

# Universidad Siglo 21



**Trabajo final de grado. Prototipado tecnológico**

**Carrera : Licenciatura en Informática**

Sistema de Digitalización de los Procesos de Atención en  
Restaurantes

**Autor:** Daniel Luciano Cassetai

**Legajo:** VINF09367

Misiones, Noviembre del 2022

## Índice

<b>Resumen</b>	<b>5</b>
<b>Abstract</b>	<b>6</b>
<b>Título</b>	<b>7</b>
<b>Introducción</b>	<b>7</b>
Antecedentes	7
Descripción del Área Problemática	8
<b>Justificación</b>	<b>8</b>
<b>Objetivo General del Proyecto</b>	<b>9</b>
<b>Objetivos Específicos del proyecto</b>	<b>9</b>
<b>Marco Teórico Referencial</b>	<b>9</b>
Dominio del Problema	9
TIC (Tecnología de la Información y la Comunicación)	10
Competencia	14
<b>Diseño Metodológico</b>	<b>15</b>
Herramientas Metodológicas	15
Herramientas de Desarrollo	15
Recolección de Datos	16
Planificación del Proyecto	17
<b>Relevamiento</b>	<b>18</b>
Relevamiento Estructural	18
Relevamiento Funcional	18
Documentación	21
<b>Procesos de Negocio</b>	<b>22</b>
<b>Diagnóstico y Propuesta</b>	<b>23</b>
Diagnóstico	23
Propuesta	24
<b>Objetivos, Límites y Alcances del Prototipo</b>	<b>25</b>
Objetivos del Prototipo	25

	2
Límites	25
Alcances	25
<b>Descripción del Sistema</b>	<b>25</b>
Product backlog	25
Historias de Usuario	26
Sprint Backlog	31
Estructura de Datos	32
Diagrama de Clases	32
Diagrama de Entidad Relación	33
Prototipos de Interfaces de Pantallas	34
Diagrama de Arquitectura	39
<b>Seguridad</b>	<b>39</b>
Acceso a la Aplicación	39
Política de Respaldo de Información	41
<b>Análisis de Costo</b>	<b>41</b>
<b>Análisis de Riesgos</b>	<b>43</b>
Plan de Contingencia	45
<b>Conclusiones</b>	<b>46</b>
<b>Demo</b>	<b>47</b>
<b>Referencias</b>	<b>48</b>
<b>Anexos</b>	<b>50</b>
Anexo 1	50
Anexo 2	53
Anexo 3	54
Anexo 4	55

## Índice de Figuras

Figura 1: Diagrama de Gantt	17
Figura 2 : Organigrama de BEERLIN RESTO - BAR	18
Figura 3: Diagrama de Modelo y Notación de Procesos de Negocio.	22
Figura 4 : Diagrama de clases	32
Figura 5 : Diagrama entidad relación.	33
Figura 6: Circuito inicial para registro e ingreso	34
Figura 7: Circuito para realizar un pedido.	35
Figura 8: Circuito para ver los pedidos realizados y solicitar la cuenta.	36
Figura 9 : Circuito para solicitar atención del mozo	37
Figura 10 : Circuito para buscar un restaurante y reservar una mesa	38
Figura 11 : Diagrama de arquitectura.	39

## Índice de Tablas

Tabla N 1 : Comparativa de empresas	15
Tabla 2: Problemas detectados del proceso de carga de precios relevado de Berlin Resto - Bar	24
Tabla 3: Problemas detectados del proceso de toma y carga de pedidos relevado de Berlin Resto - Bar	24
Tabla 4: Problemas detectados del proceso de cierre de mesa relevado de Berlin Resto - Bar	25
Tabla 5: Product backlog.	26
Tabla 6: Historia de usuario HU-001 registro de cliente.	27
Tabla 7: Historia de usuario HU-002 ver la carta	27
Tabla 8: Historia de usuario HU-003 habilitar mesa.	28
Tabla 9: Historia de usuario HU-004 realizar pedido.	28
Tabla 10: Historia de usuario HU-005 solicitar cuenta.	29
Tabla 11: Historia de usuario HU-006 ver detalle de pedidos.	29

Tabla 12: Historia de usuario HU-007 búsqueda de restaurantes.	30
Tabla 13: Historia de usuario HU-008 reserva de mesas.	30
Tabla 14: Historia de usuario HU-009 Consultar estado del pedido.	31
Tabla 15: Historia de usuario HU-010 Solicitar atención del mozo.	31
Tabla 16: Primer sprint de backlog.	32
Tabla 17 : Análisis de costos de recursos humanos	43
Tabla 18: Análisis de gastos operativos	43
Tabla 19: Total de costos	43
Tabla 20: Riesgos y sus causas	44
Tabla 21: Matriz de riesgos	44
Tabla 22: Análisis cuantitativo y grado de exposición	45
Tabla 23: Riesgos de atención inmediata con su plan de contingencia.	46

## Resumen

A pesar de los avances tecnológicos la gran mayoría de los establecimientos gastronómicos, de la provincia de Misiones, continúan utilizando el mismo formato tradicional de siempre en lo que respecta a la atención de los clientes. Los mozos, a pesar de ser el eje de la atención en los restaurantes también son el cuello de botella que puede marcar la diferencia entre una buena o una mala experiencia.

En el presente trabajo encontraremos un proyecto destinado a desarrollar una solución a los distintos problemas que surgen de la interacción entre los restaurantes y los clientes en cuanto a las demoras en la atención y la gestión del menú que se vienen prolongando hace mucho tiempo.

Es por ello que el objetivo de este proyecto fue crear una aplicación web que permitio digitalizar la atención de los restaurantes haciendo uso de las nuevas tecnologías con el fin disminuir tiempos de espera para los clientes y optimizar la calidad del servicio brindado. Esto se logró mediante el desarrollo de una herramienta que posibilitó a los clientes la interacción directa con los servicios ofrecidos por los restaurantes disminuyendo así la necesidad de intermediarios en los procesos.

**Palabras claves** : restaurantes , atención digital , auto pedidos, reservas online, menú online.

## **Abstract**

Despite technological advances, the vast majority of gastronomic establishments in the province of Misiones continue to use the same traditional format as always when it comes to customer service. The waiters, despite being the backbone of the attention in restaurants, are also the bottleneck that can make the difference between a good or a bad experience.

In this work we will find a project aimed at developing a solution to the various problems that arise from the interaction between restaurants and customers in terms of delays in service and menu management that have been going on for a long time.

That is why the objective of this project was to create a web application that allowed the restaurant to digitalize the attention making use of new technologies in order to reduce waiting times for customers and optimize the quality of service provided. This was achieved through the development of a tool that enabled customers to interact directly with the services offered by the restaurants, reducing the need for intermediaries in the processes.

**Keyword:** restaurants, digital customer service, self-ordering, online reservations, online menu.

## **Título**

Sistema de Digitalización de los Procesos de Atención en Restaurantes.

## **Introducción**

El presente proyecto fue destinado a elaborar una solución general para optimizar la interacción presencial de las personas con los distintos restaurantes de la ciudad de Posadas provincia de Misiones.

Cada vez son más los rubros que gozan de los beneficios de la transformación digital y los restaurantes no deberían de ser la excepción. Actualmente, en la provincia de Misiones, son pocos los establecimientos que han dado el salto a esta transformación manteniendo aun el mismo formato tradicional de siempre.

Hoy en día la tecnología nos abre un abanico de oportunidades para poder mejorar la calidad de los servicios, aumentar las ventas y generar mejores experiencias a los clientes.

### *Antecedentes*

En la actualidad podemos encontrar varios sistemas que se utilizan para administrar restaurantes. En su gran mayoría están enfocados en ofrecer soluciones internas a los distintos establecimientos como pueden ser el cierre de caja, control de asistencias, gestión de pedidos que toman los mozos, entre otros.

Otros tipos de sistemas que utilizan los restaurantes son los de delivery mediante el uso de apps como por ejemplo PedidosYa. Este tipo de aplicaciones permite realizar pedidos fuera del restaurante para que posteriormente sean entregados al domicilio del cliente.

Así mismo podemos encontrar en algunos establecimientos las cartas publicadas de manera digital escaneando códigos QR. Esta opción es una de las más populares pero aún no tuvo mucha adopción por parte de los clientes.

### *Descripción del Área Problemática*

Para determinar el área problemática vamos a establecer dos perspectivas distintas una desde la mirada de los clientes y otra desde los establecimientos gastronómicos.

En primer lugar nos situaremos desde la mirada de los clientes donde parte del problema es que en la actualidad deben esperar constantemente la atención de los mozos para tener que interactuar con los restaurantes ya sea para ordenar algo, solicitar la carta, hacer una reserva o pagar lo consumido. Así mismo, al no contar con información online disponible de los productos y servicios que ofrecen los restaurantes, los clientes no poseen la información necesaria acerca de los platos que son preparados, si estos son aptos para personas con alguna condición alimentaria, si el establecimiento está preparado para recibir a personas con algún tipo de discapacidad o familias con bebés, entre otros.

Por otra parte, posicionándonos en la mirada de los restaurantes, uno de los problemas centrales que se pudo identificar es la necesidad de la constante reimpresión o actualización de las cartas, ya sean por la variación de los precios, la quita o incorporación de productos al menú o porque no se cuenta con algún insumo disponible para la preparación algún platillo o bebida. Otro punto importante es el trabajo de los mozos, el cual se vuelve más arduo al tener que recordar variados pedidos de diferentes mesas, el orden de los mismos o las particularidades de estos.

Estos problemas en su mayoría generan disgustos en los clientes y la disminución del interés en estos establecimientos, lo cual se transforma en pérdidas de prestigio e incluso monetarias para los restaurantes.

### **Justificación**

El desarrollo de este sistema contribuyó a mejorar la interacción entre los restaurantes y los clientes, lo cual permitió optimizar los tiempos de atención y aumentar la productividad de los mozos.

Al contar con una carta digital administrada de manera centralizada, los restaurantes lograron evitar la reimpresión de manera constante del menú en cada variación de precios en los insumos. Así mismo los establecimientos pudieron dar a conocer las características generales del establecimiento, los servicios y productos que estos ofrecen.

Gracias al módulo de pedidos desde la mesa, la atención de los clientes se volvió más ágil y rápida permitiendo a los clientes evitar los tiempos de espera. También se logró disminuir la carga sobre los mozos al tener que acudir únicamente a entregar los pedidos, resolver algún inconveniente o alguna necesidad específica.

### **Objetivo General del Proyecto**

Implementar un sistema que permita digitalizar la atención de los restaurantes mediante una aplicación web, a fin de optimizar y reducir el tiempo de atención.

### **Objetivos Específicos del proyecto**

- Recopilar información sobre los procesos de atención al cliente en diferentes restaurantes de la ciudad de Posadas Misiones.
- Analizar diseños y formatos de cartas de los restaurantes.
- Diseñar y desarrollar una aplicación web para favorecer la interacción entre los restaurantes y los clientes.
- Incrementar la dinámica de atención al cliente.

### **Marco Teórico Referencial**

#### *Dominio del Problema*

Para comprender mejor el marco donde se realizó este proyecto, fue importante tener en cuenta algunas definiciones que nos ayudaron entender mejor el problema y evitar ambigüedades.

El concepto central de nuestro proyecto es el de restaurante, el cual “es un establecimiento público donde se sirven comidas y bebidas, mediante precio, para ser consumidas en el mismo local”. (Real Academia Española [RAE], s.f.). Es importante tener en cuenta que un restaurante puede tener distintas sucursales. Cuando hablamos de sucursal hacemos referencia a un establecimiento “situado en distinto lugar que la central de la cual depende, desempeña las mismas funciones que esta”. (RAE, s.f.)

En cuanto a la carta o menú podemos decir que en un restaurante es la “lista de platos y bebidas que se pueden elegir”. (RAE, s.f.).

El camarero o también llamado mozo en Argentina es “la persona que tiene por oficio servir consumiciones en restaurantes, bares u otros establecimientos similares.” (RAE, s.f.).

### *TIC (Tecnología de la Información y la Comunicación)*

Para la consecución del proyecto fue relevante tener en cuenta las distintas tecnologías disponibles. Entre los tipos de aplicación más utilizados podríamos dividirlos en dos categorías :

#### **Aplicación web:**

En la Ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un Servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación (Software) que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador. (Ecured, s.f.).

#### **Aplicación Móvil**

Son aplicaciones desarrolladas para teléfonos inteligentes que contienen sistemas operativos, por ejemplo Android, que permite la ejecución de

aplicaciones de diversas categorías y en su mayoría gratuitas.  
(Appdesignbook, 2013)

### **API** (*application programmer interface*)

Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. Suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor (la llamada) y el que requiere el productor (la respuesta). (redhat, 8 de Mayo 2020)

### **Lenguaje de Programación**

Para el desarrollo de aplicaciones web hoy en día contamos con varios tipos de tecnologías que nos posibilitan llegar a una solución.

JavaScript (a menudo abreviado como JS) es un lenguaje ligero, interpretado y orientado a objetos con funciones de primera clase, y mejor conocido como el lenguaje de programación para las páginas Web, pero también se utiliza en muchos entornos que no son de navegador. Es un lenguaje de scripts que es dinámico, multi paradigma, basado en prototipos y admite estilos de programación orientados a objetos, imperativos y funcionales.

JavaScript se ejecuta en el lado del cliente de la web, y se puede utilizar para estilizar/programar cómo se comportan las páginas web cuando ocurre un evento. JavaScript es un potente lenguaje de scripts y fácil de aprender, ampliamente utilizado para controlar el comportamiento de las páginas web. (MDN Web Docs, 2022)

Para el desarrollo del frontend contamos con herramientas como ReactJS el cual es “una biblioteca de JavaScript para construir interfaces de usuario” (Reactjs, s.f.).

En lo que respecta al backend, NodeJS esta comenzado a liderar el mercado gracias a que este permite el uso de Javascript del lado del servidor lo cual facilita la tarea de aprendizaje.

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google (NodeJS, s.f.).

Uno de los frameworks más utilizados en node es ExpressJS, este nos ofrece una “infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles.” (ExpressJS, s.f) además no proporciona miles de métodos para crear apis robustas y de manera sencilla.

Hoy en día las aplicaciones de celulares también pueden ser desarrolladas utilizando Javascript, para esto contamos con herramientas como ReactNative, el cual “es un Framework de JavaScript para escribir aplicaciones móviles reales y nativas para iOS y Android. Se basa en React, la biblioteca de JavaScript de Facebook para crear interfaces de usuario.” (Oreilly, s.f.)

Otra alternativa para el desarrollo de aplicación de celulares son Swift y Kotlin. “Swift es un intuitivo lenguaje de programación creado por Apple que permite diseñar apps para iOS, Mac, el Apple TV y el Apple Watch. Está pensado para dar a los desarrolladores más libertad que nunca.” (Apple, s.f.).

Para el desarrollo de aplicaciones nativas de android podemos destacar a Kotlin como el principal lenguaje utilizado actualmente.

Kotlin es un lenguaje de programación moderno pero ya maduro destinado a hacer más felices a los desarrolladores. Es conciso, seguro,

interoperable con Java y otros lenguajes, y proporciona muchas formas de reutilizar código entre múltiples plataformas para una programación productiva.” (kotlinlang, s.f.)

## **Bases de Datos**

La persistencia de los datos es uno de los puntos más importantes a tener en cuenta, para esto podemos utilizar bases de datos relaciones o no relaciones.

Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas. En una base de datos relacional, cada fila en una tabla es un registro con una ID única, llamada clave. Las columnas de la tabla contienen los atributos de los datos y cada registro suele tener un valor para cada atributo, lo que simplifica la creación de relaciones entre los puntos de datos. (Oracle, s.f.).

Los motores de base de datos relacionales más utilizados en el mercado son:

- PostgreSQL
- MySQL
- SQLServer

Las bases de datos no relaciones o NO SQL “están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala.” (Amazon Web Service [aws], s.f.). Entre las más utilizadas podemos nombrar a MongoDB y Couch DB.

## Competencia

A través de un sondeo realizado en la red, se pudieron encontrar varios sistemas informáticos que ofrecen soluciones similares, los tres más relevantes son :

- **Menuonline**<sup>1</sup>: Sistema online que permite gestionar reservas, el menú y lista de precios de restaurantes.
- **Waitry**<sup>2</sup>: Plataforma de pedidos que ofrece una aplicación de celular a los clientes para que puedan consultar los datos del restaurante y realizar sus pedidos estando en el restaurante, además ofrece la opción de pedir para delivery.
- **Pikotea**<sup>3</sup>: Software para restaurantes y hoteles que permite tomar pedidos de delivery o en el restaurante.

A continuación se presenta un cuadro comparando las distintas soluciones que ofrecen estos sistemas.

Tabla N 1 : Comparativa de empresas

Ofrece	Menuonline	Waitry	Pikotea
Menú online	X	X	X
Pedidos desde la mesa		X	X
Aplicación de celular		X	X
Reservas	X	X	
Menú con código QR	X	X	X
Búsqueda de restaurantes			
Llamar al mozo		X	
Calificar atención			X

Fuente: Elaboración propia.

<sup>1</sup> <https://monline.com.ar/>

<sup>2</sup> <https://waitry.net/>

<sup>3</sup> <https://pikotea.com/>

## **Diseño Metodológico**

### *Herramientas Metodológicas*

Para este proyecto se decidió utilizar scrum como metodología de desarrollo por la flexibilidad que este ofrece.

Scrum es un método de desarrollo ágil de software concebido por Jeff Sutherland y su equipo de desarrollo a principios de la década de 1990.

Los principios Scrum son congruentes con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades estructurales: requerimientos, análisis, diseño, evolución y entrega. Dentro de cada actividad estructural, las tareas del trabajo ocurren con un patrón del proceso llamado sprint. El trabajo realizado dentro de un sprint (el número de éstos que requiere cada actividad estructural variará en función de la complejidad y tamaño del producto) se adapta al problema en cuestión y se define —y con frecuencia se modifica— en tiempo real por parte del equipo Scrum. (Pressman, 2010, p. 69).

### *Herramientas de Desarrollo*

En cuanto a las herramientas de desarrollo que se utilizaron en el proyecto debemos destacar que tanto para los clientes como para los empleados de los restaurantes se optó por crear una aplicación web que se adapte tanto a celulares como a tablets o computadoras ya que :

- No requiere descargas.
- Las actualizaciones son instantáneas
- El proceso de desarrollo es más económico

Como lenguaje de programación principal se utilizó Javascript, debido a que una de las grandes ventajas que se obtiene al utilizarlo es su versatilidad, lo cual nos permite realizar la mayor parte del desarrollo con un mismo lenguaje.

En el desarrollo frontend se utilizó ReactJS por su facilidad para crear comportamiento dinámico y el gran soporte que tiene la comunidad.

El backend fue desarrollado utilizando NodeJS para crear la lógica de la aplicación y ExpressJS para desarrollar la API a la cual deben conectarse los distintos clientes.

Para la persistencia de datos se eligió PostgreSQL como sistema de gestión de bases de datos relacional orientado a objetos. Como ventaja de este frente a las demás opciones mencionadas podemos destacar que es gratuito, de código abierto y tiene un mejor rendimiento en alto volúmenes de datos frente a MySQL.

### *Recolección de Datos*

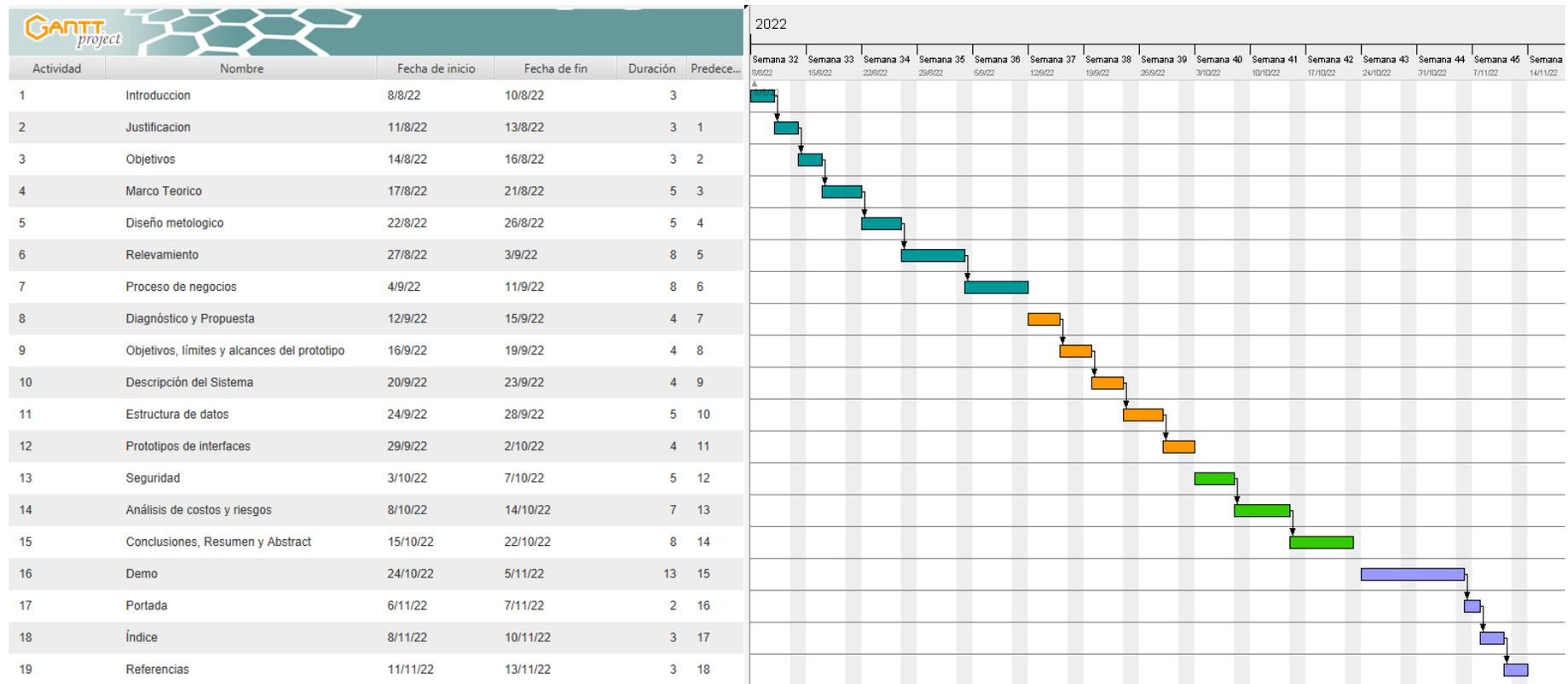
Para recabar información sobre la opinión de los clientes que asisten a los diferentes restaurantes se realizó una encuesta cerrada a través de Google Forms. (Anexo 1) Esta encuesta sirvió en principio para validar la idea principal del proyecto así como también permitió evaluar el comportamiento de los clientes.

Por otra parte se concretó una entrevista con el dueño de un bar muy reconocido de la ciudad de Posadas, además se pudo visualizar el trabajo diario para poder relevar los procesos que competen a este proyecto.

Planificación del Proyecto

A continuación se detalla en el siguiente diagrama de Gantt el plan de actividades para alcanzar los objetivos del desarrollo del trabajo final.

Figura 1: Diagrama de Gantt



Fuente: Elaboración propia.

## Relevamiento

Por tratarse de un proyecto dirigido a restaurantes en general, no es posible realizar un relevamiento estructural de cada uno de los locales gastronómicos de la ciudad de Posadas, por lo cual se optó por utilizar el restaurante “BEERLIN RESTO - BAR” como modelo.

Para este relevamiento se realizó una entrevista con uno de los dueños del local, el sr. Nicolás Eizmendi, del cual se obtuvo la siguiente información.

### *Relevamiento Estructural*

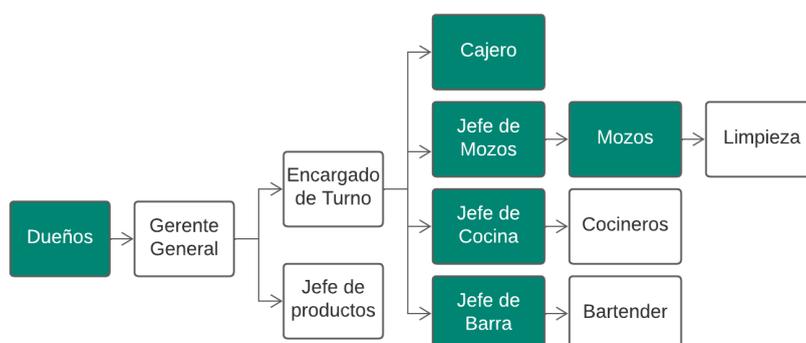
La organización cuenta con dos sucursales, una se encuentra ubicada en el ‘Balneario El Brete’ en inmediaciones de la costanera de la ciudad de Posadas, Misiones (Argentina). La otra se encuentra ubicada en la calle Bolívar 1979 , en el Posadas Plaza Shopping.

### *Relevamiento Funcional*

A continuación se presenta la estructura jerárquica de la organización tomada como modelo.

### *Organigrama*

Figura 2 : Organigrama de BEERLIN RESTO - BAR



Fuente: Elaboración propia.

### *Funciones de las Áreas*

Las áreas de color verde en el organigrama muestran los sectores alcanzados por el proyecto. Las mismas se describirán a continuación:

**Mozo:** es el encargado de interactuar con los clientes. Su función principal es tomar los pedidos, cargarlos en el sistema, enviar los pedidos a la cocina o la barra, distribuir los pedidos y enviar la cuenta.

**Jefe de barra:** Es el encargado de recibir los pedidos de tragos que envía el mozo y de dar aviso a los bartenders para su preparación. Una vez preparados se encarga de dejarlos a la vista del mozo para que los distribuya.

**Jefe de cocina:** Es el encargado de recibir los pedidos de las comidas que solicita el mozo, éste da aviso a los cocineros para que los preparen. Una vez preparados se encarga de dejarlos a la vista del mozo para que los distribuya.

**Jefe de mozos:** Es quien supervisa el trabajo de los mozos y de la limpieza. Distribuye a los mozos en función de la demanda y asigna las mesas a los clientes.

**Cajero:** Es el encargado de emitir los resúmenes de los gastos realizados y realizar los cobros.

**Dueños:** Son los responsables del control general de todas las operaciones, establecen los precios de las cartas y se encargan de distribuir estas a cada sucursal.

**Proceso:** Carga de precios

**Roles:** Dueños

**Pasos:** Cada vez que los dueños detectan una variante en los insumos o algún parámetro relevante al negocio deben realizar ajustes en los precios. El primer paso consiste en evaluar el porcentaje de incremento y cargar los precios en una planilla de excel que es enviada a una imprenta donde son reimpresas las cartas. Luego de que son recibidas las cartas con los precios nuevos los dueños se ocupan de distribuirlas a las sucursales. Una vez ahí, encargan la tarea de cambiar el papel en las cartas a los mozos.

**Proceso:** Toma y carga de pedidos.

**Roles:** Mozos, Jefe de cocina y Jefe de barra.

**Pasos:** Cada vez que llega un cliente el mozo hace entrega de uno o varios menús. Pasados unos minutos el mozo se vuelve a acercarse a la mesa para consultar al cliente lo que desea ordenar. Una vez tomado el pedido el mozo se acerca a una de las terminales donde registra el pedido. El sistema emite un ticket para las comidas solicitadas y un ticket para las bebidas solicitadas (en caso de ser bebidas que requieran preparación). Luego el mozo lleva el ticket de comidas al Jefe de cocina y el de bebidas al Jefe de barra. En caso de ser una bebida envasada el mozo no entrega ningún ticket al jefe de barra, solo toma el producto de la heladera.

**Detalle:** Los mozos no anotan los pedidos al interactuar con los clientes, estos deben recordar los pedidos a medida que atienden las distintas mesas.

**Proceso:** Distribución de pedidos

**Roles:** Mozos, Jefe de cocina y Jefe de barra.

**Pasos:** Una vez que los pedidos son preparados, el jefe correspondiente lo deja a la vista del mozo con el ticket emitido para que lo distribuya. Este último toma el pedido, verifica el ticket y lo lleva a la mesa. Como último paso se acerca a la terminal para indicar que el pedido fue entregado y deja el ticket en el mostrador donde acumulan los pedidos entregados.

**Proceso:** Cierre de mesa

**Roles:** Cliente, Mozos y Cajero

**Pasos:** Una vez que el cliente desea abonar su pedido, da aviso al mozo para que le entregue la cuenta. El mozo da aviso al cajero el cual se encarga de emitir un resumen de todos los consumos por parte del cliente y se lo entrega al mozo. El mozo lleva el resumen al cliente el cual le entrega el dinero o la tarjeta. El mozo lleva el medio de

pago al cajero, este realiza el cobro y entrega el ticket correspondiente al mozo para que se lo lleve nuevamente al cliente.

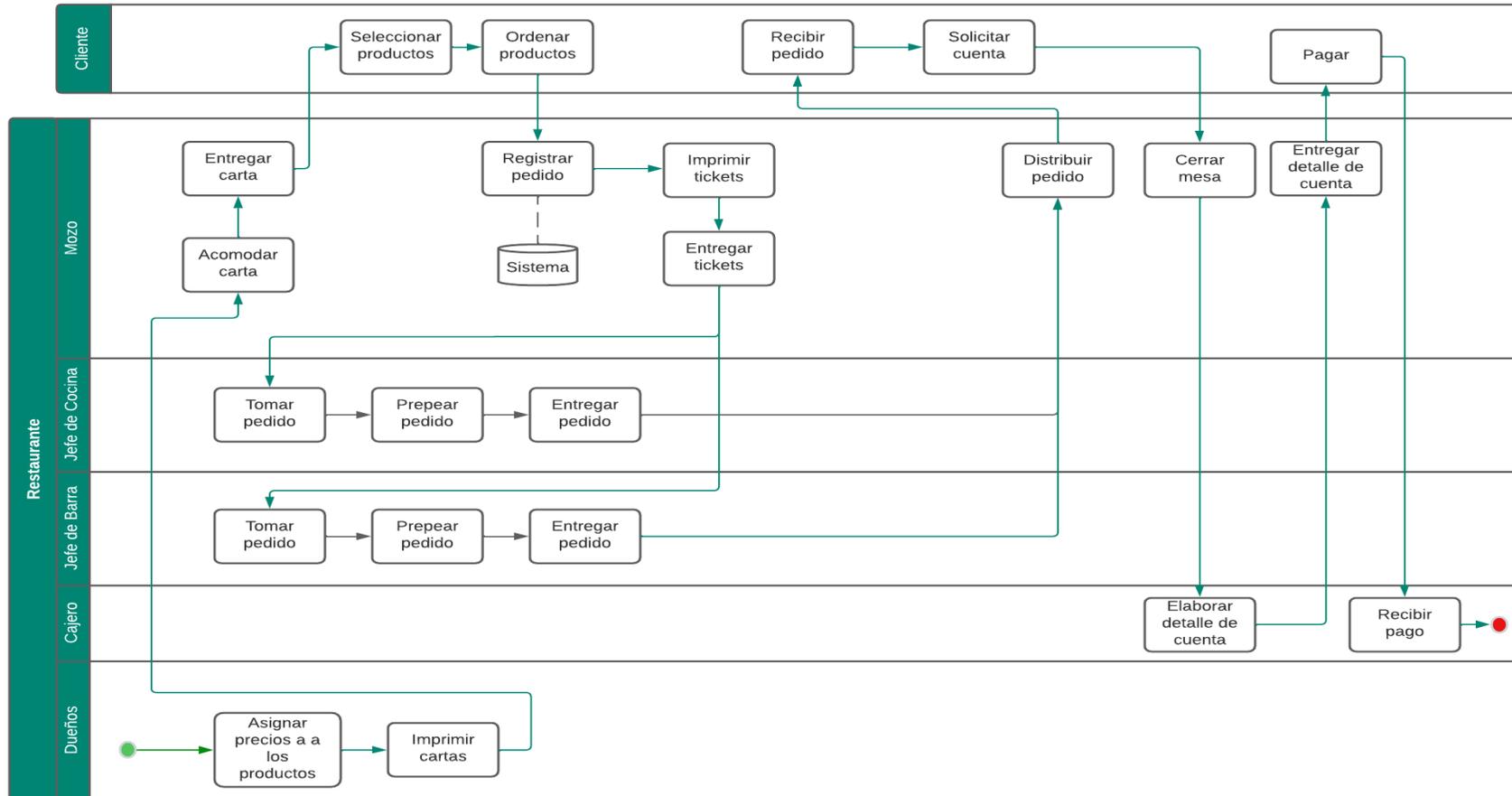
### *Documentación*

A continuación se mencionan los documentos relevados:

- **Ticket de cocina** : Es emitido por el sistema luego de que el mozo carga un pedido de comida (Anexo 2).
- **Ticket de barra**: Es emitido por el sistema luego de que el mozo carga un pedido de bebidas que requieran preparación (Anexo 3).
- **Carta**: listado de platillos y bebidas que se pueden ordenar. (Anexo 4)

### Procesos de Negocio

Figura 3: Diagrama de Modelo y Notación de Procesos de Negocio.



Fuente: Elaboración propia.

## Diagnóstico y Propuesta

### Diagnóstico

A partir de la información obtenida en el relevamiento realizado, se detallarán en las siguientes tablas los problemas encontrados en algunos de los procesos y sus respectivas causas detectadas:

Tabla 2: Problemas detectados del proceso de carga de precios relevado de Berlin Resto - Bar

Nombre del proceso: Carga de precios	
Problemas	Causas
1. Se vuelve compleja la actualización de precios en las cartas.	1. Los precios de los insumos cambian constantemente, al hacerlo hay que actualizar los precios de las cartas.
	2. Las cartas son manejadas íntegramente en papel.
	3. Tiempos de espera prolongados hasta que la imprenta tenga los impresos listos.
	4. Distribución física de cada carta en las distintas sucursales.
2. Aumento de gastos al tener que imprimir constantemente las nuevas cartas.	1. Costo de la imprenta.
	2. Costo de combustible al tener que distribuir las cartas actualizadas.

Fuente: Elaboración propia.

Tabla 3: Problemas detectados del proceso de toma y carga de pedidos relevado de Berlin Resto - Bar

Nombre del proceso: Toma y carga de pedidos	
Problemas	Causas
1. Tiempo de espera prolongado en la atención a los clientes.	1. El mozo debe acercarse reiteradas veces a cada mesa para atender a los clientes.
	2. A medida que ingresan más clientes aumenta la cantidad de pedidos.
	3. Los pedidos se cargan de manera manual en el sistema.
2. Sobrecarga de tareas del mozo	1. El mozo debe recordar cada pedido, detalle y de que mesa provino.
	2. Para reducir costos los restaurantes asignan muchas mesas a una única persona.

Fuente: Elaboración propia.

Tabla 4: Problemas detectados del proceso de cierre de mesa relevado de Berlin Resto - Bar

Nombre del proceso: Cierre de mesa	
Problemas	Causas
1. El tiempo de espera entre el pedido de la cuenta y el pago final se vuelve muy prolongado para el cliente.	1. El cliente debe esperar a que el mozo lo atienda, los días que los establecimientos son muy concurridos esto se vuelve una tarea larga.
	2. Luego que el cliente solicita el pago debe esperar hasta que el cajero genere el detalle del pedido.
	3. Nuevamente el cliente debe esperar a que el mozo en alguna de sus rondas pase por la caja, tome el detalle y se lo lleve a la mesa.
	4. El cliente entrega el medio de pago (ya sea efectivo o tarjeta) y debe esperar nuevamente a que el mozo lleve el dinero, el cajero realice el cobro y vuelva a la mesa.

Fuente: Elaboración propia.

### *Propuesta*

Con el desarrollo del sistema se logró solucionar los problemas detectados como la reimpresión constante de las cartas y los gastos que esto conlleva, los tiempos de espera elevados al realizar pedidos, la solicitud de cobros y la sobrecarga de tareas en los mozos.

Para esto se implementó un módulo en el cual los establecimientos gastronómicos pueden gestionar y publicar su menú y servicios ofrecidos a los clientes que consulten la aplicación.

Por otra parte se desarrolló un proceso en el cual los clientes pueden realizar sus propios pedidos, solicitar la cuenta y requerir la atención de los mozos desde la aplicación, disminuyendo así los tiempos de atención.

Así mismo gracias a la implementación del sistema los mozos logran disminuir la sobrecarga de trabajo acercándose a la mesa únicamente por tareas específicas como la entrega de pedidos.

## Objetivos, Límites y Alcances del Prototipo

### *Objetivos del Prototipo*

Diseñar e implementar un prototipo de sistema que permita optimizar y agilizar el proceso de atención de los clientes.

### *Límites*

Desde la búsqueda de un restaurante hasta que el cliente solicita la cuenta.

### *Alcances*

- Búsqueda de restaurante.
- Visualizar la carta.
- Registro de cliente.
- Reserva de mesa.
- Realizar pedidos.
- Verificar el estado del pedido.
- Mostrar detalle de consumo.
- Solicitar la cuenta.

## Descripción del Sistema

### *Product backlog*

Tabla 5: Product backlog.

ID	Historia de usuario	Prioridad	Puntos de historia	Dependencias
HU-001	Registro de cliente.	Alta	5	
HU-002	Visualizar carta.	Alta	5	HU-007
HU-003	Habilitar mesa.	Alta	4	
HU-004	Realizar pedido.	Alta	12	HU-001, HU-002 y HU-003
HU-005	Solicitar cuenta.	Alta	9	HU-004
HU-006	Ver detalle de pedidos.	Media	3	HU-004
HU-007	Búsqueda de restaurantes.	Media	3	
HU-008	Reserva de mesas.	Media	3	HU-001 y HU-007
HU-009	Consultar estado de pedido	Baja	3	HU-004
HU-010	Solicitar atención de mozo.	Baja	2	HU-001

Fuente: Elaboración propia.

### Historias de Usuario

A continuación, se detalla cada historia de usuario indicada en el product backlog:

Tabla 6: historia de usuario HU-001 registro de cliente.

ID	HU-001	Nombre	Registro de cliente
Descripción	Como cliente quiero registrarme en el sistema para poder ver las cartas, efectuar pedidos y realizar reservas.		
Criterios de aceptación	<ol style="list-style-type: none"> <li>1. Dado un campo vacío, cuando el cliente intente registrarse, entonces, el sistema mostrará un mensaje indicando que todos los campos son requeridos y marcará aquellos que estén vacíos.</li> <li>2. Dado un email que ya se encuentre registrado en el sistema, cuando este sea ingresado por un usuario nuevo, entonces el sistema mostrará un mensaje de error indicando que el email ya se encuentra registrado.</li> <li>3. Dada una contraseña menor a 8 dígitos alfanuméricos, cuando ésta sea ingresada, entonces se mostrará un mensaje indicando que la contraseña no es válida y los criterios que esta debe cumplir.</li> <li>4. Dado un número de documento que ya se encuentre registrado en el sistema, cuando sea ingresado por un nuevo usuario, entonces el sistema mostrará un mensaje indicando que el documento ya existe.</li> <li>5. Dado un número de documento de un cliente que se encuentre en estado "bloqueado", cuando sea ingresado por un nuevo usuario, entonces el sistema mostrará un error indicando que no se puede crear el usuario.</li> </ol>		
Prioridad	Alta	Puntos de historia estimados	5

Fuente: Elaboración propia.

Tabla 7: historia de usuario HU-002 ver la carta

ID	HU-002	Nombre	Visualizar carta.
Descripción	Como cliente quiero poder ingresar un número que identifique a la mesa o escanear un código QR para poder consultar la carta del restaurante en el cual me encuentro.		
Criterios de aceptación	<ol style="list-style-type: none"> <li>1. Dado un código QR en la mesa, cuando el cliente escanee el código, entonces el sistema deberá abrir la aplicación mostrando la carta del restaurante en el cual se encuentra el cliente.</li> <li>2. Dado un código QR de una mesa que se encuentre en estado inactivo, cuando el cliente escanee el mismo, entonces el sistema mostrará un error indicando esto.</li> <li>3. Dado un número identificador de mesa, cuando el cliente no cuente con escanner de QR, entonces éste podrá ingresar manualmente el</li> </ol>		

	identificador de la mesa. 4. Dado un número identificador de mesa invalido, cuando el cliente lo ingrese, entonces el sistema mostrará un error indicando que no se encontró la mesa. 5. Dado el número identificador de mesa de una mesa que se encuentra en estado inactivo, cuando el cliente ingrese el código, el sistema mostrará un error indicando que la mesa no se encuentra habilitada.		
<b>Prioridad</b>	Alta	<b>Puntos de historia estimados</b>	5

Fuente: Elaboración propia.

Tabla 8: historia de usuario HU-003 habilitar mesa.

ID	HU-003	Nombre	Habilitar mesa
<b>Descripción</b>	Como mozo quiero poder habilitar mesas para que cliente pueda realizar sus pedidos.		
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Dada una mozo, cuando quiera habilitar una mesa, entonces presionará la opción 'habilitar mesa' y el sistema generará un código de habilitación de 4 caracteres alfanumérico que deberá entregar al cliente.</li> <li>2. Dado un código de mesa, cuando este sea generado, se almacenará en la base de datos para utilizarlo mientras el cliente se encuentre en la mesa.</li> <li>3. Dada una mesa que se encuentre habilitada, cuando el sistema detecte este estado, entonces bloqueara el botón 'habilitar mesa'.</li> <li>4. Dado un código de mesa, cuando sea generado, entonces el sistema deberá tener la opción de imprimir el código.</li> </ol>		
<b>Prioridad</b>	Alta	<b>Puntos de historia estimados</b>	4

Fuente: Elaboración propia

Tabla 9: historia de usuario HU-004 realizar pedido.

ID	HU-004	Nombre	Realizar pedido
<b>Descripción</b>	Como cliente quiero poder seleccionar productos de la carta para poder realizar un pedido		
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Dado un cliente, cuando intente realizar un pedido, el sistema le solicitará que ingrese el código de mesa.</li> <li>2. Dado un producto de la carta, cuando el cliente lo seleccione, entonces se abrirá una ventana modal solicitando la cantidad y los detalles.</li> <li>3. Dado un código invalido, cuando el cliente intente realizar un pedido, entonces el sistema mostrará un error indicando que el código es incorrecto.</li> </ol>		
<b>Prioridad</b>	Alta	<b>Puntos de historia estimados</b>	12

Fuente: Elaboración propia.

Tabla 10: historia de usuario HU-005 solicitar cuenta.

ID	HU-005	Nombre	Solicitar cuenta.
Descripción	Como cliente quiero poder solicitar la cuenta para poder pagar y finalizar el servicio.		
Criterios de aceptación	<ol style="list-style-type: none"> <li>1. Dado un cliente, cuando presione en el menú la opción 'solicitar cuenta', entonces el sistema le mostrará el detalle de todos los pedidos realizados en la mesa.</li> <li>2. Dado un cliente que no haya realizado pedidos, cuando presione en el menú la opción 'solicitar cuenta', entonces el sistema mostrará un mensaje indicando que no hay pedidos.</li> <li>3. Dado un cliente, cuando este solicite la cuenta, entonces el sistema mostrará un combo desplegable con las opciones de pago que acepta ese establecimiento.</li> </ol>		
Prioridad	Alta	Puntos de historia estimados	9

Fuente: Elaboración propia.

Tabla 11: historia de usuario HU-006 ver detalle de pedidos.

ID	HU-006	Nombre	Ver detalle de pedidos
Descripción	Como cliente quiero tener una opción para poder ver los pedidos realizados en mi mesa.		
Criterios de aceptación	<ol style="list-style-type: none"> <li>1. Dado un cliente, cuando ingrese al menú y presione la opción 'ver pedidos', entonces se listaran todos los pedidos realizados en esa mesa.</li> <li>2. Dado un cliente que no haya realizado pedidos, cuando ingrese en la opción 'ver pedidos', entonces el sistema mostrará un mensaje indicando que no hay pedidos.</li> <li>3. Dado un pedido, cuando el cliente presione sobre este, entonces el sistema mostrará los detalles del pedido y su estado.</li> </ol>		
Prioridad	Media	Puntos de historia estimados	3

Fuente: Elaboración propia.

Tabla 12: historia de usuario HU-007 búsqueda de restaurantes.

<b>ID</b>	HU-007	<b>Nombre</b>	Búsqueda de restaurantes.
<b>Descripción</b>	Como cliente quiero una para poder ver los restaurantes que usan la aplicación.		
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Dado un cliente, cuando ingrese a la aplicación, entonces se mostrará una barra de búsqueda de restaurantes.</li> <li>2. Dado un cliente, cuando escriba el nombre de un restaurante en la barra de búsqueda, entonces el sistema listará todas las sucursales que se encuentren registradas.</li> <li>3. Dada una búsqueda, cuando el cliente ingrese un nombre que no coincida con ninguno en la base de datos, entonces el sistema mostrará un mensaje indicando que no hay coincidencias.</li> </ol>		
<b>Prioridad</b>	Media	<b>Puntos de historia estimados</b>	3

Fuente: Elaboración propia.

Tabla 13: historia de usuario HU-008 reserva de mesas.

<b>ID</b>	HU-008	<b>Nombre</b>	Reserva de mesas.
<b>Descripción</b>	Como cliente quiero poder buscar un restaurante para reservar una mesa.		
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Dado un cliente, cuando éste seleccione un restaurante , entonces el sistema mostrará una opción 'reservar mesa'.</li> <li>2. Dado un restaurante, cuando este no acepte reserva de mesas, entonces el sistema ocultará la opción 'reservar mesa'.</li> <li>3. Dando un cliente, cuando éste presione la opción 'reservar mesa', entonces el sistema solicitará que ingrese la fecha y la hora de la reserva.</li> </ol>		
<b>Prioridad</b>	Media	<b>Puntos de historia estimados</b>	3

Fuente: Elaboración propia.

Tabla 14: historia de usuario HU-009 Consultar estado del pedido.

ID	HU-009	Nombre	Consultar estado de pedido
<b>Descripción</b>	Como cliente quiero poder tener una opción para poder ver el estado de mi pedido.		
<b>Criterios de aceptación</b>	1. Dado un pedido realizado, cuando el cliente presione el botón 'ver estado' entonces se listaran los estados por los que pasó el pedido y el estado actual.		
<b>Prioridad</b>	Baja	<b>Puntos de historia estimados</b>	3

Fuente: Elaboración propia.

Tabla 15: historia de usuario HU-010 Solicitar atención del mozo.

ID	HU-010	Nombre	Solicitar atención de mozo.
<b>Descripción</b>	Como cliente quiero poder tener una opción para solicitar atención del mozo.		
<b>Criterios de aceptación</b>	<ol style="list-style-type: none"> <li>1. Dado un cliente, cuando éste presione la opción 'Llamar al mozo', entonces el sistema abrirá una ventana modal solicitando de manera opcional el motivo del llamado.</li> <li>2. Dada una solicitud de atención, cuando el cliente finalice el pedido, el sistema solicitará que ingrese el código de mesa provisto por el mozo.</li> <li>3. Dado un restaurante, cuando el cliente presione la opción 'Solicitar atención' entonces el sistema enviará una notificación al restaurante indicando el número de mesa y el motivo.</li> <li>4. Dado un pedido de atención, cuando el cliente ingrese un código de mesa invalido, el sistema mostrará un error indicando que el código no es válido.</li> </ol>		
<b>Prioridad</b>	Baja	<b>Puntos de historia estimados</b>	2

Fuente: Elaboración propia.

### Sprint Backlog

Una vez definidas las historias de usuarios, se creará el primer sprint del prototipo.

Tabla 16: primer sprint de backlog.

Sprint	Historia de usuario	ID	Tareas	Prioridad	Estimado	Estado
1	HU-001 - Registro de cliente.	1.1.1	Diseñar la pantalla de registro.	Alta	2 días	Hecho
		1.1.2	Crear la estructura de datos.	Alta	1 día	Por hacer
		1.1.3	Codificar el modulo.	Alta	3 día	Por hacer
		1.1.4	Generar casos de prueba	Alta	1 día	Por hacer
		1.1.5	Integrar módulo.	Alta	1 día	Por hacer
		1.1.6	Realizar testing unitario	Alta	1 día	Por hacer
	HU-002 - Ver la carta	1.2.1	Diseñar la pantalla de registro.	Alta	2 días	Hecho
		1.2.2	Crear la estructura de datos.	Alta	1 día	Por hacer
		1.2.3	Codificar el modulo.	Alta	5 días	Por hacer
		1.2.4	Generar casos de prueba	Alta	1 día	Por hacer
		1.2.5	Integrar módulo.	Alta	1 día	Por hacer
		1.2.6	Realizar testing unitario	Alta	1 día	Por hacer
	HU-003 - Realizar Pedido	1.3.1	Diseñar la pantalla de registro.	Alta	2 días	Hecho
		1.3.2	Crear la estructura de datos.	Alta	1 día	Por hacer
		1.3.3	Codificar el modulo.	Alta	7 días	Por hacer
		1.3.4	Generar casos de prueba	Alta	1 día	Por hacer
		1.3.5	Integrar módulo	Alta	2 días	Por hacer
		1.3.6	Realizar testing unitario	Alta	1 día	Por hacer

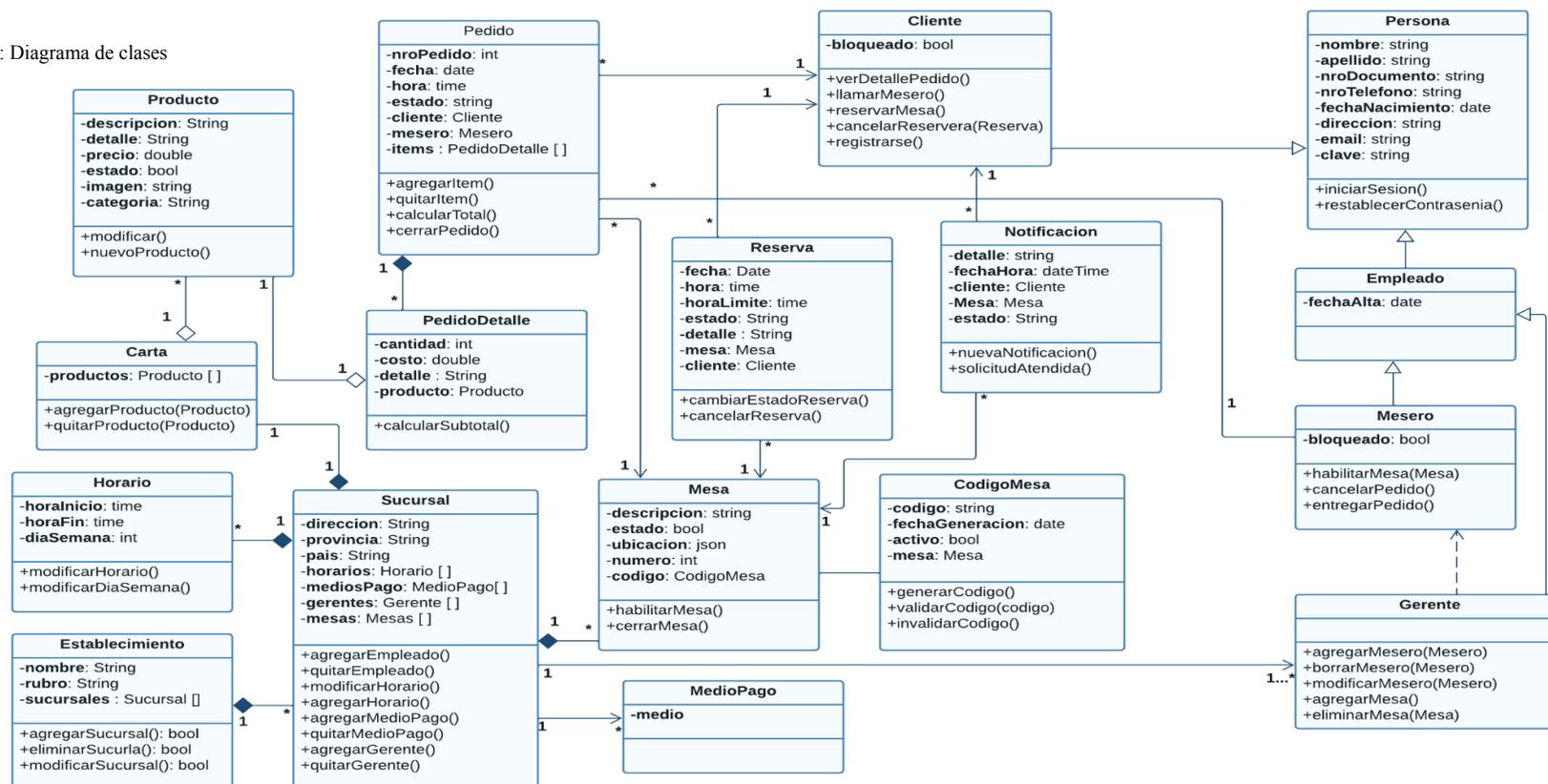
Fuente: Elaboración propia.

## Estructura de Datos

A continuación se presentan los diagramas de clases y de entidad relación, los cuales nos facilitan la comprensión de la estructura interna del proyecto y cómo serán almacenados los datos dentro de la base de datos.

## Diagrama de Clases

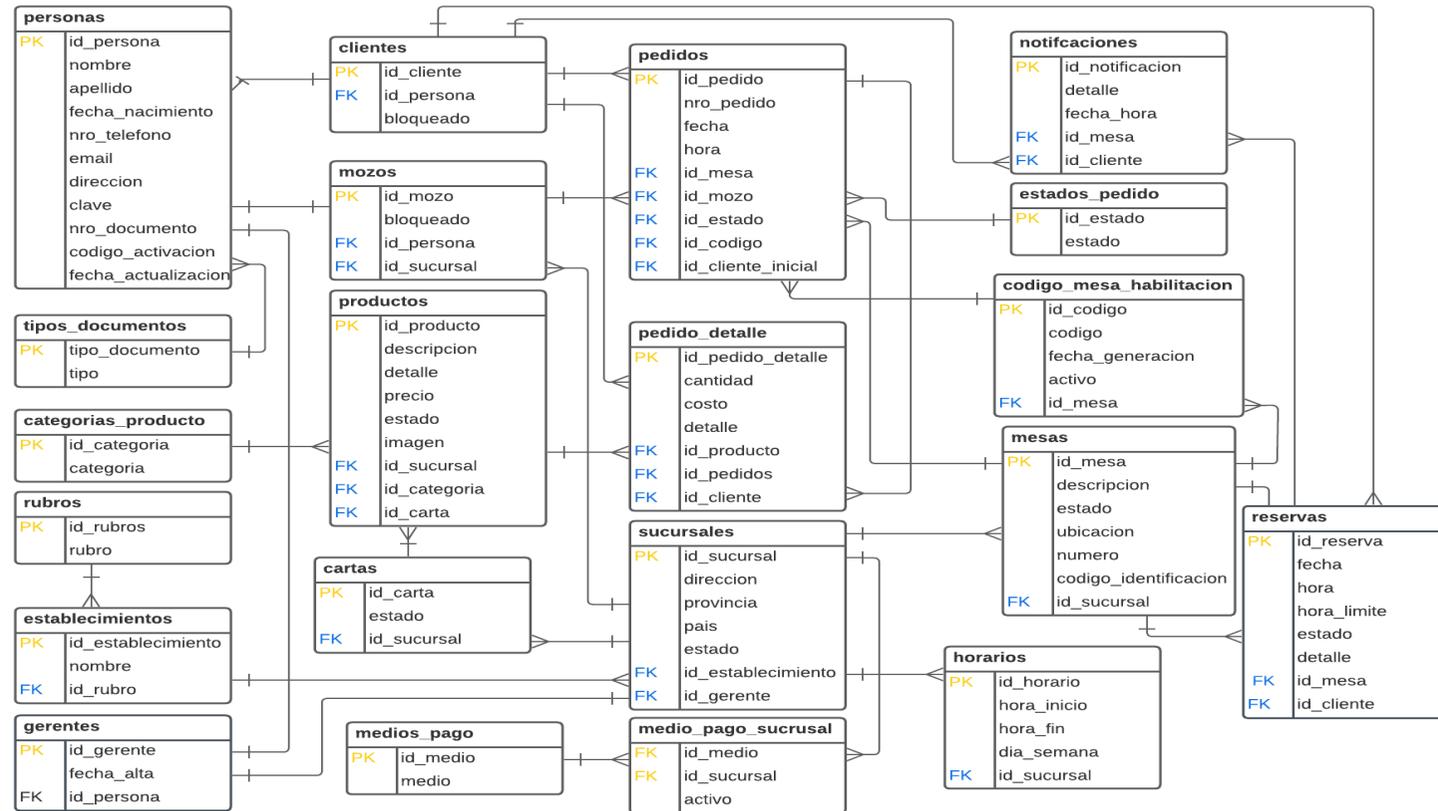
Figura 4 : Diagrama de clases



Fuente: Elaboración propia.

## Diagrama de Entidad Relación

Figura 5 : Diagrama entidad relación.

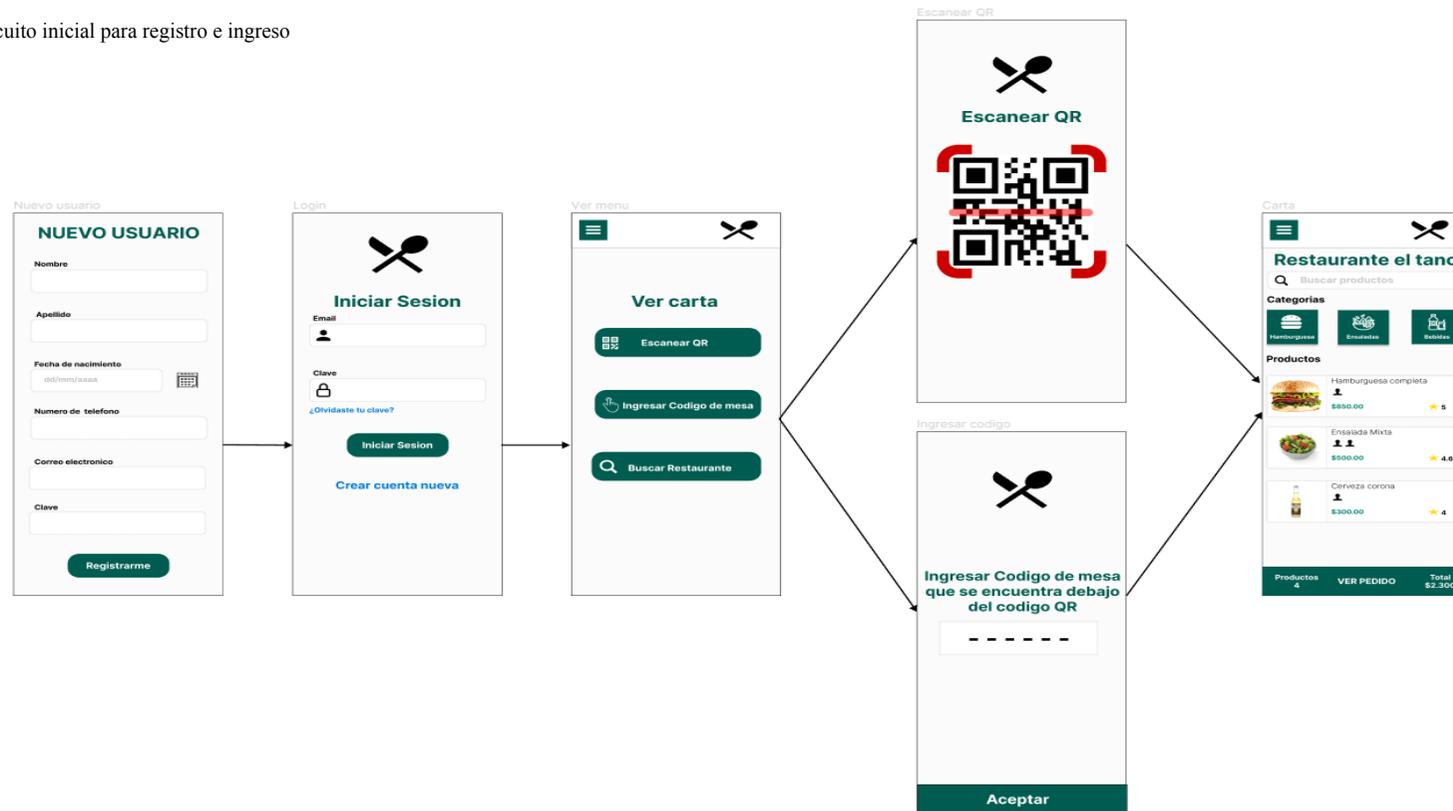


Fuente: Elaboración propia.

## Prototipos de Interfaces de Pantallas

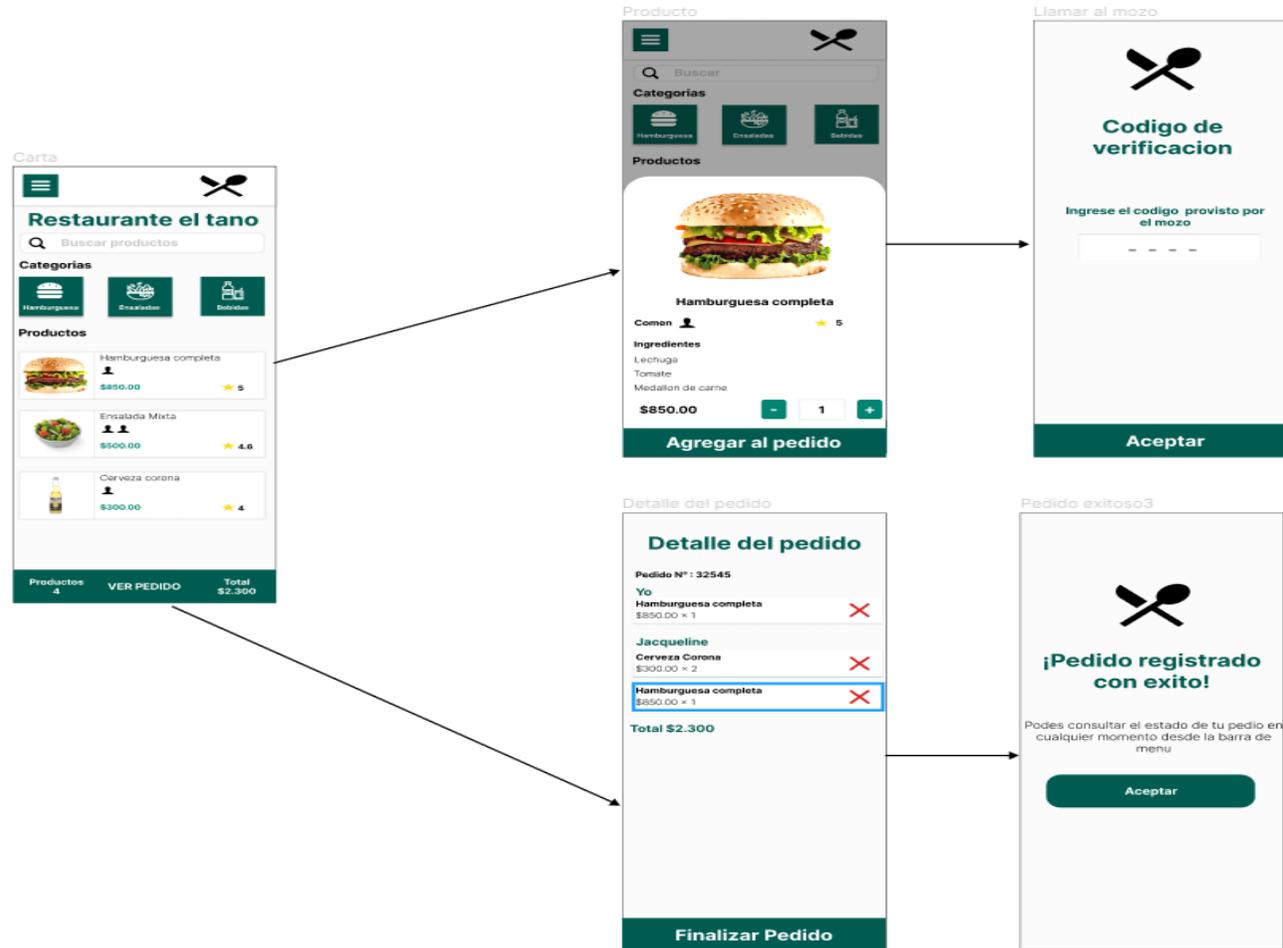
En el siguiente apartado se presenta el prototipo de los distintos circuitos con los que cuenta el sistema, para su mejor visualización se posicionarán las páginas siguientes en orientación horizontal.

Figura 6: Circuito inicial para registro e ingreso



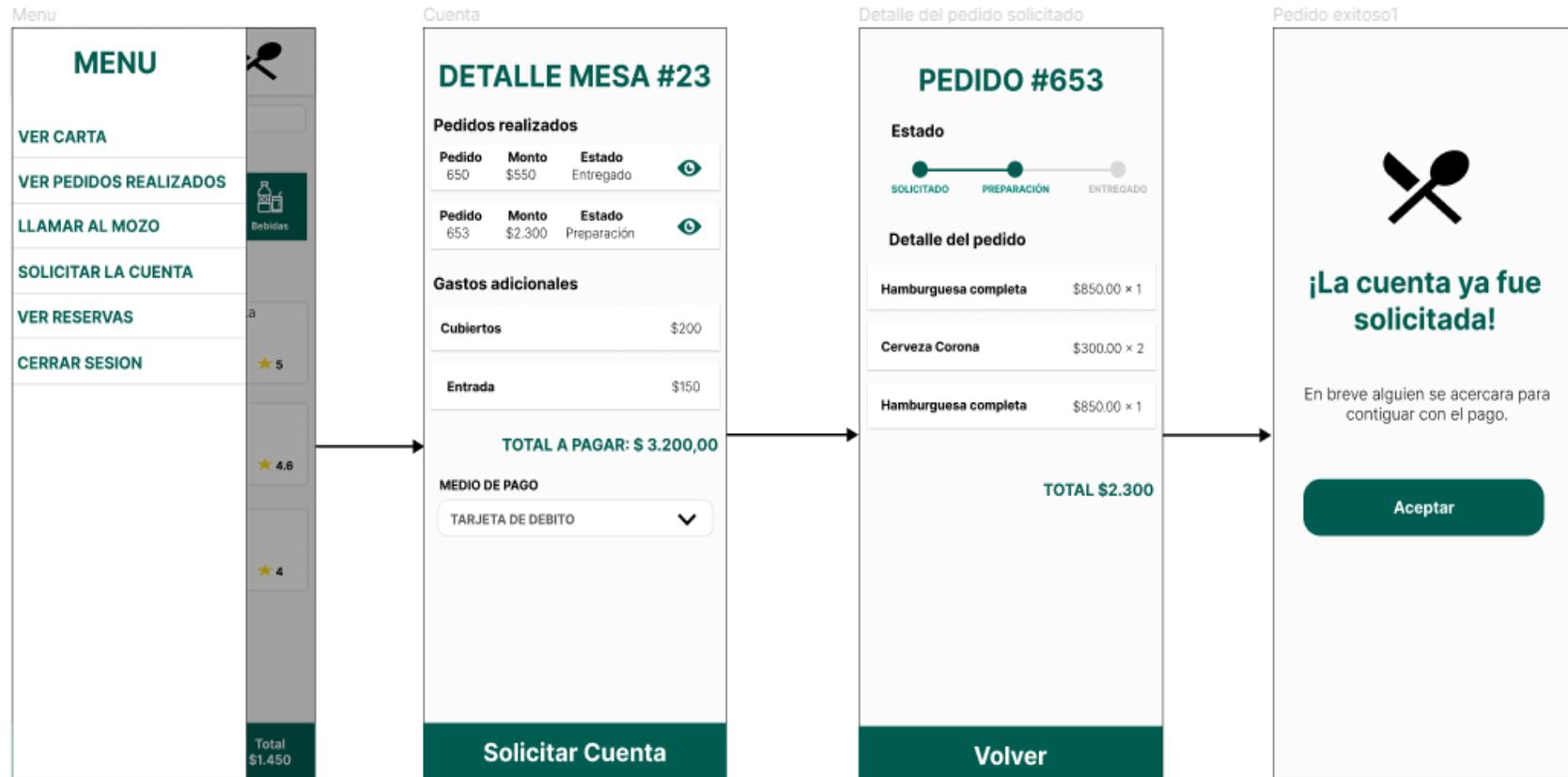
Fuente: Elaboración propia.

Figura 7: Circuito para realizar un pedido.



Fuente: Elaboración propia.

Figura 8: Circuito para ver los pedidos realizados y solicitar la cuenta.



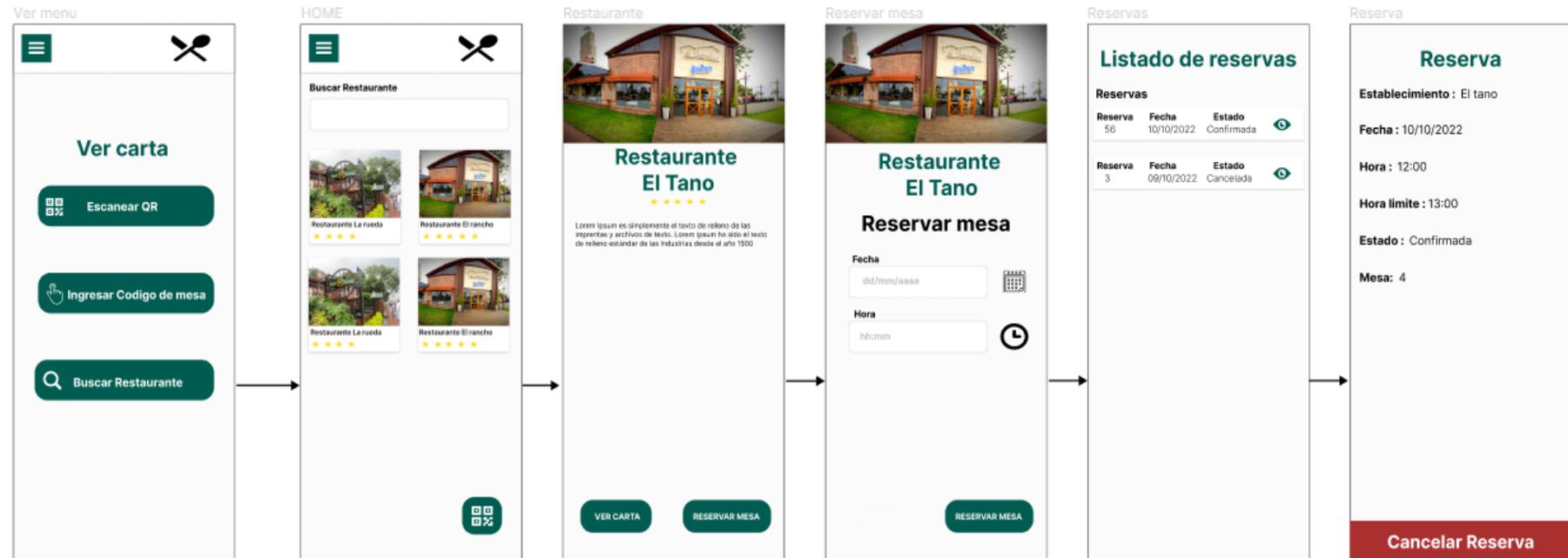
Fuente: Elaboración propia.

Figura 9 : Circuito para solicitar atención del mozo



Fuente: Elaboración propia.

Figura 10 : Circuito para buscar un restaurante y reservar una mesa

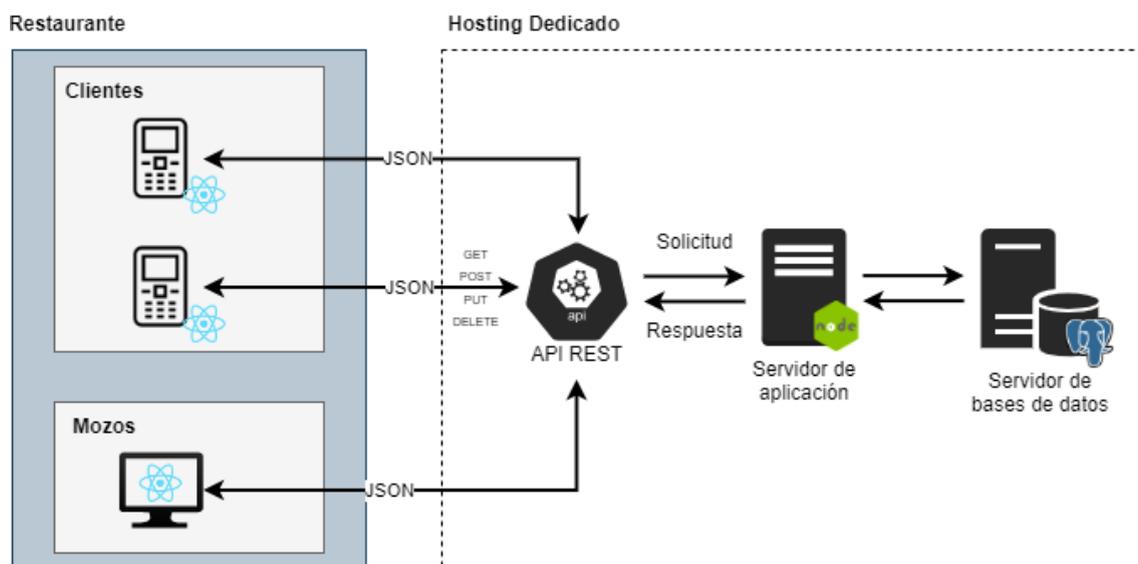


Fuente: Elaboración propia.

### Diagrama de Arquitectura

En el siguiente gráfico se describe la arquitectura para la interacción entre los clientes y los restaurantes. La aplicación interactúa con el servidor mediante solicitudes HTTPS de tipo Get, Post, Put y Delete que son enviadas en formato JSON hacia una API. Esta API se comunica con el backend el cual se encarga de procesar la solicitud e interactuar con la base de datos en caso de ser necesario. Una vez procesada la solicitud, el servidor responde al usuario nuevamente con datos en formato JSON. Cuando la aplicación del cliente tiene los datos los formatea para mostrar la información al usuario final.

Figura 11 : Diagrama de arquitectura.



Fuente: Elaboración propia.

## Seguridad

### Acceso a la Aplicación

Para el inicio de sesión el usuario debe registrar un correo electrónico, el cual no puede existir en la base de datos. Además, tiene que ingresar una contraseña que deberá cumplir con los siguiente requisitos para poder ser utilizada :

- Tener un mínimo de 8 caracteres y no más de 15.
- Contar, al menos, con un carácter especial.
- Poseer, al menos, una letra mayúscula.

- Poseer, al menos, una letra minúscula.
- Disponer de, al menos, un número.

La contraseña es encriptada utilizando la función de hashing bcrypt para poder ser almacenada de manera segura en la base de datos. Una función de hashing es “un algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija. Independientemente de la longitud de los datos de entrada, el valor hash de salida tendrá siempre la misma longitud” (kaspersky, 2014).

Una vez registrado el usuario, para activar su cuenta, debe validar su correo electrónico mediante un código de 6 dígitos alfanuméricos que se le envía a la dirección registrada.

Si el usuario intenta ingresar utilizando un correo electrónico o contraseña incorrectos, luego de dos intentos, el sistema le solicitará que complete un código captcha para realizar un nuevo intento.

Una vez dentro del sistema, para realizar pedidos los clientes deben ingresar un código de 4 dígitos alfanuméricos que es entregado por el mozo. Este código se valida contra la base de datos cada vez que el cliente intente realizar un pedido o llamar al mozo. Este proceso se realiza para evitar que cualquier persona pueda escanear el código QR y realizar pedidos en una mesa en la cual no se encuentra.

La aplicación cuenta con 4 perfiles de usuario:

- Administrador: tiene permiso de crear, modificar y eliminar los establecimientos, gestionar las sucursales para el establecimiento y designar gerentes a las sucursales.
- Gerente: es el usuario responsable de la administración general de la sucursal. Tiene permiso de administrar la carta, crear, modificar y eliminar mesas de su sucursal. Dar de alta los mozos y asignarlos a la sucursal.
- Mozo: Tiene permiso para validar la identidad de los usuarios, ver los pedidos realizados por los clientes, cambiar el estado de estos.

- Cliente: Tiene permisos de realizar pedidos, solicitar la cuenta, ver los pedidos realizados, llamar al mozo, solicitar la cuenta, buscar establecimientos y reservar mesas.

### *Política de Respaldo de Información*

Para resguardar los datos de los usuarios y los establecimientos se realizan copias completas de manera diaria en la nube de Amazon Web Service. Este proceso se realiza de manera automática en el horario de las 06:00hs ya que es el horario en el que los establecimientos presentan menor actividad. Una vez que se tienen las copias de seguridad de una semana se comenzarán a borrar los backups más antiguos para optimizar el espacio de almacenamiento.

El código de la aplicación será alojado en un repositorio de github a medida que es desarrollado. Adicionalmente se realiza una copia diaria en un servidor local en caso de que el sitio de github sufra caídas.

## **Análisis de Costo**

A continuación, se presentan los costos de recursos humanos y de gastos operativos. Estos fueron calculados el 15 de Octubre del 2022 considerando que el proyecto tendrá una duración de 4 meses. Adicionalmente, debemos tener en cuenta que el valor del dólar al día de la fecha en el Banco de la Nación Argentina es de 1 U\$\$ = \$159,50. (Banco de la Nación Argentina [BNA], 2022).

Los valores de los salarios fueron tomados del sitio web del Consejo Profesional de Ciencias Informáticas de la Provincia de Buenos Aires (CPCIBA, 2022).

Tabla 17 : Análisis de costos de recursos humanos

Rol	Salario Mensual	Cantidad	Meses Totales	Total
Lider/Jefe de Proyectos	\$400.397,00	1	4	\$1.601.588,00
Analista de Sistemas Senior	\$363.303,00	1	1	\$363.303,00
Programador Senior	\$355.120,50	1	4	\$1.420.482,00
Diseñador de Usabilidad (UX User Experience)	\$277.659,50	1	1	\$277.659,50
Administrador de Bases de datos Senior	\$360.000,00	1	4	\$1.440.000,00
Tester	\$296.206,50	1	4	\$1.184.826,00
<b>Total</b>				<b>\$6.287.858,50</b>

Fuente: Elaboración propia.

Tabla 18: Análisis de gastos operativos

Recurso	Tipo de Recurso	Fuente	Cant.	Subtotal Inicial
Notebook Lenovo IdeaPad 5i	Hardware	<a href="https://www.lenovo.com/ar">https://www.lenovo.com/ar</a>	6	\$1.170.000,00
Licencia Windows 10 Pro	Software	<a href="https://www.microsoft.com">https://www.microsoft.com</a>	6	\$173.994,00
Hosting dedicado - Hostinger VPS 6	Servicio	<a href="https://www.hostinger.com.ar/">https://www.hostinger.com.ar/</a>	1	\$107.988,00
Jetbrain IDE WebStorm	Software	<a href="https://www.jetbrains.com/">https://www.jetbrains.com/</a>	1	\$25.360,50
Github	Servicio	<a href="https://github.com/">https://github.com/</a>	1	\$0
NodeJS	Software	<a href="https://nodejs.org/">https://nodejs.org/</a>	1	\$0
React	Software	<a href="https://es.reactjs.org/">https://es.reactjs.org/</a>	1	\$0
<b>Total</b>				<b>\$1.477.342,50</b>

Fuente: Elaboración propia.

Considerando todos los costos, en la siguiente tabla se presentan los montos totales necesarios para poder llevar a cabo el proyecto.

Tabla 19: Total de costos

Tipo de Costo	Precio Final
Costos de Recursos Humanos	\$6.287.858,50
Costos Software	\$199.304,50
Costos Hardware	\$1.170.000,00
Costos Servicio	\$107.988,00
<b>Total</b>	<b>\$7.765.151,00</b>

Fuente: Elaboración propia.

## Análisis de Riesgos

Ningún proyecto está exento de riesgos. Conocer las amenazas que se pueden presentar nos ayuda a planificar las acciones para mitigar estos riesgos potenciales. En el siguiente apartado se analizan los posibles riesgos que se pueden presentar, sus causas y un plan de contingencia para aquellos que puedan afectar en mayor medida.

Tabla 20: Riesgos y sus causas

Tipo	Riesgo	Causa
Proyecto	Testeo incorrecto o escaso.	Falta de experiencia del encargado de testing y el desarrollador.
Tecnico	Caída del servidor.	Falta de mantenimiento en los servidores.
Proyecto	Escasa aceptación del usuario.	Interfaz poco intuitiva para los usuarios.
Tecnico	Pérdida de la información.	Fallas en los discos donde se encuentra alojada de la base de datos.
Tecnico	Fallo en el suministro eléctrico del restaurante.	Cortes de luz general, reparación del tendido eléctrico, fallos en los transformadores, entre otros.
Proyecto	Poco interés por parte de los restaurantes.	Muchos restaurantes prefieren mantener el estilo tradicional de atención.

Fuente: Elaboración propia.

Tabla 21: Matriz de riesgos

			Impacto				
			Muy Bajo	Bajo	Medio	Alto	Muy alto
			1	2	3	4	5
Probabilidad	Muy Alta	0,9	0,9	1,8	2,7	3,6	4,5
	Alta	0,7	0,7	1,4	2,1	2,8	3,5
	Media	0,5	0,5	1	1,5	2	2,5
	Baja	0,3	0,3	0,6	0,9	1,2	1,5
	Muy Baja	0,1	0,1	0,2	0,3	0,4	0,5

Fuente: Elaboración propia.

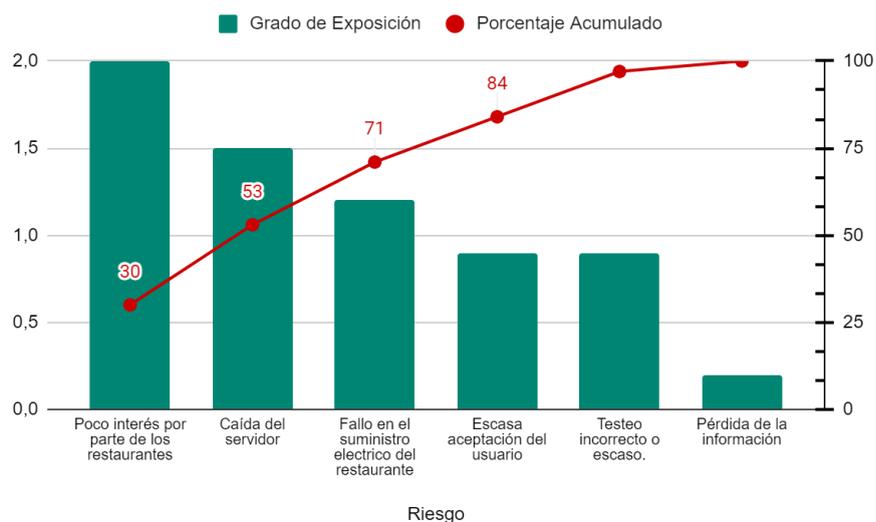
Tabla 22: Análisis cuantitativo y grado de exposición

Riesgo	Probabilidad	Impacto	Grado de Exposición	Porcentaje	Porcentaje Acumulado
Poco interés por parte de los restaurantes	Media	Alto	2	30%	30%
Caída del servidor	Baja	Muy Alto	1,5	23%	53%
Fallo en el suministro eléctrico del restaurante	Baja	Alto	1,2	18%	71%
Escasa aceptación del usuario	Baja	Medio	0,9	13%	84%
Testeo incorrecto o escaso.	Baja	Medio	0,9	13%	97%
Pérdida de la información	Muy baja	Bajo	0,2	3%	100%

Fuente: Elaboración propia.

En el siguiente gráfico se ordenaron los riesgos detectados según su probabilidad de ocurrencia de mayor a menor. Esto nos sirvió para aplicar el principio de Pareto el cual nos dice que “80% de las consecuencias proviene del 20% de las causas” (Santander Universidades, 2022).

Figura 11: Principio de Pareto de la exposición al riesgo



Fuente: Elaboración propia.

### Plan de Contingencia

A partir de la figura 11 que refleja el principio de Pareto pudimos detectar los principales riesgos que requieren de nuestra atención inmediata.

- Poco interés por parte de los restaurantes.
- Caída del servidor.
- Fallo en el suministro eléctrico del restaurante.

Para estos riesgos detectados se desarrolló el siguiente plan de contingencia para poder mitigar los posibles efectos en caso de que se presentaran.

Tabla 23: Riesgos de atención inmediata con su plan de contingencia.

Riesgos	Acciones de contingencia
Poco interés por parte de los restaurantes	Incluir en el proyecto a distintos restaurantes de la zona para que aporten su experiencia en el rubro.
	Elaborar presentaciones donde se expongan todos los beneficios que se obtienen al digitalizar estos procesos.
	Periodos de prueba gratuitos para que puedan validar el uso con sus clientes.
Caída del servidor	Contar con una operación para descargar el menú cada vez que se modifica algún dato de la carta.
	Tener un servidor adicional que almacene las cartas de cada restaurante. Adicionalmente ofrecer a los restaurantes un QR en caso de emergencia que apunte a ese servidor para que la gente pueda seguir consultando el menú.
Fallo en el suministro eléctrico del restaurante	Contar con tablets en los restaurantes con acceso a internet 4G.

Fuente: Elaboración propia.

## **Conclusiones**

El presente proyecto tuvo como objetivo proporcionar una alternativa a la atención tradicional brindada por los restaurantes con el fin de mejorar la atención y optimizar los procesos. Esto se logró mediante la implementación de una herramienta que posibilitó la interacción directa de los clientes con los restaurantes disminuyendo la necesidad de intermediarios. Como consecuencia de esto la carga que recae sobre los mozos disminuye, los clientes tienen una mejor experiencia y para los restaurantes esto se traduce en mayores utilidades.

Este proyecto propone un gran cambio cultural para los restaurantes y los clientes, donde estamos acostumbrados a un tipo de atención que data de siglos atrás. Dejar atrás los usos y costumbres supone un gran desafío pero gracias a la tecnología, con la que contamos hoy en día, podemos evolucionar estos procesos para realizar una mejora continua en la atención.

Por último, a nivel personal, considero que el proyecto presentó un gran desafío al tener que aplicar y profundizar los conocimientos académicos adquiridos con un enfoque práctico y realista. Si bien aún quedan muchos desafíos por solucionar estoy muy complacido con los resultados logrados al poder brindar una solución al rubro gastronómico.

## Demo

En el siguiente enlace podemos ver un video de demostración, en el cual se muestra el funcionamiento del sistema:

[https://drive.google.com/drive/u/0/folders/1fzFQ7jr4Ywu17YFrjayZAFc\\_qKiiVysT](https://drive.google.com/drive/u/0/folders/1fzFQ7jr4Ywu17YFrjayZAFc_qKiiVysT)

El código de la demo se encuentra publicado en Github, y puede ser consultado en el siguiente enlace:

<https://github.com/lcassettai/SeminarioFinal>

## Referencias

- Amazon. (s.f.). ¿Qué es NoSQL?. Recuperado de <https://aws.amazon.com/es/nosql/>
- Apple. (s.f.). Swift. Un lenguaje potente y abierto a todos para crear apps increíbles.  
Recuperado de <https://www.apple.com/es/swift/>
- Banco de la Nación Argentina [BNA]. (s.f.). Cotización de divisas. Recuperado de <https://www.bna.com.ar/>
- Becas-Santander. (2022). Ley de Pareto: la regla del 80/20 y el éxito empresarial.  
Recuperado de <https://www.becas-santander.com/es/blog/ley-de-pareto.html>
- Bonnie Eisenman. (2015). Learning React Native. Recuperado de <https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html>
- Ecured. (s.f.). Aplicación web. Recuperado de [https://www.ecured.cu/Aplicaci%C3%B3n\\_web](https://www.ecured.cu/Aplicaci%C3%B3n_web).
- Mozilla. (2022). Acerca de JavaScript. Recuperado de [https://developer.mozilla.org/es/docs/Web/JavaScript/About\\_JavaScript](https://developer.mozilla.org/es/docs/Web/JavaScript/About_JavaScript).
- kaspersky. (2014). ¿Qué Es Un Hash Y Cómo Funciona?. Recuperado de <https://latam.kaspersky.com/blog/que-es-un-hash-y-como-funciona/2806/>
- Kotlinlang. (s.f.). Get started with Kotlin. Recuperado de <https://kotlinlang.org/docs/getting-started.html#install-kotlin>
- Pressman Roger. (2010). Ingeniería del software un enfoque práctico (7ma ed.). Mc Graw Hill

Redhat. (2020). ¿Qué es una API de REST?. Recuperado de <https://www.redhat.com/es/topics/api/what-is-a-rest-api>

Real Academia Española. (s.f.). Carta. En Diccionario de la lengua española. Recuperado el 13 de noviembre de 2022, de <https://dle.rae.es/carta>

Real Academia Española. (s.f.). Camarero. En Diccionario de la lengua española. Recuperado el 13 de noviembre de 2022, de <https://dle.rae.es/camarero>

Real Academia Española. (s.f.). Sucursal. En Diccionario de la lengua española. Recuperado el 13 de noviembre de 2022, de <https://dle.rae.es/sucursal>

## Anexos

### Anexo 1

Encuesta realizada a través de google forms.

## Sistema de autopedidos para Restaurantes.

Esta encuesta es anónima, tus datos de usuario no quedaran registrados.

---

\*Obligatorio

1. **¿Cuál es tu edad? \***

*Marca solo un óvalo.*

- Menor a 18
- 18 años a 24 años
- 25 años a 34 años
- 35 años a 44 años
- 45 años a 54 años
- Mas de 54
- Prefiero no responder

2. **¿Con qué género te identificas? \***

*Marca solo un óvalo.*

- Masculino
- Femenino
- Otro
- Prefiero no responder



6. **Cuando vez un QR en la mesa de un restaurante, usted lo : \***

*Marca solo un óvalo.*

- Escanea  
 Ignora  
 Que es un QR?

7. **Si pudieras elegir como tener la carta del restaurante, te gustaría tenerlo en tu celular o esperar a que el mozo te acerque la carta? \***

*Marca solo un óvalo.*

- Esperar al mozo  
 Tener la carta en mi celular

8. **Como calificarías el servicio de atención general de los mozos en restaurantes : \***

*Marca solo un óvalo.*

	1	2	3	4	5	
Lento	<input type="radio"/>	Rapido				

9. **Preferís tener una aplicación descargada en tu celular para utilizar algún servicio Ej.: PedidosYa, o preferís utilizar el sitio web de PedidosYa sin tener que descargar la aplicación? \***

*Marca solo un óvalo.*

- Prefiero tener la aplicacion  
 Prefiero no descargar la aplicación y utilizar el sitio web

10. Te gustaría poder ordenar desde tu propio celular sin tener que esperar a que el mozo te atienda ? \*

Marca solo un óvalo.

Si, me gustaría poder ordenar desde mi celular

No, prefiero que me atienda el mozo

11. Te gustaría poder ver un resumen en tu celular de lo que vas a tener que abonar ? \*

Marca solo un óvalo.

Si, me gustaría ver el resumen

No, prefiero enterarme al final

12. Te gustaría poder calificar el servicio ofrecido por el restaurante ? \*

Marca solo un óvalo.

Si

No

Tal vez

13. Alguna recomendación que te gustaría incluir :

---

---

---

---

---

## Anexo 2

Ticket de cocina tomado del restaurante "Berlin Resto - Bar".

<b>COCINA: &lt;&lt;Salon&gt;&gt;</b>	
<b>Fecha: 10/09/2022</b>	<b>Hora: 19:43</b>
<b>Mozo: Uero</b>	<b>MESA: 12</b>
-----	
1 Mozarellitas	
-----	
COMBINA CON:	
1 Heineken	
1 Happy Berlin	

## Anexo 3

Ticket de barra tomado del restaurante "Berlin Resto - Bar".

<b>BARRA: &lt;&lt; Salon&gt;&gt;</b>	
<b>Fecha: 10/09/2022</b>	<b>Hora: 19:41</b>
<b>Mozo: GONZALO</b>	<b>MESA: 45</b>
-----	
1 Capuchino	☛
1 Cafe Doble	
CORT	
-----	
<b>COMBINA CON:</b>	
1 Tosado J&Q pan de Miga	

## Anexo 4

Modelo de carta tomado del restaurante “Berlin Resto - Bar”.

**Cervezas**  
 Heineken 330 cc. 350  
 Sol 330 cc 350  
 Heineken tirada 250cc 400

Escaneá el código QR  
 y probá los mejores  
 Cervezas artesanales  
 de nuestra pizarra

**Caipis**  
 Caipiroska Sernova 600  
 Upgrade Absolut 800  
 Caipiroska de frutos de estación 700  
 Caipiroska de Frutilla Sernova 700  
 Caipiriña (Cachaza, limá y azúcar) 500

**Daikiris**  
 Frutilla - Ananá - Durazno 650

**Whiskys**  
 Chivas 18 años 1300  
 Chivas Mizunara 900  
 Jameson 500  
 Jameson Caskmates IPA 600  
 Jameson Caskmates Stout 600  
 Jameson Black Barrel 600  
 Ballantines 550  
 Jim Beam Honey 500

**RESTO - BAR**  
**Berlin**  
**COFFEE & BEER**

**Vinos**

**TERRAZA DE LOS ANDES**  
 Terraza Altos del Plata Malbec 1050  
 Terrazas Altos del Plata Merlot 1050  
 Terraza Altos del Plata Cab. Sauvignon 1050

**BODEGA 33° SUR**  
 Latitud 33° Malbec 1000  
 Latitud 33° Cabernet Sauvignon 1000  
 Latitud 33° Merlot 1000

**BODEGA LUIGI BOSCA**  
 Luigi Bosca Malbec - Familia Arizu 2300

**BODEGA FINCA FLICHMAN**  
 Gran Caballero Res. Malbec 1500  
 Misterio Malbec 1100

**BODEGA NORTON**  
 Cosecha Tardia 900

**BODEGA ETCHART**  
 Cafayate Malbec 1000  
 Cafayate Torrontes 1400

**VINOS BLANCOS**  
 Misterio Sweet Chardonay 1050  
 Sta Julia chenin dulce (lata 355 cc) 450