

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA DE SISTEMAS**

### **DESARROLLO DE UN SISTEMA ASISTIDO POR VOZ Y AYUDA VISUAL ENFOCADO A ORDENES DE PEDIDOS**

#### **TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

**JEFFERSON ALEXANDER TIGSELEMA PACHECO**

jefferson.tigselema@epn.edu.ec

**DIRECTOR: MSC. VICENTE ADRIÁN EGUÉZ SARSOZA**

vicente.eguez@epn.edu.ec

**CODIRECTOR: PHD. SANDRA PATRICIA SÁNCHEZ GORDÓN**

sandra.sanchez@epn.edu.ec

**Quito, abril 2022**

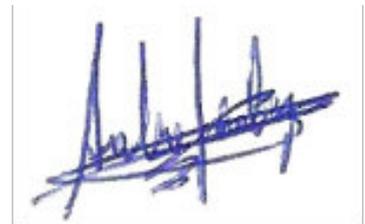
## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por **Jefferson Alexander Tigselema Pacheco**, bajo mi supervisión.



---

**MSC. Vicente Adrián Egüez Sarzosa**  
**DIRECTOR DE PROYECTO**



---

**Ph.D. Sandra Patricia Sánchez Gordón**  
**CODIRECTOR DE PROYECTO**

## DECLARACIÓN

Yo, **Jefferson Alexander Tigselema Pacheco**, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



---

**Jefferson Alexander Tigselema Pacheco**

## DEDICATORIA

A mi madre Isabel, que con su perseverancia me enseñó a luchar por mis sueños, que con su amor me enseñó la bondad y que con su sabiduría me enseñó que la vida no se trata solo del trabajo duro sino de disfrutar cada momento que de felicidad.

A mi padre Patricio, que nunca me dejó solo incluso en los días de lluvia y siempre estuvo pendiente de mí.

A mis hermanos Melissa y Daniel, quienes me enseñaron que puedo ser el ejemplo a seguir de otras personas y aunque a veces me equivoque, siempre que se preocupan por mí.

A mi novia Analy, que me apoya incondicionalmente, incluso cuando siento que no puedo más y que siempre me regala una sonrisa, un abrazo y un lugar seguro.

A mis amigos y amigas, con quienes siempre he compartido mis sueños y aún así no han pensado que estoy loco.

Este trabajo es para todos ellos. Sé que están orgullosos.

Y sobre todo me dedico este trabajo a mí mismo como muestra de que la perseverancia y el hambre de conocimiento me pueden llevar lejos.

## **AGRADECIMIENTOS**

Agradezco a toda mi familia por el apoyo brindado a lo largo de mi vida y estar pendiente de mí. Gracias por todas esas palabras de aliento de cada uno y por creer en mí.

Agradezco también a mis amigos de colegio por una tan verdadera y larga amistad. A Jeanpierre, Fátima, Martín, Jenn, Albi, Ema, Nicole cada uno es una luz cuando la vida se pone oscura.

Agradezco de forma especial a Andrés Torres, cuya amistad se remonta a las épocas de colegio y que, a pesar de las difíciles batallas que tuvimos en la universidad, siempre hubo tiempo para reír en clases o en los pasillos. Su ayuda fue fundamental en este trabajo.

A mis compañeros de Manticore, a quienes llevo en el corazón: Cris, CJ, Kevin, Ali, Óscar que vivieron conmigo la aventura de obtener el primer trabajo.

A mi director de tesis, MSc. Adrián Egüez, quién además de ser un gran maestro, es un gran amigo que comparte el conocimiento sobre la profesión. Dicho conocimiento ha sido el pilar principal en el desarrollo de este trabajo. Y a mi codirectora, Ph.D. Sandra Sanchez cuyas revisiones fueron de gran ayuda para la presentación de este proyecto de titulación.

Finalmente, agradezco a la Escuela Politécnica Nacional por sus enseñanzas y por poder conocer en ella a grandes amigos y profesionales con los cuales he establecido una gran amistad.

Gracias a todos.

# CONTENIDO

<b>Resumen</b>	<b>1</b>
<b>Abstract</b>	<b>2</b>
<b>1 INTRODUCCIÓN</b>	<b>3</b>
1.1 Descripción del problema . . . . .	3
1.2 Objetivos . . . . .	6
1.2.1 Objetivo General . . . . .	6
1.2.2 Objetivos Específicos . . . . .	6
1.3 Alcance . . . . .	6
1.4 Marco teórico . . . . .	7
1.4.1 Aplicaciones web . . . . .	7
1.4.2 Procesamiento del lenguaje natural (PNL) . . . . .	7
1.4.3 WebSockets . . . . .	7
1.4.4 Interfaces Conversacionales . . . . .	8
1.5 Herramientas . . . . .	9
1.6 Descripción de la metodología . . . . .	10
1.6.1 Descripción de Scrum . . . . .	11
1.6.2 Reuniones de Scrum . . . . .	11
1.6.3 Artefactos de Scrum . . . . .	12
1.7 Estructura del Documento . . . . .	13
<b>2 METODOLOGÍA</b>	<b>14</b>
2.0.1 Roles de Scrum . . . . .	14
2.1 Por qué SCRUM . . . . .	15
2.2 Ingeniería de requerimientos . . . . .	17
2.2.1 Recopilación y análisis de información . . . . .	17
2.2.2 Entrevista inicial . . . . .	19
2.2.3 Caso de estudio . . . . .	19
2.3 Diseño de los flujos conversacionales . . . . .	20
2.3.1 Flujos conversacionales . . . . .	20
2.3.2 Dialogflow . . . . .	24

2.4	Definición de historias de usuario . . . . .	28
2.5	Product Backlog . . . . .	28
2.6	Sprints . . . . .	30
2.6.1	Sprint 0 . . . . .	30
2.6.2	Sprint 1 . . . . .	34
2.6.3	Sprint 2 . . . . .	41
2.6.4	Sprint 3 . . . . .	49
2.6.5	Sprint 4 . . . . .	56
2.6.6	Sprint 5 . . . . .	62
2.6.7	Sprint 6 . . . . .	69
2.6.8	Sprint 7 . . . . .	73
2.6.9	Sprint 8 . . . . .	78
2.6.10	Conclusiones sobre la ejecución de los sprints . . . . .	82
<b>3</b>	<b>RESULTADOS</b>	<b>84</b>
3.1	Evaluación de usabilidad del sistema . . . . .	84
3.1.1	Descripción de la metodología SUS . . . . .	84
3.1.2	Realización de la encuesta . . . . .	85
3.1.3	Resultados individuales . . . . .	85
3.2	Producto Final . . . . .	91
<b>4</b>	<b>CONCLUSIONES</b>	<b>97</b>
4.1	Conclusiones y Recomendaciones . . . . .	97
4.2	Recomendaciones . . . . .	98
<b>5</b>	<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>99</b>
<b>6</b>	<b>ANEXOS</b>	<b>I</b>
6.1	Diseño de la aplicación del cliente final . . . . .	I
6.2	Esquema de flujos conversacionales . . . . .	I
6.3	Historias de usuario del proyecto . . . . .	I
6.4	Diagrama de base de datos . . . . .	I
6.5	Sitemap de la aplicación de gestión . . . . .	I
6.6	Encuestas de usabilidad del sistema . . . . .	II
6.7	Imágenes extras del desarrollo de los sprints . . . . .	II

## RESUMEN

En los restaurantes de primera categoría, incluidos los establecimientos de comida rápida, existe cierto nivel de insatisfacción por parte del cliente, si bien, una parte se debe al producto, otra parte se debe a la calidad del servicio que se presentan en dichos locales y a la atención al cliente. Es por esto que en el presente trabajo se presenta una solución que consiste en utilizar un asistente virtual en conjunto con una pantalla que sirva como apoyo visual para la toma de pedidos. Este tipo de aplicaciones pueden mejorar la satisfacción del cliente y también otorga valor al negocio otorgando mayor información para sus clientes.

La solución propuesta utiliza dos aplicaciones frontend utilizando Angular como framework, un backend desarrollado con el framework NestJS y una aplicación de voz desarrollada en Dialogflow con el fin de poder utilizar los dispositivos Google Home o dispositivos basados en Android para la comunicación con el cliente final. De esta forma se gestionan el menú, los productos, categorías, subcategorías, opciones y tipos de opciones que se pueden mostrar al cliente para su elección. Además, se utilizó el marco de trabajo SCRUM para el desarrollo del proyecto con el fin de tener una forma de medición y distribución del trabajo basado en la complejidad de los elementos a desarrollar

Al concluir el desarrollo de las aplicaciones, se evaluó la usabilidad, mediante el uso de la encuesta SUS (System Usability Scale), obteniendo un valor de 77.00 / 100 para la aplicación de toma de pedidos, es decir, el conjunto entre la aplicación de voz y la aplicación web que funge como ayuda visual.

**Palabras clave:** Asistente de voz, Pedidos, Angular, NestJS, Desarrollo web, Usabilidad, Scrum

## ABSTRACT

In first-class restaurants, including fast food establishments, there is a certain level of dissatisfaction on the part of the client, although, a part is due to the product, another part is due to the quality of the service that is presented in these places and the Customer Support. Therefore, the present work presents a solution that consists of using a virtual assistant in conjunction with a screen that serves as visual support for order taking. These types of applications can improve customer satisfaction and also add value to the business through more information for its customers.

The proposed solution uses two frontend applications using Angular as a framework, a backend developed using the NestJS framework and a voice application developed in Dialogflow in order to be able to use Google Home devices or Android-based devices for communication with the client. In this way, the menu, products, categories, subcategories, options and types of options that can be shown to the customer for their choice are managed. In addition, the SCRUM framework was produced for the development of the project in order to have a way of measuring and distributing the work based on the complexity of the elements to be developed.

At the end of the development of the applications, usability was evaluated by using the SUS (System Usability Scale) survey, obtaining a value of 77.00 / 100 for the order taking application, that is, the set between the application of voice and the web application that works as a visual aid.

**Keywords:** Voice assistant, Orders, Angular, NestJS, Web development, Usability, Scrum

# 1 INTRODUCCIÓN

## 1.1 DESCRIPCIÓN DEL PROBLEMA

La función principal de un menú siempre ha sido proveer a los clientes una revisión de los platillos disponibles y apoyar en la toma de decisiones presentando información sobre estos [1]. El menú es una pieza clave de un establecimiento de comida debido a que es parte de la experiencia del proceso de pedir comida, por ello, debe maximizar la satisfacción del cliente. Un menú es mucho más que un elemento de marketing, es un elemento que puede indicar oportunidades de representar la marca o el nivel de cocina del establecimiento [2].

En un restaurante generalmente existen dos formas de presentar los menús con las que el usuario puede interactuar. La forma tradicional que está basada en papel y que comúnmente muestra información sobre varios platillos de comida junto con una pequeña descripción de cada uno y otra en que la presentación de los menús está basada en tecnologías, permitiendo visualizarlos en celulares o tabletas, los cuales proveen un grado más de interactividad y que muestran la misma información que los menús tradicionales además de cierta información adicional (ej. Información nutricional) [1]. También existen otras presentaciones que, dependiendo del tipo de establecimiento, se encuentran en un agregador tales como Uber, Rappi o PedidosYa. Estas son utilizadas para compras en línea, en la cual, un tercero provee un sistema de una sola ventana en la cual el cliente tiene una gran variedad de proveedores de comida rápida y se queda con un margen del costo de la orden para llevar la comida hasta la puerta del consumidor [3]. Debido a la naturaleza de este proyecto, se descartan las presentaciones de los menús provistos por agregadores.

En este sentido, este trabajo pretende utilizar un chatbot presentado mediante una interfaz conversacional con el fin de que sea más amigable con el consumidor y sea percibido como un elemento social, teniendo en cuenta los siguientes parámetros para crear un menú digital [1]:

- ❑ Conservar la información que está disponible en papel menú también en el menú digital.
- ❑ Incluir la capacidad de navegar rápidamente a través de diferentes categorías del menú.
- ❑ Las interacciones sociales no deben estar restringidas por el menú.
- ❑ La información adicional sobre los elementos del menú debe integrarse fundamentalmente y colocarse en un lugar destacado.
- ❑ Las características novedosas deben colocarse en el menú con respecto al público objetivo.
- ❑ Evitar funciones que ejerzan presión sobre el personal o permitan que los empleados se vuelvan superfluos.
- ❑ Si un restaurante tiene una audiencia más joven, el menú digital debe estar disponible durante toda la estancia en el restaurante.

Además, este trabajo se basa en que la interactividad en un menú puede ser un elemento de disfrute con el cual el cliente puede tener una mejor experiencia en el establecimiento. Existen varios aspectos para tener en cuenta a la hora de utilizar tecnología en un menú con respecto al usuario, estas son [4]:

- ❑ La interactividad percibida por el consumidor proviene de la evaluación subjetiva del mismo al momento de interactuar con cualquier tipo de menú. La interactividad percibida en un menú es conceptualizada como el grado en el que la experiencia del cliente es de doble vía, controlable y responde a sus acciones.
- ❑ Los altos niveles de interactividad incrementan el número de veces que el usuario piensa en el producto anunciado en el menú, también conducen al usuario hacia actitudes favorables a la marca y por último, provocan una mayor intenciones de compra.
- ❑ Los menús digitales facilitan el disfrute y mejora la adopción de los productos del menú en clientes que tienen menos experiencia en el establecimiento.
- ❑ El proceso mental anticipatorio de los consumidores les permite imaginar la compra de un producto o servicio, y simular la experiencia de utilizar un producto o servicio en un futuro cercano con el fin de determinar si este uso futuro puede ser beneficioso para sus propósitos de consumo. Esto se lo conoce como visiones de consumo.
- ❑ Experiencia directa con el producto (tocarlo, verlo, sentirlo) puede hacer que los consumidores retengan información del producto y creen sus propias visiones del uso del

producto, resultando en altos niveles de confianza en cuanto al uso del producto.

## **1.2 OBJETIVOS**

### **1.2.1 Objetivo General**

Desarrollar un asistente de voz que permita realizar pedidos con ayuda visual en tiempo real usando Google Dialogflow y WebSockets.

### **1.2.2 Objetivos Específicos**

- Analizar un caso de estudio de un menú que contenga costos y descuentos para la implementación de un flujo conversacional.
- Definir los flujos conversacionales comunes de voz a implementar en base al caso de estudio analizado.
- Diseñar e implementar los flujos de voz en el sistema de pedidos a través de Google Dialogflow.
- Desarrollar una aplicación web que se vincule al sistema de pedidos a través de WebSockets.
- Realizar pruebas de usabilidad para los flujos de voz y la aplicación web.

## **1.3 ALCANCE**

En el presente proyecto integrador se desarrollará una aplicación para una interfaz conversacional con una aplicación web como apoyo visual para clientes de un establecimiento de comida rápida, así como, una aplicación de gestión para que un usuario dueño del establecimiento pueda gestionar la información que se muestra en la primera. Dada la naturaleza de este trabajo se enfocará principalmente en crear una experiencia amigable con el usuario para la toma de pedidos en un establecimiento de comida rápida.

## **1.4 MARCO TEÓRICO**

### **1.4.1 Aplicaciones web**

Una aplicación web es un programa de aplicación que se almacena en un servidor remoto y se entrega al usuario mediante un navegador web [5]. Una aplicación web también es un sitio web cuyo contenido final se determina cuando un usuario solicita una página al servidor web. La respuesta de la página varía de acuerdo con la petición del usuario y por eso también se considera una página dinámica [6].

### **1.4.2 Procesamiento del lenguaje natural (PNL)**

El procesamiento del lenguaje natural (NLP por sus siglas en inglés) es una rama de la inteligencia artificial que permite a las computadoras entender e interpretar el lenguaje humano. EL PLN incluye disciplinas diversas como las ciencias de la computación o lingüística computacional. Este tipo de tecnología se basa en técnicas de estadística y aprendizaje profundo lo cual permite a dispositivos como Alexa o Google Home interactuar con personas [7].

El procesamiento de lenguaje natural es importante porque permite extraer información de los volúmenes de datos no estructurados que se encuentran en internet. Además, permite que una computadora identifique palabras repetidas, intenciones o sentimientos de un texto creado por un humano; también una computadora dotada de PLN puede conversar con una persona entendiendo una acción y respondiendo correctamente a ella.

### **1.4.3 WebSockets**

WebSockets es una tecnología que hace posible abrir una sesión de comunicación entre el navegador del usuario y un servidor. Esta tecnología permite enviar mensajes a un servidor y recibir respuestas que son controladas por eventos, los WebSockets se utilizan para realizar aplicaciones en tiempo real como aplicaciones de chat, domótica y se pueden utilizar para mostrar información en tiempo real de sensores y otro tipo de dispositivos [8]. A continuación, se muestra en la Figura 1.1 una comparativa entre las arquitecturas de HTTP y

WebSockets.

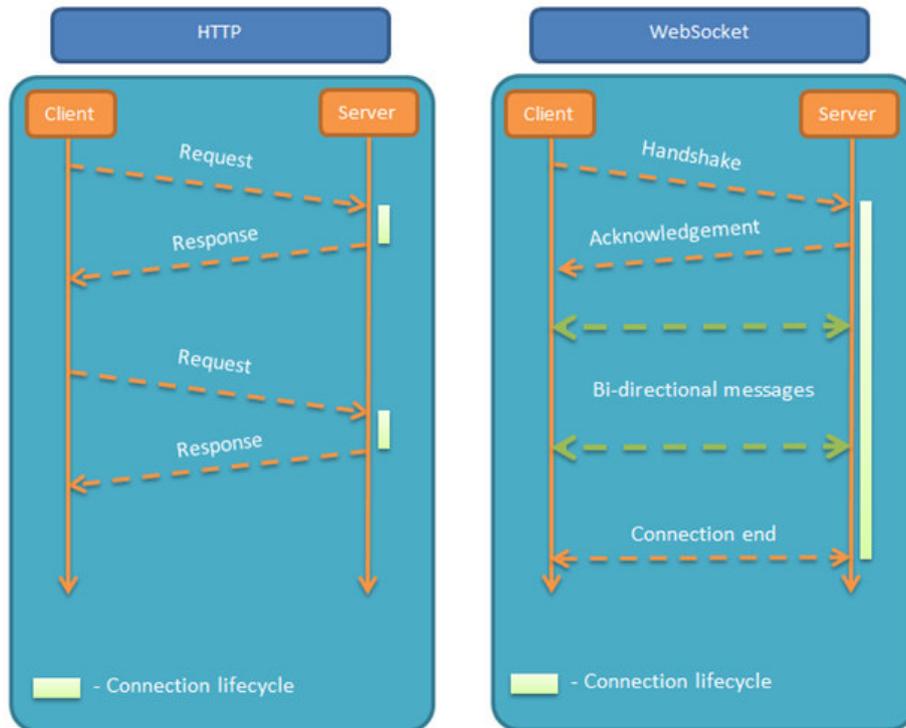
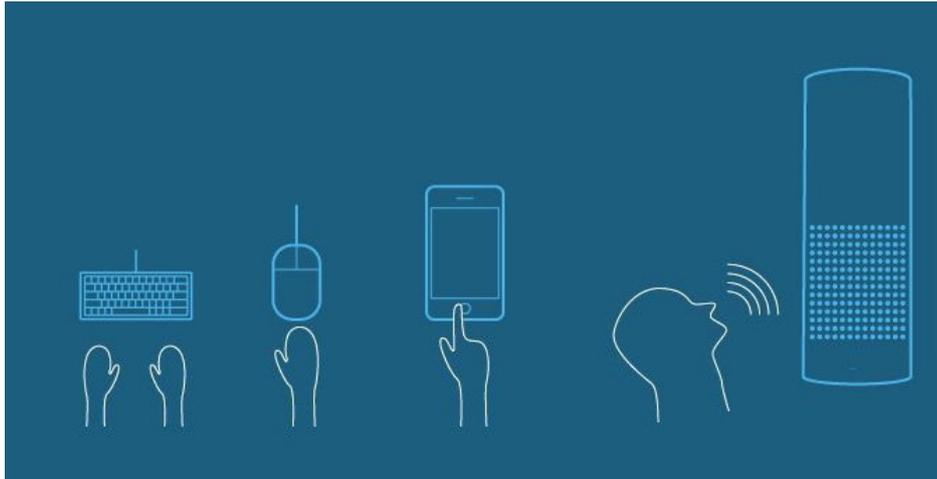


Figura 1.1: Arquitectura de comunicación HTTP y WebSockets [9].

#### 1.4.4 Interfaces Conversacionales

Las interfaces conversacionales son interfaces de usuario que simulan una conversación humana. El objetivo de este tipo de interfaces es ofrecer al usuario la información que requiere de forma intuitiva y simple. Algunos tipos de interfaces conversacionales son los asistentes de voz y los chatbots [10]. En la Figura 1.2 se puede apreciar, de manera gráfica, algunos tipos de interfaces utilizados cotidianamente.

Las interfaces conversacionales se utilizan principalmente como herramientas de atención al cliente. Debido al auge de las redes sociales, los chatbots puede solventar dudas de los clientes y esa es la razón por la que varias marcas las utilizan. Alexa o Google Home utilizan este tipo de interfaces para hacer la vida de los usuarios más fácil.



**Figura 1.2:** Tipos de interfaces [11].

## 1.5 HERRAMIENTAS

Para la implementación de las aplicaciones propuestas para el presente proyecto de titulación, se ha hecho uso de las herramientas descritas en la Tabla 1.1 a continuación:

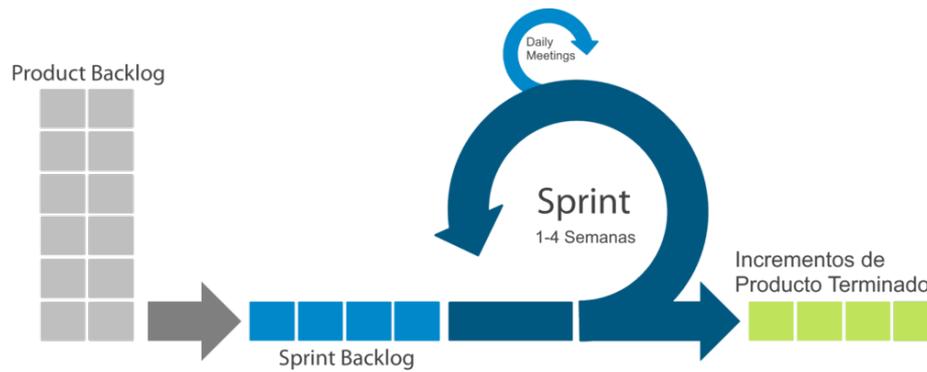
**Tabla 1.1:** Herramientas a utilizar.

Nombre	Detalle
Visual Studio Code	Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Viene con soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C++, C, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity) [12].
Google Dialogflow	Dialogflow es una plataforma con comprensión del lenguaje natural que facilita el diseño de una interfaz de usuario de conversación y su integración a tu aplicación para dispositivos móviles, aplicaciones web, dispositivos, bots, sistemas de respuesta de voz interactiva y más [13].
Gitlab	GitLab es una plataforma de desarrollo de software de código abierto de extremo a extremo con control de versiones integrado, seguimiento de problemas, revisión de código, CI / CD y más. GitLab autohospeda los proyectos en sus propios servidores, en un contenedor o en un proveedor en la nube [14].

NestJs	Nest (NestJS) es un marco para crear aplicaciones de servidor Node.js escalables y eficientes. Utiliza JavaScript progresivo, está construido con y es totalmente compatible con TypeScript (pero todavía permite a los desarrolladores codificar en JavaScript puro) y combina elementos de OOP (Programación orientada a objetos), FP (Programación funcional) y FRP (Programación reactiva funcional) [15].
Angular	Angular es un marco de diseño de aplicaciones y una plataforma de desarrollo para crear aplicaciones de una sola página eficientes y sofisticadas [16].
MySQL	MySQL es la base de datos de código abierto más popular del mundo. A pesar de sus potentes funciones, MySQL es fácil de configurar y usar [17].
Socket.IO	Socket.IO es una biblioteca que permite la comunicación en tiempo real, bidireccional y basada en eventos entre el navegador y el servidor [18].
Docker	Docker es un empaquetador de software en unidades estándar llamadas contenedores que incluyen todo lo necesario para que se ejecute el software, Docker empaqueta bibliotecas, herramientas de sistema y código. Es decir, Docker es una plataforma de software que permite crear, probar e implementar aplicaciones [19].

## 1.6 DESCRIPCIÓN DE LA METODOLOGÍA

La metodología que se utilizará en el presente trabajo es Scrum debido a que permite la organización de tareas complejas en fragmentos más pequeños llamados historias de usuario las cuales hacen que el proyecto sea más manejable y adaptativo lo que provoca que el proyecto se lleve de la mejor manera. Además, Scrum permite entregar un producto final (Increment) luego de un periodo de tiempo el cual se denomina Sprint [20]. En la Figura 1.3 se muestra el proceso de Scrum de manera gráfica, donde se enfatiza la etapa de desarrollo del sprint como una etapa iterativa.



**Figura 1.3:** Proceso de SCRUM [21].

## 1.6.1 Descripción de Scrum

Scrum es una metodología diseñada para ofrecer un valor significativo y de forma rápida durante todo el proyecto, esta metodología crea un ambiente de responsabilidad colectiva y progreso continuo mediante equipos multifuncionales y auto organizados. Scrum divide el trabajo en ciclos cortos. Algunas ventajas de usar Scrum son [22]:

- ❑ **Adaptabilidad:** Debido a las entregas iterativas los proyectos se adaptan y están abiertos a cambios.
- ❑ **Retroalimentación continua:** La retroalimentación se realiza mediante las reuniones de seguimiento del proyecto que son diarias por parte del equipo.
- ❑ **Entrega continua de valor:** Al término de cada sprint se hace una entrega al cliente.
- ❑ **Resolución de problemas de forma rápida:** La colaboración y formación de equipos multifuncionales resuelven los problemas con mayor rapidez.

## 1.6.2 Reuniones de Scrum

### Sprint planning

Las reuniones de sprint planning son conducidas por el product owner, los miembros del equipo y el scrum master. El product owner es responsable de priorizar las tareas y así definir cuáles son las tareas más importantes. Esta reunión se enfoca en lo que hay que hacer y como hay que hacerlo [23].

### Sprint

Un sprint se define como una iteración que dura de una a seis semanas. Los cambios no son aceptados mientras el sprint está en curso [23].

### **Daily Scrum**

Los miembros del equipo y el Scrum master entre ellos reportan las actividades realizadas el día anterior. Estas reuniones deben durar alrededor de 15 minutos y se deben contestar las siguientes preguntas [23]:

- ¿Qué hicieron ayer?
- ¿Qué tienen planeado hacer hoy?
- ¿Que dificultades pueden encontrar en el cumplimiento de sus tareas?

### **Sprint review**

El equipo de Scrum presenta los entregables del Sprint al product owner con el fin de que este lo revise y los compare con los criterios de aceptación. Este los acepta o rechaza de acuerdo con las historias de usuario previamente acordadas. Esta reunión se realiza después de cada sprint [22].

### **Sprint retrospective**

Se discute sobre las experiencias y los problemas que se tuvo en el sprint y también se discute sobre lo que se debe y no se debe seguir haciendo, además de lo que se puede mejorar. Esta reunión dura por lo general entre 15 y 30 minutos se realiza después de cada sprint y participa todo el equipo [23].

## **1.6.3 Artefactos de Scrum**

### **Product Backlog**

El product backlog es una lista de requerimientos por producto que está ordenada por la prioridad decidida por el product owner del proyecto. El product owner puede volver a priorizar la lista al final de cada sprint. Esto es transparente para todos los interesados [23].

### **Sprint backlog**

El sprint backlog es una lista de trabajo que el equipo de desarrollo debe abordar durante el sprint. Estos requerimientos no deben cambiar durante la ejecución del sprint [23].

### **Burndown chart**

El sprint burndown chart es un gráfico que muestra el trabajo pendiente en el backlog del sprint, este gráfico es actualizado cada día a medida que el sprint progresa. Esto provee una visualización rápida del trabajo que se está realizando [23].

## **1.7 ESTRUCTURA DEL DOCUMENTO**

El presente trabajo está organizado de la siguiente manera: En la sección 2 se describe la metodología utilizada. En la sección 3, se presentan los resultados y la discusión por cada uno de los prototipos desarrollados. Finalmente, en la sección 4 se presentan las conclusiones y las recomendaciones del presente trabajo de titulación.

## 2 METODOLOGÍA

### 2.0.1 Roles de Scrum

Un Scrum team es una unidad pequeña, un equipo de personas en el cual no existen subequipos ni jerarquías, el Scrum team es una unidad cohesionada de profesionales enfocados en un objetivo a la vez. Los roles dentro de un Scrum Team son [24]:

- ❑ **Developers:** Son las personas del Scrum Team que deben crear cualquier aspecto de un Increment en un Sprint. Los miembros del equipo pueden ser programadores, diseñadores, testers o deben tener algún perfil que ayude al equipo.
- ❑ **Product Owner:** Es la persona responsable del maximizar el valor del Increment mediante la gestión efectiva del conjunto de tareas (Product Backlog). Es el encargado de definir las características del producto y también decide la fecha del lanzamiento del producto junto con el contenido, también es el responsable de la aprobación final y de su aceptación o rechazo de los resultados del trabajo [23] .
- ❑ **Scrum Master:** Es la persona responsable de establecer Scrum dentro del Scrum team y de esta forma lograr efectividad mediante la mejora de las prácticas. Ayuda a los miembros del equipo y se responsabiliza por remover obstáculos. Tiene un rol de líder sobre los miembros del equipo y los protege de interferencias externas [23].

Los roles en el presente proyecto se encuentran definidos tal como se muestran en la Tabla 2.1:

**Tabla 2.1:** Roles Scrum

<b>Rol Scrum</b>	<b>Responsable</b>
Developer	Alexander Tigselema
Scrum Master	Ing. Cristian Lara
Product Owner	Msc. Adrián Egúez

## 2.1 POR QUÉ SCRUM

A continuación, se hará el análisis de Scrum frente a otras metodologías ágiles conocidas para así aclarar el por qué se ha utilizado Scrum. En la Tabla 2.2 se presenta un breve resumen de distintas metodologías que se consideraron para la implementación del presente proyecto.

**Tabla 2.2:** Tabla comparativa de metodologías ágiles [25]

Metodología	Puntos clave	Características especiales	Experiencia
<b>Desarrollo de Software Adaptativo (ADS)</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Adaptado a la cultura organizacional</li> <li><input type="checkbox"/> Colaborativo</li> <li><input type="checkbox"/> Componentes basados en la misión (objetivo)</li> <li><input type="checkbox"/> Desarrollo iterativo</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Las organizaciones son vistas como sistemas adaptativos</li> </ul>	Sin experiencia práctica con la metodología
<b>Desarrollo basado en funcionalidades (FDD)</b>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Proceso de cinco pasos</li> <li><input type="checkbox"/> Orientado a objetos</li> <li><input type="checkbox"/> Desarrollo basado en componentes</li> </ul>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Método simple</li> <li><input type="checkbox"/> El diseño y la implementación de sistemas se realizan por características</li> <li><input type="checkbox"/> Modelado de objetos</li> </ul>	Sin experiencia práctica con la metodología

<p><b>Scrum</b></p>	<ul style="list-style-type: none"> <li>❑ Equipos de desarrollo independientes, pequeños y auto organizados.</li> <li>❑ Ciclos de desarrollo de 30 días</li> </ul>	<ul style="list-style-type: none"> <li>❑ Impone un cambio de paradigma de “definido y repetitivo” a “nueva visión de desarrollo de productos”</li> </ul>	<ul style="list-style-type: none"> <li>❑ Experiencia práctica con la metodología en proyectos reales</li> <li>❑ Experiencia con ciclos de desarrollo de 14 días utilizando la metodología</li> </ul>
<p><b>XP</b></p>	<ul style="list-style-type: none"> <li>❑ Desarrollo orientado al cliente</li> <li>❑ Equipos pequeños</li> <li>❑ Compilaciones diarias</li> </ul>	<ul style="list-style-type: none"> <li>❑ Utiliza la técnica del refactoring sobre diseño actual del sistema para mejorar su rendimiento y también su capacidad de respuesta</li> </ul>	<p>Sin experiencia práctica con la metodología</p>

Como se aprecia en la tabla anterior, entre las razones principales por las cuales se ha escogido Scrum se tienen la experiencia en proyectos reales que se tiene con la metodología y la adaptabilidad de los tiempos de desarrollo que tiene la misma. Se descartaron las metodologías ADS y FDD debido a que eran más ligadas a la organización, en el primer caso, y que los diseños debían hacerse de manera individual para cada funcionalidad, en el segundo caso. La metodología XP se la descartó debido a que la metodología se adapta mejor en desarrollos de mínimo un equipo de dos personas.

## 2.2 INGENIERÍA DE REQUERIMIENTOS

La meta del proceso de ingeniería de requerimientos es crear y mantener un documento de requerimientos del sistema [26]. Para el presente proyecto, el documento de requerimientos estará compuesto por las historias de usuario, con sus respectivos pesos y criterios de aceptación. También se realizó recopilación de información para conocer el estado actual de la tecnología aplicada en menús incluido los menús tradicionales.

### 2.2.1 Recopilación y análisis de información

Se recopiló información sobre los distintos tipos de menús, y las diferentes tecnologías que se aplican a estos. Para esto se realizó revisión de la literatura tanto sobre menús tradicionales así como en menús que se presentan en aplicaciones móviles además de un apartado sobre menús digitales que hacen uso chatbots, los cuales, no han sido explorados en su totalidad todavía.

Los menús tradicionales son menús basados en papel en los cuales el usuario puede ver los platillos que ofrece el establecimiento, junto con una pequeña descripción de estos [1]. Para interactuar con el menú el usuario solamente debe pasar la página y ver un ítem que le interese y leer la mucha o poca información que se le provea [4]. Los menús tradicionales tienen varios problemas que se pueden mencionar [1]:

- ❑ El menú cambia frecuentemente lo que aumenta los costos y el esfuerzo, es por ello que los restaurantes muchas veces optan por no hacer cambios en el menú [1].
- ❑ Debido a la localización del restaurante no se puede tener un menú plurilingüe tal y como se querría, para captar el interés de usuarios que no hablen el mismo idioma. Por lo cual es una opción que los menús se encuentren en el idioma nativo del país además del idioma inglés.
- ❑ El menú basado en papel no incluye demasiadas fotografías de los platillos con el fin de mantenerlo compacto y barato.
- ❑ Los dueños de los establecimientos identificaron [1] que existe una falta de información en el menú por ejemplo el origen y la información nutricional. La falta de esta información hace que el consumidor no tenga en cuenta diferentes aspectos como la salud.

Los menús digitales, por otro lado, son la nueva apuesta de los restaurantes con el fin de aumentar sus ventas, por ejemplo McDonalds ha aumentado un 3.5 % las ventas en sus kioscos digitales solamente modificando los productos que se visualizan dependiendo de la condición del tiempo. Existe una gran aceptación de los menús digitales por parte de los usuarios debido a que les permite buscar un producto más fácilmente, también obtienen más información sobre un platillo que en un menú tradicional. Se ha encontrado que los menús digitales tienen los siguientes efectos en los consumidores [4]:

- ❑ Una mayor interactividad proporciona a los consumidores las oportunidades para acceder a más información para evocar y participar en experiencias más cognitivas, lo que resulta en un mejor aprendizaje del menú.
- ❑ El proceso de buscar comida en tablet requiere un aprendizaje previo de cómo manejar aparatos electrónicos con sus gestos (swiping, double tapping, pinch to zoom) mientras que en un menú tradicional solo se cambia de página.
- ❑ Los hallazgos mostraron que los menús digitales, en comparación con los menús tradicionales en papel, pueden tener un efecto adverso o nulo en las evaluaciones de los consumidores de alimentos experimentados en el establecimiento.
- ❑ El diseño web tiene una incidencia en la facilidad de uso percibida [27].
- ❑ La información entregada a los consumidores debe ser precisa y las descripciones detalladas [27].

También existen menús que han sido poco explorados como son los menús en chatbots. En los pocos estudios académicos que existen se puede notar que [28]:

- ❑ Los chatbots tienen una ventaja de comunicación sobre los otros métodos de pedidos porque permiten a los clientes omitir pasos en el proceso u ordenar de manera tan orgánica como lo harían con un servicio personal.
- ❑ Los consumidores que utilizan el chatbot para ordenar en restaurantes de servicio rápido informaron actitudes cognitivas más altas en comparación con aquellos que ordenaron en restaurantes de servicio completo.
- ❑ El método de chatbot se utilizó para elementos de menú simples y productos básicos. El método de teléfono es adecuado para elementos especiales o más complicados, y el método en línea para elementos y complementos más costosos.
- ❑ Los chatbots se utilizan a menudo para responder solicitudes simples, mientras que

los humanos tienen que reemplazarlos cuando las situaciones se vuelven más complejas.

- ❑ Los chatbots no son percibidos tan sociales como los seres humanos.
- ❑ Los restaurantes de servicio rápido con diseños de menú simples pueden convertirse en los primeros en adoptar los chatbots debido a que poseen una asignación presupuestaria más alta para tecnología.

## **2.2.2 Entrevista inicial**

Para la obtención del documento de requerimientos (historias de usuario) se hizo uso de la entrevista como proceso de ingeniería de requerimientos, además de la información recopilada sobre menús digitales y establecimientos de comida descritos en secciones anteriores. La razón por la cual no se hizo uso netamente de la entrevista se debe a que se tiende a omitir información esencial [26] durante la misma y por esto, solo será utilizada como complemento a la información ya recopilada. La reunión se realizó con el product owner en la cual se definieron los siguientes criterios para el desarrollo de este proyecto.

- ❑ El dueño de un establecimiento debe gestionar los menús mediante una aplicación web.
- ❑ El cliente final puede hacer un pedido mediante su voz
- ❑ El cliente final puede tener un apoyo visual de su pedido cuando está conversando con la aplicación de voz.
- ❑ El sistema debe ser estable para brindar confianza al consumidor final.

## **2.2.3 Caso de estudio**

En el caso de los menús tradicionales, los clientes del establecimiento usan un menú físico para escoger los productos de su preferencia. Si existen descuentos o promociones, se informa al cliente al momento de tomar la orden o en otros casos, se modifican los costos del menú con los valores promocionales. De acuerdo con lo anteriormente mencionado, el caso de estudio que se propone para el presente proyecto se enfoca netamente en los menús digitales a través de la creación de una aplicación de voz para realizar órdenes de pedidos.

Para la realización de la aplicación de voz, se pretende partir de los productos de un menú tradicional perteneciente a un local de comida rápida y adaptarlo de tal manera que pueda ser utilizado como ayuda visual para la aplicación de voz. De acuerdo con la literatura consultada, este tipo de aplicaciones debe estar dirigida hacia personas que frecuentemente consumen en el local de comida rápida y tengan buenos conocimientos del menú de forma previa. Esto con el fin de disminuir la incomodidad de los usuarios que utilizan aplicaciones de voz.

Al pasar de un menú tradicional a uno digital, se presenta la ventaja de que, en caso de descuentos o promociones, se puede actualizar fácilmente los valores de los productos y el cliente conocerá de antemano dichos valores promocionales.

Adicional a lo previamente dicho, existirá una aplicación de gestión la cual se enfoca en la creación del menú, categorías, subcategorías, productos y opciones, con el fin de tener una forma de gestionar la aplicación que se muestra al cliente final.

La aplicación de voz funcionará en conjunto con la aplicación de ayuda visual las cuales harán uso de WebSockets para comunicarse entre sí; además, ambas aplicaciones compartirán el backend conjuntamente con la aplicación de gestión.

## **2.3 DISEÑO DE LOS FLUJOS CONVERSACIONALES**

Los flujos conversacionales son muy importantes en las aplicaciones que incluyen un asistente de voz, se definen como los temas o conversaciones con una finalidad específica que puede tener un aplicación de voz [29]. Por ejemplo: una aplicación de voz para realizar pedidos puede tener un flujo para navegar por el menú y otro flujo para realizar el pedido. Cada uno de estos flujos se definen por la pregunta que hace el usuario y la respuesta que obtiene de la aplicación.

### **2.3.1 Flujos conversacionales**

Los flujos conversacionales para este proyecto se pueden ver más a detalle en el Anexo **6.2**.

Este proyecto contará con tres flujos de conversación: el primer flujo es el flujo de navegación por el menú y compra de un producto, el segundo flujo será utilizado para las acciones

que se realicen con el carrito y el tercer y último flujo es para terminar compra.

Para llamar al bot se utilizará el comando: “Ok Google, hablar con tesis restaurante“. Este será la llamada de invocación de la aplicación de voz mediante el asistente de Google. En la Figura 2.1 a continuación, se muestra el diagrama correspondiente al inicio del flujo.



**Figura 2.1:** Inicio del flujo

Antes de describir más a detalle los distintos flujos, es necesario aclarar que la notación con el símbolo “\$“ hace referencia a variables. Por ejemplo, al decir \$producto se refiere a cualquier producto existente en el menú más no a un producto en específico; la misma lógica se aplica con cualquier otra variable que aparezca a continuación.

### **Primer flujo**

Una vez que se ha llamado a la aplicación se tendrán los siguientes comandos para navegar por la aplicación y para realizar compras dentro de la aplicación:

- Ir al menú
- Subir
- Bajar
- Que tiene en \$categoria, \$subcategoria
- Quiero ver más información de \$producto
- Quiero comprar \$cantidad de \$producto con \$opciones[]

## ☐ Ayuda

En la Figura 2.2 se puede observar el flujo construido en base a los comandos anteriormente descritos.

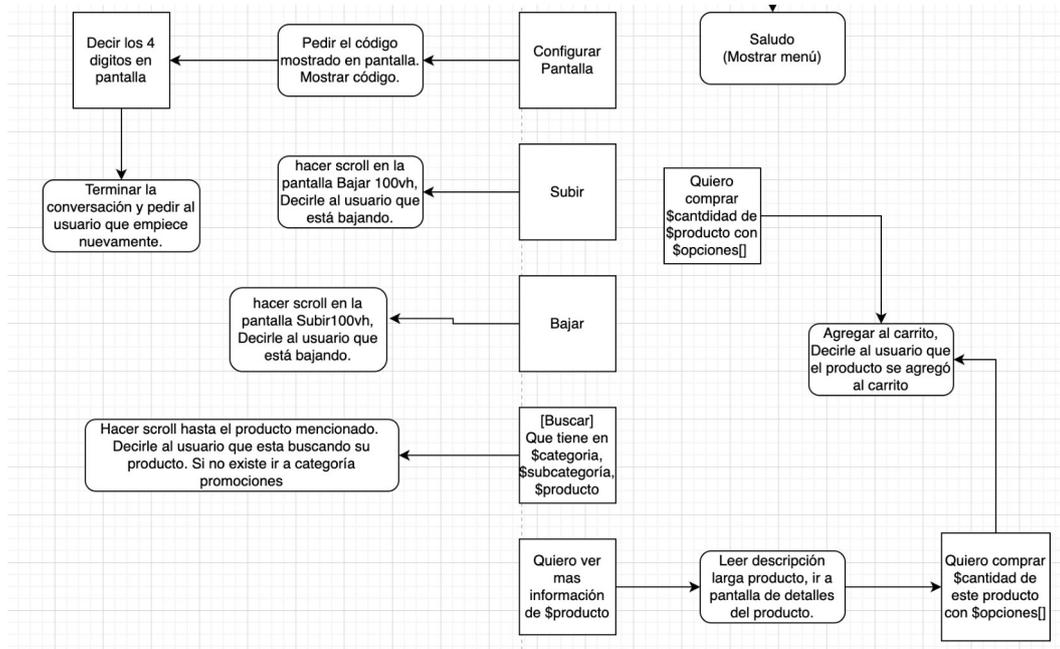


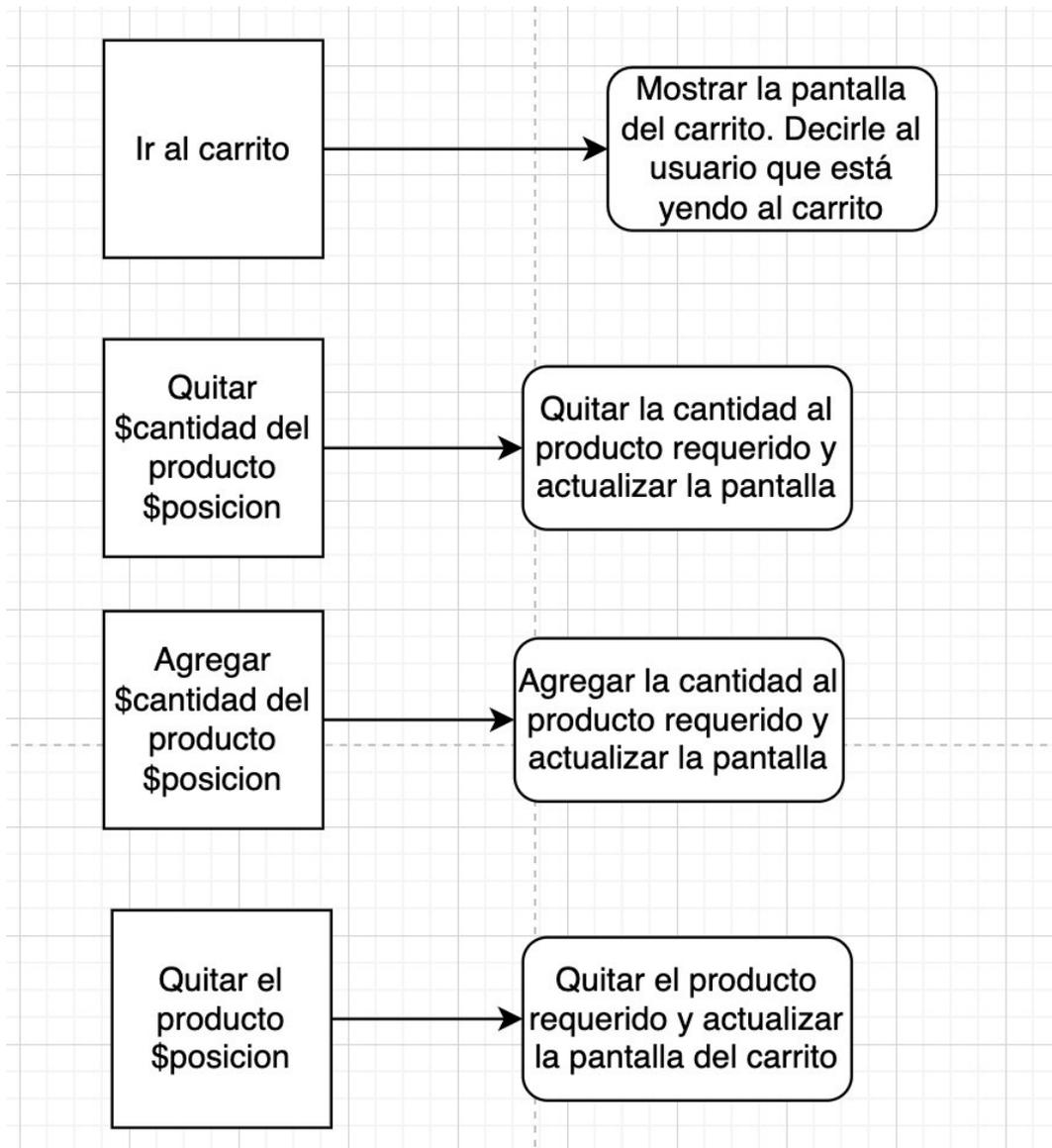
Figura 2.2: Primer flujo

## Segundo flujo

Luego de navegar por el menú y agregar uno o varios productos al carrito, se puede navegar hacia la pantalla del carrito nuevamente e interactuar mediante los siguientes comandos:

- ☐ Ir al carrito
- ☐ Quitar \$cantidad del producto \$posición
- ☐ Agregar \$cantidad del producto \$posición
- ☐ Quitar el producto \$posición

En la Figura 2.3 se aprecia el diagrama correspondiente a las acciones relacionadas con la cantidad del producto.



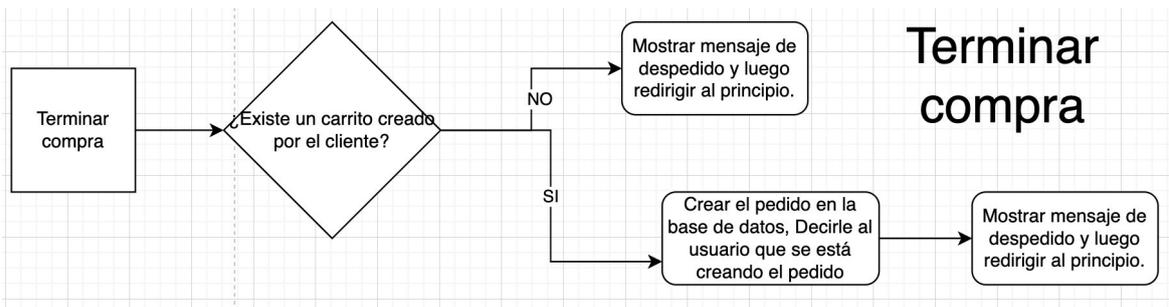
**Figura 2.3:** Segundo flujo

### Tercer flujo

Por último, una vez que el cliente final está de acuerdo con su carrito y ya desea terminar su compra, podrá hacerlo con el siguiente comando:

- ❑ Terminar compra

En la Figura 2.4 se indica el diagrama para terminar la compra; en caso de que el cliente haya agregado productos al carrito, se guardará a nivel de base de datos y se indicará que se está creando el pedido, caso contrario, se mostrará un mensaje de despedida y se volverá al principio.



**Figura 2.4:** Tercer flujo

### 2.3.2 Dialogflow

Dialogflow es una plataforma con comprensión del lenguaje natural que facilita el diseño de una interfaz de usuario de conversación y una integración a una aplicación para dispositivos móviles, aplicaciones web, dispositivos, bots, sistemas de respuesta de voz interactiva y más [13].

Antes de explicar el proceso de creación de una aplicación de voz utilizando Dialogflow, es necesario familiarizarse con algunos de los conceptos que maneja:

- ❑ **Entidad:** Una entidad es una palabra que se utiliza para detectar coincidencias para presentar los datos de forma personalizada o en su defecto datos como fechas, colores o nombres que son entidades que vienen predeterminados por el sistema.
- ❑ **Intent:** Un intent clasifica la intención de un usuario final para un turno de conversación.
- ❑ **Fulfillment:** Es el modo de entrega de respuesta que se define por cada intent. Cuando se activa la opción de fulfillment la aplicación sabe que debe hacer una llamada mediante webhook al servicio destinado para responder a la aplicación.
- ❑ **Webhook:** El webhook se refiere a un servicio externo que generalmente contiene la lógica empresarial. Los webhooks permiten usar los datos extraídos de la aplicación de Dialogflow a fin de generar respuestas dinámicas, validar datos recopilados o activar acciones en el backend.
- ❑ **Frases de entrenamiento:** Las frases de entrenamiento son frases de ejemplo de algo que podrían decir o escribir los usuarios finales, conocidas como entradas del

usuario final. Cuando la entrada del usuario final se parece a una de estas frases, Dialogflow hace una coincidencia con el intent.

Para crear una nueva aplicación de voz en Dialogflow primero se debe poner el nombre de la aplicación y elegir el idioma predeterminado con el que se va a comunicar la aplicación. En la Figura 2.5 se muestra la pantalla de creación de una nueva aplicación en Dialogflow.

tesis-bot CREATE

---

DEFAULT LANGUAGE ⓘ DEFAULT TIME ZONE

Spanish – es (GMT-5:00) America/New\_York

Primary language for your agent. Other languages can be added later. Date and time requests are resolved using this timezone if not provided in the API requests.

---

GOOGLE PROJECT

Create a new Google project ▼

Enables Cloud functions, Actions on Google and permissions management.

---

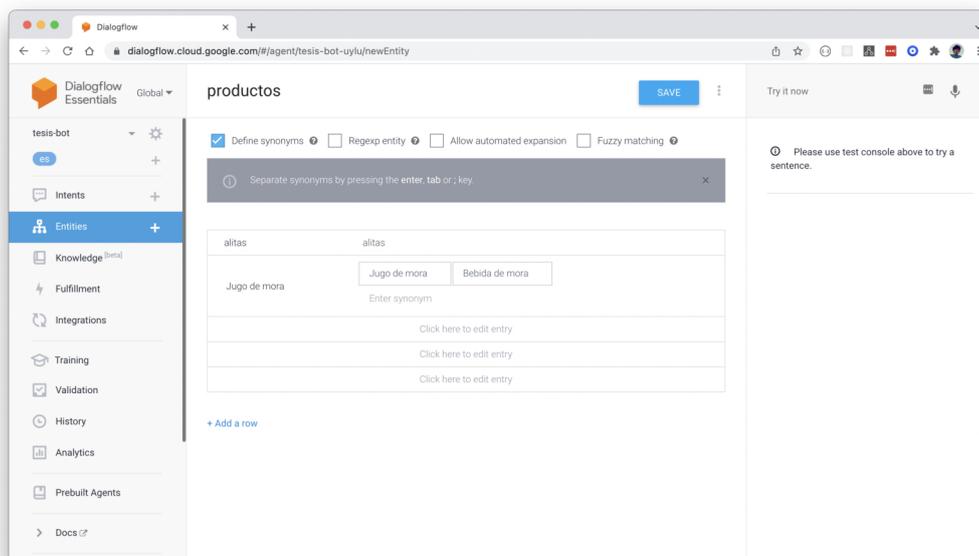
AGENT TYPE

Set as Mega Agent

Combine multiple Dialogflow agents (i.e. sub agents) into a single agent (i.e. [mega agent](#)).

**Figura 2.5:** Crear aplicación en Dialogflow

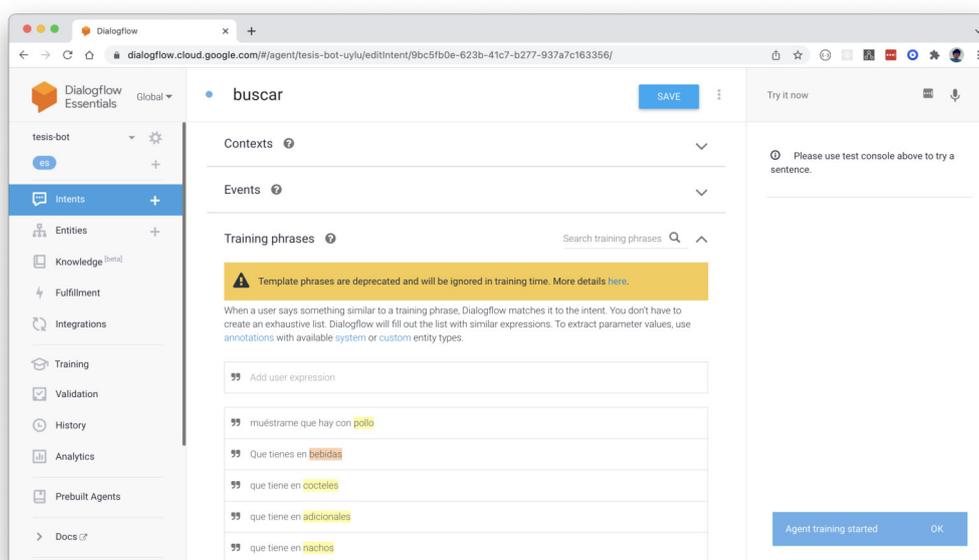
Luego se debe configurar la lista de *entidades* especiales, como se muestra en la Figura 2.6, que la aplicación debe reconocer esto incluye la lista de nombres de categorías, sub-categorías y el nombre de los productos así como el nombre de las opciones para que la aplicación pueda reconocer e interactuar con ellas.



**Figura 2.6:** Entidades

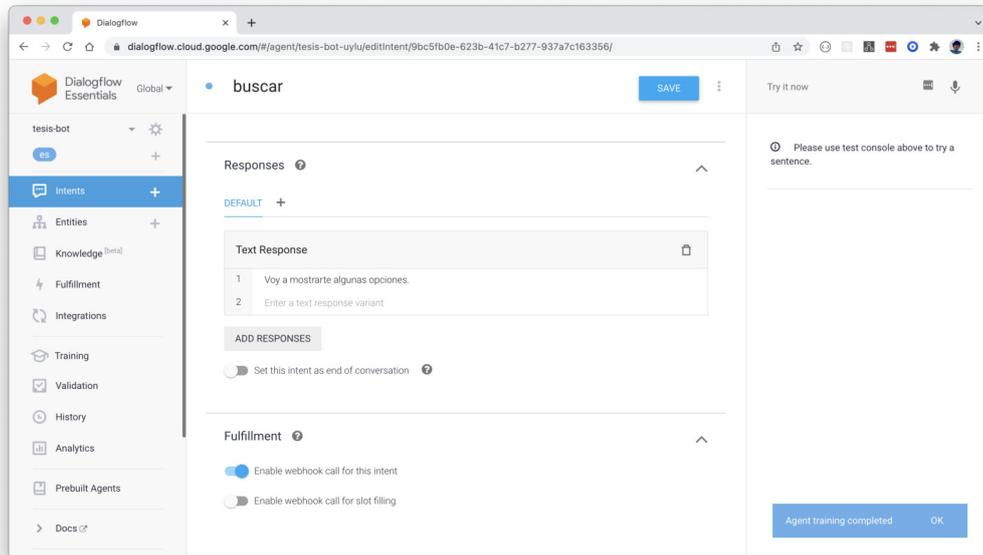
También se deben configurar los *intent* de acuerdo con los flujos definidos previamente. Los *intents* en su parte principal son las frases de entrenamiento en las cuales se definen las frases que el usuario puede decir para activarlo.

Una vez que se definen las frases de entrenamiento se deben definir las *frases con la que la aplicación puede responder*, en la Figura 2.7 se puede observar un Ejemplo.



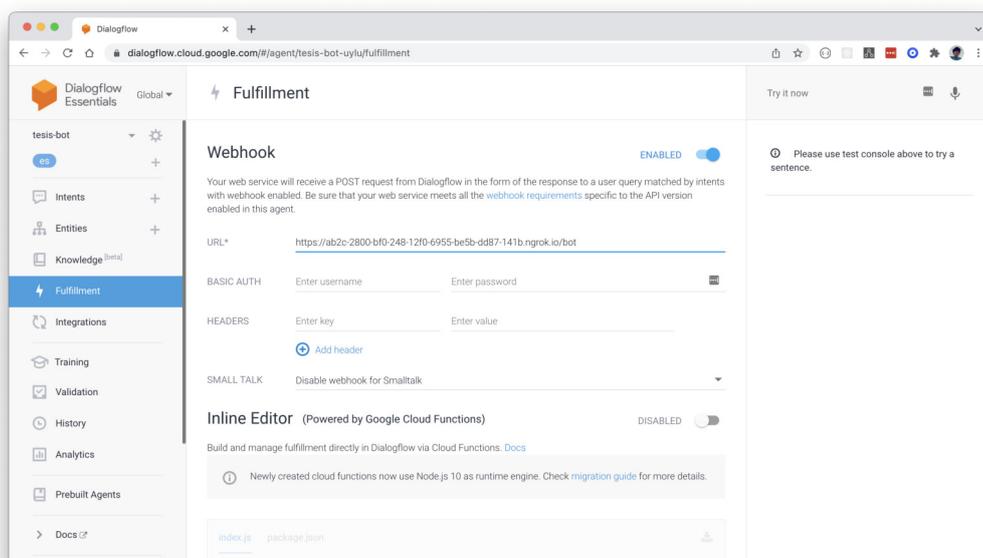
**Figura 2.7:** Frases de entrenamiento

Sin embargo, en este caso se activará la opción de activar llamada a *webhook* para este *intent*. En la Figura 2.8 se puede observar dicha opción en la interfaz de usuario.



**Figura 2.8:** Opción fulfillment

Por último, se debe configurar la url para la llamada del *webhook*, tal como se muestra en la Figura 2.9. Para esto de acuerdo a la documentación de Dialogflow se debe levantar un endpoint con el método POST el cual va a ser llamado por cada *intent* que tenga activada la opción de llamada de *webhook*. La url debe estar sobre el protocolo HTTPS.



**Figura 2.9:** Url en fulfillment

## 2.4 DEFINICIÓN DE HISTORIAS DE USUARIO

Con la información obtenida anteriormente, tanto el scrum master como el product owner expresaron las necesidades, las cuales han sido descritas en las siguientes historias de usuario. Debido a la complejidad del sistema se dividieron las historias de usuario en historias de usuario para la aplicación de gestión e historias de usuario para la aplicación de voz y apoyo visual.

## 2.5 PRODUCT BACKLOG

A continuación, en la Tabla 2.3 se muestra el product backlog generado en base a las historias de usuario obtenidas; las historias de usuario completas se encuentran en el anexo 6.3. Para la descripción de la misma, se ha hecho uso de Gherkin Lang debido a que maneja una sintaxis que puede ser entendida tanto por perfiles de negocio (usuarios) como por perfiles tecnológicos (desarrolladores). El peso y la prioridad de las historias de usuario han sido estimadas por el desarrollador, en conjunto con el scrum master y el product owner, utilizando la técnica de Planning Poker.

**Tabla 2.3:** Product backlog

Código	Descripción	Prioridad	Peso
HU01	Yo como usuario deseo crear, editar y deshabilitar un "menú".	Alta	13
HU02	Yo como usuario deseo crear, editar y deshabilitar "categorías"	Alta	8
HU03	Yo como usuario deseo crear, editar y borrar una "subcategoría" de una "categoría"	Alta	5
HU04	Yo como usuario deseo crear, editar y deshabilitar "tipos de opciones"	Alta	5
HU05	Yo como usuario deseo crear, editar y deshabilitar un "menú detalle" de un "menú"	Alta	13
HU06	Yo como usuario deseo crear, editar y deshabilitar una "opción" de un "tipo opción"	Alta	5
HU07	Yo como usuario deseo crear, editar y deshabilitar "opciones disponibles" de un "menú detalle"	Alta	5

HU08	Yo como usuario deseo añadir y quitar una "opción" de una "opción disponible"	Alta	5
HU09	Yo como usuario deseo buscar "menú detalle" del menú	Media	5
HU10	Yo como usuario deseo ver los "menú detalle" que busco en la pantalla de apoyo al buscar "menú detalle" del menú	Media	5
HU11	Yo como usuario deseo que si no encuentro nada en la búsqueda de categorías, subcategorías, promociones o por nombre del "menú detalle" se mencionen las promociones	Media	5
HU12	Yo como usuario deseo que si no encuentro nada en la búsqueda de "categorías", "subcategorías", "promociones" o por nombre del "menú detalle" muestre las promociones en la pantalla de apoyo	Media	5
HU13	Yo como usuario deseo obtener más detalles de un "menú detalle" que mencione	Alta	5
HU14	Yo como usuario deseo ver más detalles de un "menú detalle" que mencioné en la pantalla de apoyo	Alta	5
HU15	Yo como cliente deseo agregar un "menú detalle" al carrito	Alta	20
HU16	Yo como cliente deseo que al agregar un "menú detalle" al carrito se pueda observar el icono de carrito lleno en la pantalla de apoyo	Alta	3
HU17	Yo como usuario deseo listar mi carrito de compras	Alta	13
HU18	Yo como usuario deseo ver mi carrito de compras en la pantalla de apoyo	Alta	13
HU19	Yo como usuario deseo modificar un "menú detalle" de mi carrito de compras	Alta	13
HU20	Yo como usuario deseo que al modificar un "menú detalle" de mi carrito de compras se visualice la información del "menú detalle" y los cambios que se realizaron en la pantalla de apoyo	Alta	8
HU21	Yo como usuario deseo terminar el proceso de compra	Alta	3

HU22	Yo como usuario deseo que al terminar el proceso de compra se muestre un mensaje en la pantalla de apoyo	Alta	5
------	--	------	---

## 2.6 SPRINTS

Esta sección detalla el trabajo realizado en cada Sprint. En el anexo Anexo **6.7** se encuentran imágenes relacionadas con el desarrollo de los sprints.

### 2.6.1 Sprint 0

#### **Sprint planning**

Este es el sprint inicial antes de comenzar el desarrollo, tiene como objetivo levantar los ambientes para el desarrollo, organizar y priorizar las historias de usuario, realizar el modelo de base de datos y también crear el diagrama de la aplicación de voz para el asistente virtual.

#### **Ejecución del sprint**

Se instalaron las herramientas de desarrollo como lo son Webstorm, MySQL, Redis. Estas herramientas permitirán desarrollar una aplicación web las cuales están instaladas sobre Docker como se muestra en la Figura 2.10.

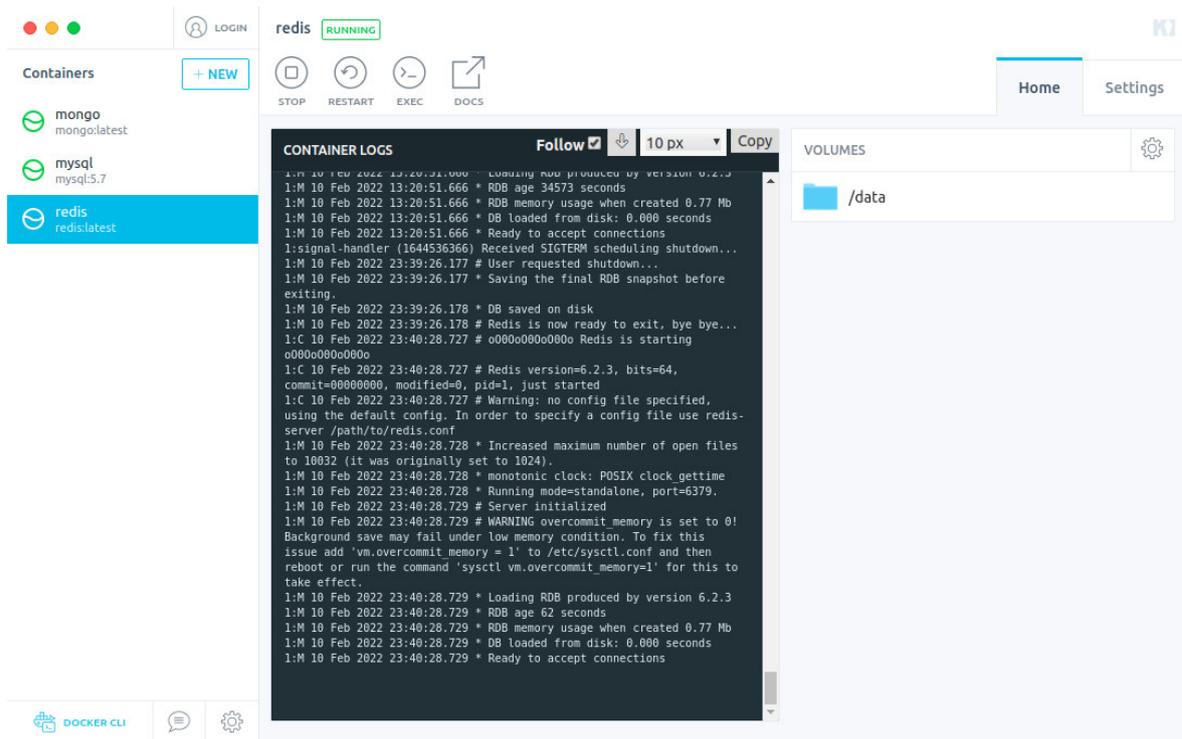


Figura 2.10: Herramientas instaladas

También se procedió a descargar el proyecto base para realizar una gestión en back y una gestión en front. En la Figura 2.11 se muestran los repositorios que fueron provistos por el product owner.

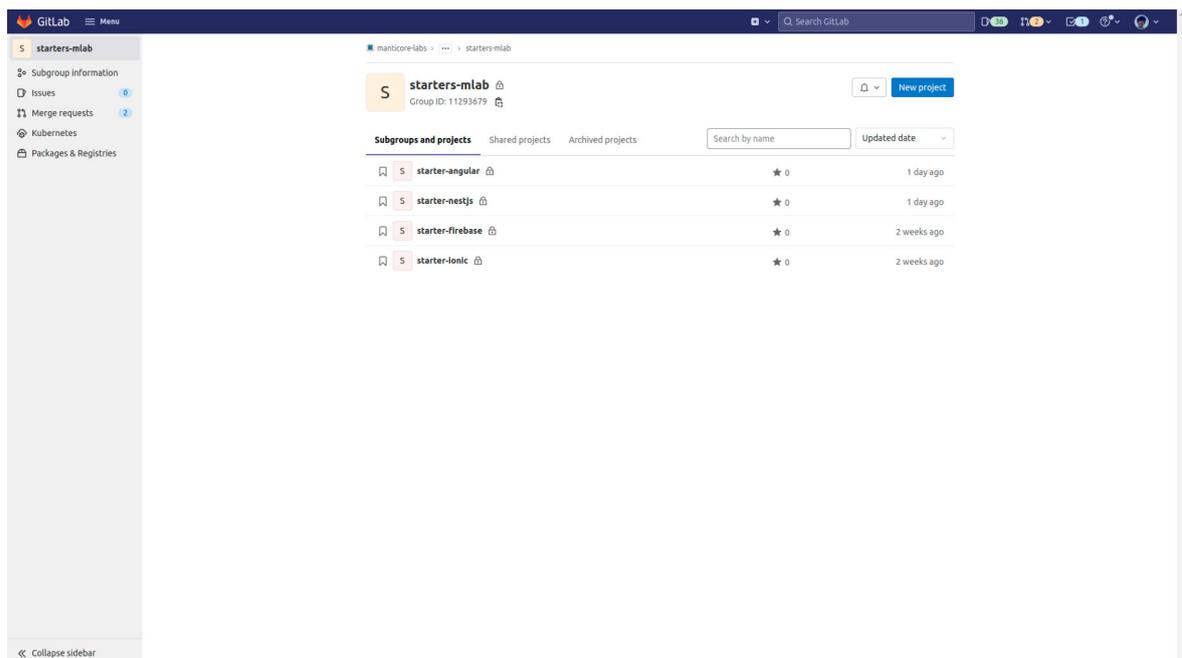
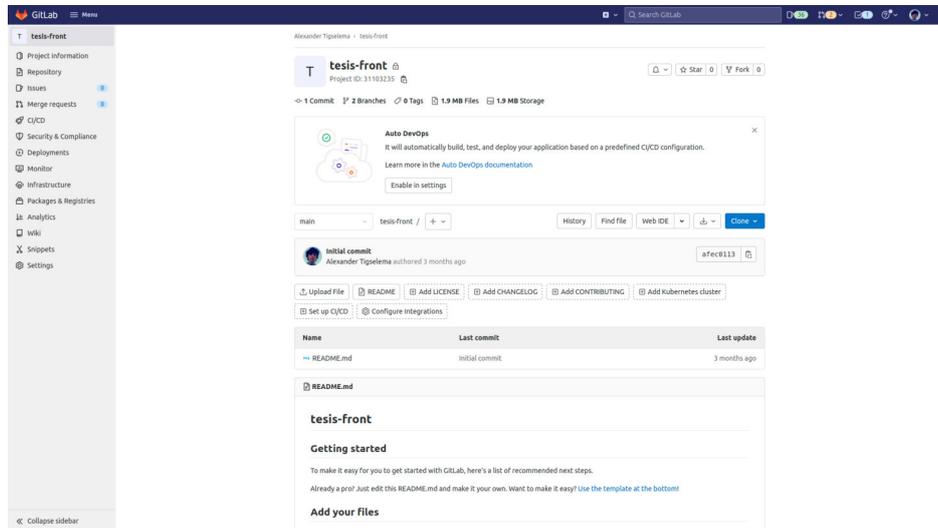


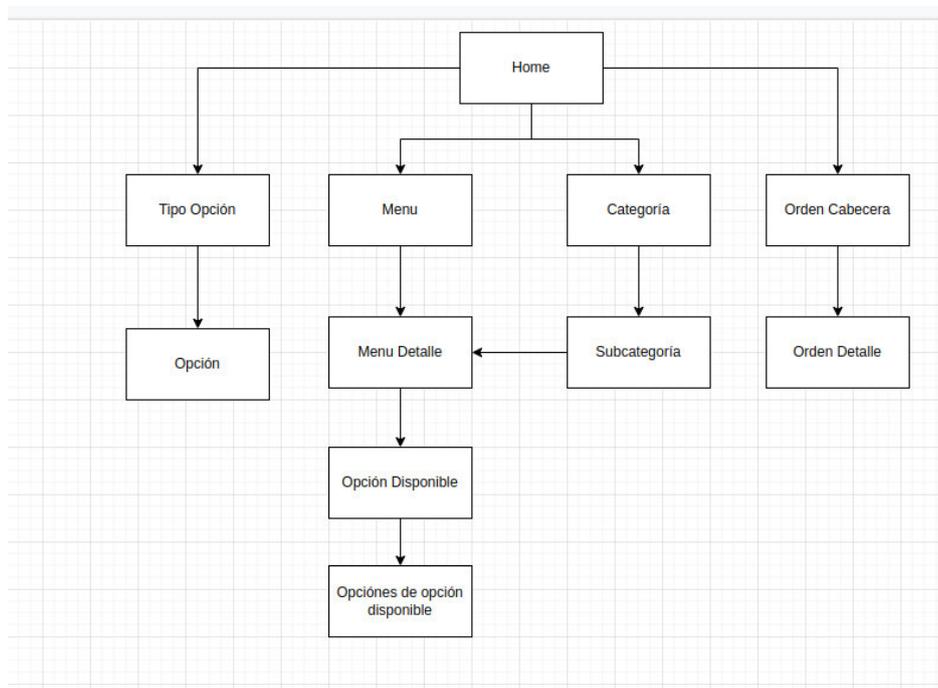
Figura 2.11: Repositorios provistos

En la Figura 2.12 se muestra otro repositorio que se creó, en el cual, se almacenará el código de la aplicación web que se mostrará al cliente final junto con la aplicación de voz.



**Figura 2.12:** Repositorio de la aplicación de voz frontend

La Figura 2.13 muestra un mapa del sitio que se creó para conocer los módulos que se implementarán en la aplicación de gestión desde el cual se generará la información para la aplicación de voz.



**Figura 2.13:** Mapa del sitio de la aplicación de gestión.

Se realizó un prototipo de bajo nivel, que se muestra en la Figura 2.14, de la aplicación web

que se mostrará al cliente final mientras realiza el pedido por la aplicación de voz.

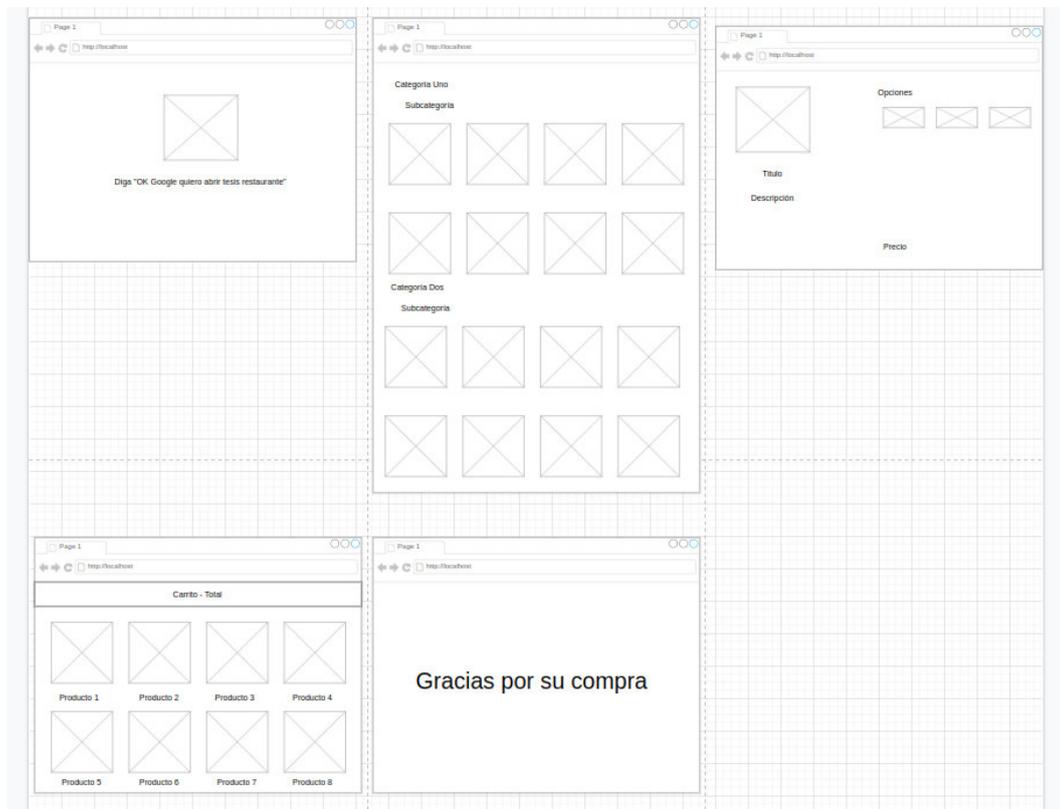


Figura 2.14: Prototipo de bajo nivel

## Sprint retrospective

### ¿Qué salió bien en la iteración?

- Se implementó el ambiente de desarrollo
- Se crearon los diagramas necesarios para la ejecución del proyecto.
- Se crearon los repositorios para almacenar el código del proyecto.
- Se realizó un prototipo para que sirva como guía de la parte visual de la aplicación del cliente.

### ¿Se alcanzaron los objetivos propuestos?

Si

### ¿Qué se puede mejorar?

N/A

## 2.6.2 Sprint 1

### Sprint planning

#### Objetivos del sprint

- Crear el módulo “menú” en el proyecto de gestión en backend y frontend
- Crear el módulo “categoría” en el proyecto de gestión en backend y frontend
- Crear el módulo “subcategoría” en el proyecto de gestión en backend y frontend

### Sprint backlog

Para el backlog del sprint 1 se piensa trabajar las historias de usuario 01, 02 y 03 cuyas tareas se encuentran listadas en la Tabla 2.4.

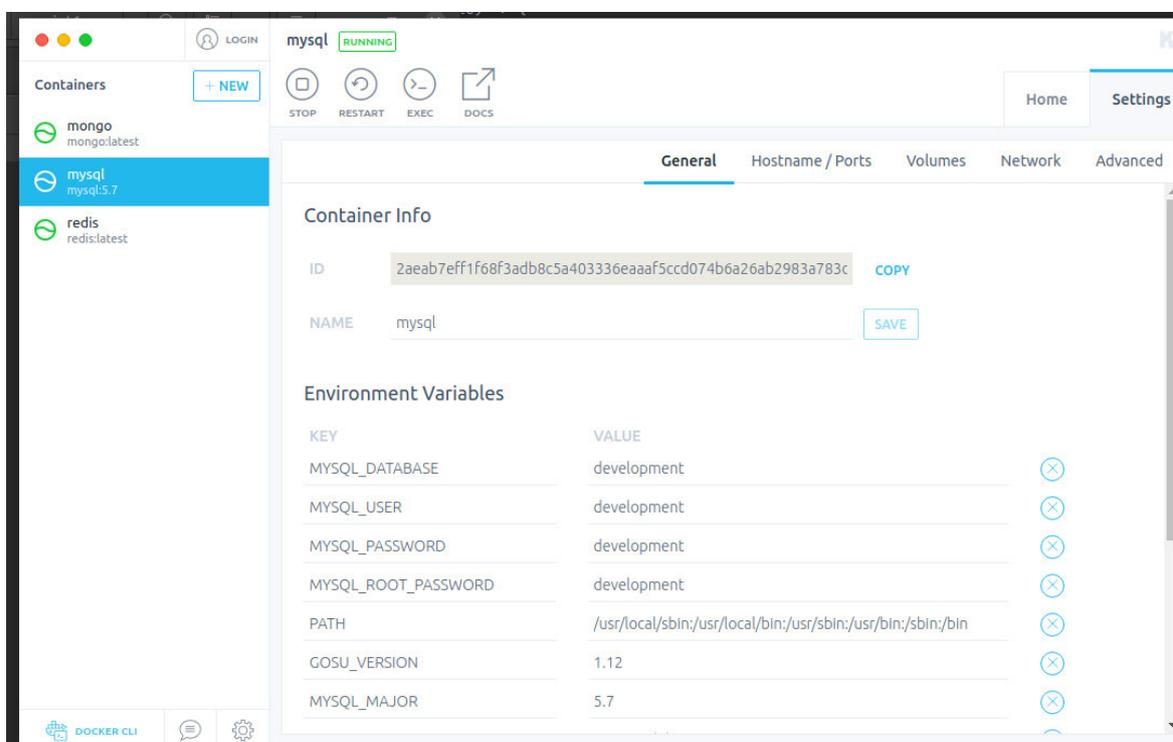
**Tabla 2.4:** Sprint 1 backlog

Código	Historia de usuario	Tareas del sprint
HU01	Yo como usuario deseo crear, editar y deshabilitar un “menú”.	<ul style="list-style-type: none"><li><input type="checkbox"/> Crear servicio REST de “menú” en backend.</li><li><input type="checkbox"/> Crear tabla de “menú” en la base de datos</li><li><input type="checkbox"/> Crear ruta “menú” en el frontend.</li><li><input type="checkbox"/> Cargar el “menú” en las entidades reconocidas por el bot para poderlas llamar por su nombre.</li></ul>
HU02	Yo como usuario deseo crear, editar y deshabilitar “categorías”.	<ul style="list-style-type: none"><li><input type="checkbox"/> Crear servicio REST de “categoría” en backend.</li><li><input type="checkbox"/> Crear tabla de “categoría” en la base de datos.</li><li><input type="checkbox"/> Crear ruta “categoría” en el frontend.</li><li><input type="checkbox"/> Crear formulario de creación y edición “categoría” en el frontend.</li><li><input type="checkbox"/> Implementar campos de búsqueda.</li></ul>

HU03	Yo como usuario deseo crear, editar y borrar una "subcategoría" de una "categoría".	<ul style="list-style-type: none"> <li><input type="checkbox"/> Crear servicio REST de "subcategoría" en backend.</li> <li><input type="checkbox"/> Crear tabla de "subcategoría" en la base de datos.</li> <li><input type="checkbox"/> Crear ruta "subcategoría" en el frontend.</li> <li><input type="checkbox"/> Crear formulario de creación y edición "subcategoría" en el frontend.</li> <li><input type="checkbox"/> Implementar campos de búsqueda.</li> </ul>
------	---	---

## Ejecución del sprint

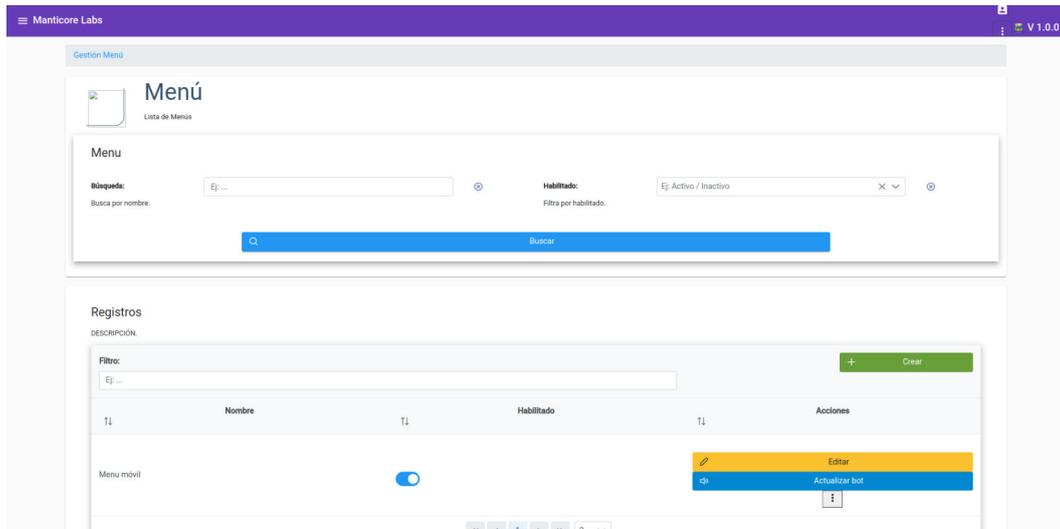
1. Se creó la base de datos utilizando la herramienta Docker, como se muestra en la Figura 2.15, junto con Docker Compose, esto permite tener todo el software de la base de datos en un ambiente virtualizado para poder ocuparlo como se requiera.



**Figura 2.15:** Herramientas virtualizadas

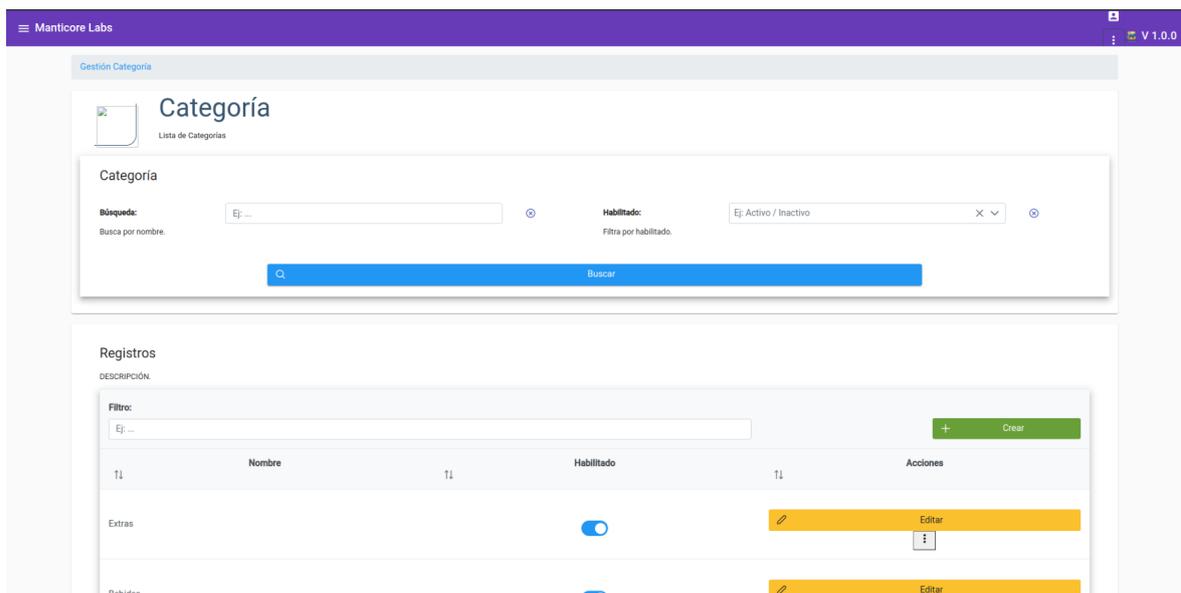
2. Se creó el módulo "menú" tanto en frontend como en backend. En la Figura 2.16 se

muestra el módulo “menú” en frontend.



**Figura 2.16:** Módulo “menú”

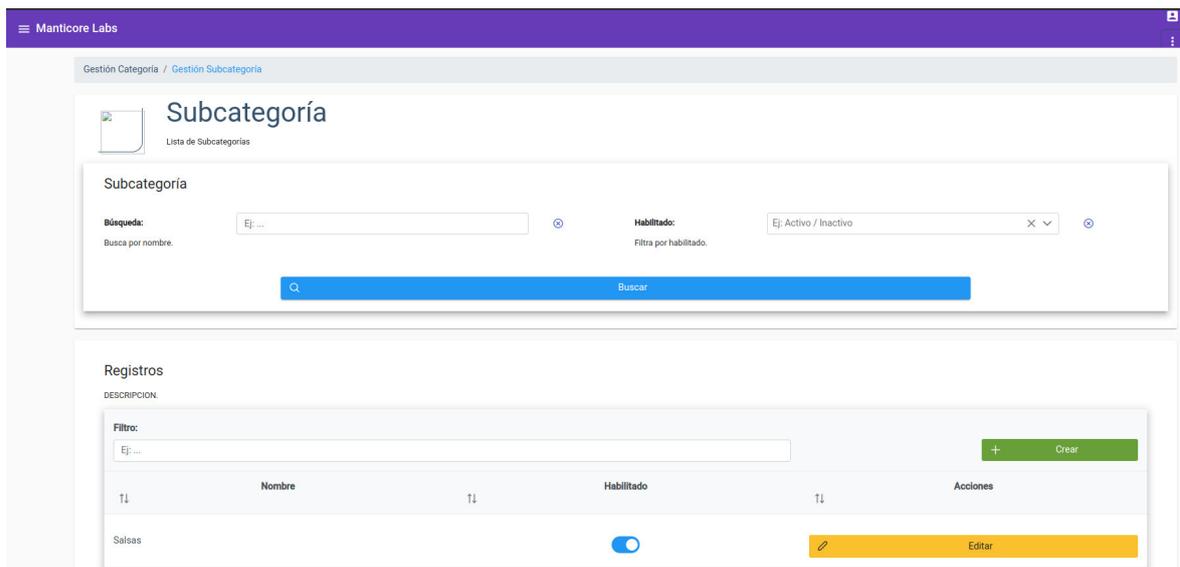
- a) Se creó el módulo “menú” en backend.
  - b) Se creó la base de datos del módulo “menú” en el motor de base de datos.
  - c) Se crearon los campos de búsqueda en el módulo “menú”.
  - d) Se creó el formulario del módulo “menú” utilizando la librería de generación de código.
3. Se creó el módulo “categoría” tanto en frontend como en backend. En la Figura 2.17 se muestra el módulo “categoría” en frontend.



**Figura 2.17:** Módulo “categoría”

- a) Se creó el módulo “categoría” en backend.
- b) Se creó la base de datos del módulo “categoría” en el motor de base de datos.
- c) Se crearon los campos de búsqueda en el módulo “categoría”.
- d) Se creó el formulario del módulo “categoría” utilizando la librería de generación de código.

4. Se creó el módulo “subcategoría” tanto en frontend como en backend. En la Figura 2.18 se muestra el módulo “subcategoría” en frontend.



**Figura 2.18:** Módulo “subcategoría”

- a) Se creó el módulo “subcategoría” en backend.
- b) Se creó la base de datos del módulo “subcategoría” en el motor de base de datos.
- c) Se crearon los campos de búsqueda en el módulo “subcategoría”.
- d) Se creó el formulario del módulo “subcategoría” utilizando la librería de generación de código.

## Sprint review

Una vez terminado el sprint 1, se procede a realizar el sprint review, el cual muestra el cumplimiento de las historias de usuario desarrolladas, este resultado se encuentra en la Tabla 2.5.

**Tabla 2.5:** Sprint 1 review

Historia	Criterio de aceptación	Cumplimiento
----------	------------------------	--------------

<p>HU01</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> El sistema permitirá crear, listar y editar un “menú”</li> <li><input type="checkbox"/> Solo existirá un “menú” en el sistema.</li> <li><input type="checkbox"/> Se podrá sincronizar el “menú” con la aplicación de voz desde esta pantalla con el fin de que la lista de productos se mantenga actualizada.</li> <li><input type="checkbox"/> Los campos que se mostrarán en el formulario de crear o editar son los siguientes: <ul style="list-style-type: none"> <li>✧ Nombre (Puede ingresar letras y números) (Editable) (Obligatorio).</li> <li>✧ mensajeFinal (Puede ingresar letras y números) (Editable) (Obligatorio).</li> </ul> </li> <li><input type="checkbox"/> No existirán campos de búsqueda.</li> </ul>	<p>Parcialmente</p>
<p>HU02</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> El sistema permitirá crear, listar, editar y deshabilitar una “categoría”.</li> <li><input type="checkbox"/> Solo se puede crear “categorías” para un “menú” ya existente.</li> <li><input type="checkbox"/> Las “categorías” se podrán buscar por su estado (habilitado o deshabilitado).</li> <li><input type="checkbox"/> Las “categorías” se podrán buscar por el campo nombre.</li> <li><input type="checkbox"/> Los campos que se mostrarán en el formulario de crear o editar son los siguientes: <ul style="list-style-type: none"> <li>✧ Nombre (Puede ingresar letras y números)(Editable)(Obligatorio).</li> <li>✧ Descripción (Puede ingresar letras y números)(Editable)(Obligatorio).</li> </ul> </li> </ul>	<p>Si</p>

HU03	<ul style="list-style-type: none"> <li><input type="checkbox"/> El sistema permitirá crear, listar, editar y deshabilitar una “subcategoría” de una “categoría”.</li> <li><input type="checkbox"/> Solo se puede crear una “subcategoría” para una “categoría” ya existente.</li> <li><input type="checkbox"/> Se navegará a esta pantalla mediante al hacer click en la opción ir a “subcategorías” de una “categoría” seleccionada.</li> <li><input type="checkbox"/> Las “subcategorías” se podrán buscar por su estado (habilitado o deshabilitado).</li> <li><input type="checkbox"/> Las “subcategorías” se podrán buscar nombre.</li> <li><input type="checkbox"/> Los campos que se mostrarán en el formulario de crear o editar son los siguientes: <ul style="list-style-type: none"> <li>✧ Nombre (Puede ingresar letras y números) (Editable) (Obligatorio).</li> <li>✧ Descripción (Puede ingresar letras y números) (Editable) (Obligatorio).</li> </ul> </li> </ul>	Si
------	--	----

En la Tabla 2.6 se puede ver el cumplimiento del sprint 1 por tareas.

**Tabla 2.6:** Sprint 1 cumplimiento

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo No o parcial
HU01	Crear servicio REST de “menú” en backend.	Si	
	Crear tabla de “menú” en la base de datos	Si	
	Crear ruta “menú” en el frontend	Si	

	Cargar el "menú" en las entidades reconocidas por el bot para poderlas llamar por su nombre	No	Las entidades del "menú" deben ser cargadas una vez se haya terminado de construir la base de datos con todas las entidades y el "menú" esté generado y guardado
HU02	Crear servicio REST de "categoría" en backend.	Si	
	Crear tabla de "categorías" en la base de datos	Si	
	Crear ruta "categoría" en el frontend.	Si	
	Crear formulario de creación y edición "categoría" en el frontend.	Si	
	Implementar campos de búsqueda.	Si	
HU03	Crear servicio REST de "subcategoría" en backend.	Si	
	Crear tabla de "subcategorías" en la base de datos.	Si	
	Crear ruta "subcategoría" en el frontend.	Si	
	Crear formulario de creación y edición "subcategoría" en el frontend.	Si	
	Implementar campos de búsqueda.	Si	

### **Sprint retrospective**

#### **¿Se alcanzaron los objetivos propuestos?**

Parcialmente

#### **¿Qué salió bien en la iteración?**

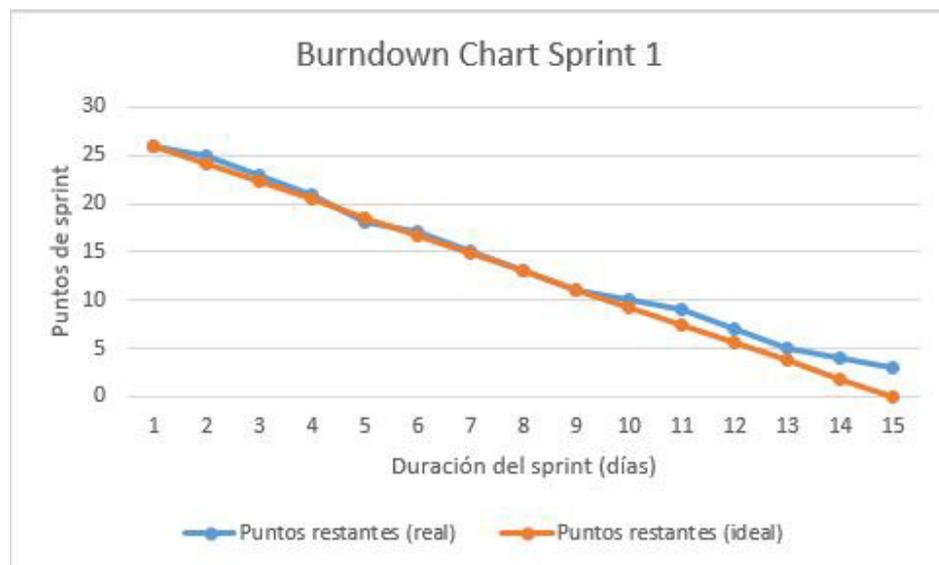
- Se cumplieron los objetivos principales del sprint.

- ❑ Se lograron crear tres módulos CRUD de gestión para el cliente interno de los establecimientos.
- ❑ Se configuró la aplicación backend con la base de datos MySQL.

### ¿Qué se puede mejorar?

- ❑ Se debió planear un poco mejor el sprint debido a que el bot debe tener toda la información del menú antes de crear las entidades del bot.
- ❑ No se conocía a profundidad la librería generadora de código por lo que hubo un poco de retraso en la creación de formularios.

Como se puede observar en la Figura 2.19, no se consiguió completar los puntos planeados para el sprint, pues, como se menciona anteriormente, para cargar el “menú” en las entidades reconocidas por el bot es necesario tener la base de datos completa.



**Figura 2.19:** Burndown Chart del Sprint 1

## 2.6.3 Sprint 2

### Sprint planning

#### Objetivos del sprint

- ❑ Crear el módulo “Tipos de opciones” en el proyecto de gestión en backend y frontend

- Crear el módulo “Opciones disponibles” de un “menú detalle” en el proyecto de gestión en backend y frontend.
- Crear el módulo “Opción” en el proyecto backend y frontend

## Sprint backlog

Para el backlog del sprint 2 se piensa trabajar las historias de usuario 04, 06 y 07 cuyas tareas se encuentran listadas en la Tabla 2.7.

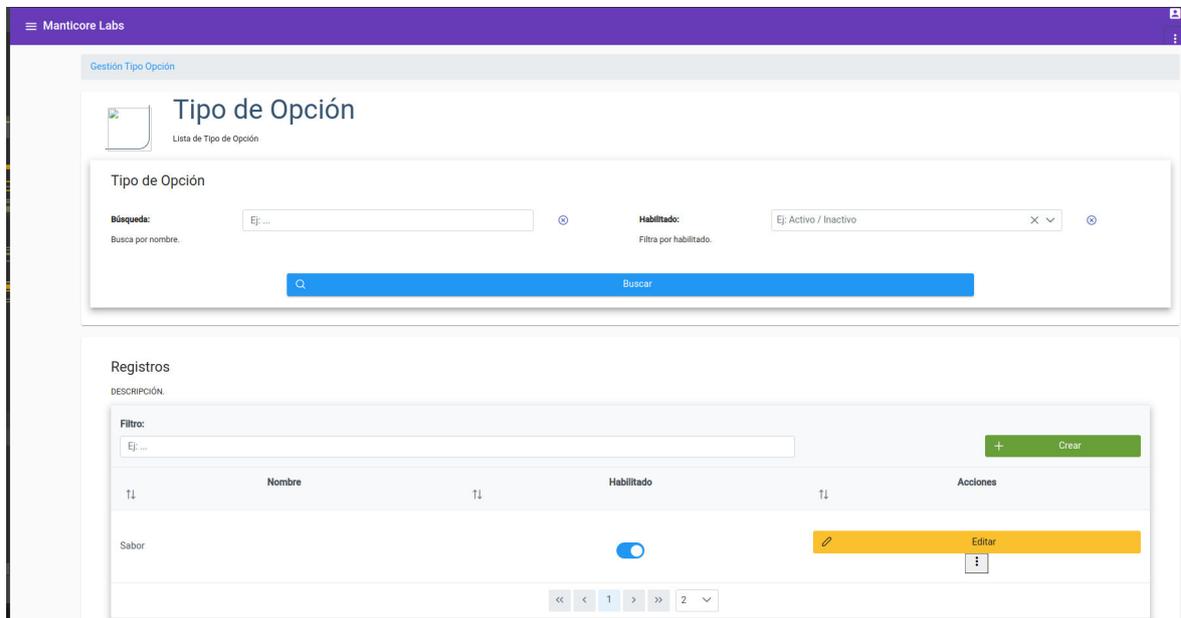
**Tabla 2.7:** Sprint 2 backlog

Código	Historia de usuario	Tareas del sprint
HU04	Yo como usuario deseo crear, editar y deshabilitar “tipos de opciones”.	<ul style="list-style-type: none"> <li><input type="checkbox"/> Crear servicio REST de “tipos de opciones” en backend.</li> <li><input type="checkbox"/> Crear tabla de “tipos de opciones” en la base de datos</li> <li><input type="checkbox"/> Crear ruta “tipos de opciones” en el frontend.</li> <li><input type="checkbox"/> Crear formulario de creación y edición de “tipos de opciones” en el frontend.</li> <li><input type="checkbox"/> Implementar campos de búsqueda.</li> </ul>
HU06	Yo como usuario deseo crear, editar y deshabilitar un “opción” de un “tipo opción”.	<ul style="list-style-type: none"> <li><input type="checkbox"/> Crear servicio REST de “opción” en backend.</li> <li><input type="checkbox"/> Crear tabla de “opción” en la base de datos.</li> <li><input type="checkbox"/> Crear ruta “opción” en el frontend.</li> <li><input type="checkbox"/> Crear formulario de creación y edición de “opción” en el frontend.</li> <li><input type="checkbox"/> Implementar campos de búsqueda.</li> </ul>

<p>HU07</p>	<p>Yo como usuario deseo crear, editar y deshabilitar “opciones disponibles” de un “menú detalle”.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Crear servicio REST de “opciones disponibles” en backend.</li> <li><input type="checkbox"/> Crear tabla de “opciones disponibles” en la base de datos.</li> <li><input type="checkbox"/> Crear ruta “opciones disponibles” en el frontend.</li> <li><input type="checkbox"/> Crear formulario de creación y edición de “opciones disponibles” en el frontend.</li> <li><input type="checkbox"/> Implementar campos de búsqueda.</li> </ul>
-------------	--	--

## Ejecución del sprint

1. Se creó el módulo “tipos de opciones” tanto en frontend como en backend. En la Figura 2.20 se muestra el módulo “menú” en frontend.

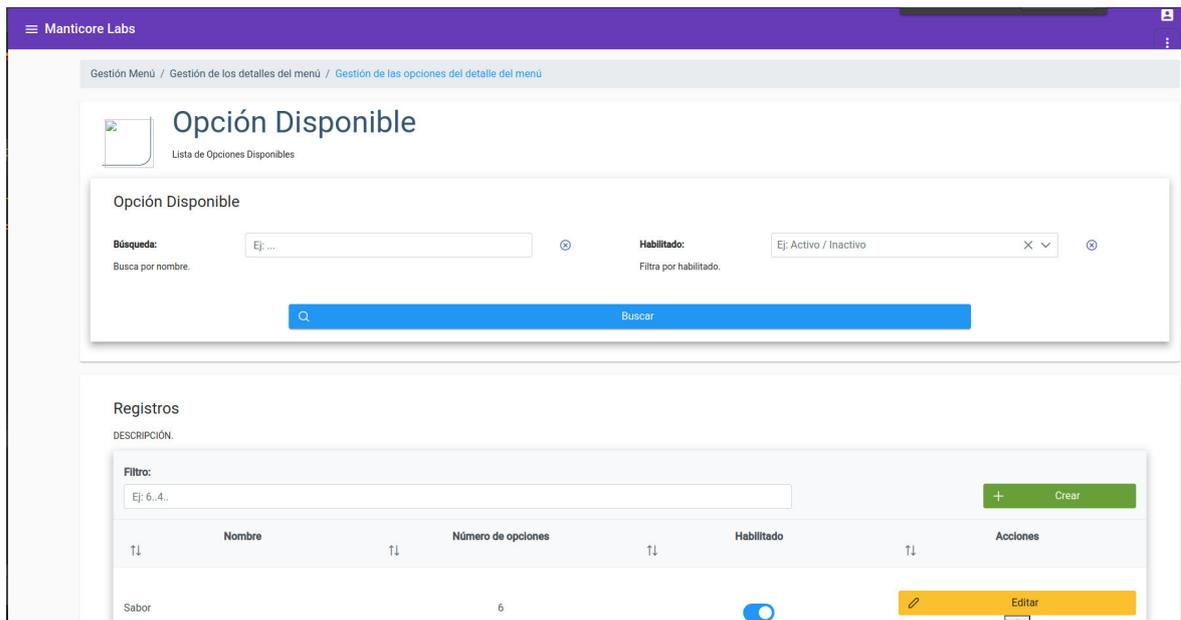


**Figura 2.20:** Modulo “tipos de opciones”

- a) Se creó el módulo “tipos de opciones” en backend.
- b) Se creó la base de datos del módulo “tipos de opciones” en el motor de base de datos.

- c) Se crearon los campos de búsqueda en el módulo “tipos de opciones”.
- d) Se creó el formulario del módulo “tipos de opciones” utilizando la librería de generación de código

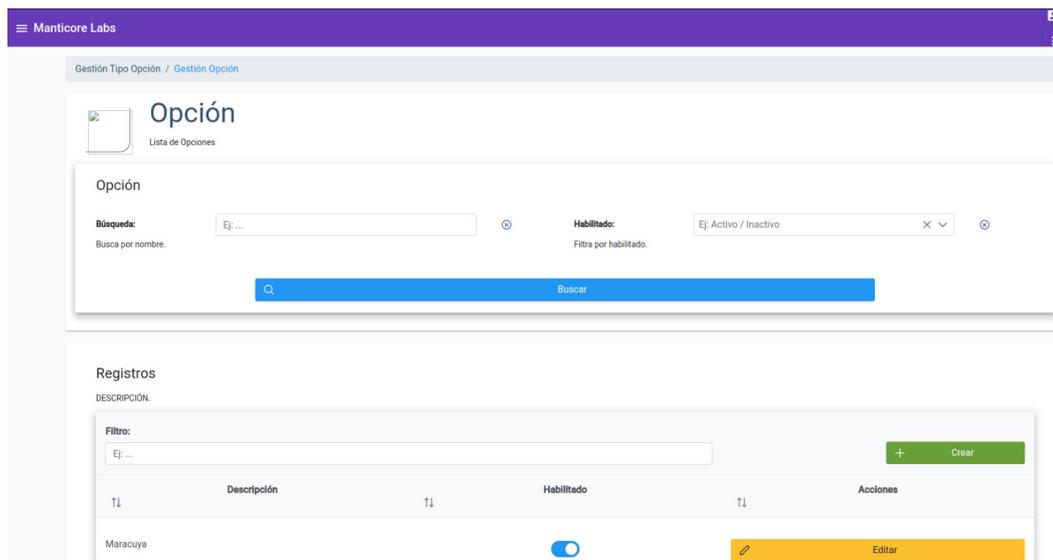
2. Se creó el módulo “opciones disponibles” tanto en frontend como en backend. En la Figura 2.21 se muestra el módulo “menú” en frontend.



**Figura 2.21:** Módulo “opciones disponibles”

- a) Se creó el módulo “opciones disponibles” de un “menú detalle” en backend.
- b) Se creó la base de datos del módulo “opciones disponibles” de un “menú detalle” en el motor de base de datos
- c) Se crearon los campos de búsqueda en el módulo “opciones disponibles” de un “menú detalle”.
- d) Se creó el formulario del módulo “opciones disponibles” de un “menú detalle” utilizando la librería de generación de código.

3. Se creó el módulo “opción” tanto en frontend como en backend. En la Figura 2.22 se muestra el módulo “opción” en frontend.



**Figura 2.22:** Módulo “opción”

- Se creó el módulo “opción” en backend.
- Se creó la base de datos del módulo “opción” en el motor de base de datos.
- Se crearon los campos de búsqueda en el módulo “opción”.
- Se creó el formulario del módulo “opción” utilizando la librería de generación de código.

## Sprint review

Una vez terminado el sprint 2, se procede a realizar el sprint review, el cuál que muestra el cumplimiento de las historias de usuario desarrolladas, este resultado se encuentra en la Tabla 2.8.

**Tabla 2.8:** Sprint 2 review

Historia	Criterio de aceptación	Cumplimiento
----------	------------------------	--------------

<p>HU04</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> El sistema permitirá crear, listar, editar y deshabilitar “tipos de opciones”.</li> <li><input type="checkbox"/> Los “tipos de opciones” se podrán buscar por su estado (habilitado o deshabilitado).</li> <li><input type="checkbox"/> Los “tipos de opciones” se podrán buscar nombre.</li> <li><input type="checkbox"/> Esta pantalla permitirá navegar hacia la pantalla de “opciones”.</li> <li><input type="checkbox"/> Los campos que se mostrarán en el formulario de crear o editar son los siguientes: <ul style="list-style-type: none"> <li>✧ Nombre (Puede ingresar letras y números) (Editable) (Obligatorio).</li> </ul> </li> </ul>	<p>Si</p>
<p>HU06</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Solo se puede crear “opciones” para un “menú detalle” ya existente.</li> <li><input type="checkbox"/> El sistema permitirá crear, listar, editar y deshabilitar una “opción” de un “tipo opción”.</li> <li><input type="checkbox"/> La “opción” se podrá buscar por su estado (habilitado o deshabilitado).</li> <li><input type="checkbox"/> La “opción” detalle se podrá buscar nombre.</li> <li><input type="checkbox"/> Los campos que se mostrarán en el formulario de crear o editar son los siguientes: <ul style="list-style-type: none"> <li>✧ Descripción (Puede ingresar letras y números)(Editable) (Obligatorio).</li> <li>✧ Precio (Puede ingresar números)(Editable)(Obligatorio).</li> </ul> </li> </ul>	<p>Si</p>

HU07	<ul style="list-style-type: none"> <li><input type="checkbox"/> El sistema permitirá crear, listar, editar y deshabilitar una “opción disponible” de un “menú detalle”.</li> <li><input type="checkbox"/> Solo se puede crear “opciones disponibles” para un “menú detalle” ya existente.</li> <li><input type="checkbox"/> Las “opciones disponibles” se podrán buscar por su estado (habilitado o deshabilitado).</li> <li><input type="checkbox"/> Las “opciones disponibles” se podrán buscar por el tipo de opción.</li> <li><input type="checkbox"/> Los campos que se mostrarán en el formulario de crear o editar son los siguientes: <ul style="list-style-type: none"> <li>✧ Número de opciones (Puede ingresar números)(Editable) (Obligatorio).</li> <li>✧ Tipo de opción (Puede ingresar números)(No editable) (Obligatorio).</li> </ul> </li> </ul>	Si
------	---	----

En la Tabla 2.9 se puede ver el cumplimiento del sprint 2 por tareas.

**Tabla 2.9:** Sprint 2 cumplimiento

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo No o parcial
HU04	Crear servicio REST de “tipos de opciones” en backend.	Si	
	Crear tabla de “tipos de opciones” en la base de datos.	Si	
	Crear ruta “tipos de opciones” en el frontend.	Si	
	Crear formulario de creación y edición de “tipos de opciones” en el frontend.	Si	
	Implementar campos de búsqueda.	Si	
HU06	Crear servicio REST de “opción” en backend.	Si	

	Crear tabla de "opción" en la base de datos.	Si	
	Crear ruta "opción" en el frontend.	Si	
	Crear formulario de creación y edición de "opción" en el frontend.	Si	
	Implementar campos de búsqueda.	Si	
HU07	Crear servicio REST de "opciones disponibles" en backend.	Si	
	Crear tabla de "opciones disponibles" en la base de datos.	Si	
	Crear ruta "opciones disponibles" en el frontend.	Si	
	Crear formulario de creación y edición de "opciones disponibles" en el frontend.	Si	
	Implementar campos de búsqueda.	Si	

## Sprint retrospective

### ¿Se alcanzaron los objetivos propuestos?

Si

### ¿Qué salió bien en la iteración?

- Se logró crear el CRUD del módulo de "tipos de opción".
- Se logró crear el CRUD del módulo de "opciones" de un "tipo de opción".
- Se logró crear el CRUD del módulo de "opciones disponibles".

### ¿Qué se puede mejorar?

- La solución del backend para las "opciones disponibles" de un menú detalle puede no ser la más adecuada.

En la Figura 2.23 se puede apreciar el esfuerzo realizado durante los días del sprint, como se puede observar, se consiguió completar los puntos planeados para el presente sprint

sin mayores inconvenientes. Como se observa en la gráfica, los puntos restantes reales del sprint tienen una tendencia distinta a la proyectada, esto se debe a que para cumplir con los objetivos del sprint, se completaron en promedio 2 o 3 puntos diariamente. Al acercarse al final de sprint, se completó tan solo un punto diariamente.

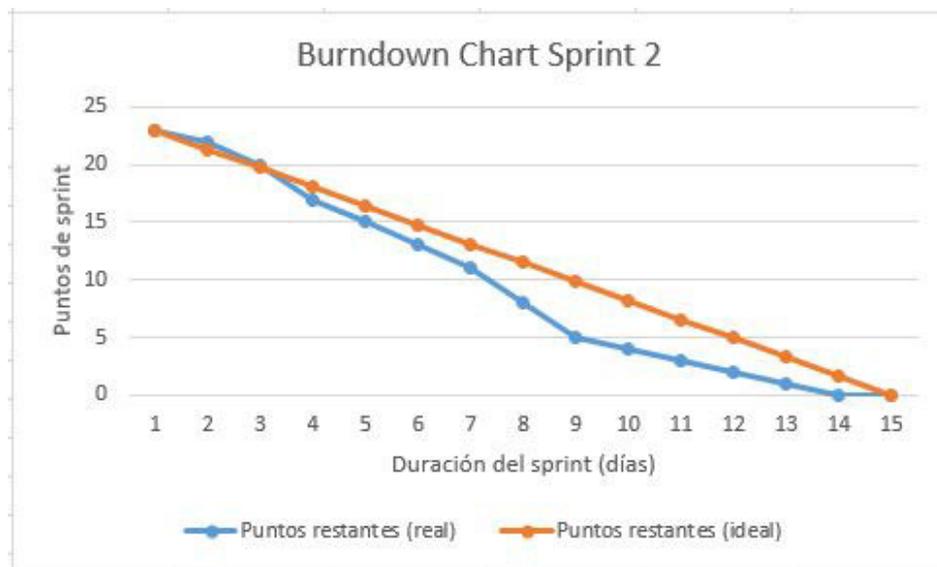


Figura 2.23: Burndown Chart del Sprint 2

## 2.6.4 Sprint 3

### Sprint planning

#### Objetivos del sprint

- Crear el módulo "menú detalle" en el proyecto de gestión en backend y frontend
- Agregar funcionalidad de carga de imágenes para el "menú detalle".
- Crear el módulo "opción" de "opción disponible" en el proyecto de gestión en backend y frontend.
- Crear un bucket en Google Cloud para almacenamiento de "imágenes"
- Conectar el servicio de almacenamiento de "imágenes" con la aplicación.

#### Sprint backlog

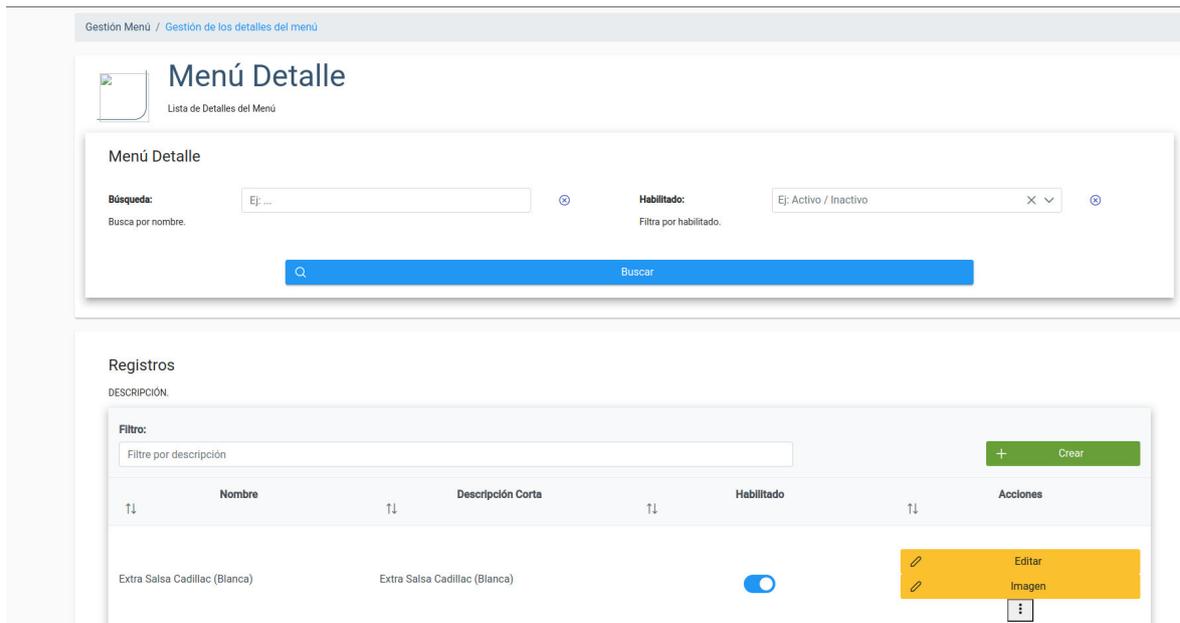
Para el backlog del sprint 1 se piensa trabajar las historias de usuario 05 y 08 cuyas tareas se encuentran listadas en la Tabla 2.10

**Tabla 2.10:** Sprint 3 backlog

Código	Historia de usuario	Tareas del sprint
HU05	Yo como usuario deseo crear, editar y deshabilitar un “menú detalle” de un “menú”.	<ul style="list-style-type: none"> <li><input type="checkbox"/> Crear servicio REST de “menú detalle” en backend.</li> <li><input type="checkbox"/> Crear servicio REST para subir una “imagen” para el “menú detalle” en backend.</li> <li><input type="checkbox"/> Crear tabla de “menú detalle” en la base de datos.</li> <li><input type="checkbox"/> Crear ruta “menú detalle” en el frontend.</li> <li><input type="checkbox"/> Crear formulario de creación y edición de “menú detalle” en el frontend.</li> <li><input type="checkbox"/> Implementar campos de búsqueda.</li> <li><input type="checkbox"/> Implementar funcionalidad para subir “imagenes”.</li> </ul>
HU08	Yo como usuario deseo añadir y quitar una “opción” de una “opción disponible”.	<ul style="list-style-type: none"> <li><input type="checkbox"/> Crear servicio REST de una “opción” para una “opción disponible” en backend.</li> <li><input type="checkbox"/> Crear tabla de “opción” de una “opción disponible” en la base de datos.</li> <li><input type="checkbox"/> Crear ruta “opciones” de una “opción disponible” en el frontend.</li> <li><input type="checkbox"/> Crear tabla de selección de “opciones” para una “opción disponible”.</li> </ul>

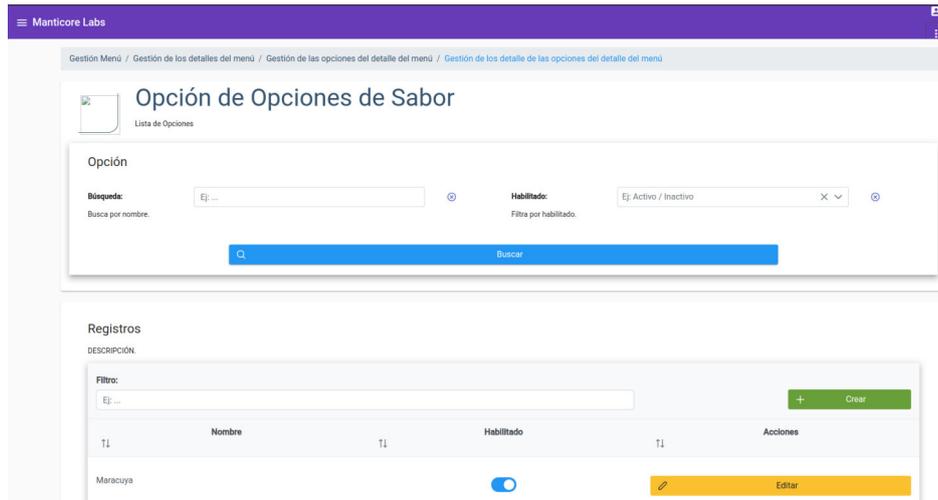
### Ejecución del sprint

1. Se creó el módulo “menú detalle” de un “menú” tanto en frontend como en backend. En la Figura 2.24 se muestra el módulo “menú detalle” en frontend.



**Figura 2.24:** Módulo “menú detalle”

- a) Se creó el módulo “menú detalle” de un “menú” en backend
  - b) Se creó la base de datos del módulo “menú detalle” de un “menú” en el motor de base de datos.
  - c) Se crearon los campos de búsqueda en el módulo “menú detalle” de un “menú”.
  - d) Se creó el formulario del módulo “menú detalle” de un “menú” utilizando la librería de generación de código.
  - e) Se agregó la funcionalidad para subir “imágenes” del “menú detalle”.
2. Se creó el módulo “opción” de una “opción disponible” tanto en frontend como en backend. En la Figura 2.25 se muestra el módulo “opción” en frontend.



**Figura 2.25:** Módulo “opción” de una “opción disponible”

- Se creó el módulo “opción” de una “opción disponible” en backend.
- Se creó la base de datos del módulo “opción” de una “opción disponible” en el motor de base de datos.
- Se creó el formulario del módulo “opción” de una “opción disponible”.
- Se creó el formulario del módulo “opción” de una “opción disponible” utilizando la librería de generación de código.

## Sprint review

Una vez terminado el sprint 3, se procede a realizar el sprint review, el cuál que muestra el cumplimiento de las historias de usuario desarrolladas, este resultado se encuentra en la Tabla 2.11.

**Tabla 2.11:** Sprint 3 review

Historia	Criterio de aceptación	Cumplimiento
----------	------------------------	--------------

<p>HU05</p>	<ul style="list-style-type: none"> <li>❑ El sistema permitirá crear, listar, editar y deshabilitar un “menú detalle” de un “menú”.</li> <li>❑ Solo se puede crear “menú detalles” para un “menú” ya existente.</li> <li>❑ Los “menú detalle” se podrán buscar por su estado (habilitado o deshabilitado).</li> <li>❑ Los “menú detalle” se podrán buscar nombre.</li> <li>❑ Esta pantalla permitirá navegar hacia la pantalla de “opciones disponibles” del “menú”.</li> <li>❑ El sistema permitirá subir una “imagen” para mostrarse en el “menú detalle”.</li> <li>❑ Los campos que se mostrarán en el formulario de crear o editar son los siguientes: <ul style="list-style-type: none"> <li>✧ Nombre (Puede ingresar letras y números) (Editable) (Obligatorio).</li> <li>✧ DescripcionCorta (Puede ingresar letras y números) (Editable) (Obligatorio).</li> <li>✧ DescripcionLarga (Puede ingresar letras y números) (Editable) (Obligatorio).</li> <li>✧ Precio (Puede ingresar números) (Editable) (Obligatorio).</li> </ul> </li> </ul>	<p>Si</p>
-------------	---	-----------

HU08	<ul style="list-style-type: none"> <li><input type="checkbox"/> El sistema permitirá añadir y quitar una “opción” de una “opción disponible”.</li> <li><input type="checkbox"/> Solo se puede añadir o quitar “opciones” de una “opción disponible”.</li> <li><input type="checkbox"/> Las “opciones” de una “opción disponible” se podrán buscar por su estado (habilitado o deshabilitado).</li> <li><input type="checkbox"/> Las “opciones” de una “opción disponible” se podrán buscar por el nombre de la opción.</li> <li><input type="checkbox"/> En la subventana para añadir o quitar “opciones” de las “opciones disponibles” se visualizará una tabla en la que se podrá marcar las “opciones” que desea añadir o se desmarcarán las “opciones” que se deseen quitar.</li> </ul>	Si
------	---	----

En la Tabla 2.12 se puede ver el cumplimiento del sprint 3 por tareas.

**Tabla 2.12:** Sprint 3 cumplimiento

<b>Código</b>	<b>Tarea</b>	<b>Cumplimiento (Si/No/Parcial)</b>	<b>Motivo No o parcial</b>
HU05	Crear servicio REST de “menú detalle” en backend.	Si	
	Crear servicio REST para subir una “imagen” para el “menú detalle” en backend.	Parcialmente	Se debe configurar este servicio en google console.
	Crear tabla de “menú detalle” en la base de datos.	Si	
	Crear ruta “menú detalle” en el frontend.	Si	
	Crear formulario de creación y edición de “menú detalle” en el frontend.	Si	
	Implementar campos de búsqueda.	Si	

	Implementar funcionalidad para subir "imagenes".	Parcialmente	Se encuentra configurado el módulo para subir "imagenes" en frontend pero aún el enlace a google cloud no está listo.
HU08	Crear servicio REST de una "opción" para una "opción disponible" en backend.	Si	
	Crear tabla de "opción" de una "opción disponible" en la base de datos.	Si	
	Crear ruta "opciones" de una "opción disponible" en el frontend.	Si	
	Crear tabla de selección de "opciones" para una "opción disponible".	Si	

## Sprint retrospective

### ¿Se alcanzaron los objetivos propuestos?

Parcialmente

### ¿Qué salió bien en la iteración?

- Se logró crear el CRUD de "menú detalle" con sus filtros.
- Se logró maquetar la pantalla para subir "imagenes" para el "menú detalle"
- Se logró crear el CRUD para enlazar las "opciones" con el conjunto de "opciones disponibles" para utilizar en un "menú detalle".

### ¿Qué se puede mejorar?

- La tabla de "opciones" de una "opción disponible" debería mostrar solo las "opciones disponibles" faltantes y no las que ya fueron elegidas.

En la Figura 2.26 se puede apreciar el esfuerzo realizado durante los días del sprint. Similar al sprint anterior, en promedio se completaron 2 puntos diariamente, sin embargo, se sobre

dimensionó un poco el esfuerzo del sprint y es por eso que la línea real de puntos restantes muestra una tendencia similar al sprint anterior.

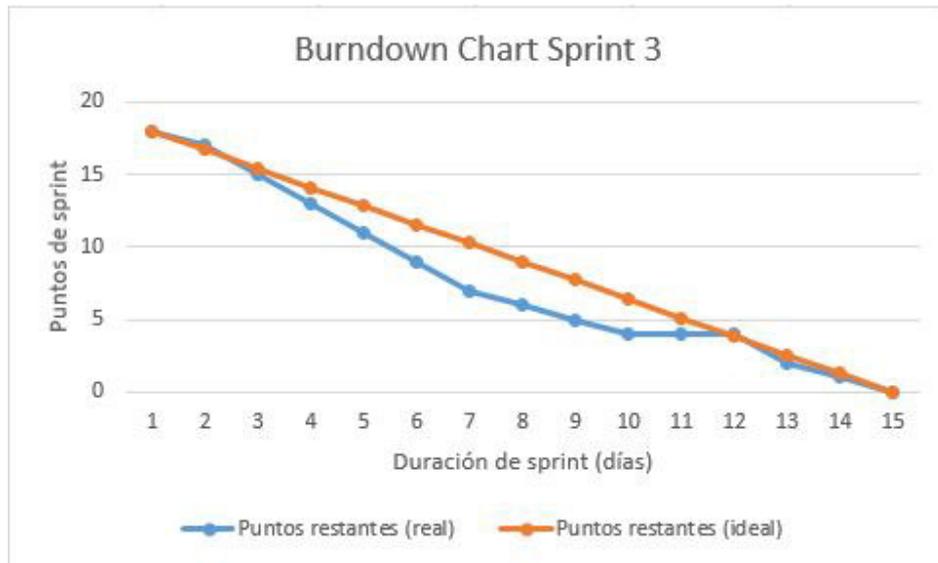


Figura 2.26: Burndown Chart del Sprint 3

## 2.6.5 Sprint 4

### Sprint planning

#### Objetivos del sprint

- Conectar la aplicación frontend en tiempo real con backend para que responda a las interacciones por voz y eventos disparados desde el backend.
- Crear los intents de navegación y búsqueda de “menú detalle” en la aplicación por voz.
- Obtener la información para mostrar el “menú” y hablar con el cliente final.
- Crear ruta de “menú” para mostrar el “menú” al cliente final.
- Realizar las configuraciones finales a la aplicación de voz en dialogflow.

### Sprint backlog

Para el backlog del sprint 4 se piensa trabajar las historias de usuario 09, 10, 11, 12 y 01 cuyas tareas se encuentran listadas en la Tabla 2.13

**Tabla 2.13:** Sprint 4 backlog

Código	Historia de usuario	Tareas del sprint
HU09	Yo como usuario deseo buscar "menú detalle" del "menú".	<ul style="list-style-type: none"> <li><input type="checkbox"/> Crear intent de búsqueda para que el bot conteste de forma adecuada.</li> <li><input type="checkbox"/> Conectar el intent con el backend para que pueda realizar las búsquedas de productos.</li> <li><input type="checkbox"/> Conectar el bot a una pantalla.</li> <li><input type="checkbox"/> Configurar el comando para llamar al bot mediante las interfaces de google.</li> </ul>
HU10	Yo como usuario deseo ver los "menú detalles" que busco en la pantalla de apoyo al buscar "menú detalles" del "menú".	<ul style="list-style-type: none"> <li><input type="checkbox"/> Conectar frontend al servicio REST para obtener el "menú".</li> <li><input type="checkbox"/> Conectar el websocket para que escuche la navegación y búsqueda de productos.</li> <li><input type="checkbox"/> Crear ruta "opciones" de una "opción disponible" en el frontend.</li> <li><input type="checkbox"/> Crear lógica de navegación en front al escuchar un evento.</li> </ul>
HU11	Yo como usuario deseo que si no encuentro nada en la búsqueda de "categorías", "subcategorías", promociones o por nombre del "menú detalle" se mencionen las promociones.	<ul style="list-style-type: none"> <li><input type="checkbox"/> Modificar el intent de búsqueda para que si no encuentra un producto, dispare un evento al frontend para dirigir a las promociones.</li> <li><input type="checkbox"/> Modificar el intent para que el dispositivo lea los productos de la "categoría" promociones.</li> </ul>

HU12	Yo como usuario deseo que si no encuentro nada en la búsqueda de “categorías”, “subcategorías”, promociones o por nombre del “menú detalle” muestre las promociones en la pantalla de apoyo.	<input type="checkbox"/> Al recibir un evento de producto “categoría” o “subcategoría” no encontrado la aplicación frontend deberá desplazarse hacia las promociones.
HU01	Yo como usuario deseo crear, editar y deshabilitar un “menú”.	<input type="checkbox"/> Cargar el “menú” en las entidades reconocidas por el bot para poderlas llamar por su nombre.

## Ejecución del sprint

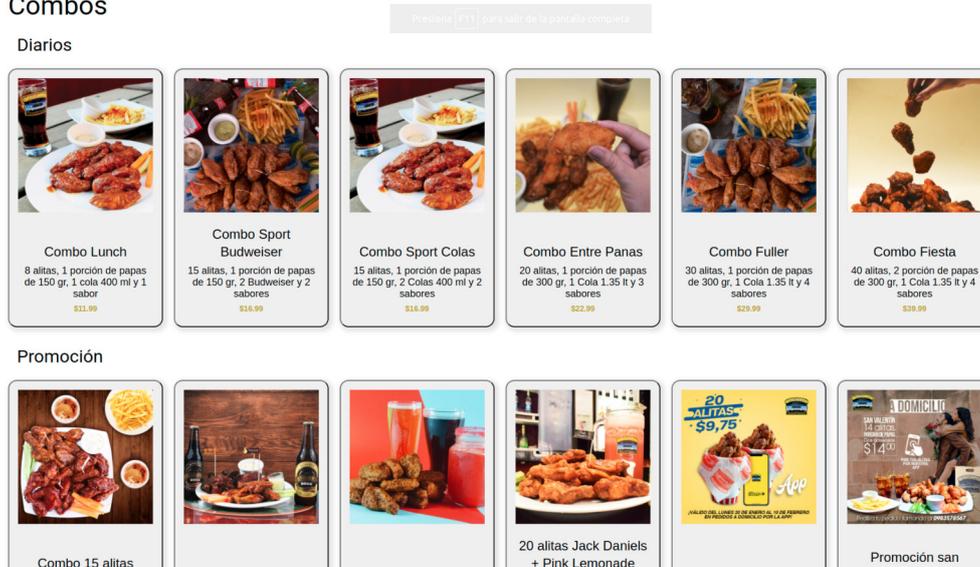
1. Crear el webhook para conectarlo con el bot mediante REST.

a) Se creó el módulo “bot” en el backend.

b) Se cargaron los “menú detalles” y “opciones” como las entidades reconocidas por el bot.

2. Crear intent de búsqueda en Dialogflow. En la Figura 2.27 se puede observar la pantalla que reacciona a la interacción con este intent.

## Combos



**Figura 2.27:** Intent de búsqueda

- Crear intent de búsqueda y navegación en la pantalla de menú.
- Configurar el comportamiento del webhook en back cuando se invoca el intent y encuentra "menu detalles" en búsqueda.
- Configurar el comportamiento del webhook en back y no se encuentran "menú detalles" en la búsqueda.
- Maquetar el menú en la aplicación frontend cliente, se deben mostrar los productos del "menú" obtenidos desde el backend.
- Conectar la aplicación frontend cliente con los eventos del intent, mediante el websocket.

## Sprint review

Una vez terminado el sprint 4, se procede a realizar el sprint review, el cuál que muestra el cumplimiento de las historias de usuario desarrolladas, este resultado se encuentra en la Tabla 2.14.

**Tabla 2.14:** Sprint 4 review

Historia	Criterio de aceptación	Cumplimiento
----------	------------------------	--------------

HU09	<input type="checkbox"/> Puedo buscar “categorías”. <input type="checkbox"/> Puedo buscar “subcategorías”. <input type="checkbox"/> Puedo buscar por nombre del “menú detalle”.	Si
HU10	<input type="checkbox"/> Puedo ver la lista de “menú detalles” de la “categoría”, “subcategoría” o promoción buscada. <input type="checkbox"/> Puedo ver el detalle del “menú detalle” buscado.	Si
HU11	<input type="checkbox"/> Se leerán la descripción corta de los productos de la “categoría” de promociones.	Si
HU12	<input type="checkbox"/> Se mostrará la lista de productos de la “categoría” de promociones.	Si

En la Tabla 2.15 se puede ver el cumplimiento del sprint 4 por tareas.

**Tabla 2.15:** Sprint 4 cumplimiento

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo No o parcial
HU09	Crear intent de búsqueda para que el bot conteste de forma adecuada.	Si	
	Conectar el intent con el backend para que pueda realizar las búsquedas de “menú detalles”.	Si	
	Conectar el bot a una pantalla.	Si	
	Configurar el comando para llamar al bot mediante las interfaces de google	Si	

HU10	Conectar frontend al servicio REST para obtener el menú.	Si	
	Conectar el websocket para que escuche la navegación y búsqueda de "menú detalle".	Si	
	Crear lógica de navegación en front al escuchar un evento.	Si	
HU11	Modificar el intent de búsqueda para que si no encuentra un "menú detalle", dispare un evento al frontend para dirigir a las promociones.	Si	
	Modificar el intent para que el dispositivo lea los "menú detalles" de la "categoría" promociones.	Si	
HU12	Al recibir un evento de producto "categoría" o "subcategoría" no encontrado la aplicación frontend deberá desplazarse hacia las promociones.	Si	
HU01	Cargar el menú en las entidades reconocidas por el bot para poderlas llamar por su nombre.	Si	

## **Sprint retrospective**

### **¿Se alcanzaron los objetivos propuestos?**

Si

### **¿Qué salió bien en la iteración?**

- Se logró configurar el bot mediante la aplicación web Dialogflow.
- Se logró conectar el bot con la aplicación backend mediante un servicio REST utilizando el servicio fulfillment de Dialogflow.
- Se creó la interacción de búsqueda del menú mediante voz.

## ¿Qué se puede mejorar?

- ❑ El servicio de Dialogflow para comunicar las entidades del bot no está bien documentada por lo cual se perdió gran cantidad de tiempo en esta tarea.
- ❑ Se actualizó la librería de terceros para utilizar WebSocket por lo cual se tuvo que leer la documentación nuevamente.

En la Figura 2.28 se puede apreciar el esfuerzo realizado durante los días del sprint, como se puede observar, se consiguió completar los puntos planeados para el presente sprint sin mayores inconvenientes. Los puntos planeados para el presente sprint se completaron a pesar de las dificultades descritas anteriormente.

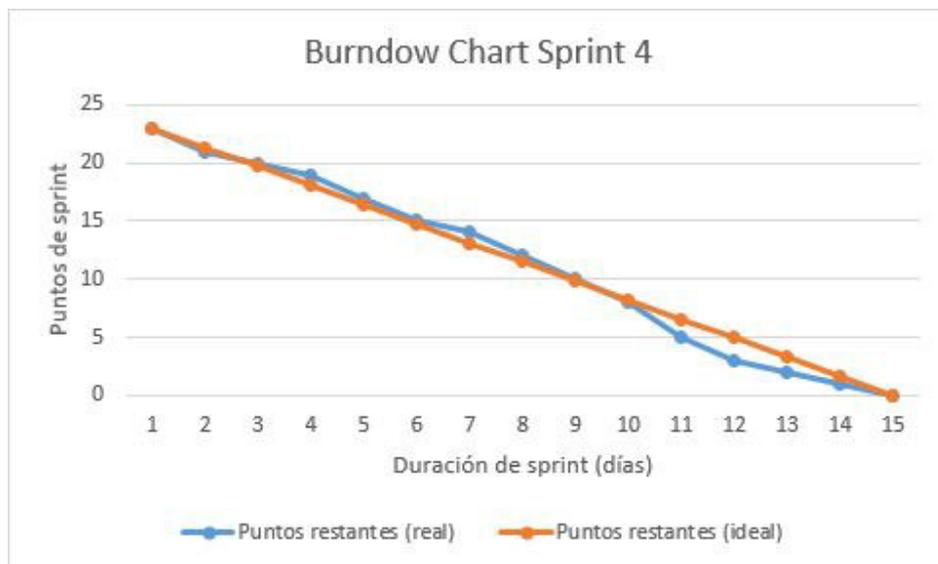


Figura 2.28: Burndown Chart del Sprint 4

## 2.6.6 Sprint 5

### Sprint planning

#### Objetivos del sprint

- ❑ Navegar por el “menú” utilizando la voz y la pantalla de apoyo
- ❑ Obtener la información necesaria desde el backend para mostrarla información del “menú detalle” de interés.

- Crear la conversación para ver el “detalle” de un “menú detalle”.
- Navegar hacia la pantalla de “detalle de un “menú detalle” y también hacia la pantalla del carrito.
- Obtener la información del carrito desde el backend para mostrarla en la pantalla y en la conversación

### Sprint backlog

Para el backlog del sprint 5 se piensa trabajar las historias de usuario 13, 14, 17 y 18 cuyas tareas se encuentran listadas en la Tabla

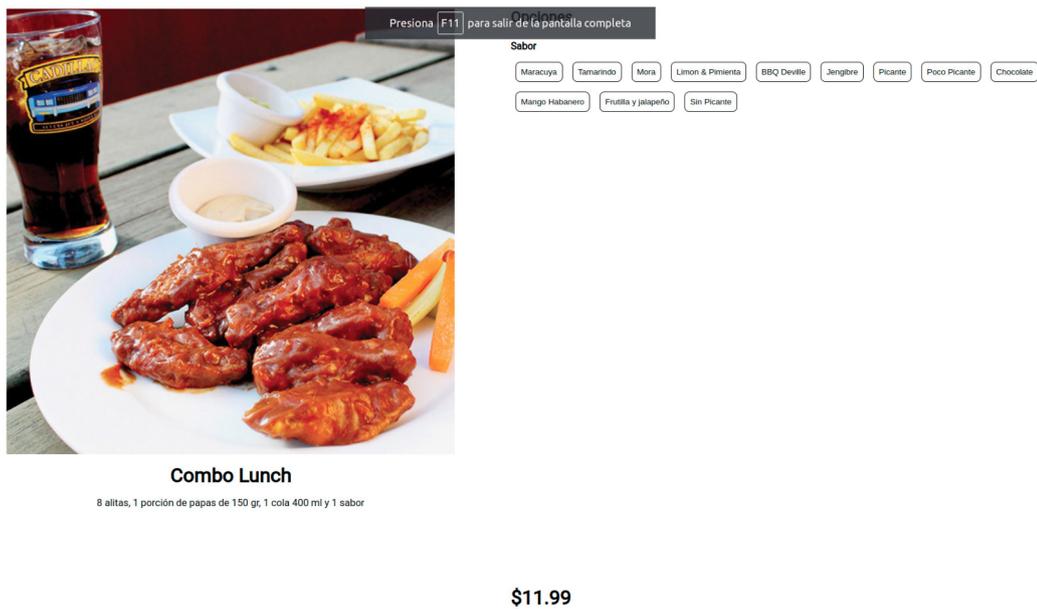
**Tabla 2.16:** Sprint 5 backlog

Código	Historia de usuario	Tareas del sprint
HU13	Yo como usuario deseo obtener más detalles de un “menú detalle” que mencione.	<ul style="list-style-type: none"> <li><input type="checkbox"/> Crear el intent para decir el detalle de un “menú detalle” mediante el dispositivo de voz.</li> <li><input type="checkbox"/> Conectar el intent con el backend para que pueda obtener el detalle del “menú detalle”.</li> <li><input type="checkbox"/> Configurar la respuesta del bot, cuando encuentra o no encuentra el “menú detalle” seleccionado.</li> </ul>
HU14	Yo como usuario deseo ver más detalles de un “menú detalle” que mencioné en la pantalla de apoyo.	<ul style="list-style-type: none"> <li><input type="checkbox"/> Obtener la información del “menú detalle” desde el backend para mostrarlo en la pantalla de apoyo.</li> <li><input type="checkbox"/> Maquetar la pantalla del detalle del “menú detalle”.</li> <li><input type="checkbox"/> Navegar hacia la pantalla del carrito cuando el intent dispare el evento adecuado.</li> </ul>

<p>HU17</p>	<p>Yo como usuario deseo listar mi carrito de compras.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Crear el intent para listar el detalle del “menú detalle” cuando el cliente quiera ir a su carrito de compras.</li> <li><input type="checkbox"/> Obtener el carrito de compras del usuario desde el backend de la aplicación.</li> <li><input type="checkbox"/> Configurar la respuesta del bot cuando el carrito tiene “menú detalles”.</li> <li><input type="checkbox"/> Configurar la respuesta del bot cuando el carrito no tiene “menú detalles”.</li> </ul>
<p>HU18</p>	<p>Yo como usuario deseo ver mi carrito de compras en la pantalla de apoyo.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Obtener la información del carrito desde el backend y mostrarlo en la aplicación de la pantalla de apoyo.</li> <li><input type="checkbox"/> Maquetar el carrito cuando este tiene “menú detalle”.</li> <li><input type="checkbox"/> Maquetar el carrito cuando este no tiene “menú detalle”.</li> <li><input type="checkbox"/> Navegar hacia el carrito cuando el intent dispare el evento de ir al carrito.</li> </ul>

### Ejecución del sprint

1. Crear intent de más detalles de un “menú detalle” en Dialogflow. En la Figura 2.29 se puede observar la pantalla que reacciona a la interacción con este intent.



**Figura 2.29:** Pantalla para ver más detalles

- a) Crear intent de más detalles de un “menú detalle”.
  - b) Configurar el comportamiento del webhook en backend cuando se invoca el intent de más detalles de un “menú detalle”.
  - c) Maquetar la pantalla en la aplicación frontend cliente a que muestra un “menú detalle”, la descripción corta y larga, la imagen, las opciones y el precio.
  - d) Conectar la aplicación frontend cliente con los eventos del intent de ver más detalles de un “menú detalle”, mediante el webhook.
2. Crear intent para ir al carrito de compras en Dialogflow. En la Figura 2.30 se puede observar la pantalla que reacciona a la interacción con este intent.



**Figura 2.30:** Pantalla de carrito con “menú detalle”

- a) Crear intent para ir al carrito de compras en Dialogflow.
- b) Configurar el comportamiento del webhook en back cuando se invoca el intent para ir al carrito de compras
- c) Maquetar la pantalla en la aplicación frontend cliente cuando el carrito de compras está vacío.
- d) Maquetar la pantalla en la aplicación frontend cliente cuando el carrito de compras tiene productos.
- e) Conectar la aplicación frontend cliente con los eventos del intent ir al carrito de compras mediante el webhook.

### Sprint review

Una vez terminado el sprint 5, se procede a realizar el sprint review, el cuál que muestra el cumplimiento de las historias de usuario desarrolladas, este resultado se encuentra en la Tabla 2.17

**Tabla 2.17:** Sprint 5 review

Historia	Criterio de aceptación	Cumplimiento
HU13	<input type="checkbox"/> Se leerá la descripción larga del “menú detalle” mencionado.	Si

HU14	<input type="checkbox"/> Se mostrará la descripción larga del “menú detalle” mencionado. <input type="checkbox"/> Se mostrarán las opciones disponibles del “menú detalle” mencionado.	Si
HU17	<input type="checkbox"/> Se leerá la cantidad de los “menú detalle” agregados al carrito y su descripción corta.	Si
HU18	<input type="checkbox"/> Se mostrará la lista de “menú detalle” agregados al carrito junto con sus cantidades.	Si

En la 2.18 se puede ver el cumplimiento del sprint 5 por tareas.

**Tabla 2.18:** Sprint 5 cumplimiento

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo No o parcial
HU13	Crear el intent para decir el detalle de un “menú detalle” mediante el dispositivo de voz.	Si	
	Conectar el intent con el backend para que pueda obtener el detalle del “menú detalle”.	Si	
	Configurar la respuesta del bot, cuando encuentra o no encuentra el “menú detalle” seleccionado.	Si	
HU14	Obtener la información del “menú detalle” desde el backend para mostrarlo en la pantalla de apoyo.	Si	
	Maquetar la pantalla del detalle del “menú detalle”.	Si	

	Navegar hacia la pantalla del carrito cuando el intent dispare el evento adecuado.	Si	
HU17	Crear el intent para listar el detalle del "menú detalle" cuando el cliente quiera ir a su carrito de compras.	Si	
	Obtener el carrito de compras del usuario desde el backend de la aplicación.	Si	
	Configurar la respuesta del bot cuando el carrito tiene "menú detalle".	Si	
	Configurar la respuesta del bot cuando el carrito no tiene "menú detalle".	Si	
HU18	Obtener la información del carrito desde el backend y mostrarlo en la aplicación de la pantalla de apoyo.	Si	
	Maquetar el carrito cuando este tiene "menú detalle".	Si	
	Maquetar el carrito cuando este no tiene "menú detalle".	Si	
	Navegar hacia el carrito cuando el intent dispare el evento de ir al carrito.	Si	

### **Sprint retrospective**

#### **¿Se alcanzaron los objetivos propuestos?**

Si

#### **¿Qué salió bien en la iteración?**

- Se creó la navegación a la pantalla de detalles de un "menú detalle" y al carrito
- Se logró realizar búsquedas en el módulo del servicio REST del bot.
- Se crearon las respuestas para la interfaz de voz mediante el servicio REST del bot.

## ¿Qué se puede mejorar?

- Se puede utilizar lenguaje de marcación de síntesis de voz (SSML) para una mejor conversación del bot.

En la Figura 2.31 se puede apreciar el esfuerzo realizado durante los días del sprint, durante el sprint, se empezó completando las tareas con pesos entre 1 y 2 puntos diariamente; conforme se avanzaba el sprint, se tuvieron que completar tareas con pesos entre 3 y 4 puntos diariamente para así cumplir con lo planificado para el sprint.

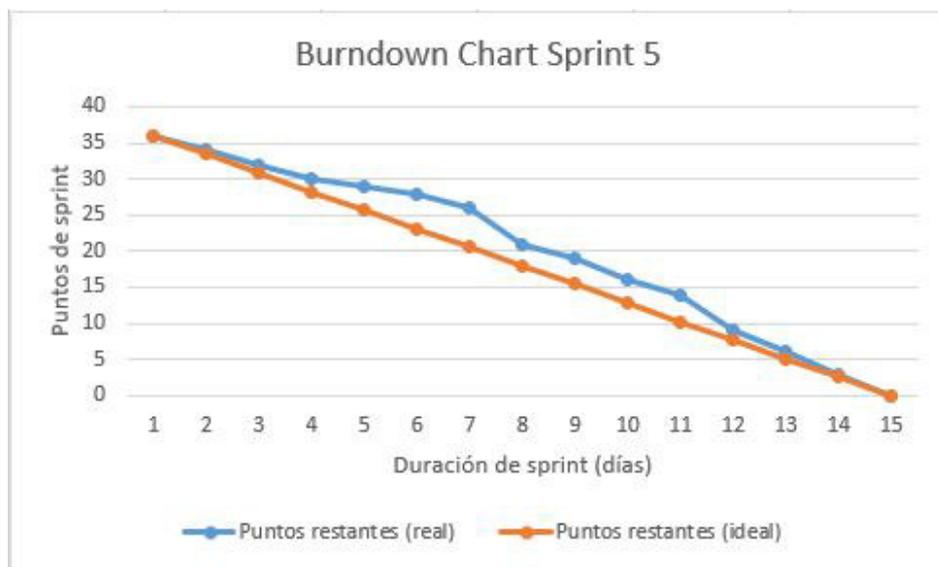


Figura 2.31: Burndown Chart del Sprint 5

## 2.6.7 Sprint 6

### Sprint planning

#### Objetivos del sprint

- Agregar “menú detalle” al carrito utilizando el asistente de voz
- Obtener la información necesaria del cliente para crear el “menú detalle” en el carrito con su cantidad y “opciones”.
- Crear la conversación en caso de que el cliente no proporcione la información completa

- Escuchar el carrito del usuario para maquetar un indicador visual en la pantalla de apoyo

### Sprint backlog

Para el backlog del sprint 6 se piensa trabajar las historias de usuario 15 y 16 cuyas tareas se encuentran listadas en la Tabla 2.19.

**Tabla 2.19:** Sprint 6 backlog

Código	Historia de usuario	Tareas del sprint
HU15	Yo como cliente deseo agregar un “menú detalle” al carrito.	<ul style="list-style-type: none"> <li><input type="checkbox"/> Obtener la información del “menú detalle” requerido por el usuario.</li> <li><input type="checkbox"/> Preguntar al cliente por la cantidad a agregar del “menú detalle”.</li> <li><input type="checkbox"/> Preguntar al cliente por las “opciones” que se le agregan al “menú detalle”.</li> <li><input type="checkbox"/> Verificar que la información recolectada sea correcta.</li> <li><input type="checkbox"/> Enviar la información al backend para que registre un “menú detalle” en el carrito del usuario.</li> </ul>
HU16	Yo como cliente deseo que al agregar un “menú detalle” al carrito se pueda observar el icono de carrito lleno en la pantalla de apoyo.	<ul style="list-style-type: none"> <li><input type="checkbox"/> Escuchar al carrito del usuario para maquetar un indicador visual en la aplicación web de la pantalla de apoyo.</li> </ul>

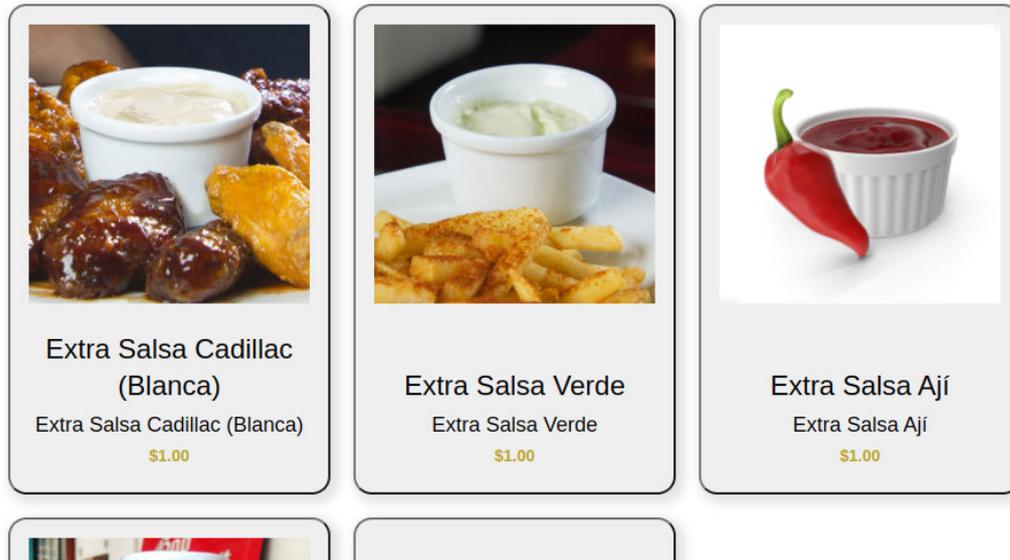
### Ejecución del sprint

1. Crear intent de agregar un “menú detalle” al carrito en Dialogflow. En la Figura 2.32 se puede observar la pantalla que reacciona a la interacción con este intent.



## Extras

### Salsas



**Figura 2.32:** Pantalla de “menú” con el ícono del carrito

- Crear intent de agregar un “menú detalle” al carrito en Dialogflow.
- Configurar el comportamiento del webhook en back cuando se invoca el intent de agregar un “menú detalle” al carrito.
- Configurar el comportamiento del webhook en back cuando se invoca el intent de agregar un “menú detalle” al carrito y tiene opciones.
- Maquetar el icono de carrito con un punto rojo cuando hay “menú detalle” en el carrito en la aplicación frontend cliente.
- Conectar la aplicación frontend cliente con los eventos del intent de agregar un “menú detalle” al carrito, mediante el webhook.

### Sprint review

Una vez terminado el sprint 6, se procede a realizar el sprint review, el cuál que muestra el cumplimiento de las historias de usuario desarrolladas, este resultado se encuentra en la Tabla 2.20.

**Tabla 2.20:** Sprint 6 review

Historia	Criterio de aceptación	Cumplimiento
----------	------------------------	--------------

HU15	<input type="checkbox"/> Puedo agregar un “menú detalle” sin opciones. <input type="checkbox"/> Puedo agregar un “menú detalle” con opciones.	Si
HU16	<input type="checkbox"/> Se mostrará un icono de carrito lleno cuando se agregue un “menú detalle” al carrito.	Si

En la Tabla 2.21 se puede ver el cumplimiento del sprint 6 por tareas.

**Tabla 2.21:** Sprint 6 cumplimiento

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo No o parcial
HU15	Obtener la información del “menú detalle” requerido por el usuario.	Si	
	Preguntar al cliente por la cantidad a agregar del “menú detalle”.	Si	
	Preguntar al cliente por las “opciones” que se le agregan al “menú detalle”.	Si	
	Verificar que la información recolectada sea correcta.	Si	
	Enviar la información al backend para que registre un “menú detalle” en el carrito del usuario.	Si	
HU16	Escuchar al carrito del usuario para maquetar un indicador visual en la aplicación web de la pantalla de apoyo.	Si	

### Sprint retrospective

**¿Se alcanzaron los objetivos propuestos?**

Si

### ¿Qué salió bien en la iteración?

- Se logró añadir un “menú detalle” al carrito con su cantidad y sus opciones.
- Se logró hacer que la pantalla muestre un indicador cuando el carrito tiene “menú detalles”.
- Se logró validar el “menú detalle” antes de agregarlo al carrito.

### ¿Qué se puede mejorar?

- La interacción por voz con el bot aún no es la adecuada pero se puede mejorar entrenando el bot.

En la Figura 2.33 se puede apreciar el esfuerzo realizado durante los días del sprint. A pesar de la carga de puntos por completar para este sprint fue pesada debido a la integración del bot con el carrito de compras, se pudo completar los puntos planificados sin mayores inconvenientes.

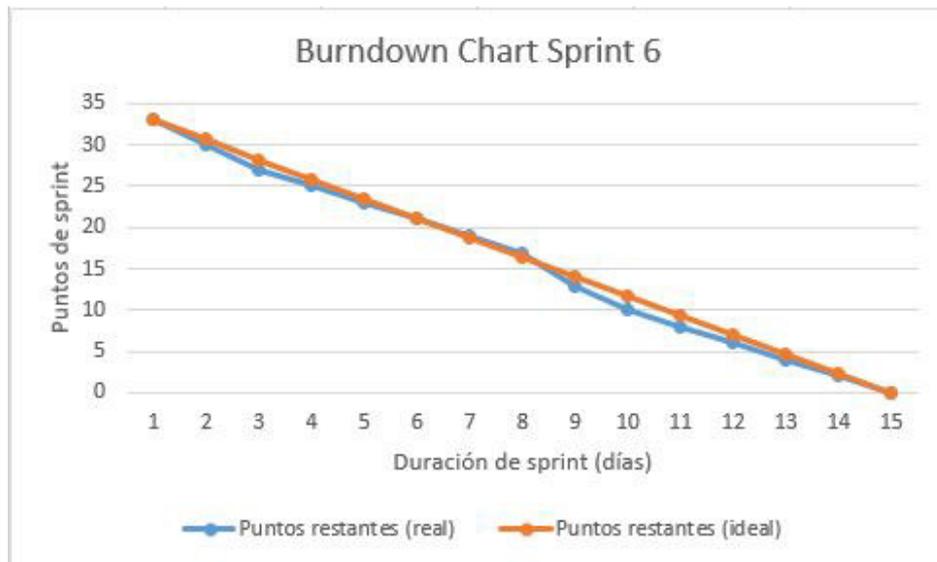


Figura 2.33: Burndown Chart del Sprint 6

## 2.6.8 Sprint 7

### Sprint planning

### Objetivos del sprint

- Aumentar la cantidad de un “menú detalle” del carrito de compras mediante la aplicación de voz.
- Disminuir la cantidad de un “menú detalle” del carrito de compras mediante la aplicación de voz.
- Quitar un “menú detalle” del carrito de compra mediante la aplicación de voz.
- Recargar la página una vez que se hagan cambios en el carrito para reflejar los cambios.

### Sprint backlog

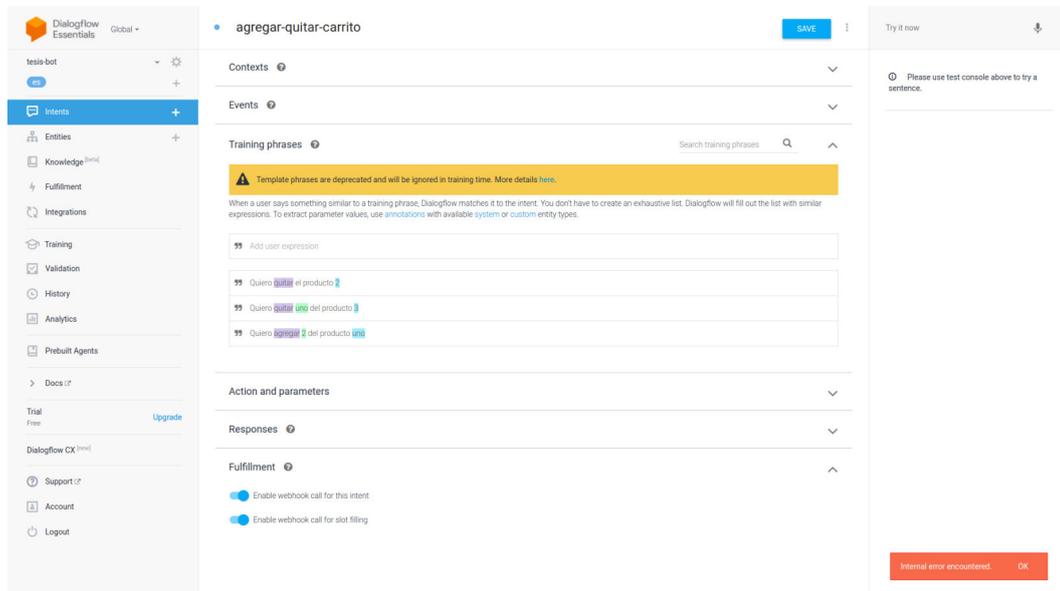
Para el backlog del sprint 7 se piensa trabajar las historias de usuario 19 y 20 cuyas tareas se encuentran listadas en la Tabla 2.22.

**Tabla 2.22:** Sprint 7 backlog

Código	Historia de usuario	Tareas del sprint
HU19	Yo como usuario deseo modificar un “menú detalle” de mi carrito de compras.	<ul style="list-style-type: none"> <li><input type="checkbox"/> Crear intent para manipular las cantidades de un “menú detalle” cuando el usuario lo desee.</li> <li><input type="checkbox"/> Obtener los “menú detalles” del carrito.</li> <li><input type="checkbox"/> Configurar la respuesta del bot para cuando el cliente añade o disminuye la cantidad de un “menú detalle”.</li> <li><input type="checkbox"/> Configurar la respuesta del bot cuando el cliente quita un “menú detalle” del carrito.</li> <li><input type="checkbox"/> Enviar la información al backend para que estas acciones se vean reflejadas en el carrito.</li> </ul>
HU20	Yo como usuario deseo que al modificar un “menú detalle” de mi carrito de compras se visualice la información del “menú detalle” y los cambios que se realizaron en la pantalla de apoyo.	<ul style="list-style-type: none"> <li><input type="checkbox"/> Una vez que el usuario añada o disminuya la cantidad de un “menú detalle” se recargará la página para obtener la nueva información del carrito.</li> </ul>

## Ejecución del sprint

1. Crear intent para editar los “menú detalles” del carrito en Dialogflow. En la Figura 2.34 se puede observar la pantalla de configuración de este intent.



**Figura 2.34:** Intent para editar los “menú detalle” del carrito

- a) Crear intent para editar los “menú detalles” del carrito en Dialogflow.
- b) Configurar el comportamiento del webhook en back cuando se invoca el intent para añadir o disminuir la cantidad de un “menú detalle” del carrito.
- c) Configurar el comportamiento del webhook en back cuando se invoca el intent para quitar un “menú detalle” del carrito.
- d) Conectar la aplicación frontend cliente para refrescar la pantalla del carrito una vez que se hicieron los cambios.

## Sprint review

Una vez terminado el sprint 7, se procede a realizar el sprint review, el cuál que muestra el cumplimiento de las historias de usuario desarrolladas, este resultado se encuentra en la Tabla 2.23.

**Tabla 2.23:** Sprint 7 review

Historia	Criterio de aceptación	Cumplimiento
----------	------------------------	--------------

HU19	<input type="checkbox"/> Se leerá la descripción corta del “menú detalle” agregado al carrito. <input type="checkbox"/> Puedo modificar la cantidad del “menú detalle”. <input type="checkbox"/> Puedo modificar las opciones asociadas al “menú detalle”.	Si
HU20	<input type="checkbox"/> Se mostrará la pantalla de información del “menú detalle” y la cantidad seleccionada del producto. <input type="checkbox"/> Se mostrarán las “opciones” elegidas por el usuario.	Si

En la Tabla 2.24 se puede ver el cumplimiento del sprint 7 por tareas.

**Tabla 2.24:** Sprint 7 cumplimiento

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo No o parcial
HU19	Crear intent para manipular las cantidades de un “menú detalle” cuando el usuario lo desee.	Si	
	Obtener los “menú detalles” del carrito.	Si	
	Configurar la respuesta del bot para cuando el cliente añade o disminuye la cantidad de un “menú detalle”.	Si	
	Configurar la respuesta del bot cuando el cliente quita un producto del “menú detalle”.	Si	
	Enviar la información al backend para que estas acciones se vean reflejadas en el carrito.	Si	

HU20	Una vez que el usuario añade o disminuya la cantidad de un “menú detalle” se recargará la página para obtener la nueva información del carrito.	Si	
------	---	----	--

## **Sprint retrospective**

### **¿Se alcanzaron los objetivos propuestos?**

Si

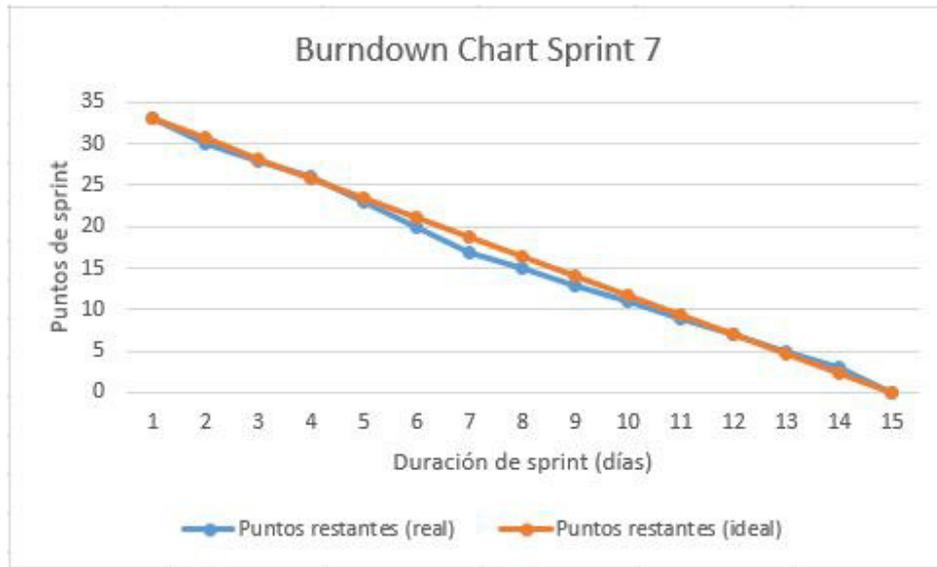
### **¿Qué salió bien en la iteración?**

- Se logró interactuar con el carrito añadiendo y quitando la cantidad de los “menú detalles”.
- Se logró refrescar la página cuando se edita el carrito para que se visualicen los cambios.
- El backend y el frontend se comunican en tiempo real cuando se dispara el evento de edición del carrito.

### **¿Qué se puede mejorar?**

- Cuando el cliente quiere cambiar las “opciones” del “menú detalle” es mejor que el cliente borre el “menú detalle” del carrito y lo agregue nuevamente con las nuevas “opciones”.
- La pantalla se refresca totalmente cuando se edita el carrito, puede ser que exista una mejor solución.

En la Figura 2.35 se puede apreciar el esfuerzo realizado durante los días del sprint. Los puntos planificados para este sprint tuvo una carga similar a la del sprint anterior y dejando de lado el problema de que la pantalla se refresca totalmente al editar el carrito, se cumplieron con todos los puntos planificados para el sprint.



**Figura 2.35:** Burndown Chart del Sprint 7

## 2.6.9 Sprint 8

### Sprint planning

#### Objetivos del sprint

- Finalizar la conversación del asistente de voz.
- Enviar el pedido si el cliente quiere enviar el pedido.
- Eliminar el carrito si el cliente no quiere terminar la conversación sin hacer un pedido.
- Mostrar un mensaje en la pantalla cuando el cliente termine la interacción por voz.

#### Sprint backlog

Para el backlog del sprint 8 se piensa trabajar las historias de usuario 21 y 22 cuyas tareas se encuentran listadas en la Tabla 2.25.

**Tabla 2.25:** Sprint 8 backlog

Código	Historia de usuario	Tareas del sprint
--------	---------------------	-------------------

<p>HU21</p>	<p>Yo como usuario deseo terminar el proceso de compra.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Obtener el pedido del cliente.</li> <li><input type="checkbox"/> Enviar el pedido si el usuario así lo requiere.</li> <li><input type="checkbox"/> Eliminar el carrito si el cliente no quiere enviar el carrito.</li> <li><input type="checkbox"/> Cambiar el estado del pedido.</li> <li><input type="checkbox"/> Obtener el menú de la aplicación.</li> <li><input type="checkbox"/> Configurar la respuesta del bot con el mensaje configurado en el menú.</li> </ul>
<p>HU22</p>	<p>Yo como usuario deseo que al terminar el proceso de compra se muestre un mensaje en la pantalla de apoyo.</p>	<ul style="list-style-type: none"> <li><input type="checkbox"/> Mostrar un mensaje de despedida para el usuario.</li> <li><input type="checkbox"/> Redirigir nuevamente a la pantalla de inicio luego de 10 segundos.</li> </ul>

### Ejecución del sprint

1. Crear intent para terminar la compra en Dialogflow. En la Figura 2.36 se puede observar la pantalla que reacciona a la interacción con este intent.

Gracias por venir a este restaurante.

**Figura 2.36:** Pantalla con mensaje final

- a) Crear intent para terminar la compra en Dialogflow.
- b) Configurar el comportamiento del webhook en back cuando se invoca el intent para terminar la compra cuando se realizó un pedido.
- c) Configurar el comportamiento del webhook en back cuando se invoca el intent para terminar la compra cuando no se realizó un pedido.
- d) Maquetar la pantalla final con el mensaje fijado en el menú.
- e) Conectar la aplicación frontend cliente con los eventos del intent para terminar la compra, mediante el webhook.

### **Sprint review**

Una vez terminado el sprint 8, se procede a realizar el sprint review, el cuál que muestra el cumplimiento de las historias de usuario desarrolladas, este resultado se encuentra en la Tabla 2.26.

**Tabla 2.26:** Sprint 8 review

<b>Historia</b>	<b>Criterio de aceptación</b>	<b>Cumplimiento</b>
HU21	<input type="checkbox"/> Puedo finalizar el proceso de compra.	Si

HU22	<input type="checkbox"/> Puedo finalizar el proceso de compra. <input type="checkbox"/> Se mostrará un mensaje final al terminar la compra	Si
------	---	----

En la Tabla 2.27 se puede ver el cumplimiento del sprint 8 por tareas.

**Tabla 2.27:** Sprint 8 cumplimiento

Código	Tarea	Cumplimiento (Si/No/Parcial)	Motivo No o parcial
HU21	Obtener el pedido del cliente.	Si	
	Enviar el pedido si el usuario así lo requiere.	Si	
	Eliminar el carrito si el cliente no quiere enviar el carrito.	Si	
	Cambiar el estado del pedido.	Si	
	Obtener el menú de la aplicación.	Si	
	Configurar la respuesta del bot con el mensaje configurado en el menú.	Si	
HU22	Mostrar un mensaje de despedida para el usuario.	Si	
	Redirigir nuevamente a la pantalla de inicio luego de 10 segundos.	Si	

### **Sprint retrospective**

#### **¿Se alcanzaron los objetivos propuestos?**

Si

#### **¿Qué salió bien en la iteración?**

- Se logró terminar la conversación cuando el cliente realiza un pedido.
- Se logró terminar la conversación cuando el cliente no realiza ningún pedido y solo quiere terminar la interacción.

- ❑ Se logró mostrar el mensaje adecuado cuando se termina la interacción con el asistente virtual.

### ¿Qué se puede mejorar?

- ❑ Al terminar la interacción se debe borrar el carrito en el caso que el cliente no haya realizado un pedido.
- ❑ Ajustar el tiempo que se muestra en la pantalla de mensaje final.

El en la figura 2.37 se puede apreciar el esfuerzo realizado durante los días del sprint, como se puede observar, se consiguió completar los puntos planeados para el presente sprint sin mayores inconvenientes. Debido a que la cantidad de puntos planificados para este sprint era pequeña en comparación a sprints anteriores, la tendencia de puntos restantes real de la gráfica se muestra con esa tendencia en zig zag, pues hubo días en los que se pulía el sistema y otros en los que se completaban los puntos planificados.

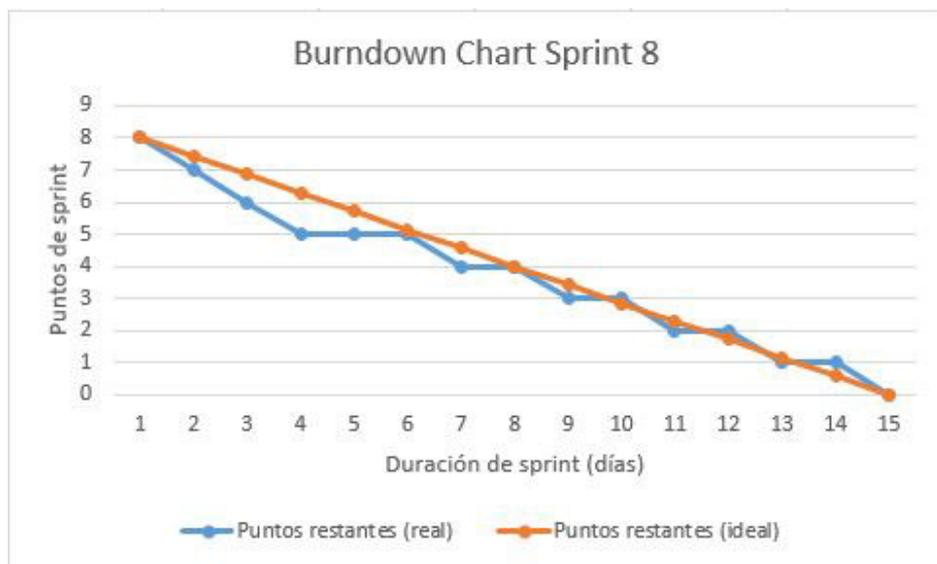
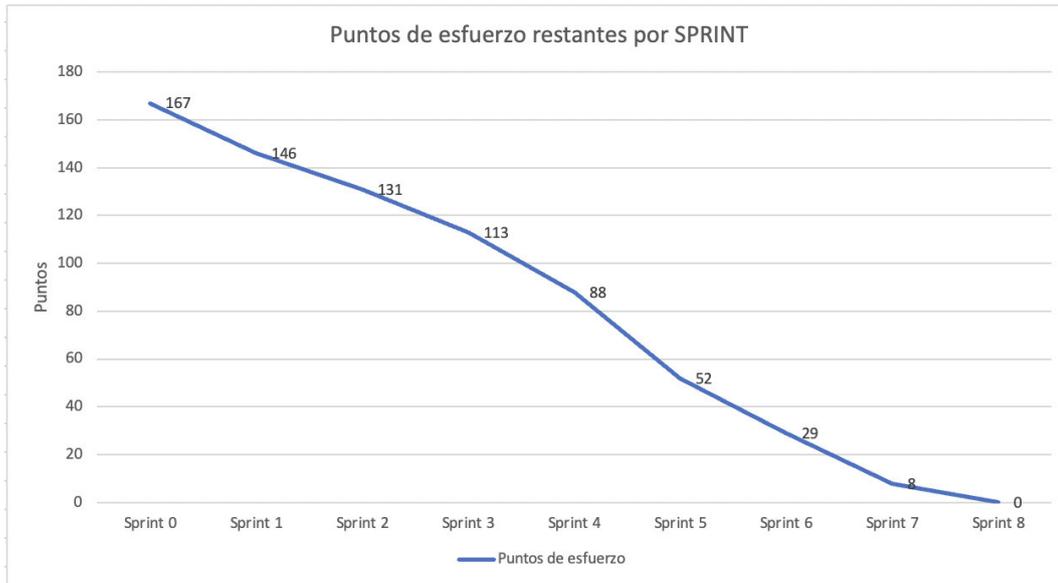


Figura 2.37: Burndown Chart del Sprint 8

## 2.6.10 Conclusiones sobre la ejecución de los sprints

El periodo de desarrollo concluyó con el cumplimiento de los objetivos planteados. En la Figura 2.38 se muestra el avance de los sprints de acuerdo con los puntos de esfuerzo que restaron por cada sprint.



**Figura 2.38:** Gráfico de avance: Puntos de esfuerzo / Sprint

Como se indica en el gráfico el avance durante el desarrollo fue constante y sin contratiempos.

#### ¿Qué salió bien en la iteración?

- Se lograron alcanzar los objetivos planteados en cada sprint mediante los criterios de aceptación.
- La carga de trabajo fue repartida de manera adecuada para cada sprint.
- La metodología permitió iterar rápidamente sobre la solución y se ajustó bien a las necesidades del proyecto.

#### ¿Qué se puede mejorar?

- Algunas interacciones pueden ser mejoradas ajustando el flujo de acuerdo a las necesidades de la aplicación. Se recomienda hacer un estudio sobre la aceptación que tienen este tipo de aplicaciones en la sociedad.
- La estimación de algunas tareas no fue exacta debido a la complejidad que representaban con respecto al tiempo asignado. Se recomienda conocer mejor las tecnologías y realizar las estimaciones con una persona que tenga mayor experiencia en este tipo de aplicaciones.

## **3 RESULTADOS**

### **3.1 EVALUACIÓN DE USABILIDAD DEL SISTEMA**

La evaluación de los resultados del sistema se realizó a través de la escala de usabilidad del sistema (SUS por sus siglas en inglés)

Esta evaluación permite obtener una visión general de la usabilidad del sistema de forma rápida y efectiva. Se escogió esta metodología por ser una herramienta confiable para la medición de la facilidad de uso [30].

#### **3.1.1 Descripción de la metodología SUS**

La metodología SUS consiste en recopilar comentarios de un mínimo de cinco personas para obtener comentarios confiables mediante la utilización de un cuestionario de 10 preguntas cuyas respuestas pueden ser desde “totalmente en desacuerdo” hasta “totalmente de acuerdo”.

Las preguntas que están en el cuestionario son las siguientes:

1. Creo que usaría esta aplicación frecuentemente.
2. Encuentro esta aplicación innecesariamente compleja.
3. Creo que la aplicación fue fácil de usar.
4. Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar esta aplicación.
5. Las funciones de esta aplicación están bien integradas.
6. Creo que la aplicación es muy inconsistente.
7. Imagino que la mayoría de la gente aprendería a usar esta aplicación de forma muy rápida.
8. Encuentro que la aplicación es muy difícil de usar.

9. Me siento confiado al usar la aplicación.
10. Necesité aprender muchas cosas antes de ser capaz de usar esta aplicación.

El cálculo de los resultados se realiza de la siguiente forma:

- Sume el puntaje total de todas las preguntas impares, luego reste 5 del total para obtener (X).
- Sume la puntuación total de todas las preguntas pares, luego reste ese total de 25 para obtener (Y).
- Sume la puntuación total de los nuevos valores (X+Y) y multiplíquela por 2,5.

En la metodología SUS, la calificación media de usabilidad es de 68 puntos. Al comparar esta calificación con la calificación obtenida en el sistema que se está evaluando se obtiene una idea inmediata de la usabilidad general de la aplicación desarrollada[30].

### **3.1.2 Realización de la encuesta**

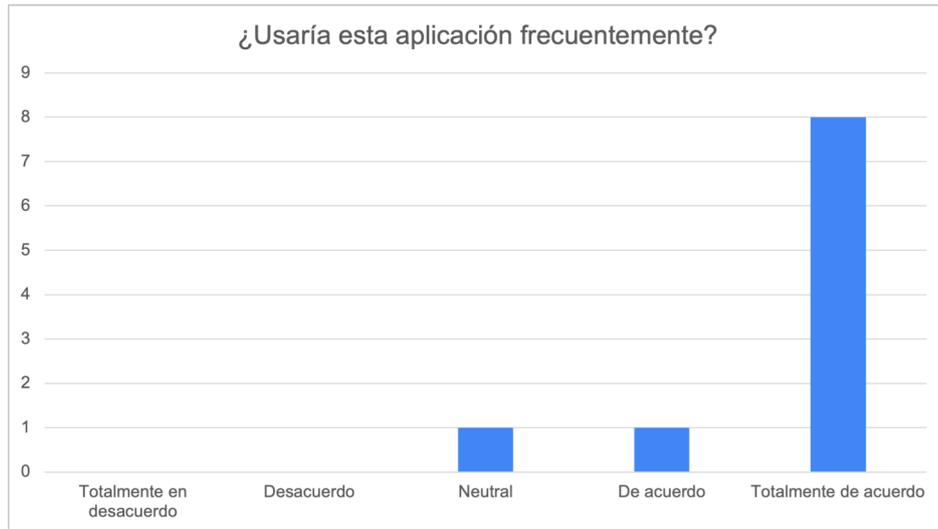
La encuesta se realizó a 10 personas que frecuentan los restaurantes de comida rápida de diferentes cadenas.

A todos los encuestados se les pidió realizar 15 tareas, las cuales cubren varios aspectos sobre la aplicación del cliente final para medir la funcionalidad del sistema, tras lo cual se le pidió responder la encuesta SUS.

Las pruebas cubren las actividades correspondientes a las que realiza el usuario final al realizar una compra mediante la aplicación de voz y estas son: navegar por el menú, realizar una compra, ver el carrito y modificarlo y finalmente terminar la compra.

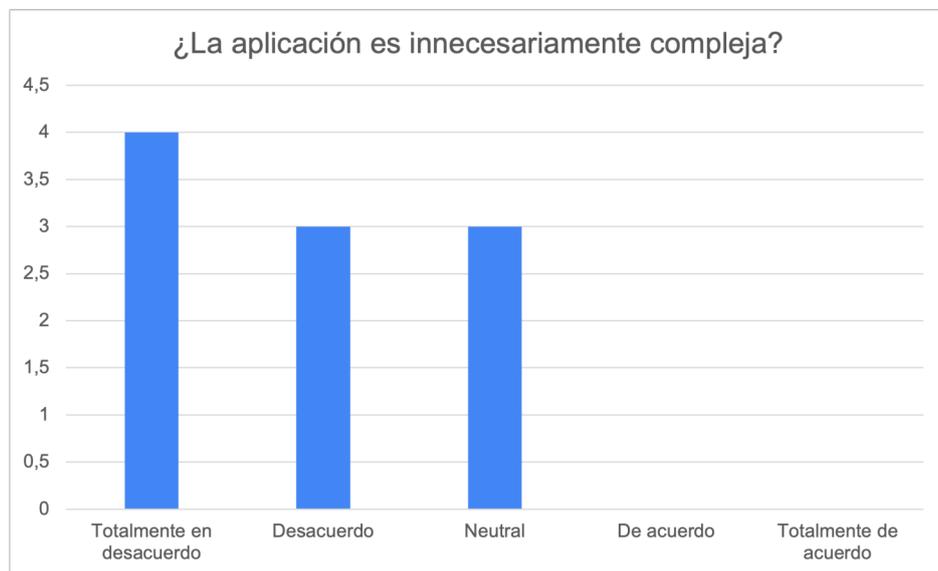
### **3.1.3 Resultados individuales**

Se obtuvieron los siguientes resultados.



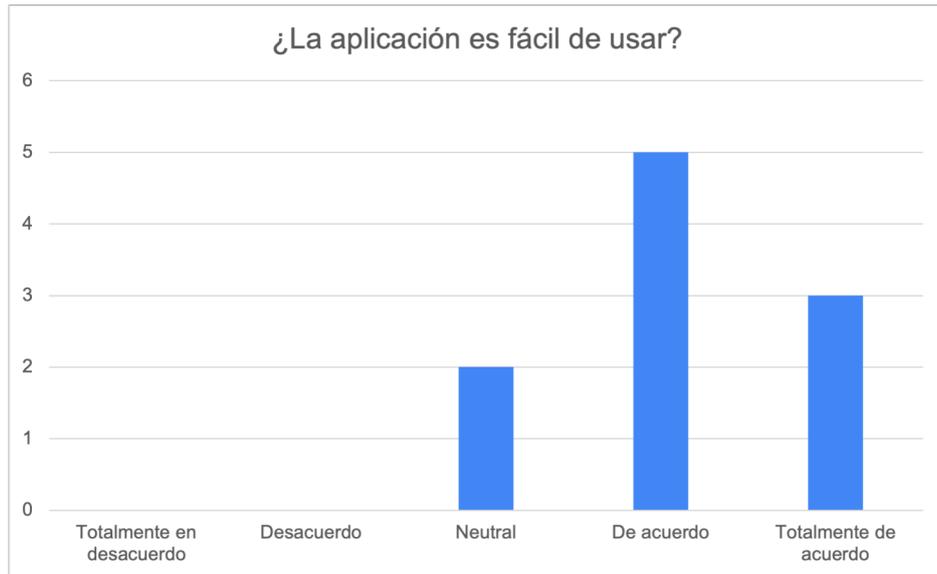
**Figura 3.1:** Resultados de la pregunta 1

En la Figura 3.1 se observa que la mayoría de los usuarios creen que esta aplicación puede ser utilizado por ellos frecuentemente y que el producto podría agilizar la toma de pedidos.



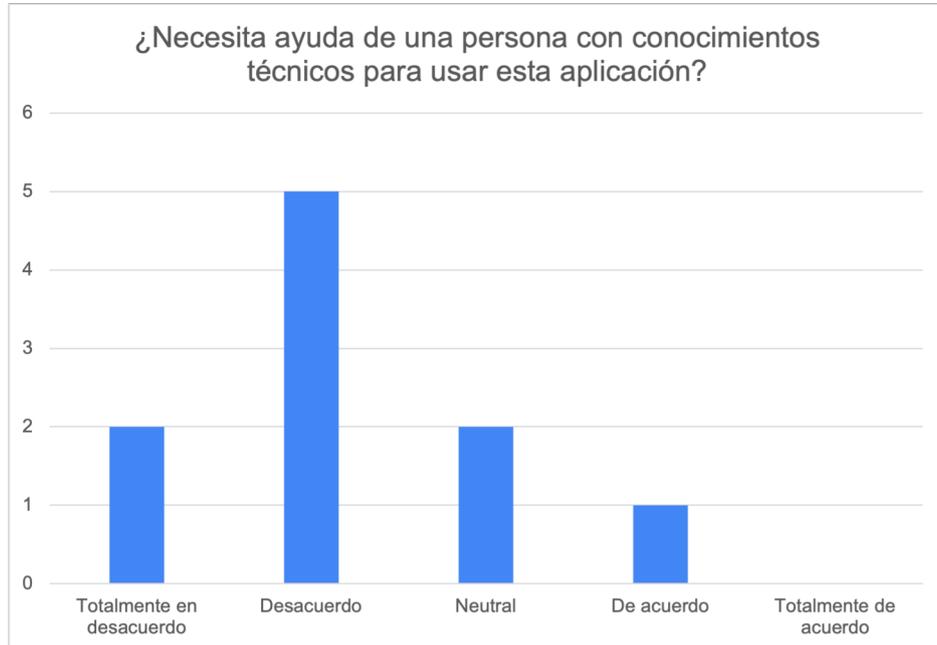
**Figura 3.2:** Resultados de la pregunta 2

En la Figura 3.2 se observa que los usuarios encontraron poca o ninguna dificultad al utilizar el sistema de pedidos, sin embargo, existe un grupo considerable que considera que la complejidad del sistema por lo menos existe.



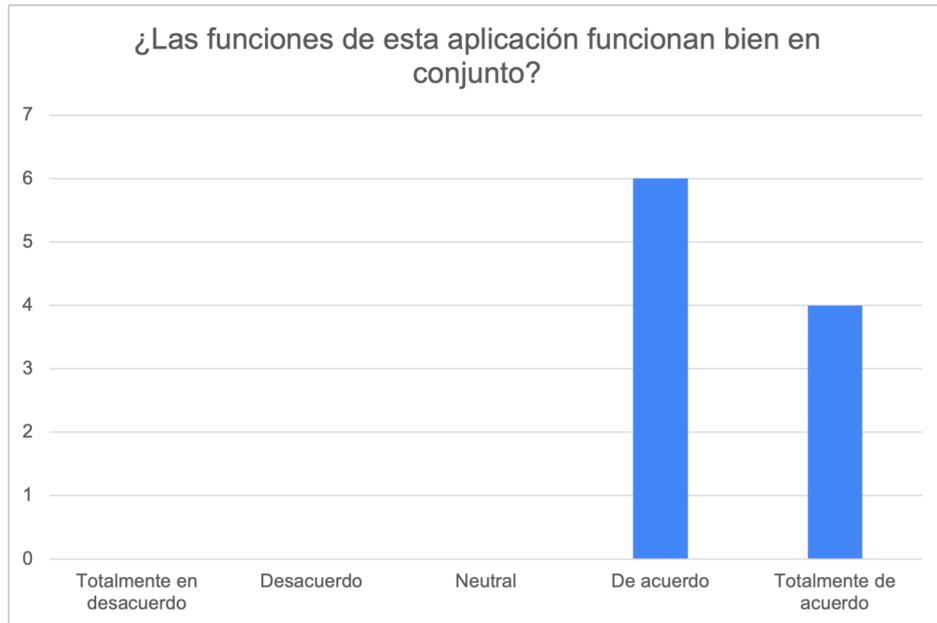
**Figura 3.3:** Resultados de la pregunta 3

En la Figura 3.3 se observa que la mayoría de usuarios considera que la aplicación para realizar pedidos es fácil de usar de esta forma se refuerzan las respuestas de la primera pregunta.



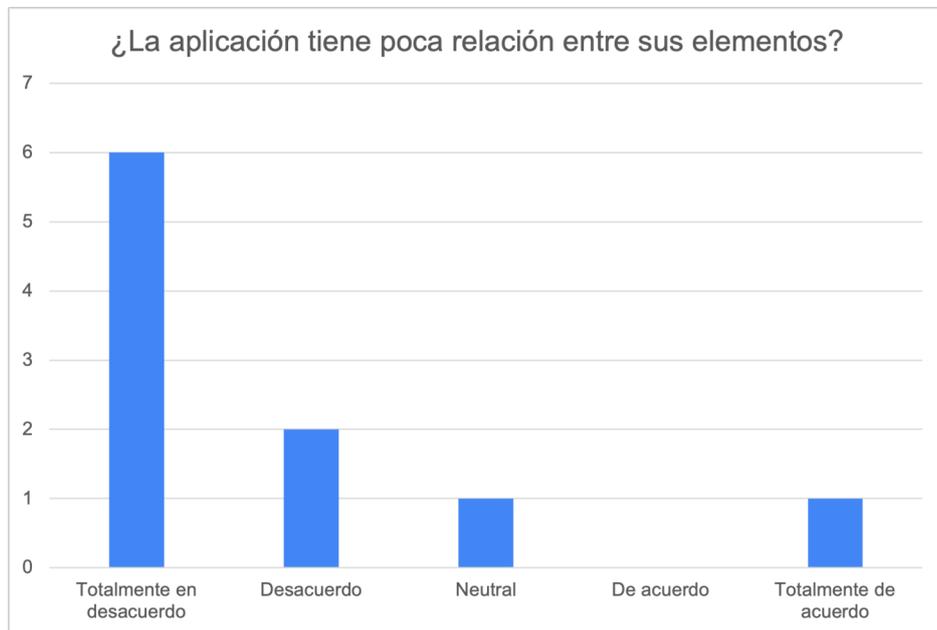
**Figura 3.4:** Resultados de la pregunta 4

En la Figura 3.4 se observa que las respuestas para esta pregunta son diversas, debido a que la mayoría de usuarios pudo utilizar la aplicación por sus medios pero un pequeño grupo se apoyó en una persona con conocimientos técnicos para realizar las pruebas.



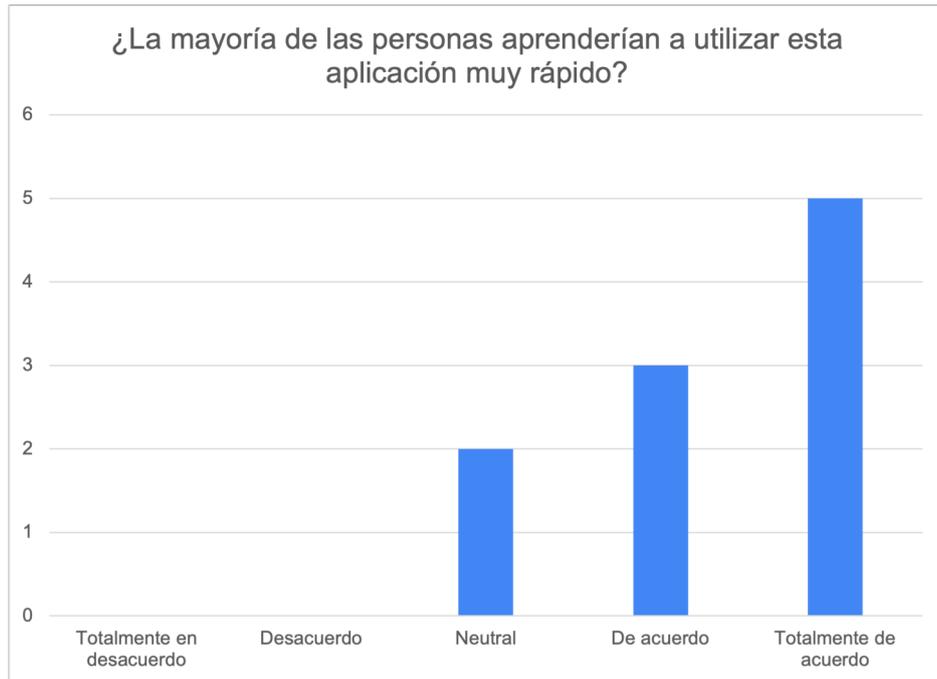
**Figura 3.5:** Resultados de la pregunta 5

En la Figura 3.5 se observa que los usuarios consideran que las funciones de la aplicación son coherentes entre sí.



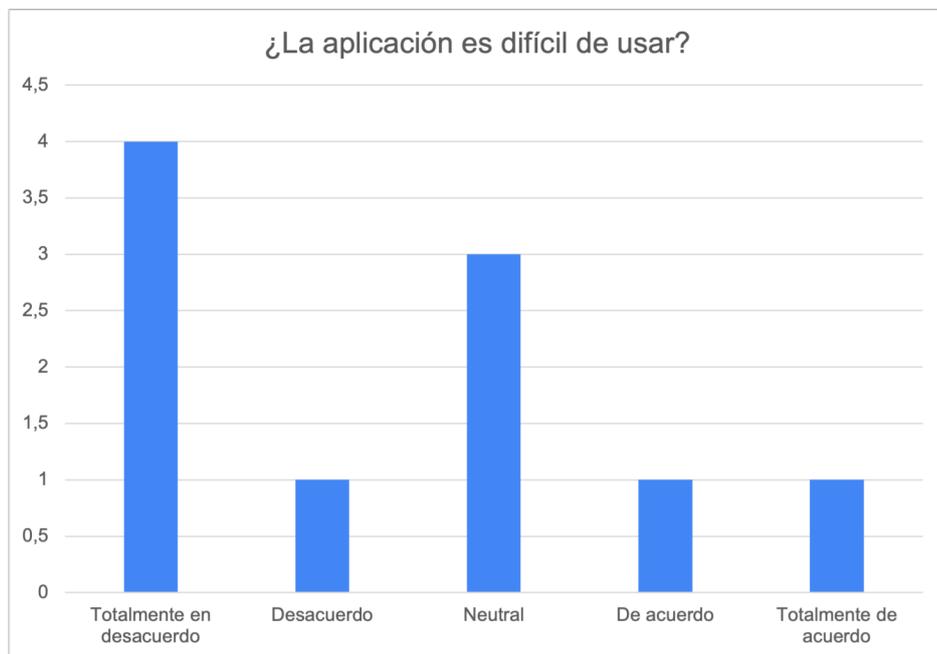
**Figura 3.6:** Resultados de la pregunta 6

En la Figura 3.6 se observa que pocos usuarios consideran que los elementos de la aplicación tienen poca relación entre sí, sin embargo este puede ser un punto a mejorar.



**Figura 3.7:** Resultados de la pregunta 7

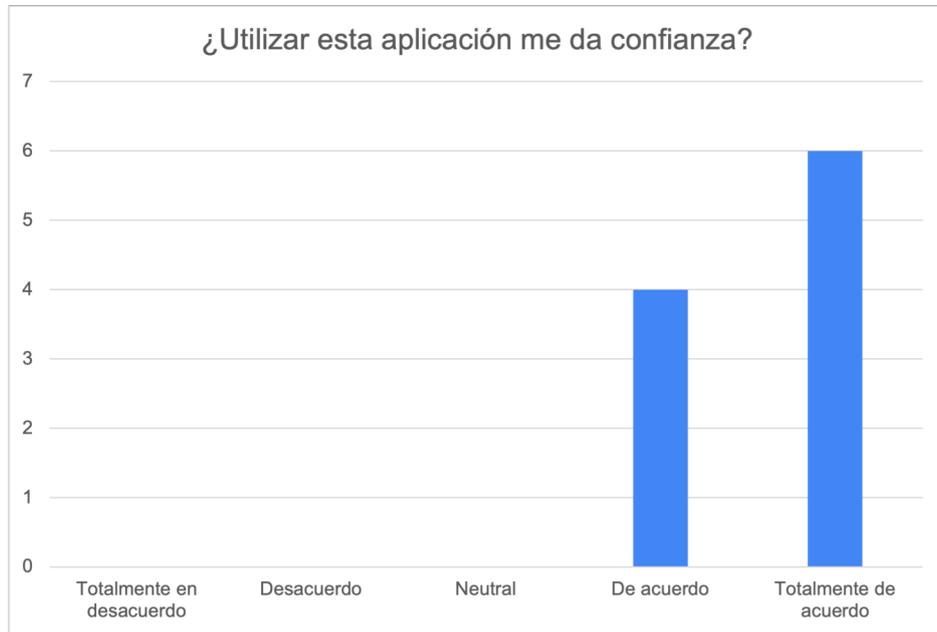
En la Figura 3.7 se observa que la mayoría de los usuarios considera que aprendió a utilizar esta aplicación de forma muy rápida sin embargo existe un pequeño grupo de usuarios que se mantiene neutral en esta pregunta.



**Figura 3.8:** Resultados de la pregunta 8

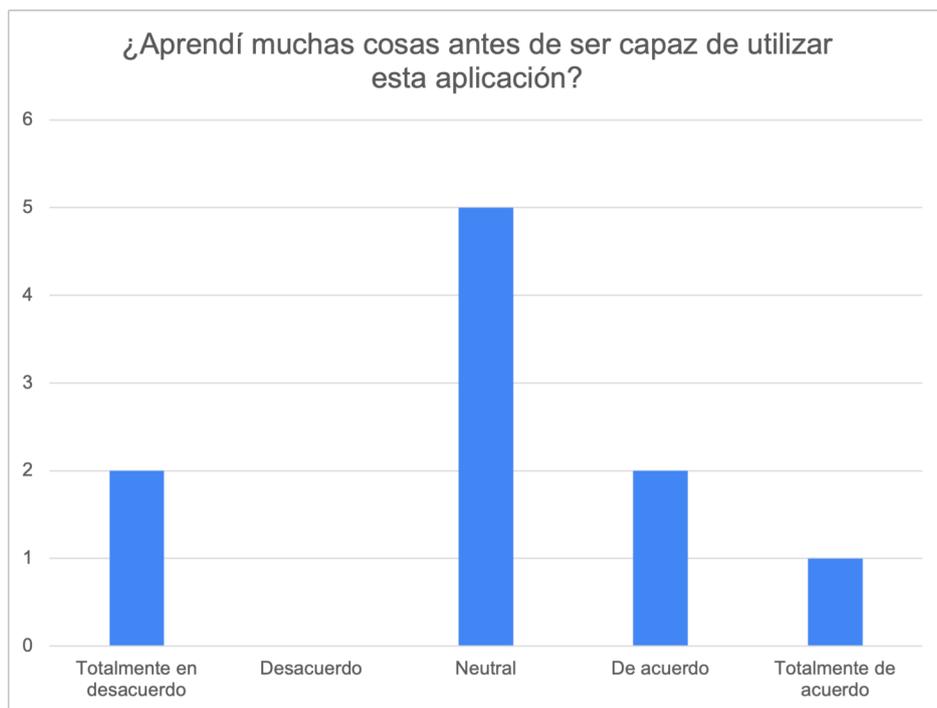
En la Figura 3.8 se observa que las respuestas para esta pregunta se encuentran divididas,

sin embargo un grupo importante de usuarios considera que la aplicación no es difícil de usar mientras que otro grupo importante se mantiene neutral en cuanto a esta pregunta.



**Figura 3.9:** Resultados de la pregunta 9

En la Figura 3.9 se observa que todos los usuarios consideran que la aplicación para realizar pedidos les genera confianza.



**Figura 3.10:** Resultados de la pregunta 10

En la Figura 3.10 se observa que la mayoría de usuarios piensa que necesitó aprender muchas cosas antes de utilizar la aplicación, mientras que un grupo pequeño consideró que no aprendió nada nuevo.

A continuación se muestra, en la Figura 3.11, un consolidado de los resultados por pregunta.

P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
4	2	3	2	4	2	3	3	4	3
3	3	3	4	4	5	3	3	4	3
5	2	5	2	4	3	5	3	5	5
5	1	5	2	5	1	5	1	5	3
5	1	4	3	4	1	5	1	5	3
5	1	5	1	4	1	4	1	4	4
5	3	4	2	5	1	4	4	5	3
5	3	4	2	5	2	5	1	5	1
5	2	4	3	4	1	4	2	4	4
5	1	4	1	5	1	5	5	5	1

**Figura 3.11:** Resultados generales

Luego de realizar los cálculos pertinentes para las encuestas SUS, el resultado fue 77/100, teniendo en cuenta que la media de usabilidad es 68 podemos concluir que la aplicación necesita mejoras menores pero que la usabilidad en general es buena. Como se menciona anteriormente, hubo algunos usuarios que necesitaron de apoyo para poder utilizar la aplicación y varios usuarios consideraron que es difícil de usar. Estos aspectos de la aplicación podrían mejorarse con una presentación más dinámica del uso de la misma, como por ejemplo, hacer uso de vídeos explicativos para que de esta manera tengan un idea clara de como funciona y que pueden esperar de la aplicación. Otro aspecto que podría mejorarse es el diseño de las pantallas, pues hubo usuarios que consideraron que la aplicación tiene poca relación entre sus elementos; una distribución más atractiva de los elementos que conforman la aplicación puede ser una buena alternativa para mejorar este punto.

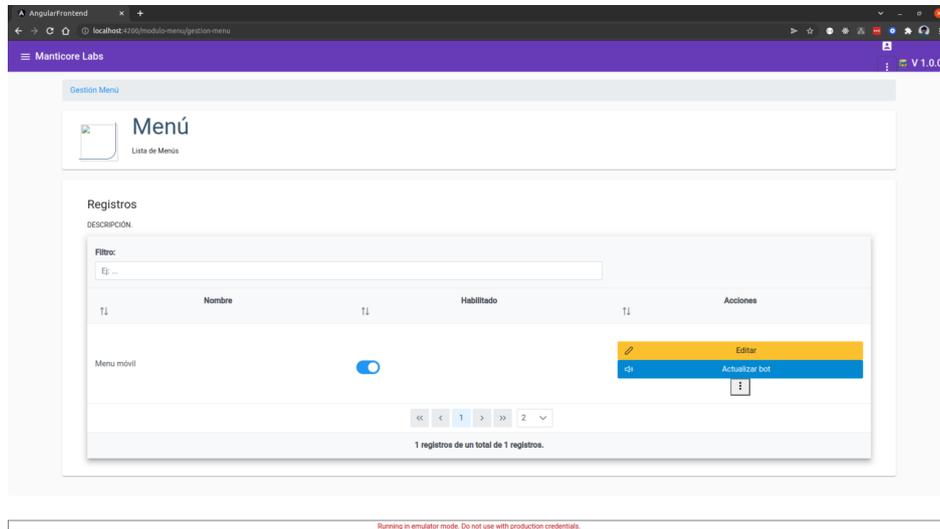
## 3.2 PRODUCTO FINAL

El sistema consta de nueve módulos principales además de un módulo en backend para utilizar el bot e interactuar con el frontend del cliente final. A continuación se detalla cada una de estas partes:

### 1. Módulo Menú

En la Figura 3.12 se muestra el módulo menú. Este módulo presenta la lista de menús que siempre será de máximo un menú, aquí se permite sincronizar todas las catego-

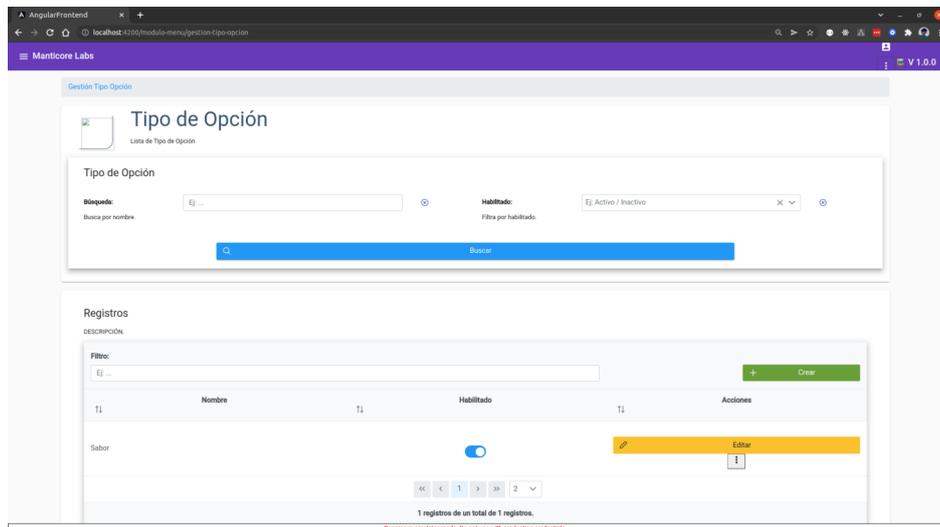
rías, subcategorías y productos con Dialogflow para que el asistente de voz reconozca estos nombres. Además, permite navegar hacia la pantalla de menú detalles.



**Figura 3.12:** Pantalla del módulo menú

## 2. Módulo Tipo opción

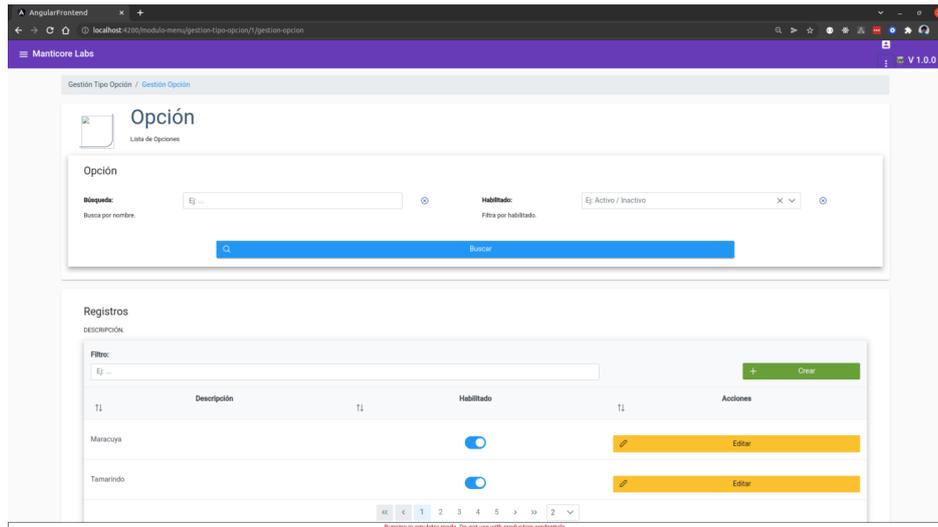
En la Figura 3.13 se muestra el módulo tipo opción. Este módulo permite gestionar los tipos de opciones que puedan existir como por Ej. Sabores de las salsas.



**Figura 3.13:** Pantalla del módulo tipo opción

## 3. Módulo Opción

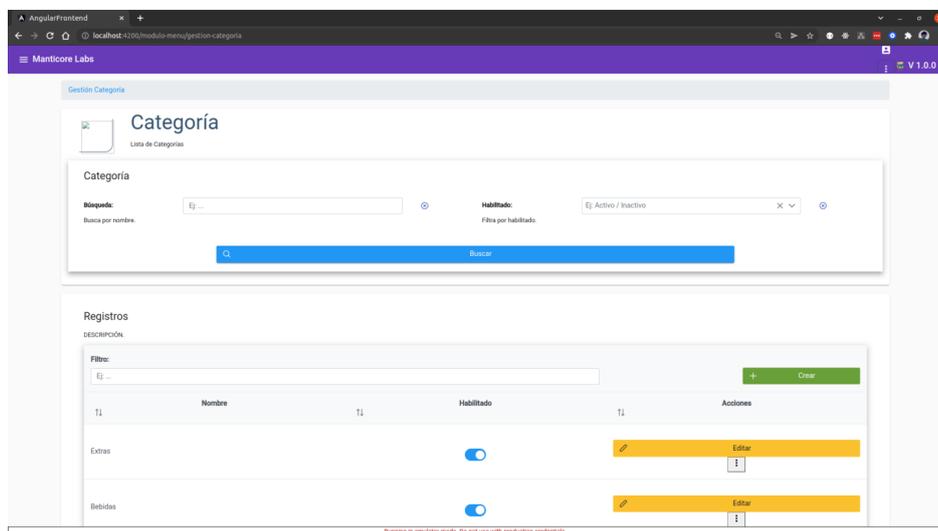
En la Figura 3.14 se muestra el módulo opción. Este módulo permite registrar todas las opciones que pueden ser parte de un tipo de opción por Ej. Mora o Tamarindo.



**Figura 3.14:** Pantalla del módulo opción

#### 4. Módulo Categoría

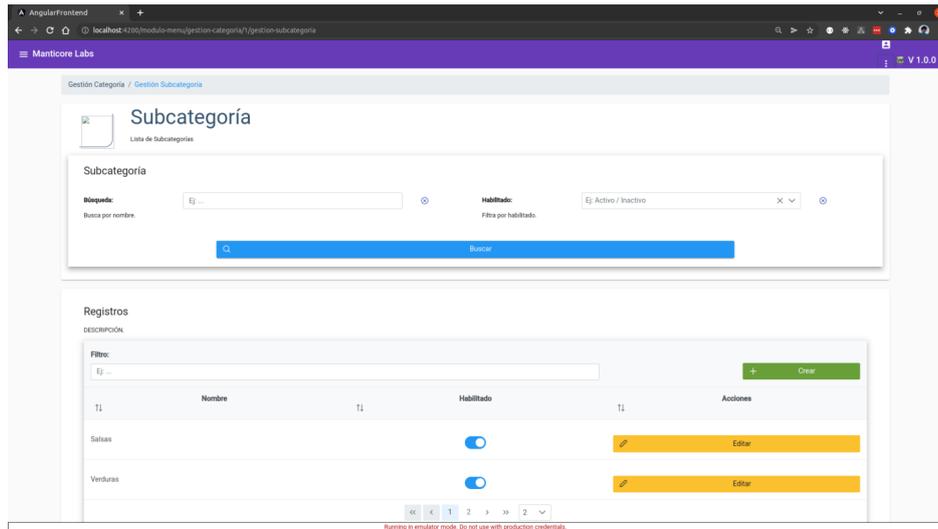
En la Figura 3.15 se muestra el módulo categoría. Este módulo permite registrar todas las categorías que son parte del menú por Ej. Bebidas o promociones



**Figura 3.15:** Pantalla del módulo categoría

#### 5. Módulo Subcategoría

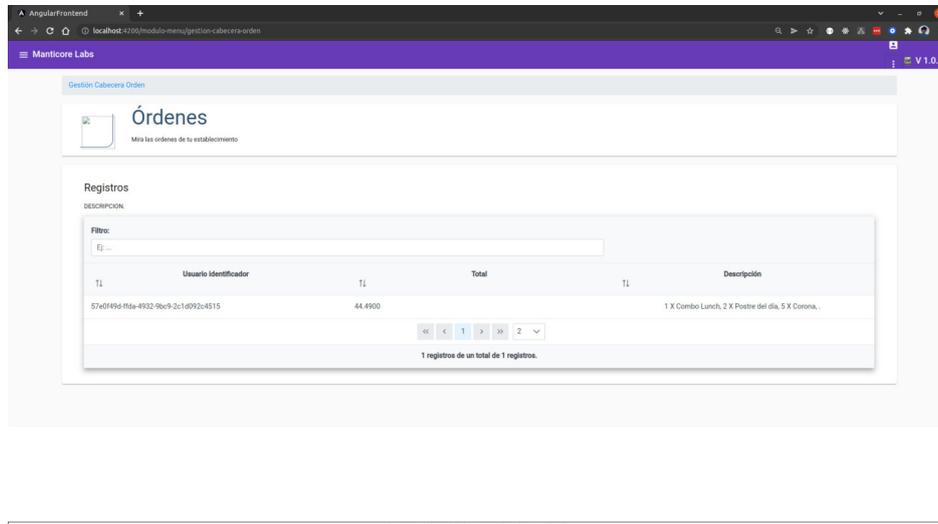
En la Figura 3.16 se muestra el módulo subcategoría. Este módulo permite registrar todas las subcategorías que son parte de una subcategoría por Ej. Jugos o gaseosas.



**Figura 3.16:** Pantalla del módulo subcategoría

## 6. Módulo Órdenes

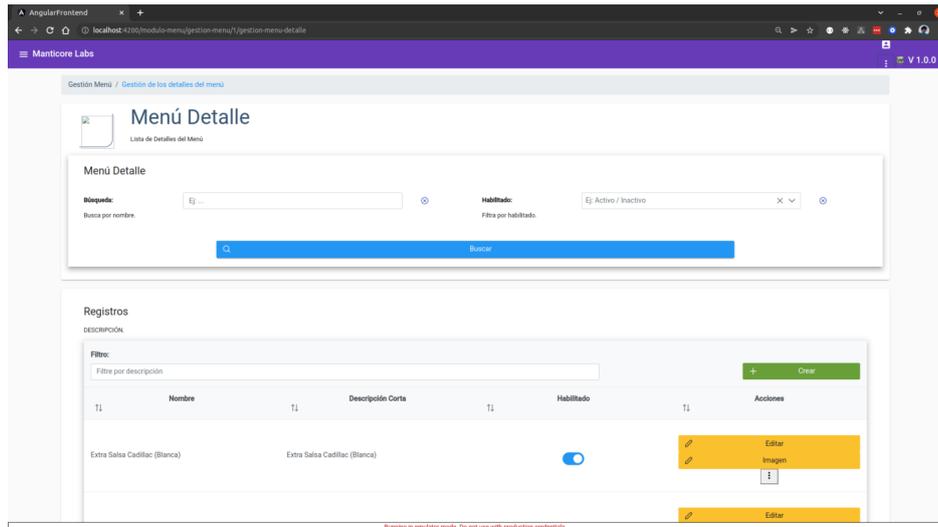
En la Figura 3.17 se muestra el módulo órdenes. Este módulo permite registrar todas las subcategorías que son parte de una subcategoría por Ej. Jugos o gaseosas.



**Figura 3.17:** Pantalla del módulo órdenes

## 7. Módulo Menú detalle

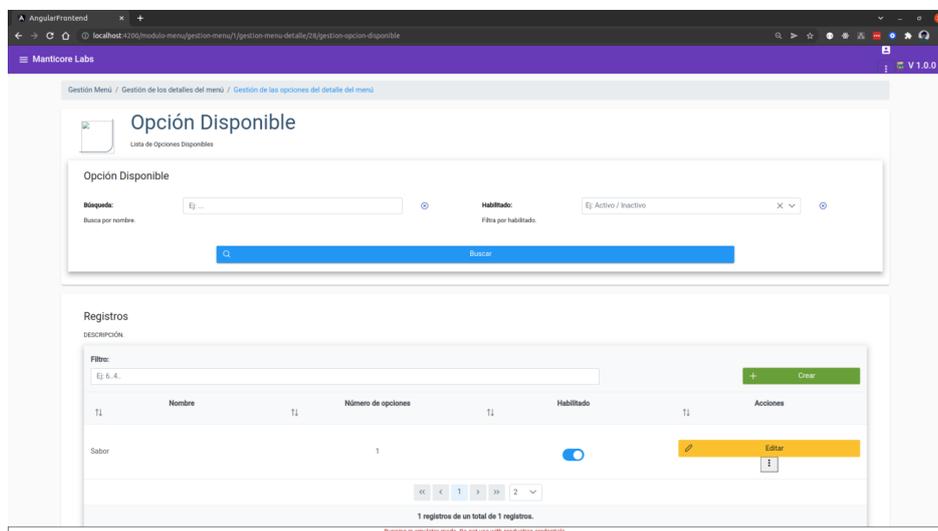
En la Figura 3.18 se muestra el módulo menú detalle. Este módulo permite registrar todos los items de un menú que se quiere visualizar al momento de realizar el pedido. En esta pantalla también se pueden cambiar las imágenes de los productos y también navega hacia las opciones disponibles para ese producto.



**Figura 3.18:** Pantalla del módulo menú detalle

## 8. Módulo Opciones disponibles

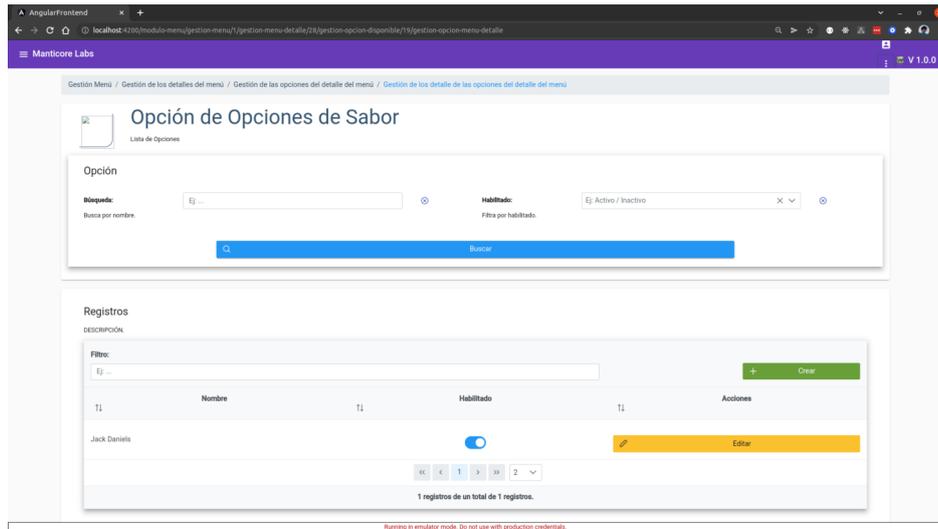
En la Figura 3.19 se muestra el módulo opciones disponibles. Este módulo sirve para ligar un grupo opciones a un producto en específico. Aquí se puede ligar el tipo de opción y la cantidad de ese producto que el cliente final puede elegir.



**Figura 3.19:** Pantalla del módulo opción disponible

## 9. Módulo Opción de una opción disponible

En la Figura 3.20 se muestra el módulo opción de una opción disponible. Este módulo sirve para indicar cuales opciones serán elegibles para ese producto con el fin de segmentar las opciones del universo de opciones que existen en el tipo de opción.



**Figura 3.20:** Pantalla del módulo opción de una opción disponible

## 10. Frontend cliente final

El cliente final podrá ver el menú en una pantalla de apoyo para poder realizar sus pedidos, también podrá ver una pantalla de carrito, así como del detalle de un menú. En la Figura 3.21 se puede ver la pantalla de inicio de la aplicación del cliente final. Al interactuar con el asistente de voz estas pantallas pueden ir interactuando con el cliente para que la conversación con el asistente virtual sea reforzada gráficamente.



**Figura 3.21:** Pantalla de la aplicación del cliente final

## 4 CONCLUSIONES

### 4.1 CONCLUSIONES Y RECOMENDACIONES

- ❑ Se logró desarrollar la aplicación para el asistente virtual el cual tiene conexión con una aplicación web que sirve como apoyo visual para la toma de pedidos.
- ❑ Se lograron realizar las pruebas de usabilidad con la metodología SUS obteniendo que los usuarios que participaron percibieron la aplicación como útil aunque aún existen aspectos por mejorar. En estas pruebas se observó que las personas fueron mostrando mayor confianza mientras más tiempo utilizaban la aplicación, a pesar de que decían sentirse inseguros cuando se les presentaba la aplicación por primera vez.
- ❑ Se logró desarrollar una aplicación en Angular que sirve como aplicación de gestión de categorías, subcategorías, menús opciones y detalles de un menú con el fin de hacer dinámico todo el proceso de productos que se pueden mostrar al cliente. Utilizar el framework de Angular fue de gran utilidad debido a que se puede controlar un estado global de la aplicación mediante la capa de servicios, esto fue de mucha ayuda al momento de realizar la escucha de los eventos emitidos por los usuarios mediante WebSockets.
- ❑ Se diseñó de manera adecuada la solución de arquitectura tecnológica para la realización del proyecto, el cual cuenta con un backend, dos aplicaciones clientes programadas en Angular y una aplicación de terceros (Dialogflow) que se comunica mediante un webhook. Existió gran dificultad al utilizar el API REST de Dialogflow debido a que existe muy poca documentación en la pagina oficial de Dialogflow.
- ❑ La librería de generación de código de Manticore Labs fue una pieza fundamental para la realización del proyecto porque optimizó el tiempo en el que se genera un CRUD para la gestión de los módulos.

- ❑ Al realizar la búsqueda de información sobre las diferentes presentaciones de menús se encontró bastante documentación sobre menús físicos y menús que se presentan en aplicaciones móviles, sin embargo, no se encontró publicaciones que presenten información relevante sobre los menús que se presentan en chatbots y menos aún, información sobre menús presentados en aplicaciones de voz.
- ❑ La experiencia previa de trabajo con la metodología Scrum permitió mantener sprints con carga de trabajo balanceada, a pesar de que el último sprint tuvo pocas actividades, el proyecto no tuvo retrasos.
- ❑ Para evitar que todos los dispositivos que estén escuchando al backend mediante websockets respondan a la vez cuando se realiza una interacción con la aplicación de voz se utilizaron namespaces. No se utilizaron rooms debido a una incompatibilidad con las versiones de la librería Socket.IO y la versión de NestJS.

## 4.2 RECOMENDACIONES

- ❑ Se recomienda que la aplicación del asistente de voz sea entrenada en ambientes de producción con el fin de admitir las frases más utilizadas por los clientes.
- ❑ Debido a la complejidad de la documentación para utilizar el SDK de Dialogflow. Se recomienda utilizar los servicios REST para poner en uso todas las capacidades de conexión con Dialogflow y Google Cloud.
- ❑ Dialogflow tiene una librería para Nodejs que cubren la interacción básica entre el usuario con los *intents* de la aplicación de voz con la cual se programaron las respuestas en el backend. Sin embargo, para interacciones más avanzadas se recomienda utilizar los servicios REST de Dialogflow, lo cual resulta difícil debido a la poca documentación que existe de los mismos.
- ❑ Para configurar los dispositivos que se conectan con el asistente de Google. Se recomienda iniciar sesión en cada uno de ellos con diferentes cuentas de Google para que puedan guardar información dentro de los dispositivos.
- ❑ Debido a la naturaleza del asistente de voz, se recomienda que las personas que utilicen la aplicación para la toma de pedidos, sean quienes tienen más conocimientos del menú del establecimiento.

## 5 REFERENCIAS BIBLIOGRÁFICAS

- [1] P. Lessel, M. Böhmer, A. Kröner y A. Krüger, «User requirements and design guidelines for digital restaurant menus,» en *Proceedings of the 7th Nordic Conference on Human-Computer Interaction Making Sense Through Design - NordiCHI '12*, Copenhagen, Denmark: ACM Press, 2012, pág. 524, ISBN: 9781450314824. DOI: 10.1145/2399016.2399096. dirección: <http://dl.acm.org/citation.cfm?doid=2399016.2399096> (visitado 27-12-2021).
- [2] Y. Wang, «Designing Restaurant Digital Menus to Enhance User Experience,» Master of Fine Arts, Iowa State University, Digital Repository, Ames, 2012. DOI: 10.31274/etd-180810-2790. dirección: <https://lib.dr.iastate.edu/etd/12781/> (visitado 27-12-2021).
- [3] «Digital food delivery apps revolutionizing food products marketing in india,» *International Journal of Recent Technology and Engineering*, vol. 8, n.º 2, págs. 662-665, 16 de sep. de 2019, ISSN: 2277-3878. DOI: 10.35940/ijrte.B1126.0782S619. dirección: <https://www.ijrte.org/wp-content/uploads/papers/v8i2S6/B11260782S619.pdf> (visitado 27-12-2021).
- [4] M. Y.-C. Yim y C. Y. Yoo, «Are digital menus really better than traditional menus? the mediating role of consumption visions and menu enjoyment,» *Journal of Interactive Marketing*, vol. 50, págs. 65-80, mayo de 2020, ISSN: 10949968. DOI: 10.1016/j.intmar.2020.01.001. dirección: <https://linkinghub.elsevier.com/retrieve/pii/S1094996820300013> (visitado 27-12-2021).
- [5] (). «¿Qué es una aplicación web? GUÍA [2022],» dirección: <https://www.crehana.com/ec/blog/desarrollo-web/aplicacion-web-que-es/> (visitado 10-01-2022).
- [6] (). «¿Qué son las aplicaciones web y las páginas web dinámicas?» Dirección: <https://helpx.adobe.com/es/dreamweaver/using/web-applications.html> (visitado 10-01-2022).

- [7] SAS. (). «Qué es el Procesamiento de Lenguaje Natural - Natural Language Processing?» Dirección: [https://www.sas.com/es\\_ar/insights/analytics/what-is-natural-language-processing-nlp.html](https://www.sas.com/es_ar/insights/analytics/what-is-natural-language-processing-nlp.html) (visitado 26-11-2020).
- [8] Mozilla. (). «WebSockets,» Documentación web de MDN, dirección: [https://developer.mozilla.org/es/docs/Web/API/WebSockets\\_API](https://developer.mozilla.org/es/docs/Web/API/WebSockets_API) (visitado 26-11-2020).
- [9] ProgramadorClick, *HTTP y websocket*, [Online; accesado Marzo 27, 2022]. dirección: <https://images2.programmerclick.com/301/16/16bea2d864c0d9ea330a484ecf23bf1d.png>.
- [10] Hubtype. (). «Mejora la experiencia de uso con interfaces conversacionales,» dirección: <https://www.hubtype.com/es/blog/mejora-experiencia-usuario-interfaces-conversacionales/> (visitado 30-11-2020).
- [11] Evaluandocrm.com, *Interfaces conversacionales*, [Online; accesado Marzo 27, 2022], 2018. dirección: <https://www.evaluandocrm.com/wp-content/uploads/2018/07/286-CRM-Interfaces-conversacionales-04.png>.
- [12] Microsoft. (). «Documentation for visual studio code,» Getting Started, dirección: <https://code.visualstudio.com/docs> (visitado 19-11-2020).
- [13] Google. (). «Documentación de Dialogflow,» Google Cloud, dirección: <https://cloud.google.com/dialogflow/docs?hl=es-419> (visitado 19-11-2020).
- [14] Gitlab. (). «GitLab.org / GitLab,» GitLab, dirección: <https://gitlab.com/gitlab-org/gitlab> (visitado 19-11-2020).
- [15] NestJS. (). «Documentation | NestJS - a progressive node.js framework,» Documentation | NestJS - A progressive Node.js framework, dirección: <https://docs.nestjs.com> (visitado 19-11-2020).
- [16] Google. (). «Angular - Introduction to the Angular Docs,» dirección: <https://angular.io/docs> (visitado 19-11-2020).
- [17] Oracle. (). «MySQL :: Getting Started with MySQL,» dirección: <https://dev.mysql.com/doc/mysql-getting-started/en/> (visitado 19-11-2020).
- [18] D. Arrachequesne. (12 de nov. de 2020). «Introduction,» Socket.IO, dirección: <https://socket.io/docs/v3/index.html> (visitado 19-11-2020).
- [19] (). «Contenedores de Docker | ¿Qué es Docker? | AWS,» Amazon Web Services, Inc. Dirección: <https://aws.amazon.com/es/docker/> (visitado 08-01-2022).

- [20] Atlassian. (). «Scrum: qué es, cómo funciona y por qué es excelente,» Atlassian, dirección: <https://www.atlassian.com/es/agile/scrum> (visitado 26-11-2020).
- [21] IEBS, *Cómo funciona la Metodología Scrum: Qué es y cómo utilizarla*, [Online; accesado Marzo 27, 2022], 2001. dirección: <https://www.iebschool.com/blog/wp-content/uploads/2021/04/como-funciona-scrum.png>.
- [22] *A guide to the Scrum Body of knowledge (SBOK Guide)*, 2013 edition. Phoenix, Arizona: SCRUMstudy, A brand of VMEdU, Inc, 2013, ISBN: 9780989925204.
- [23] M. Mahalakshmi y D. M. Sundararajan, «International Journal of Emerging Technology and Advanced Engineering,» English, *Traditional SDLC Vs Scrum Methodology – A Comparative Study*, vol. 3, n.º 6, págs. 192-196, jun. de 2013, ISSN: 2250-2459. dirección: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.2992&rep=rep1&type=pdf>.
- [24] K. S. Sutherland. (nov. de 2020). «La Guía de Scrum,» Scrum Guides, dirección: <https://www.scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf> (visitado 29-11-2020).
- [25] M. A. Awad, «A Comparison between Agile and Traditional Software Development Methodologies,» 2005.
- [26] I. Sommerville y M. I. Alfonso Galipienso, *Ingeniería del software*. Madrid: Pearson Educación, 2005, OCLC: 971756510, ISBN: 9788478290741.
- [27] E.-Y. Lee, S.-B. Lee e Y. J. J. Jeon, «Factors influencing the behavioral intention to use food delivery apps,» *Social Behavior and Personality: an international journal*, vol. 45, n.º 9, págs. 1461-1473, 3 de oct. de 2017, ISSN: 0301-2212. DOI: 10.2224/sbp.6185. dirección: <https://www.ingentaconnect.com/content/10.2224/sbp.6185> (visitado 27-12-2021).
- [28] X. Y. Leung y H. Wen, «Chatbot usage in restaurant takeout orders: A comparison study of three ordering methods,» *Journal of Hospitality and Tourism Management*, vol. 45, págs. 377-386, dic. de 2020, ISSN: 14476770. DOI: 10.1016/j.jhtm.2020.09.004. dirección: <https://linkinghub.elsevier.com/retrieve/pii/S1447677020302102> (visitado 27-12-2021).
- [29] (). «Conceptos básicos de Dialogflow CX,» Google Cloud, dirección: <https://cloud.google.com/dialogflow/cx/docs/basics?hl=es-419> (visitado 19-03-2022).

[30] (). «The System Usability Scale & How it's Used in UX | Adobe XD Ideas,» Ideas, dirección: <https://xd.adobe.com/ideas/process/user-testing/sus-system-usability-scale-ux/> (visitado 02-03-2022).

## **6 ANEXOS**

### **6.1 DISEÑO DE LA APLICACIÓN DEL CLIENTE FINAL**

<https://drive.google.com/file/d/1zg0D-uVK-lqgYCFH4qxizRSmOWnRCqEC/view?usp=sharing>

### **6.2 ESQUEMA DE FLUJOS CONVERSACIONALES**

<https://drive.google.com/file/d/1D8ZE9uc0mJUmdU-TLA8NnlwqRgQ2svVb/view?usp=sharing>

### **6.3 HISTORIAS DE USUARIO DEL PROYECTO**

<https://docs.google.com/document/d/1OhDnnlhLIWNbHKeZtcL9rQAtpgfj0cTe6QgSAkkMmQo/edit?usp=sharing>

### **6.4 DIAGRAMA DE BASE DE DATOS**

<https://drive.google.com/file/d/1xGs3isxm6W8gMiCGeFGe6nH-5KNGhKt0/view?usp=sharing>

### **6.5 SITEMAP DE LA APLICACIÓN DE GESTIÓN**

<https://drive.google.com/file/d/1XPQ17nOwPW02XVPLtM-ntBKrQxr4-BmJ/view?usp=sharing>

## **6.6 ENCUESTAS DE USABILIDAD DEL SISTEMA**

<https://docs.google.com/spreadsheets/d/16Vr31C4mGj5hIWbQTefzXDarwAk8vdia/edit?usp=sharing&oid=113778906571444530782&rtpof=true&sd=true>

## **6.7 IMÁGENES EXTRAS DEL DESARROLLO DE LOS SPRINTS**

[https://docs.google.com/document/d/1RpYv\\_p4dW1Ob-qLAguUwWd1CkJV0UKIx/edit?usp=sharing&oid=113778906571444530782&rtpof=true&sd=true](https://docs.google.com/document/d/1RpYv_p4dW1Ob-qLAguUwWd1CkJV0UKIx/edit?usp=sharing&oid=113778906571444530782&rtpof=true&sd=true)