app.listen(3000);
```

- **package.json:** En este archivo es donde están definidas todas las dependencias y donde se encuentra la configuración de la aplicación Node js. A continuación se muestra el código de este archivo como ejemplo:

```
{
  "name": "sopin",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "nodemon app.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "cors": "^2.8.5",
    "dotenv": "^10.0.0",
    "express": "^4.17.1",
    "mongoose": "^6.0.5",
    "nodemon": "^2.0.12",
    "path": "^0.12.7",
```

```
    "stripe": "^8.174.0"  
  }  
}
```

- **test.rest:** Este documento es un archivo REST que tiene un propósito de pruebas del documento. En él se han ido haciendo pruebas de las conexiones entre el Back-end y el resto de servicios mediante los metodos POST y GET. Se explicará con más detalle su función en el capítulo de Pruebas 7.

## Capítulo 7

# Pruebas

En este capítulo del documento se explicarán las pruebas llevadas a cabo en la última etapa del proyecto y durante su desarrollo. Todas las pruebas que se han realizado son de caja negra y tendrán el objetivo de validar el correcto funcionamiento del proyecto.

### 7.1. Pruebas de Caja Negra

<b>CN-01</b>	Realización del proceso de compra completo utilizando comandos vocales buscando un resultado satisfactorio.
<b>Propósito</b>	Comprobar si el sistema mantiene un flujo fluido conversacional entre el usuario y el sistema a la hora de realizar una compra satisfactoria.
<b>Prerrequisitos</b>	El usuario utiliza los comandos de voz necesarios para realizar una compra completa satisfactoria
<b>Datos de Entrada</b>	Datos de los productos almacenados en el carrito e información del pedido.
<b>Resultado Esperado</b>	El sistema tramita el pedido correctamente.
<b>Resultado Obtenido</b>	El sistema tramita el pedido correctamente.

Tabla 7.1: Prueba de Caja Negra CN-01: Realización del proceso de compra completo utilizando comandos vocales con objetivo satisfactorio.

<b>CN-02</b>	Realización del proceso de compra completo mediante el uso de periféricos tradicionales buscando un resultado satisfactorio.
<b>Propósito</b>	Comprobar el correcto funcionamiento de la pasarela de pago y del trámite de compra de un pedido.
<b>Prerrequisitos</b>	El usuario realiza una compra y procede a pagarla.
<b>Datos de Entrada</b>	Datos de los productos almacenados en el carrito e información del pedido junto con los datos de la tarjeta de pago.
<b>Resultado Esperado</b>	El sistema tramita el pedido correctamente mostrando al usuario un mensaje.
<b>Resultado Obtenido</b>	El sistema tramita el pedido correctamente mostrando al usuario un mensaje.

Tabla 7.2: Prueba de Caja Negra CN-02: Realización del proceso de compra completo utilizando periféricos tradicionales con objetivo satisfactorio.

<b>CN-03</b>	Realización del proceso de compra completo utilizando comandos vocales buscando un resultado erróneo.
<b>Propósito</b>	Comprobar si el sistema mantiene un flujo fluido conversacional entre el usuario y el sistema a la hora de realizar una compra frustrada.
<b>Prerrequisitos</b>	El usuario utiliza los comandos de voz necesarios para realizar una compra completa erróneamente
<b>Datos de Entrada</b>	Datos de los productos almacenados en el carrito e información del pedido.
<b>Resultado Esperado</b>	El sistema no tramita el pedido.
<b>Resultado Obtenido</b>	El sistema no tramita el pedido.

Tabla 7.3: Prueba de Caja Negra CN-03: Realización del proceso de compra completo utilizando comandos vocales con objetivo de fallo.

<b>CN-04</b>	Realización del proceso de compra completo mediante el uso de periféricos tradicionales buscando un resultado erróneo.
<b>Propósito</b>	Comprobar el correcto funcionamiento de la pasarela de pago y del trámite de compra de un pedido en caso de fallo.
<b>Prerrequisitos</b>	El usuario realiza una compra y procede a pagarla pero introduciendo valores erróneos.
<b>Datos de Entrada</b>	Datos de los productos almacenados en el carrito e información del pedido junto con los datos de la tarjeta de pago.
<b>Resultado Esperado</b>	El sistema no tramita el pedido correctamente y muestra un mensaje de error.
<b>Resultado Obtenido</b>	El sistema no tramita el pedido correctamente y muestra un mensaje de error.

Tabla 7.4: Prueba de Caja Negra CN-04: Realización del proceso de compra completo utilizando periféricos tradicionales con objetivo de fallo.

## 7.2. Pruebas conexiones Back-End

A parte de las pruebas de caja negra y como se ha comentado en el capítulo de implementación 6, a lo largo del desarrollo del proyecto Back-end se han realizado pruebas de conexión con los distintos servicios para comprobar su correcto funcionamiento. Dichas pruebas tenían relación con los métodos de GET y POST. En el documento 'test.rest' es donde pueden encontrarse. A continuación se muestra el código de dicho documento:

```
GET http://localhost:3000/

###
GET http://localhost:3000/items

###
GET http://localhost:3000/Order

###
GET http://localhost:3000/items/item
Content-Type: application/json

{
  "itemId": "613dbac56300b7884c1332ca"
}
```

```
###  
GET http://localhost:3000/img/patata.jpg
```

```
###  
POST http://localhost:3000/items  
Content-Type: application/json
```

```
{  
  "id": "df72e507-7988-451a-9e20-0cf3ef3a52af",  
  "title": "Sonic",  
  "price": 6.99,  
  "image": "sonic.jpg",  
  "offer": false  
}
```

```
###  
POST http://localhost:3000/create-payment-intent  
Content-Type: application/json
```

```
[  
  {  
    "id": "613dbac56300b7884c1332ca",  
    "qty": 2  
  }  
]
```

```
###
POST http://localhost:3000/order
Content-Type: application/json

{
  "nombre": "Pedido 15",
  "email": "test@test.com",
  "fecha": "12/09/2021",
  "cantProductosTotal": 2,
  "priceTotal": 15.59,
  "items": [
    {
      "id": "613dbac56300b7884c1332ca",
      "image": "sonic.jpg",
      "title": "Sonic",
      "price": 6.99,
      "qty": 1
    },
    {
      "id": "613dba376300b7884c1332bf",
      "image": "patata.jpg",
      "title": "Patatas",
      "price": 1.12,
      "qty": 5
    },
    {
      "id": "613dba9f6300b7884c1332c2",
      "image": "cebolla.jpg",
      "title": "Cebollas",
      "price": 0.23,
      "qty": 3
    }
  ]
}
```

Al realizar dichas pruebas el servidor podía devolver dos tipos de mensajes. Erróneos y correctos. Si la respuesta era errónea algunos mensaje devueltos podían ser algo como 7.1 y 7.2, en cambio si la respuesta devuelta era correcta, ésta era algo como 7.3.

```
HTTP/1.1 404 Not Found
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Security-Policy: default-src 'none'
X-Content-Type-Options: nosniff
Content-Type: text/html; charset=utf-8
Content-Length: 150
Date: Mon, 20 Sep 2021 16:01:35 GMT
Connection: close

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>Cannot GET /items/item5</pre>
</body>
</html>
```

Figura 7.1: 404 Not Found message.

```
HTTP/1.1 400 Bad Request
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Security-Policy: default-src 'none'
X-Content-Type-Options: nosniff
Content-Type: text/html; charset=utf-8
Content-Length: 1233
Date: Mon, 20 Sep 2021 16:02:28 GMT
Connection: close

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>SyntaxError: Unexpected token } in JSON at position 146<
```

Figura 7.2: 400 Bad Request message.

```
HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 105
ETag: W/"69-uKEmI1BLUpgadySTqU2ou9jesXc"
Date: Mon, 20 Sep 2021 16:00:28 GMT
Connection: close

{
  "_id": "613dbac56300b7884c1332ca",
  "title": "Sonic",
  "price": 6.99,
  "image": "sonic.jpg",
  "offer": false,
  "__v": 0
}
```

Figura 7.3: 200 OK message.



## Capítulo 8

# Mejoras

En este capítulo se expondrán algunas mejoras que la aplicación podría tener. Algunas de ellas factibles en una segunda versión de la aplicación y otras que conllevarían la creación de un proyecto nuevo. Estas mejoras menores podrían haberse incluido en el desarrollo del proyecto si no fuera por la limitación temporal.

Las mejoras menores serían las siguientes:

- Sistema de login utilizando la voz
- En el Historial de Pedidos se podrían añadir estadísticas, como por ejemplo, El producto más comprado, el total de pedidos realizados o el total de dinero gastado en los pedidos.
- En el Historial de Pedidos se podría añadir un filtro. De esta manera se podrían buscar Pedidos en una determinada fecha o que contengan un determinado Producto.
- En el Historial de Pedidos se podría ofrecer la posibilidad de repetir uno de los Pedidos y volverlo a comprar.
- Realizar un despliegue del Back-end en un servicio en la nube, como Docker.
- Adaptación de la aplicación para dispositivos móviles.

Si pensamos más a lo grande se podría incrementar notablemente el valor y la utilidad de esta aplicación si le añadimos el desarrollo de un modelo de inteligencia artificial basado en el reconocimiento de lenguaje natural. De esta manera el modelo sería capaz de que cuando el usuario quiera referirse a una funcionalidad concreta de la aplicación, el sistema podría reconocer a que funcionalidad se refiere aunque el usuario lo diga de muchas formas distintas. Esto elimina el problema de tener que aprenderse comandos específicos para el uso de todas las funcionalidades de la aplicación.



## Capítulo 9

# Conclusiones

A lo largo de esta memoria se ha realizado la documentación técnica del proyecto que se ha desarrollado como Trabajo Fin de Grado. En este proyecto se ha desarrollado una aplicación que utiliza la voz como entrada de datos para el manejo completo de las funcionalidades que una web de compras online puede ofrecer. El nombre elegido para el proyecto ha sido SOPIN. Este nombre nace de convertir al español utilizando la fonética la palabra en inglés Shopping, que significa compras.

Los objetivos personales de este proyecto eran aprender nuevas tecnologías de desarrollo y realizar un proyecto completo, permitiéndome ampliar mis conocimientos como informático. Se ha realizado una parte de Análisis y una parte de desarrollo que han servido para satisfacer los principales objetivos propuestos. Es por eso que considero que la mayor parte de estos objetivos han sido cumplidos.

Para poder cumplirlos se realizó un primer análisis del problema que se presentaba, que era realizar la creación de un asistente virtual que pudiera realizar el proceso de compra con el uso de la Voz. Después de ese primer análisis se comenzó con una elicitación de los primeros requisitos hasta conformar una buena lista de posibles funcionalidades que la aplicación tendría que suplir. Por último comenzó la parte de desarrollo en la que todos estos requisitos fueron tomando forma y convirtiéndose en funcionalidad real.

Además, he adquirido conocimientos de Node JS, del desarrollo de un proyecto utilizando back-end y front-end, he aprendido a utilizar nuevas herramientas como Sublime, yEd Graph Editor, Visual Studio Code, etc. Todos estos conocimientos nuevos, los podré utilizar en los años venideros y seguro que me serán de gran ayuda.

Sin embargo, considero que la aplicación tiene un gran margen de mejora, como he comentado en el Capítulo 8 - Mejoras. Creo que algunas de las funcionalidades ahí descritas podrían haberse llegado a efectuar si las condiciones hubieran sido otras. Aún así creo que los objetivos principales han sido correctamente suplidos y se adaptan a lo que esperaba conseguir al comienzo de este desarrollo.



# Bibliografía

- [1] Tal Ater. annyang readme, (visitado 19-09-2021). <https://github.com/TalAter/annyang/blob/master/README.md>.
- [2] Ayudaley. Base de datos no relacional. ¿Qué es? Características y ejemplos, (visitado 18-09-2021). [https://biblioteca.sistedes.es/submissions/uploaded-files/JISBD\\_2017\\_paper\\_71.pdf](https://biblioteca.sistedes.es/submissions/uploaded-files/JISBD_2017_paper_71.pdf).
- [3] MDN contributors. ¿Qué es JavaScript?, (visitado 19-09-2021). [https://developer.mozilla.org/es/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript).
- [4] Manuel Echeverría. La tecnología transformó tu forma de comprar, también la de vender, (visitado 18-09-2021). <https://blog.growxco.com/articulos/la-tecnologia-transformo-tu-forma-de-comprar-tambien-la-de-vender>.
- [5] Frogx3. Annyang, increíble librería JavaScript para reconocimiento de voz, (visitado 19-09-2021). <https://www.frogx3.com/2013/09/17/annyang-increible-libreria-javascript-para-reconocimiento-de-voz/>.
- [6] Aurelio Gandarillas. Marco de trabajo, (visitado 18-09-2021). <https://metodologia.es/marco-de-trabajo/>.
- [7] Sonia Garcia. Asistentes virtuales, una ayuda para las personas dependientes, (visitado 18-09-2021). <https://planetachatbot.com/asistentes-virtuales-ayuda-personas-dependientes-voicebots/>.
- [8] Carlos Betancur Gálvez. Comercio Electrónico: Cómo ha cambiado con el paso del tiempo, (visitado 18-09-2021). <https://btodigital.com/comercio-electronico-como-ha-cambiado-con-el-paso-del-tiempo/>.
- [9] Alberto Hernández Chillón, Severino Feliciano Morales, Jesús García Molina, Diego Sevilla Rui. Visualización de Esquemas en Bases de Datos NoSQL basadas en documentos\*, (visitado 18-09-2021). [https://biblioteca.sistedes.es/submissions/uploaded-files/JISBD\\_2017\\_paper\\_71.pdf](https://biblioteca.sistedes.es/submissions/uploaded-files/JISBD_2017_paper_71.pdf).
- [10] Nick Karnik. Introduction to Mongoose for MongoDB, (visitado 19-09-2021). <https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/>.
- [11] kuombo. Asistentes virtuales: la nueva atención al cliente online, (visitado 18-09-2021). <https://kuombo.com/asistentes-virtuales-nueva-atencion-al-cliente-online/>.

- [12] kuombo. Ecommerce conversacional, ¿cuánto hablas con tus clientes?, (visitado 18-09-2021). <https://kuombo.com/ecommerce-conversacional-cuanto-hablas-con-tus-clientes-3/>.
- [13] Node.js. Acerca de Node.js®), (visitado 19-09-2021). <https://nodejs.org/es/about/>.
- [14] Singular. Interfaces de Voz, (visitado 18-09-2021). <https://www.singular.com/wp-content/uploads/2019/05/VUI-Research-2019-sngular.pdf>.
- [15] Wikipedia. Asistente virtual, (visitado 18-09-2021). <https://kuombo.com/ecommerce-conversacional-cuanto-hablas-con-tus-clientes-3/>.
- [16] Wikipedia. MongoDB, (visitado 19-09-2021). <https://es.wikipedia.org/wiki/MongoDB>.
- [17] Wikipedia. Node.js, (visitado 19-09-2021). <https://es.wikipedia.org/wiki/Node.js>.
- [18] wrike. PROJECT MANAGEMENT GUIDE FAQ, (visitado 18-09-2021). <https://www.wrike.com/es/project-management-guide/faq/que-es-la-metodologia-agile-en-gestion-de-proyectos/>.

# Apéndices



# Apéndice A

## Despliegue

En este primer apartado del Apéndice vamos a realizar todo lo necesario para poder realizar el despliegue de la aplicación a partir del código fuente que tendríamos del proyecto. Este despliegue estará hecho únicamente para Windows, debido a que es el SO utilizado durante el desarrollo de este proyecto.

El primer paso que vamos a realizar es la instalación de Visual Studio Code con todos sus paquetes:

- **Paso 1:** Primero vamos a dirigirnos a la página de Microsoft Visual Studio Code (<https://code.visualstudio.com/>) y vamos a descargar la versión para Windows.

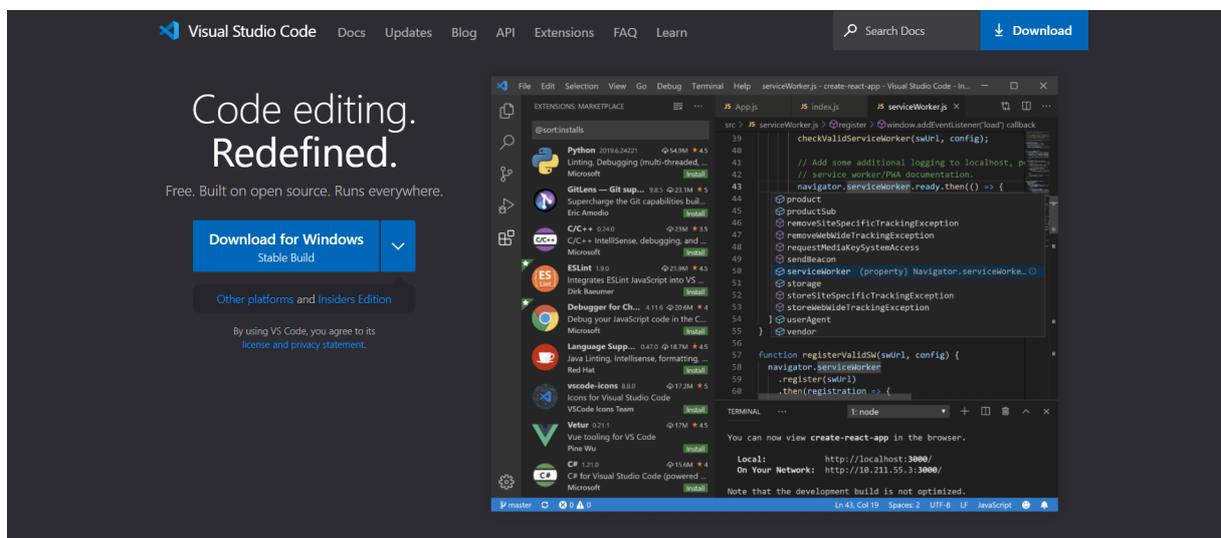


Figura A.1: Página de descarga de Visual Studio Code

- **Paso 2:** Ahora se deberá abrir el archivo .exe que se ha descargado y tendremos que seguir todos los pasos pulsando el botón siguiente y dejando la configuración predeterminada.

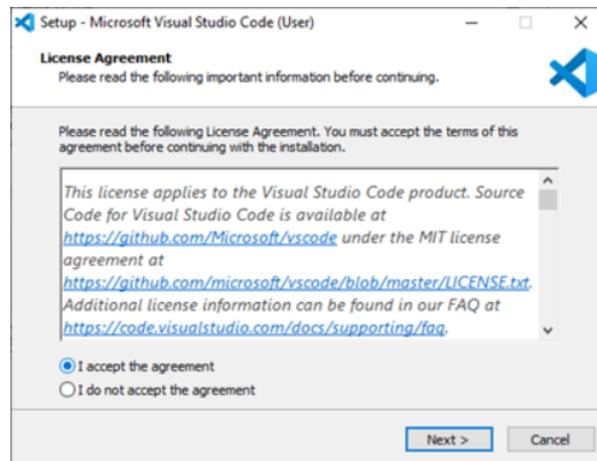


Figura A.2: Primer paso en la instalación de Visual Studio Code

- **Paso 3:** Cuando haya finalizado le saldrá una ventana como la siguiente. Pulse en finalizar y la instalación habrá terminado.

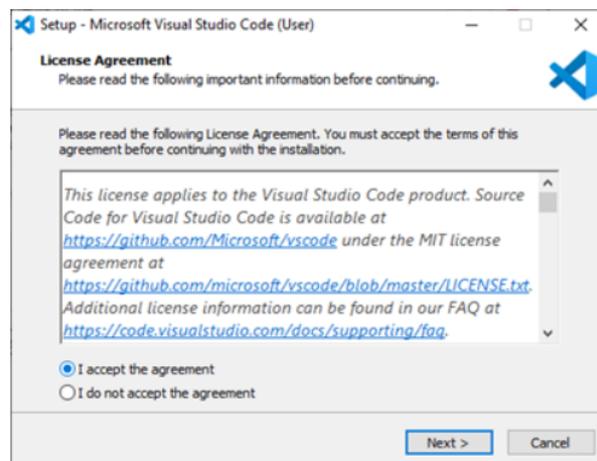


Figura A.3: Último paso en la instalación de Visual Studio Code

- **Paso 4:** Por último vamos a abrir Visual Studio Code y vamos a instalar todos los paquetes necesarios. Para ello, una vez tenemos abierto visual estudio vamos a buscar en la barra lateral izquierda el icono de extensiones. En la foto que se puede ver a continuación es el icono seleccionado (Ver Figuras A.4 y A.5 ). Es recomendable la instalación de todas las extensiones presentadas. De todas formas, las únicas indispensables para el correcto funcionamiento de la aplicación son: ESLint, Live Server, REST Client y Terminal. Para poder instalar cada extensión use el buscador que aparece en la parte superior de la columna donde se encuentran todas las extensiones e introduzca el nombre de ellas. Después de haber encontrado la extensión pulse en el botón 'install'. Cuando haya realizado la instalación de todas las extensiones reinicie Visual Studio Code.

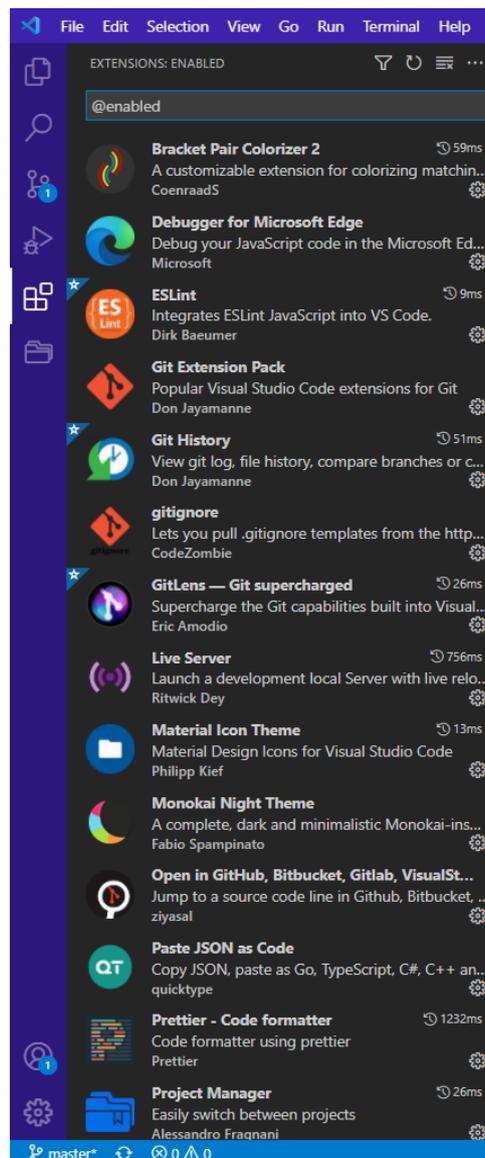


Figura A.4: Visualización de las extensiones necesarias 1

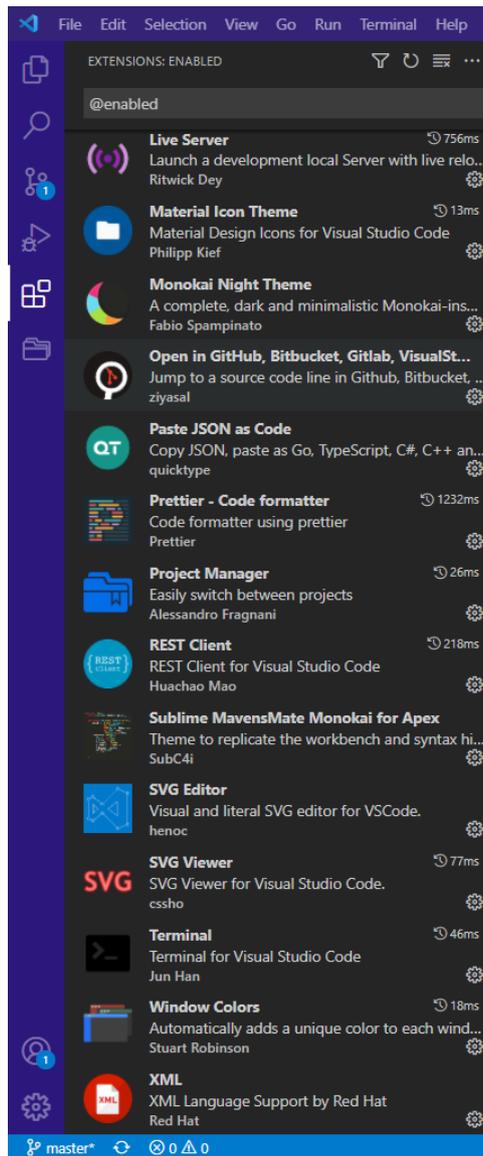


Figura A.5: Visualización de las extensiones necesarias 1

Una vez tenemos instalado Visual Studio Code vamos a comenzar con la puesta en marcha de la aplicación. Tendremos el paquete con el código dividido en dos proyectos o carpetas: Front-end y Back-end. Debemos abrir con Visual Studio Code por separado, cada carpeta como si fueran 2 proyectos distintos. Una vez hecho esto tendremos que tener 2 ventanas distintas de Visual Studio Code con los dos proyectos abiertos (Ver Figura A.6).

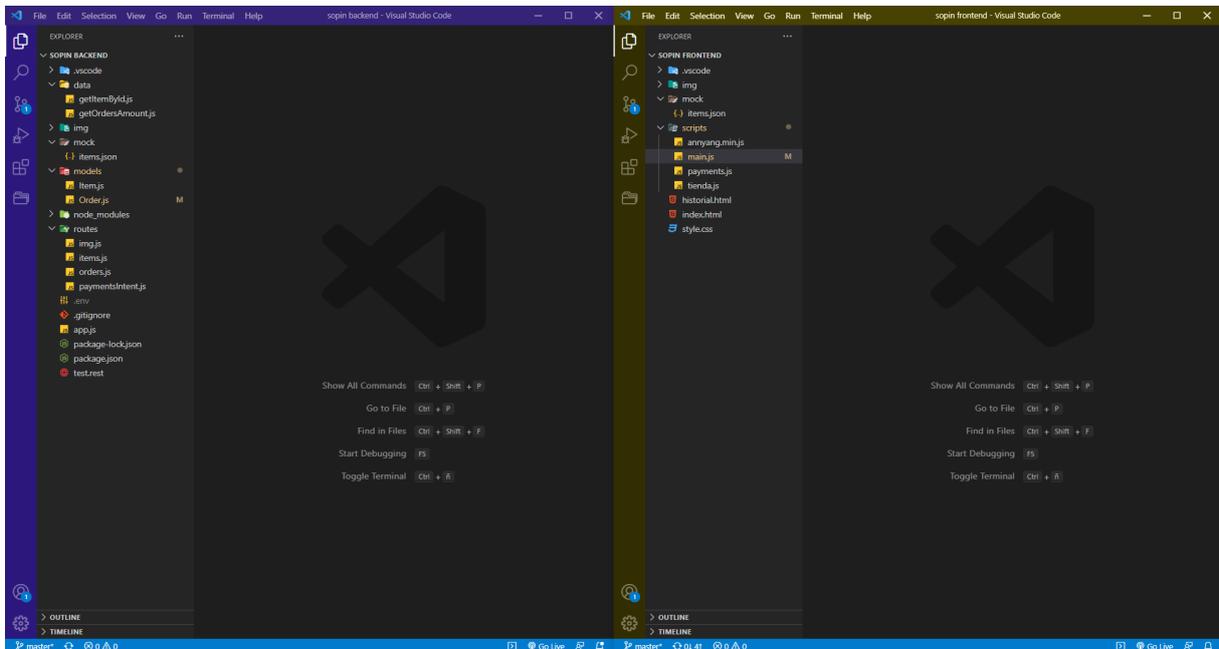


Figura A.6: Visualización de los dos proyectos abiertos (Back-end y Fron-end).

Después de que ya tenemos el código fuente del proyecto organizado, vamos a comenzar con la instalación del servidor. Para ello vamos a irnos a la página oficial de Node JS (<https://nodejs.org/es/>) y vamos a instalar la versión recomendada para la mayoría (Ver Figura A.7



Figura A.7: Instalación de Node JS.

A continuación, abra el archivo .exe descargado y siga todos los pasos al igual que en el caso de la instalación de Visual Studio Code.

Cuando lo tengamos instalado vamos a irnos al proyecto back-end y vamos a abrir una Terminal pulsando Ctr+Shift+ñ. Cuando tengamos abierta una Terminal o consola, introducimos el comando: 'npm init -y'. Esto lo que hará es inicializar todos los paquetes y dependencias del proyecto. Una vez hecho esto escribiremos en la consola el siguiente comando: 'npm start'. Esto lo que hará es lanzar el servidor e iniciarlo. Si todo ha salido bien el resultado debería ser como el siguiente (Ver Figura A.8). Si al intentar iniciar el servidor diera algún error por falta de paquetes, se deberán instalar individualmente utilizando el comando 'npm install {nombre del paquete}' sin las llaves y poniendo en su lugar el nombre del paquete que se quiera instalar.

```
PS C:\Users\carlo\Google Drive\UNI\CUARTO\SEGUNDO CUATRIMESTRE\TFG\sopin\sopin backend> npm start
> sopin@1.0.0 start C:\Users\carlo\Google Drive\UNI\CUARTO\SEGUNDO CUATRIMESTRE\TFG\sopin\sopin backend
> nodemon app.js

[nodemon] 2.0.12
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Conectado a BBDD
```

Figura A.8: Inicio del servidor mediante el comando 'npm start'

Después de esto tendremos lanzado nuestro servidor. Ahora lo que nos queda es configurar el Front-end. Para ello, vamos a abrir la ventana de Visual Studio Code donde teníamos el Front-end y vamos a pulsar en el botón que se encuentra en la barra inferior y que dice 'Go Live' (Ver Figura A.9). Cuando pulsemos automáticamente se abrirá una pestaña en el navegador con la aplicación lista para funcionar.



Figura A.9: Inicio de la aplicación a través del proyecto Front-end.

Cuando hayamos realizado todos los pasos descritos anteriormente el resultado será el siguiente (Ver Figura A.10) y habremos desplegado correctamente la aplicación.

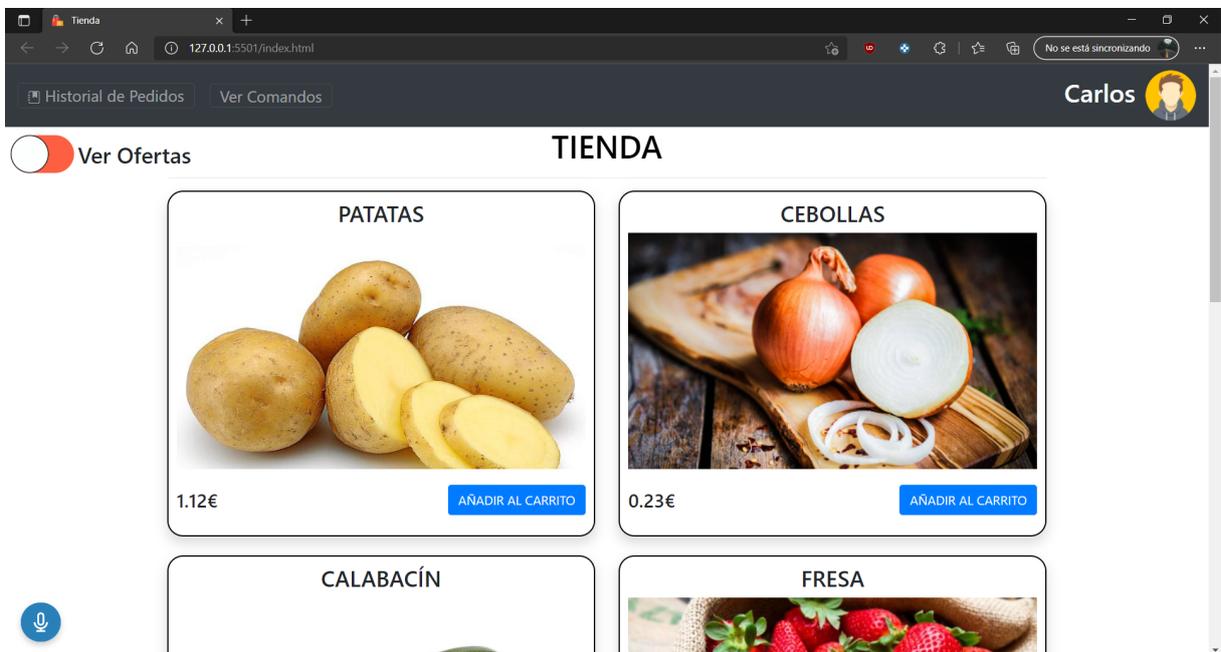


Figura A.10: Aplicación desplegada correctamente.



## Apéndice B

# Manual de Usuario

En esta sección se presenta un Manual de usuario donde se detallan todas las funcionalidades de la aplicación con la intención de que el usuario tenga conocimiento sobre como hacer un correcto uso de ésta. A continuación se dividen las diferentes pantallas de las que dispone la aplicación. Es reseñable saber que en cualquier punto de la aplicación si se quieren utilizar las funcionalidades del asistente virtual, deberá pulsarse el botón azul con un micrófono que se encuentra en la parte inferior izquierda de la pantalla en todo momento.

### B.1. Tienda Catálogo

En esta pantalla el usuario podrá ver todos los productos que se ofrecen en el catálogo. Dispondrán de un nombre, un precio y la posibilidad de añadirlos al carrito. (Ver Figura B.1)

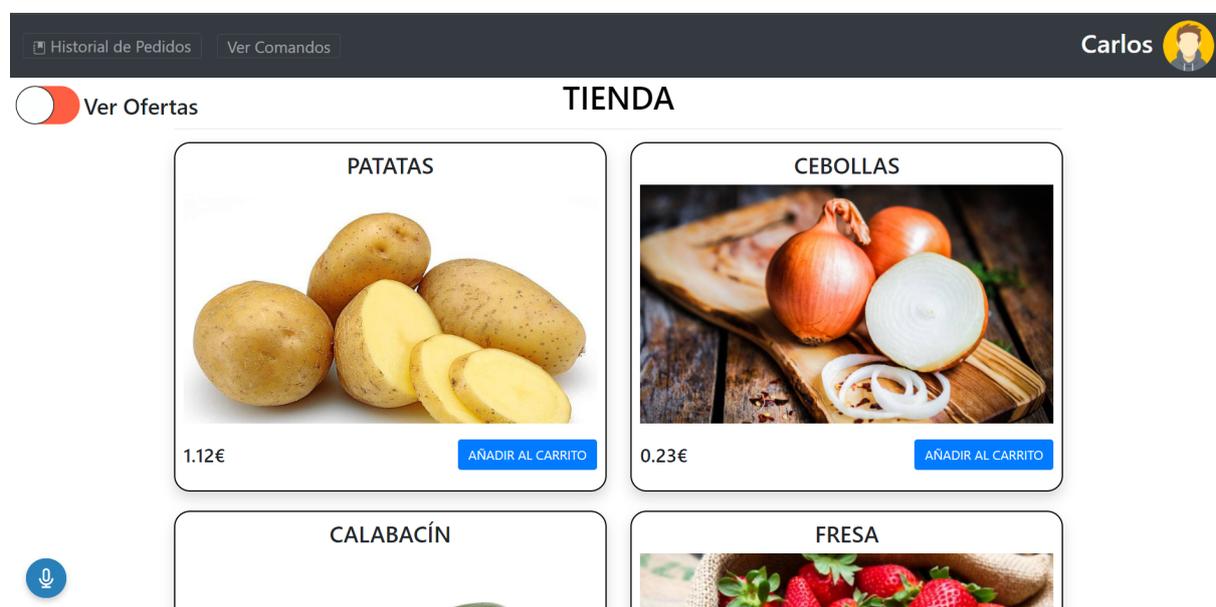


Figura B.1: Tienda Catálogo.

## B.2. Tienda Ofertas

En esta pantalla el usuario podrá ver todos los productos que se ofrecen en el catálogo de ofertas. Dispondrán de un nombre, un precio y la posibilidad de añadirlos al carrito. (Ver Figura B.2)

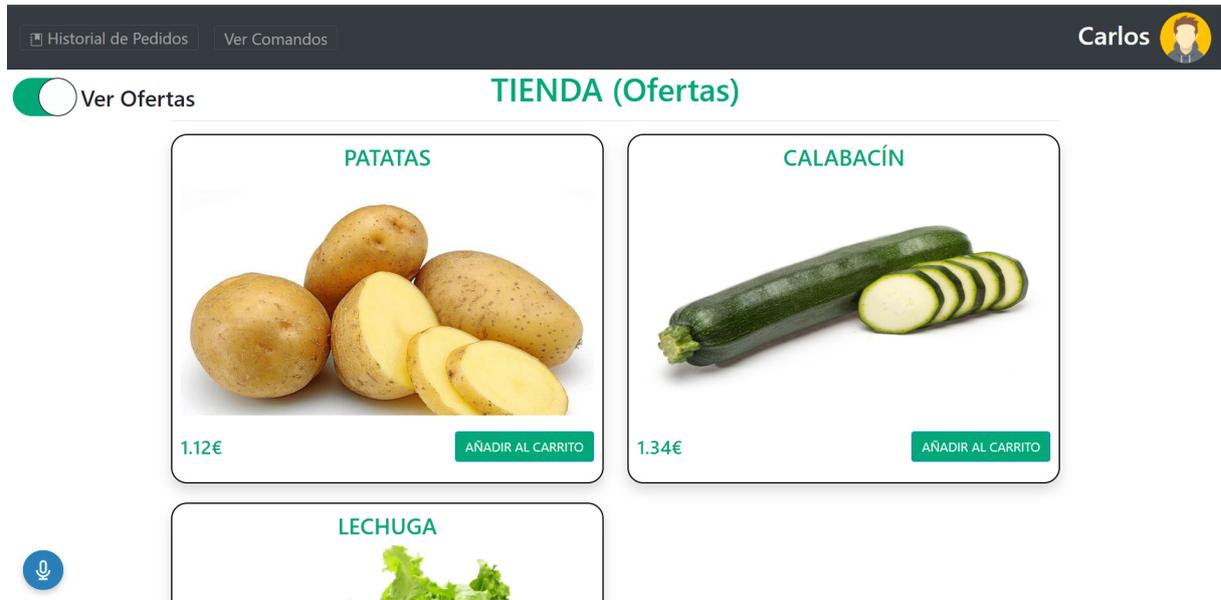


Figura B.2: Tienda Ofertas.

### B.3. Carrito de la Compra

En esta pantalla el usuario podrá ver todos los productos añadidos en el carrito. Dispondrá de la información de cada producto añadido: su foto, nombre, precio y cantidad y el precio total acumulado dentro del pedido. Además, se podrán eliminar individualmente los productos del carrito o vaciar el carrito completo pulsando vaciar. Por último también se ofrece la posibilidad de finalizar el pedido y comprar. (Ver Figura B.3)

CARRITO			
Producto	Precio	Cantidad	
 Calabacín	1.34€	<input type="text" value="1"/>	<input type="button" value="X"/>
 Fresa	3.5€	<input type="text" value="5"/>	<input type="button" value="X"/>
 Lechuga	2€	<input type="text" value="3"/>	<input type="button" value="X"/>
 Patatas	1.12€	<input type="text" value="3"/>	<input type="button" value="X"/>
Total	28.20€		<input type="button" value="Vaciar"/> <input type="button" value="Comprar"/>

Figura B.3: Carrito de la compra.

## B.4. Historial de Pedidos.

En esta pantalla el usuario podrá ver todos los pedidos que ha realizado a lo largo del tiempo. En cada pedido se registra información como el nombre del pedido, email, fecha, Número de productos, precio total del pedido y la lista con una pequeña imagen, el nombre de los productos y su cantidad. (Ver Figura B.4)



Figura B.4: Historial de Pedidos.

## B.5. Tutorial con comandos.

En esta pantalla el usuario podrá ver todos los comandos que existen en la aplicación para poder utilizarlos. Esta pantalla se trata de un modal que se abre en cualquier punto de la aplicación y sirve como tutorial o guía de uso. (Ver Figuras B.5 y B.6)

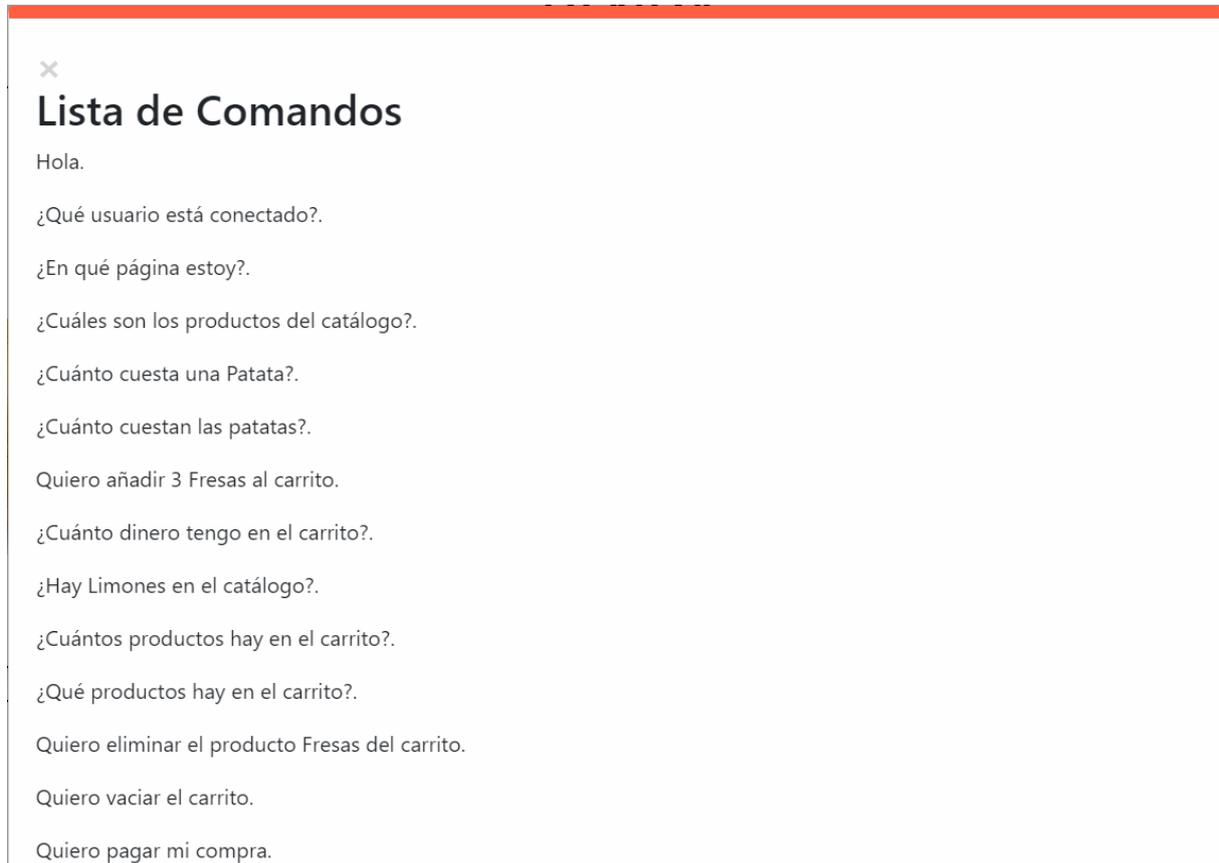


Figura B.5: Comandos disponibles.

¿Cuáles son los productos en oferta del catálogo?.

Quiero ver los productos en oferta del catálogo.

Quiero ver todos los productos del catálogo.

Quiero ir a la tienda.

Quiero ir al historial de pedidos.

¿Cuáles son los pedidos del historial?.

¿Cuáles son los datos del pedido Barbacoa?.

¿Cuáles son los productos del pedido Barbacoa?.

Figura B.6: Comandos disponibles 2.