

# INFORME FINAL

Sistema de gestión gastronómica - lacarta

## **Alumnos**

- Masuero Luciano: [luchimasuero@gmail.com](mailto:luchimasuero@gmail.com)
- Taborda Andrés Maximiliano: [andresmaximilianotaborda@gmail.com](mailto:andresmaximilianotaborda@gmail.com)

**Director:** Mg. Lisandro Vrancken

Universidad Tecnológica Nacional – Facultad Regional Santa Fe – Noviembre 2019

## Contents

1. Introducción.....	2
2. Metodología.....	11
3. Tecnologías.....	15
4. Testing.....	19
5. Arquitectura.....	23
6. Seguridad.....	27
7. Funcionalidades y Manual de usuario.....	29
8. Gestión de riesgos.....	37
9. Estimación vs. Realidad.....	39
10. Conclusiones.....	42
11. Bibliografía.....	44
12. Anexo I - Glosario.....	46
13. Anexo II: Product Backlog.....	48
14. Anexo III: Gestión del desarrollo.....	52
15. Anexo IV: Historias de usuario.....	55
16. Anexo V: Manual de usuario.....	71

# 1. Introducción

En la actualidad, gran parte de los comercios gastronómicos llevan a cabo la gestión de sus actividades de forma manual y tradicional, perdiéndose así la posibilidad de obtener provecho en la utilización de sistemas de información, situación que alienta la generación de aplicaciones que permitan optimizar e incluir mejoras en el soporte a los procesos llevados a cabo. Es notable también, que en la gran mayoría de los casos que cuentan con tecnología de información (de ahora en más TI) aplicada, la gestión se realiza utilizando herramientas que no se ajustan a sus requerimientos y/o expectativas, o que fueron adaptadas a las mismas de forma pobre o incorrecta. Además, cabe destacar que a medida que crecen dichos comercios (en capacidad, clientes, recursos humanos, sucursales, etc.), la problemática se profundiza sin la asistencia de un sistema adecuado para la administración.

Si bien la incorporación de un sistema de gestión no se traduce necesaria y directamente en una mejora de la eficiencia de todos los procesos del negocio, sin duda alguna, aporta control y un conjunto de métricas que permiten a los administradores de los comercios identificar los puntos claves a mejorar y/o ajustar.

El trabajo realizado en el proyecto consistió en el diseño y desarrollo de un sistema destinado a brindar soporte a la gestión de comercios gastronómicos para que estos puedan controlar los pedidos realizados por sus clientes, confeccionar su menú en forma digital, brindar acceso masivo a su oferta gastronómica, obtener métricas precisas, prever la demanda y estimar su capacidad para condicionar así su estructura de forma más eficiente, permitiéndoles mejorar su desempeño y crecimiento de una forma más ordenada.

El objetivo principal que guió este proyecto fue entonces, el diseño y desarrollo de una aplicación (de uso gratuito) que permita a comercios gastronómicos realizar una gestión de los productos ofrecidos en sus menús, y dar tratamiento y monitoreo a las órdenes de pedidos realizadas por los comensales desde una aplicación móvil<sup>1</sup> o directamente por el panel administrador del sistema.

---

<sup>1</sup> Esta aplicación móvil fue desarrollada por otro equipo de desarrolladores. El diseño de la misma fue realizado coordinando entre ambos equipos de desarrollo para llegar a una integración satisfactoria.

## 1.1. Fundamentación

La forma mediante la cual los comercios gastronómicos gestionan la actividad de los comensales dentro de sus locales y brindan acceso a sus distintas ofertas gastronómicas no responde adecuadamente a las pretensiones de los nuevos usuarios, como por ejemplo, la alta accesibilidad.

Actualmente en la mayoría de los casos, los comensales no cuentan con la posibilidad de acceder de forma remota a la oferta gastronómica disponible en los distintos comercios, ni tampoco la facilidad de realizar pedidos a través de dispositivos móviles y evitar así, una espera en la atención del personal.

En forma análoga, dentro de la misma problemática, los comercios no tienen la posibilidad de monitorizar o controlar las órdenes en el preciso momento en el cual los comensales han tomado su decisión, agregando así un tiempo de espera innecesario en el servicio.

En este contexto, se evidencian carencias que se pretenden resolver mediante la implementación del sistema desarrollado, para permitir:

### A los comensales:

- Tener acceso remoto a la oferta gastronómica de los distintos restaurantes disponibles y sus respectivos precios.
- Realizar pedidos mediante la aplicación móvil una vez situados en el comercio<sup>2</sup>.
- Disponer de un registro de sus pedidos y/o gastos para llevar control de los mismos.
- Brindar feedback sobre atención y calidad de servicio.

### A los comercios:

- Brindar acceso masivo y de calidad a su oferta gastronómica.
- Monitorizar y administrar las órdenes de los comensales dentro de sus locales.
- Realizar un seguimiento de las distintas órdenes de los comensales.
- Optimizar y utilizar en forma más eficiente sus recursos humanos.

---

<sup>2</sup> Lugar físico donde se desarrolle la actividad gastronómica

- Prever la demanda y adaptar su estructura de acuerdo a la misma
- Servir a los comensales con mayor celeridad y obtener así un incremento en la rotación y disponibilidad de mesas.
- Obtener feedback de sus clientes.
- Llevar un control y gestión de caja directamente desde el sistema.

## *1.2. Ámbito y Población*

Se espera que el sistema de gestión desarrollado sea implementado en los comercios gastronómicos que deseen adquirirlo, comenzando inicialmente por un grupo predeterminado, ubicados en la ciudad de Santa Fe y zonas aledañas.

Por motivos de cercanía y costos se decidió inicialmente promocionar el sistema en un conjunto de restaurantes situados en zonas cercanas a la ciudad de Santa Fe y Santo Tomé, lugares que se destacan por tener una amplia y diversa oferta gastronómica.

A partir de datos obtenidos de la web oficial de la municipalidad de Santa Fe sabemos que la ciudad cuenta actualmente<sup>3</sup> con una oferta gastronómica de 100 restaurantes. El objetivo inicial era que la implementación del sistema cubra un 20% del total en un plazo de 6 meses luego de la salida al público.

Adicionalmente, se planteó como objetivo secundario buscar incluir a futuros emprendimientos que surjan luego del lanzamiento del sistema captándolos a medida que comienzan su paso en el rubro gastronómico.

## *1.3. Contexto de informatización gastronómico*

Para describir el estado actual de informatización en lo que respecta al ambiente gastronómico se planteó una segregación de los mismos en dos grupos que van desde los nuevos emprendedores hasta aquellos comercios ya establecidos.

Para aquellos comercios enmarcados bajo el primer grupo y caracterizados por su temprana iniciación en el mercado se puede evidenciar una precariedad notable en lo que respecta a sistemas. Además se pudo constatar que hoy en día, el acceso a alguno de estos es altamente costoso y complicado de financiar, lo que nos brinda una ventaja competitiva con este grupo debido a la oferta sin costo de nuestra aplicación.

---

<sup>3</sup> Relevamiento desde <http://www.santafeturismo.gov.ar/web/info-util/gastronomia/restoranes.php>

Con respecto al grupo que denota una continuidad en el tiempo de su emprendimiento y se encuentra establecido con algunos años de operación, pudimos verificar que la mayoría de estos cuenta con al menos un sistema de gestión, que tiende a ser el mismo en los distintos casos.

Una de las soluciones informáticas más utilizada a la que se hace referencia anteriormente es MaxiREST<sup>4</sup>. La misma representa una aplicación de gran envergadura y larga trayectoria en el rubro. Cuenta con una amplia variedad de características y funcionalidades que tienden a convertirlo en una herramienta compleja de administrar y utilizar para ciertos escenarios. Otro detalle a destacar del mismo es que conlleva un costo inicial considerable, tanto para adquirirlo como así también para capacitar al personal, lo que deja a una gran cantidad de emprendedores al margen de su utilización.

#### 1.4. Terminología

Si bien la terminología que existe en lo que respecta a un negocio gastronómico no es para nada compleja ni desconocida, creemos conveniente dejar una breve lista de términos que son utilizados en el día a día y una definición de los mismos. Corresponde aclarar que esta lista surge como el resultado del análisis y relevamiento de los productos de la competencia, como así también, de reuniones con usuarios de los mismos.

- Comensal: son aquellas personas que concurren a los comercios gastronómicos para hacer uso de sus servicios y consumir productos de los mismos.
- Mesa: corresponde al lugar físico ocupado por uno o varios comensales en el momento que consumen servicios dentro del comercio. Son ocupadas durante un lapso finito de tiempo.
- Carta: lista de productos ofrecidos por el comercio gastronómico.
- Mozo: persona encargada de atender a los comensales en las distintas solicitudes de productos.
- Adición: corresponde a una orden relacionada por uno o varios comensales y que consta de una lista de productos con sus respectivas cantidades.
- Comanda; impresión de la adición utilizando la comandera.

---

<sup>4</sup> Detalles sobre el sistema se pueden encontrar en <https://www.maxisistemas.com.ar/>

- Comandera: herramienta utilizada para generar una comanda y enviar a quienes se encarguen de su elaboración. Normalmente corresponde a un artefacto similar a una impresora pequeña que imprime en un papel la comanda.
- Cuenta: listado de productos ordenados por uno o varios comensales donde se puede ver los detalles de los mismos como así también el total a abonar por lo consumido.
- Apertura de caja: corresponde a la actividad que realiza el encargado de la caja a la hora de comenzar su turno. Consiste en dejar asentado el estado actual de la caja al momento de iniciar su gestión sobre la misma.
- Cierre de caja: es el equivalente a la apertura que se realiza a la hora de la finalización del turno del encargado.
- Rotación de mesa: indicador que refleja el tiempo promedio en el que una mesa es ocupada por comensales y liberada para ser utilizada por otros. Tiene a contemplar varios factores como tiempo de cocina, preparación de mesa, tamaño de mesa, etc... Ejemplo: un restaurante de comida italiana tendrá un índice de rotación de mesa más bajo que uno de comida rápida.
- Encargado: empleado encargado de la administración y el funcionamiento de un comercio gastronómico durante un turno o período de tiempo preestablecido.
- Cubiertos: indica la cantidad de comensales que se encuentran en una mesa en particular.

## 1.5. Relevamiento sobre la competencia

En esta sección se encuentran los detalles más importantes que pudimos relevar acerca de los competidores más directos del producto. Estos son:

- **MaxiRest** (<https://www.maxisistemas.com.ar/>)
- **Nucleo IT Gastronomico** (<http://www.nucleo-it.com.ar/Software>)
- **Mr Comanda** (<http://www.mrcomanda.com/>)
- **OASS** (<http://www.oass.net/>)

El relevamiento de información sobre la competencia fue llevado a cabo mediante diversos métodos, entre los cuales podemos destacar el acceso a versiones gratuitas de los sistemas, entrevistas informales a usuarios de estos, visualización de videos de ayuda sobre los mismos y hasta una reunión organizada con vendedores de uno de estos.

Con respecto al competidor MaxiRest, se pudo organizar una reunión directa con un proveedor del mismo, en la que se tuvo que simular que se abriría un comercio gastronómico en la zona. El sistema es el más usado en el rubro gastronómico y uno de los que más trayectoria tiene.

Con respecto a Núcleo IT se pudo llevar a cabo una entrevista con un usuario administrador de dicho sistema donde pudimos obtener algunos detalles del mismo.

En cuanto a Mr Comanda y OASS, la información recolectada y las conclusiones obtenidas son resultado de una investigación utilizando internet como medio principal.

### 1.5.1. Entrevista con proveedor de MaxiRest

Fue sin dudas la reunión más beneficiosa y esclarecedora de todas. En esta detectamos varias cuestiones, que luego fueron corroboradas, como la nomenclatura y dialecto usado en el negocio, las funciones más importantes y/o necesarias en el día a día, la forma de darle tratamiento a ciertos aspectos que suelen escaparse a la hora de encarar algunos comportamientos o escenarios habituales, etc.

La característica más determinante de MaxiRest es su gran envergadura como así también la importante cantidad de módulos o funcionalidades que ofrece, cuestión que en ciertas ocasiones lo hacen complicado de usar y/o configurar.



Otra de las características a destacar de este sistema, es que aparenta ser costoso de implementar y mantener, lo que hace que no sea adquirido por la mayoría de emprendimientos jóvenes hasta tanto logren establecerse (a menos de que cuenten con una inversión inicial considerable).

### 1.5.2. Entrevista con usuario de Núcleo IT

Como principal característica que detectamos en este, es que a simple vista pareciera ser un sistema simple y funcional que ofrece lo necesario para poder llevar a cabo el control y gestión de pedidos dentro de un comercio gastronómico. De todas formas, la usabilidad del mismo no es demasiado intuitiva y se necesita una cierta capacitación o prueba y error para ciertas ocasiones.

Uno de los defectos más importantes es que el sistema era del estilo aplicación de escritorio, lo que no permitía al usuario poder realizar una gestión mediante internet desde su casa o tener la información resguardada en otro lugar que no sea su pc.

En esta revisión, confirmamos que utilizan la misma nomenclatura base que MaxiRest y las funcionalidades en su gran mayoría se asemejan (las más básicas y necesarias al menos).

### 1.5.3. Cuadro comparativo

A continuación, se puede encontrar un cuadro comparativo que llevamos a cabo entre ciertos aspectos que nos interesan y cómo se podían describir en cada uno de los sistemas.

Aspecto	MaxiRest	Núcleo IT Gastronómico	Mr Comanda	OASS
Integración Mobile	<b>MaxiPad:</b> ofrecen la posibilidad de que los mozos carguen los pedidos desde su celular o dispositivo móvil.	No	<b>MRC Express Tablet</b> (similar a maxi rest)	Si: Los mozos cargan pedidos desde celular o tablet.
Accesibilidad Web	<b>MaxiWeb:</b> Ver/Controlar la actividad del local desde lejos	No	Módulo centralizador	Si (Monitoreo online)
Delivery	<b>MaxiDeli</b>	Si	<b>My-delivery</b>	Si
Reservas	<b>MaxiBook</b>	Pareciera no ofrecer módulo de	Solo las versiones más	Pareciera no ofrecer módulo de reservas

		reservas	caras	
Cupones	-	-	-	-
Comandas de cocina	<b>MaxiKDS:</b> monitor en la cocina con los pedidos -Impresión de "tickets" con impresora fiscal	Si	Si	Electrónicas (No se encontró la opción de imprimirlas)
Tickets Fiscales	Si	Si	Si	Si
Costo	-	-	\$1500 (mensual)	U\$S400 (Versión Full)
Forma de conseguirlo	Distribuidores zonales	Distribuidores	Internet, Proveedores	Internet
Versión Free	No (Se puede solicitar una demostración presencial)	No	Si	No
Vista Mesas como plano	Si	Si	Si	Si
Gestión de stock	Si	Si	-	-
Proveedores	Si	Si	-	-
Llamadores inalámbricos	Si	-	-	-
Capacitaciones	Si	-	-	-
Estadísticas	Si (módulo extra)	Si	-	Si
Gestión de recetas	-	Si	-	-
Gestión de personal	-	Si	Si	-

*Tabla 1: Cuadro comparativo entre features/producto*

## 1.6. Organización y estructura del informe

Habiendo descrito la temática principal y objetivo de este proyecto, en las siguientes secciones del presente informe, se describe:

- **Metodología:** descripción de la metodología con la que este proyecto fue abordado y las etapas principales del mismo.
- **Tecnologías:** lista de herramientas y tecnologías usadas para el desarrollo y gestión del proyecto en cuestión.
- **Testing:** descripción de la estrategia de testing utilizada.
- **Arquitectura:** descripción de la arquitectura diseñada e implementada para la aplicación.
- **Seguridad:** detalle de los mecanismos de seguridad contemplados en los distintos ámbitos de la aplicación.
- **Funcionalidades y Manual de usuario:** lista de funcionalidades más destacadas de la aplicación junto con imágenes que permiten una descripción gráfica.
- **Gestión de riesgos:** detalle de la gestión de riesgos propuesta y utilización de la misma en el desarrollo del proyecto
- **Estimación vs Realidad:** descripción del resultado esperado y el obtenido en la finalización del desarrollo junto con las cuestiones que creemos fueron de mayor relevancia en el mismo.
- **Conclusiones:** detalle de conclusiones acerca de lo que fuimos aprendiendo y relevando a lo largo del proyecto.
- **Anexos:** lista de anexos que creemos útiles y necesarios para una mejor comprensión de lo realizado.

## 2. Metodología

Dada la configuración de los integrantes del proyecto, como así también la naturaleza del mismo, optamos por la aplicación de diversas actividades de distintas metodologías ágiles, en su gran mayoría provenientes de las conocidas bajo el nombre de Extreme Programming (de ahora en más XP) y Scrum Programming (de ahora en más Scrum). En pocas palabras, la forma de proceder para el diseño y desarrollo de la aplicación, como así también la gestión del proyecto, fue utilizando un híbrido entre estas dos metodologías, seleccionando los conceptos que mejor se ajustaban de ambas a nuestro contexto y conocimientos, e integrándolos para sacarle el máximo provecho.

La elección de optar por el uso de prácticas disponibles en ambas metodologías se debió a varios factores entre los cuales podemos destacar, por un lado, la naturaleza de requerimientos cambiantes, una falta explícita de plazo de entrega del software y la necesidad imperiosa de tener incrementos funcionales de la aplicación y, por otro lado, favorecer prácticas que se centren en los conocimientos y destrezas de los individuos involucrados como así también la motivación de los mismos.

A continuación, se detallarán los conceptos y actividades más importantes que fuimos utilizando a lo largo de este proyecto, segregados por metodología.

### 2.1. *Extreme Programming*

De esta metodología vimos conveniente la aplicación del concepto de Programación en parejas, para poder hacer frente a la falta de conocimientos sobre tecnologías y herramientas relacionadas a la programación web de uno de los integrantes.

Utilizando esta práctica, se realizó una fase inicial de programación en conjunto, que permitió al integrante del equipo adquirir las habilidades requeridas para poder continuar en una segunda parte del desarrollo de forma más independiente y organizada.

Además, el desarrollo de a pares nos permitió llegar de forma implícita, mediante la aplicación diaria de la misma, a un consenso en la forma en la que desarrollamos y gestionamos distintas actividades del proyecto.

## 2.2. Scrum Programming

Esta metodología fue la que más se destacó en cuanto a conceptos utilizados, primero debido a la experiencia que se cuenta en aplicarla y segundo a la gran utilidad de los mismos en proyectos de este estilo. Algunos de estos fueron:

- **Daily Scrum:** se realizaron reuniones periódicas con los product owners donde se llevaron a cabo las pertinentes revisiones y se definieron cuestiones de diversa índole acerca del proyecto. Este tipo de concepto no fue aplicado al pie de la letra como indica la metodología en su estado puro, sino que fue alterada la periodicidad de estas reuniones como así también su duración dado de que nos resultaron más útiles de esta forma.
- **Sprint - Sprint Backlog:** se decidió trabajar en conjuntos de funcionalidades (Sprint Backlog) organizadas y agrupadas bajo Sprints para poder obtener resultados incrementales de la aplicación a lo largo del proyecto que nos permitieran medir la evolución del mismo como así también, corregir el rumbo en casos donde lo requería.
- **Product backlog:** se decidió contar con un Product backlog donde teníamos el listado de funcionalidades que se requerían de la aplicación, con su respectiva prioridad definida por los product owners, para que puedan ir siendo incluidas en los distintos Sprints que se encaraban.

## 2.3. Etapas del proyecto

Se decidió encarar el proyecto en una serie de etapas que pudimos identificar en base a las actividades que fuimos realizando y que no necesariamente ocurrieron de forma secuencial y ordenada en el tiempo.

### 2.3.1. Etapa 1: Investigación de campo.

En esta etapa se realizó una búsqueda de aplicaciones con características similares, que existían en el mercado local, para estudiar sus funcionalidades, fortalezas y debilidades.

Además, se analizaron las diferentes opciones utilizadas por algunos de los comercios para confeccionar y gestionar sus menús, como así también sus formatos de presentación al público, en busca de llegar a un diseño que permita de una forma rápida y sencilla volcarlos en nuestra aplicación sin restricciones o al menos con la menor cantidad de estas.

El resultado de esta etapa fue el perfeccionamiento de nuestros conocimientos acerca del proceso de negocio involucrado, como así también un relevamiento más fino y tangible de información sobre el desempeño de los distintos actores que participan del mismo. Medios para lograr esto fueron reuniones que se concretaron con uno de los proveedores de la competencia como así también con gente encargada de distintos restaurantes.

A su vez elaboramos una sección en este informe con los datos relevados en las reuniones e investigaciones sobre las competencias existentes más directas. Se puede encontrar en la sección “1.5 [Relevamiento sobre la competencia](#)” y allí destacamos los detalles de interés de cada una estas como así también sus fortalezas y debilidades.

### 2.3.2. Etapa 2: Análisis y Requerimientos.

En esta etapa se trabajó en definir y detallar las funcionalidades y requerimientos necesarios para el desarrollo e implementación del sistema.

El resultado de esta etapa fue el Product Backlog del sistema que fue modificado durante todo el proyecto y adaptado a las necesidades cambiantes del mismo.

Se puede ver en el “[Anexo II](#)”, un documento que hace referencia al product backlog desde comienzos hasta fines del proyecto donde se detallan los ítems que se encontraron y fueron agregando como así también sus estados.

### 2.3.3. Etapa 3: Análisis y Diseño de Arquitectura.

Teniendo en cuenta la naturaleza del sistema y los requerimientos obtenidos en la Etapa 2 (Análisis y Requerimientos), se realizó el diseño y definición de la arquitectura más adecuada a utilizar.

Para esto elaboramos un documento donde se pueden encontrar conceptos básicos de la arquitectura deseada, junto con sus restricciones y premisas más importantes, como así también un diagrama que la describe en alto nivel. Los detalles de esta son descritos en la sección “5. [Arquitectura](#)”.

#### 2.3.4. Etapa 4: Desarrollo.

Siguiendo los lineamientos de Scrum, dividimos los requerimientos detallados en el Product Backlog en diversas tareas las cuales fueron estimadas y agrupadas bajo Sprints que definieron hitos y entregables de la aplicación.

En esta etapa también se generaron distintos casos de pruebas automáticos que contemplaron alguno de los flujos más importantes y críticos del sistema.

En cuestiones relacionadas al desarrollo se establecieron los siguientes criterios:

- El tiempo promedio aproximado de cada Sprint fue de 3 semanas.
- En la primera semana (o días) de cada Sprint realizábamos una planificación (Sprint planning) revisando el Product Backlog y decidiendo qué tareas se incluían en base a la prioridad, urgencias de las mismas y tiempo disponible del Sprint (Sprint Backlog). Durante este tiempo, además, llevábamos a cabo tareas de documentación, tanto del proyecto como del sistema.
- Al final de cada Sprint realizábamos una reunión (Sprint Review) donde hacíamos una pequeña demostración de lo implementado en el mismo y recolectábamos posibles nuevas tareas, como así también errores o cambios, que eran incluidas dentro del Product Backlog.

La principal herramienta para la administración de esta etapa fue Waffle.IO, la cual era utilizada como nuestro Scrum Board y donde llevábamos un control de las tareas, los sprints y los estados entre los mismos como así también dejábamos detalles y/o comentarios sobre diversos aspectos. En el "[Anexo III](#)" se puede ver un detalle de esta herramienta y cómo la utilizamos.

## 3. Tecnologías

Se utilizó una extensa y diversa cantidad de tecnologías para poder concretar el proyecto, tanto para el desarrollo como así también para la gestión y organización del mismo.

La selección de las tecnologías a emplear estuvo guiada por las siguientes premisas:

- Software libre y gratuito.
- Documentación detallada y accesible vía web.
- Uso probado y masivo (comunidad de desarrolladores que la usan de tamaño considerable e interacción frecuente).
- Simplicidad en aprendizaje.
- Accesibilidad desde múltiples plataformas y de forma online.

A continuación, se detallan y se justifica la elección de las mismas, categorizadas según el área de utilización.

### 3.1. Comunicación

Un factor clave en el desarrollo del proyecto era atacar la falta de disponibilidad horaria de los integrantes del equipo para la realización del proyecto en forma conjunta y la fuerte necesidad de mantener una comunicación fluida y constante, fundamental para la utilización de toda metodología ágil. Para hacer frente a estas problemáticas se utilizaron las siguientes tecnologías:

- Email ([Gmail](#) principalmente).
- Comunicación por canales de chat y equipos: [Slack](#)<sup>5</sup>
- Video llamadas con [Appear.In](#)<sup>6</sup>

---

<sup>5</sup> <https://slack.com/>

<sup>6</sup> <https://appear.in/>



### 3.2. Gestión de Proyecto

Para la ejecución de algunas actividades relacionadas a la gestión del proyecto se necesitaba de un conjunto de herramientas que permitiera trabajar en equipo de forma remota y tener control total sobre las distintas versiones que se iban construyendo.

Además, era sumamente importante contar con una herramienta de gestión de proyectos que nos permitiera tener una visión global de las tareas y el estado de las mismas.

La elección final de estas tecnologías fue la siguiente:

- Como sistema de control de versiones utilizamos [Git](#)<sup>7</sup> y la herramienta de colaboración [GitHub](#)<sup>8</sup>.
- Como herramienta de gestión de proyecto se utilizó [Waffle.io](#)<sup>9</sup> integrándolo directamente a GitHub.
- Como herramienta para el seguimiento de tiempo en tareas se utilizó [Hamster indicator](#)<sup>10</sup>.

### 3.3. Documentación

Las herramientas que se utilizaron para la documentación del proyecto fueron las siguientes:

- Para la confección de documentos de texto: Google Docs.
- Documentación del software y entorno de desarrollo: Wiki de GitHub.

Se eligieron estas herramientas por su simpleza y capacidad de edición online, lo cual permite construir la documentación en forma compartida accediendo a los cambios en forma instantánea y desde cualquier lugar. Otra importante característica fue el resguardo de la información ante cualquier pérdida material.

---

<sup>7</sup> GIT: <https://git-scm.com/>

<sup>8</sup> GITHub: <https://github.com/>

<sup>9</sup> WaffleIO: <https://waffle.io/>

<sup>10</sup> Hamster Indicator: <https://projecthamster.wordpress.com/about/>

### 3.4. Desarrollo

#### 3.4.1. Base de datos

Para la elección del motor de base de datos, se tuvieron en cuenta aquellas opciones que cumplan con los siguientes requisitos: gratuito, relacional, experiencia sobre la misma por parte de los integrantes y adecuado para la envergadura del proyecto. De entre las opciones disponibles, optamos por la utilización del motor [MySQL](https://www.mysql.com/)<sup>11</sup>, que cumplía con todos los requisitos citados anteriormente.

#### 3.4.2. Aplicación

Una de las premisas que debía satisfacer el sistema desarrollado era la de respetar los conceptos considerados en el estilo [Single Page Application](http://en.wikipedia.org/wiki/Single-page_application)<sup>12</sup>. A grandes rasgos en este estilo, la aplicación web debe mostrar el contenido sin necesidad de recargar la página adaptándose de forma dinámica para reflejar las acciones realizadas por los usuarios.

Para cumplir con esta premisa optamos por la utilización del framework [AngularJS](https://angularjs.org/)<sup>13</sup> debido a su sencilla extensibilidad, facilidad de integración con otras librerías, amplia documentación, rapidez de desarrollo y vasta utilización que lo convierte en una herramienta muy popular hoy en día (lo cual representaba además, una importante oportunidad de adquirir conocimientos).

A su vez toda esta estructura fue montada sobre 3 simples lenguajes con los que llegamos a construir el aplicativo cliente: HTML5, CSS3 y JavaScript. Como framework de CSS/HTML utilizamos [Bootstrap](http://getbootstrap.com/)<sup>14</sup>.

#### 3.4.3. Procesamiento y obtención de la información

Debido al requisito de cumplir con el estilo SPA detallado anteriormente fue necesario contar con herramientas para obtener y procesar la información de las acciones de los usuarios en tiempo real<sup>15</sup>. Para esto se optó por la utilización del servicio [Pusher](https://pusher.com/)<sup>16</sup>, el cual a través de una API basada en eventos nos permitió mantener la información de la aplicación constantemente actualizada.

---

<sup>11</sup> MySQL: <https://www.mysql.com/>

<sup>12</sup> SPA: [http://en.wikipedia.org/wiki/Single-page\\_application](http://en.wikipedia.org/wiki/Single-page_application)

<sup>13</sup> AngularJS: <https://angularjs.org/>

<sup>14</sup> Bootstrap: <http://getbootstrap.com/>

<sup>15</sup> Real Time: Real time, [http://en.wikipedia.org/wiki/Real-time\\_web](http://en.wikipedia.org/wiki/Real-time_web)

<sup>16</sup> Pusher: <https://pusher.com/>

La tecnología detrás de este servicio de terceros es conocida como Websockets<sup>17</sup> y representa un concepto muy utilizado e importante de la actualidad teniendo en cuenta como las aplicaciones se han transformado en grandes administradores de información, la cual fluye en varios sentidos y formas.

#### 3.4.4. Testing

Decidimos incluir en los flujos más importantes o críticos del sistema el uso de pruebas funcionales utilizando la metodología [E2E](#)<sup>18</sup> y como tecnología la herramienta [Protractor](#)<sup>19</sup>.

#### 3.4.5. Entorno de desarrollo

Para realizar el desarrollo del proyecto contamos con computadoras cuyo sistema operativo base era Ubuntu y donde corríamos máquinas virtuales para las diferentes componentes del sistema bajo la herramienta Vagrant<sup>20</sup>.

Encuanto al IDE de desarrollo optamos por el uso de Netbeans y en algunas ocasiones, dependiendo del lenguaje, Sublime Text.

---

<sup>17</sup> WebSocket: <https://es.wikipedia.org/wiki/WebSocket>

<sup>18</sup> E2E: <https://docs.angularjs.org/guide/e2e-testing>

<sup>19</sup> Protractor: <http://angular.github.io/protractor/#/>

<sup>20</sup> Vagrant: <https://www.vagrantup.com/>

## 4. Testing

En la estrategia diseñada para realizar las pruebas sobre la aplicación desarrollada debemos diferenciar 2 grupos:

- Por un lado, las pruebas automáticas, implementadas a nivel código y mediante la herramienta Protractor. Estas tienen las características de seguir la metodología End-to-end, en la cual los diferentes casos simulan la interacción de un usuario real con el sistema, validando por supuesto, los resultados obtenidos de las acciones ejecutadas.
- Por otro lado, las pruebas manuales, donde también es posible diferenciar entre las llevadas a cabo por los desarrolladores basándose en las historias de usuarios y sus casos de pruebas, como así también las integrales a cargo de los product owners en las entregas incrementales de cada uno de los sprints.

A continuación, se van a dar más detalles sobre estos dos enfoques y cómo fueron llevados a cabo en el proyecto.

### 4.1. Pruebas automatizadas

Inicialmente se decidió hacer testing automático para cada nueva funcionalidad que era desarrollada en los diferentes issue del sprint. Es decir, por cada nueva funcionalidad perteneciente a las historias de usuario, se implementaba una prueba end-to-end que se encargaba de validar la misma. Estas eran ejecutadas a demanda, al momento de realizar la integración de las porciones desarrolladas con el resto de la aplicación.

Al correr del proyecto, el re-trabajo que nos causó esta decisión se comenzó a volver importante dado que cada nueva funcionalidad desarrollada terminaba repercutiendo en otras ya probadas, y cambios mínimos hacían que los casos de pruebas tengan que ser reprogramados.

Debido a esta complicación en el mantenimiento de escenarios de pruebas y el considerable tiempo que nos insumían, se decidió generar solamente casos para los caminos críticos en cada incremento producido en los distintos Sprints. De esta forma el testing y su mantenimiento, fue más razonable y nos permitió estar más cerca de cumplir los tiempos de entregas pactados en su momento.

#### 4.1.1. Ejemplo de pruebas end-to-end

A continuación se procede a demostrar, mediante un ejemplo, el espíritu de las pruebas end-to-end, utilizando un seudocódigo para que se tenga una noción más representativa de las mismas.

Como se detalló anteriormente, la idea de esta metodología es realizar una prueba simulando la interacción de un usuario, y recorriendo como tal, el flujo completo de una acción desde un punto del sistema como lo es el “frontend”, hacia otro como lo es el “backend”. De esta forma uno puede asegurarse que todos los componentes del sistema funcionan correctamente en los diferentes escenarios que se desee.

Ejemplo de un flujo: “Autenticación de usuario”

- Accedo a la página inicial de la aplicación
- Presionó botón “Entrar”
- Ingreso datos de cuenta administrador: email y contraseña
- Presionó botón “Aceptar”
- Verifico que me encuentro en la página “Administración de mesas y pedidos”

Cada uno de estos pasos representa una instrucción de código que brinda la herramienta “Protractor” y que nos permite ir interactuando con el sistema como si fuéramos un usuario en particular.

## 4.2. Pruebas manuales

En las pruebas del tipo manual se pueden diferenciar 2 tipos:

- Por un lado, tenemos aquellas llevadas a cabo en forma cruzada por los desarrolladores del proyecto luego de implementar cada funcionalidad. Es decir, cuando un desarrollador del equipo terminaba una tarea, está pasaba a la etapa de testing, en donde el otro desarrollador, siguiendo la historia de usuario y su caso de prueba, validaba manualmente de que se encuentre todo de acuerdo a lo esperado. Cabe destacar que estas se comenzaron a hacer de forma cruzada cuando la programación en pareja dejó de existir, antes eran ejecutadas por ambos.
- Por el otro lado, tenemos las integrales realizadas por los product owners en los ambientes de prueba donde se desplegaban los incrementos en las finalizaciones de cada Sprint. Estas eran las validaciones más importantes internamente debido a que hacían visible el trabajo realizado y la evolución de la aplicación.

En ambos casos, por cada problema o defecto que se encontraban, se procedía a reportar mediante la creación de “Issues” en Waffle.IO (ver [Anexo III](#)). Cada uno de estos, tenía un significado y era tratado de una forma diferente, detallada a continuación:

- **Issues reportados en pruebas de desarrolladores:** estos se creaban bajo la etiqueta “Error” y eran atendidos en la brevedad por el desarrollador del Issue bajo prueba. No requerían una replanificación ni cambios en el Backlog.
- **Issues reportados en pruebas de product owners:** estos se creaban bajo la etiqueta de “Bug” y debido a que surgían luego de la finalización de los Sprints que incluían las diferentes funcionalidades, requerían una modificación del Backlog necesitando incluirlos en los Sprints siguientes para poder atacarlos y darles el tratamiento correspondiente.

### 4.3. Pruebas sobre REST-API

Además del esquema de pruebas detallado anteriormente es necesario aclarar que también desde la aplicación REST-API se desarrolló un set de pruebas automáticas que se encargan de validar los diferentes endpoints expuestos.

En cada ejecución de estas pruebas se realizan los siguientes pasos:

- Creación de base de datos: al comenzar la ejecución se lleva a cabo la creación del esquema de base de datos junto con las tablas mediante los comandos de migración.
- Seed de datos: el siguiente paso es la ejecución de lo que se conoce como seed de datos, donde se inserta un conjunto de datos mínimos necesarios para realizar las pruebas de los diferentes endpoints.
- Ejecución de test cases: se comienzan a ejecutar los diferentes casos creados para los recursos y sus respectivos endpoints, realizando las validaciones pertinentes. Cada caso de prueba ejecuta una solicitud (Request POST/GET/PUT/PATCH) hacia el API y analiza el Response validando el resultado esperado.
- Borrado de base de datos: la etapa final consiste en la eliminación de la base de datos con el fin de dejar el ambiente preparado para una próxima ejecución.

Mediante estas y las anteriormente nombradas lográbamos asegurar un correcto funcionamiento de los sub-sistemas por sí mismos y el sistema en su totalidad.

## 5. Arquitectura

La siguiente sección busca detallar los aspectos más importantes de la arquitectura utilizada en la aplicación como así también los motivos del uso de las mismas.

Luego, por medio de un diagrama junto con las pertinentes descripciones se brinda un detalle que explicará las ventajas y desventajas del modelo utilizado.

### 5.1. *Objetivos y restricciones de la arquitectura*

La arquitectura diseñada fue basada en una serie de requisitos, restricciones y objetivos que fueron definidos y considerados necesarios para el éxito de la aplicación como también así su factibilidad. A continuación, se listan los mismos:

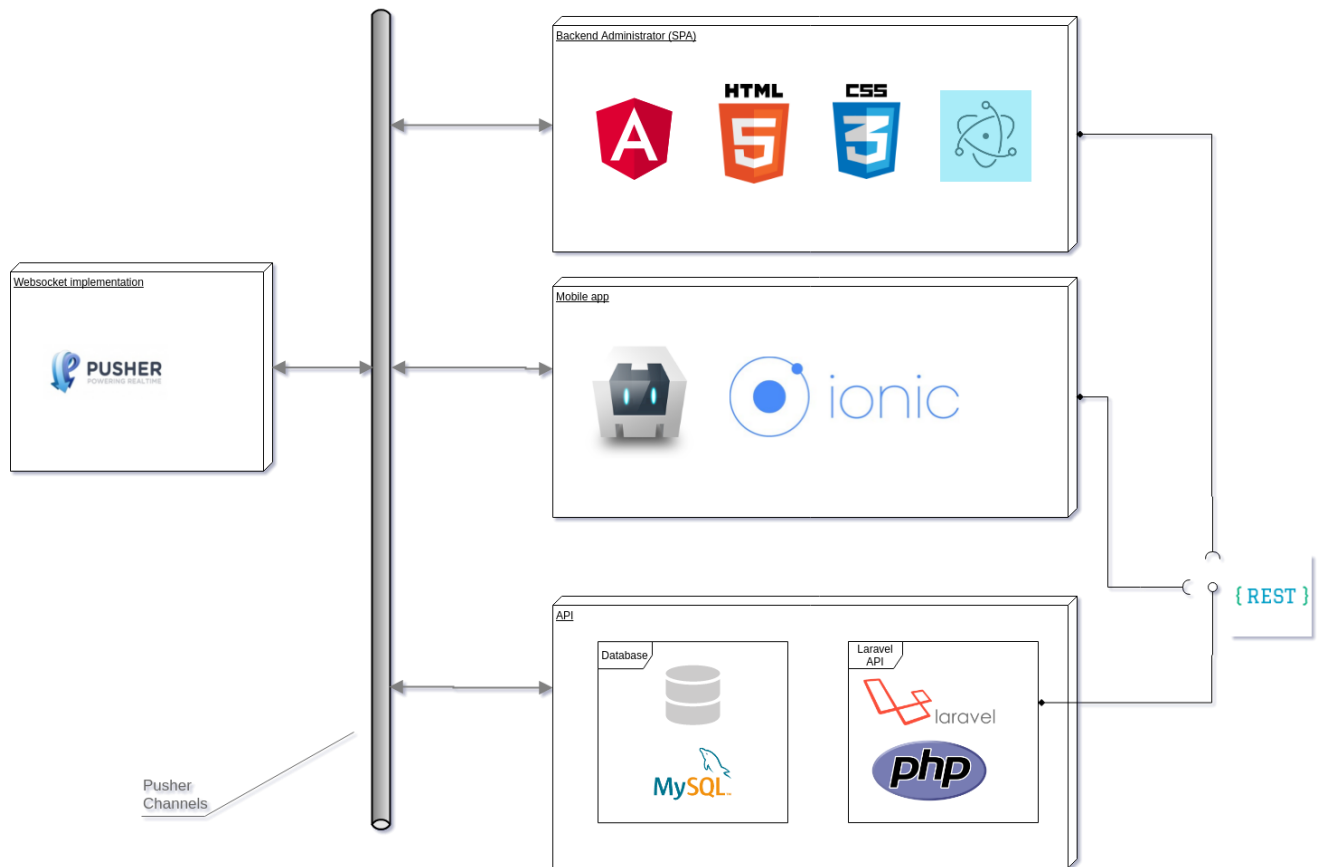
- La aplicación debía ofrecer una altísima accesibilidad dado que es necesario que sea utilizada desde cualquier dispositivo ya sea móvil o no.
- Las herramientas utilizadas en la construcción de la aplicación debían de ser de índole libre y gratuita para evitar caer en gastos innecesarios.
- La organización de la aplicación como sus componentes debían ser de naturaleza distribuida para permitir escalar de forma sencilla en caso de necesitarlo.
- Se debían usar tecnologías que permitan gestionar la información en tiempo real.
- Los requerimientos del Hardware no debían ser determinantes y la solución desarrollada tenía que ser compatible con mayoría de las configuraciones más populares.
- La performance de la aplicación era un atributo importante debido a la deseada rapidez de respuesta de las funcionalidades expuestas por la aplicación.

### 5.2. *Diseño de la Arquitectura de la Solución*

A continuación se encuentra una imagen que describe a alto nivel la arquitectura diseñada e implementada para el ecosistema de aplicaciones, detallando las tecnologías



utilizadas en cada una y los canales por los cuales se comunican entre sí.



*Imagen 1: Arquitectura de la solución*

El objetivo en sí de este proyecto fue el diseño y desarrollo de la aplicación **Backend Administrator** y la integración con los demás sistemas. El desarrollo final constituye una **SPA** (Single Page Application) construida con la utilización de **AngularJS**, **HTML5** y **CSS3**. Luego utilizando la herramienta **Electron-Builder**<sup>21</sup> generamos un instalador ejecutable para ser utilizado en los despliegues a usuarios finales en caso de ser necesario (aplicación de escritorio).

Esta SPA hace uso de una **REST API** que ofrece un conjunto de accesos a distintos recursos (endpoints), utilizados para hacerse de información necesaria para su correcto funcionamiento. Esta API fue desarrollada utilizando el framework **Laravel**<sup>22</sup> y el motor de base de datos **MySQL**.

<sup>21</sup> Electron-Builder: <https://www.electron.build/>

<sup>22</sup> Laravel: <https://laravel.com/>

A su vez, existe una mobile APP que también interactúa con la **REST API**. Esta aplicación fue desarrollada utilizando **Apache Cordova**<sup>23</sup> y la herramienta **Ionic**<sup>24</sup>.

Todas estas aplicaciones hacen uso además de la tecnología conocida como **Websockets**. En la implementación actual se cuenta con un servicio gratuito de notificaciones provisto por **Pusher**, con una oferta limitada de canales, suscripciones y publicaciones. Las distintas aplicaciones se suscriben a un canal (1 canal por resto) donde comienzan a publicar y recibir mensajes llegando a simular una interacción en tiempo real entre todas.

Esta arquitectura nos permitió llevar a cabo un desarrollo en paralelo e independiente entre los distintos grupos encargados de los subsistemas. Además, nos dio la posibilidad de lograr uno de los principales objetivos deseados como es la performance, dado que contamos con una configuración distribuida que permite ser escalada fácilmente y replicada sin costos elevados.

### *5.3. Caso de uso de implementación de Websockets*

A continuación, se describe un ejemplo de cómo se implementó la tecnología de Websockets en el proyecto, destacando el papel de cada una de las aplicaciones y cómo interactúan entre sí mediante los canales de comunicación.

Cuando una instancia de **Backend Administrator** se ejecuta, por ejemplo, mediante la autenticación de un administrador de restó, se crea un canal específico para la misma (o se reutiliza uno creado previamente) y además se suscribe a este.

Luego, cuando una instancia de **Mobile APP** se ejecuta, por ejemplo, mediante la apertura de la misma desde un celular de un comensal que se autentica y abre una cuenta en una mesa del restó, la misma se suscribe al canal que el **Backend Administrator** ha creado (como se describió anteriormente).

Una vez decidida su orden, el comensal genera un pedido desde la **Mobile APP**, lo que termina siendo un mensaje enviado a la **REST Api** en uno de sus endpoints ofrecidos. Luego

---

<sup>23</sup> Apache cordova: <https://cordova.apache.org/>

<sup>24</sup> Ionic Framework: <https://ionicframework.com/>

desde la **REST Api** se llevan a cabo las acciones necesarias y se termina generando un evento que publica en el canal del restó dicha acción para que los involucrados lo procesen.

Este mensaje es recibido por el **Backend Administrator** y se comienza a dar curso al pedido realizado desde a **Mobile APP** ejecutada por el comensal. Tener en cuenta que mediante el uso de esta tecnología no es la aplicación quien busca el mensaje en el servidor, sino que el mensaje le llega a la misma mientras se encuentra corriendo en el lado cliente. Esto evita sobrecargas en servidores, que se tendría si se utilizará una estrategia de polling<sup>25</sup>.

Este ejemplo corresponde a uno de los tantos escenarios donde se hace un uso claro de la tecnología de Websockets.

Como se mencionó anteriormente hoy se hace uso de un servicio provisto por un software de terceros (Pusher), el cual es gratuito pero limitado. La ventaja de la implementación actual es que el reemplazo del proveedor por un servicio propio u otro proveedor puede ser llevada a cabo sin impedimentos en un futuro.

---

<sup>25</sup> Polling: <https://www.ably.io/concepts/long-polling>

## 6. Seguridad

En la siguiente sección se detallan las políticas de seguridad definidas para asegurar el correcto resguardo de la información y evitar ataques o dar respuesta a vulnerabilidades conocidas que permitan acciones maliciosas contra la aplicación. Para poder describirlas correctamente se las segregó en diferentes categorías a continuación listadas.

### 6.1. Accesos a Servidor y Aplicación

La aplicación está alojada en un servidor virtual privado (VPS<sup>26</sup>) provisto por DigitalOcean<sup>27</sup>. El sistema operativo del servidor es Ubuntu Server y la única forma de acceder al mismo es utilizando el protocolo SSH, con previa configuración de la SSH-KEY en el servidor. El acceso vía usuario y contraseña está deshabilitado para reforzar la seguridad.

Adicionalmente, el servidor cuenta con un firewall (UFW<sup>28</sup>) configurado para inhabilitar el acceso desde el exterior a los servicios que no son necesarios para el servicio de la web. Dicho de otro modo, sólo está habilitado el servicio web de nginx (https) y openSSH.

La comunicación entre las distintas aplicaciones se realiza mediante el protocolo HTTPS. Por medio de este, la información viaja encriptada de extremo a extremo. El certificado instalado que hace posible la encriptación fue generado por el Certificate Authority Let's Encrypt<sup>29</sup>.

Las conexiones remotas a la base de datos están deshabilitadas. Solo puede conectarse a la base de datos la aplicación que reside en el servidor accediendo de forma local a esta.

---

<sup>26</sup> VPS: Virtual Private Server - [https://es.wikipedia.org/wiki/Servidor\\_virtual\\_privado](https://es.wikipedia.org/wiki/Servidor_virtual_privado)

<sup>27</sup> DigitalOcean: <https://www.digitalocean.com/>

<sup>28</sup> UFW - Uncomplicated Firewall: <https://help.ubuntu.com/community/UFW>

<sup>29</sup> Let's Encrypt: <https://letsencrypt.org/>

## 6.2. Autenticación

Para proveer de autenticación tanto en la aplicación Mobile como el Backend Administrator se utiliza el protocolo OAuth2.0<sup>30</sup>. La única forma de llevar a cabo la misma es mediante el consumo de endpoints expuestos por la API REST.

Además, se encuentra configurado un tiempo de expiración para las sesiones. De esta forma se fuerza la renovación de las mismas ante un lapso de inactividad, agregando seguridad al sitio.

## 6.3. Monitoreo

Se tiene configurado un servicio de monitoreo de la aplicación provisto por New Relic<sup>31</sup> el cual nos otorga diversos tipos de métricas útiles (días de mayor uso, tiempos de respuesta, tasa de errores, consumos de CPU, uso de memoria, etc.), como así también caídas de servicios.

Este tipo de métricas nos permiten realizar mejoras y detectar anomalías para brindar un mejor servicio a los usuarios.

Se cuenta con un desarrollo de logging<sup>32</sup> de fallas almacenados en la base de datos que nos brinda identificación y seguimiento de errores como así también notificación de las mismas vía email.

## 6.4. Datos

Se tiene configurada una política de backups de datos y archivos que se ejecuta semanalmente y resguarda tanto el estado actual de la base de datos como los archivos generados o accesibles desde la aplicación (ej.: imágenes, documentos PDF, etc.).

Las contraseñas son almacenadas en base de datos con algoritmos de encriptación que evitan la posibilidad de ser descifradas, aun en caso de tener acceso a la misma.

Los datos provistos por el usuario siempre pasan por una etapa en la que se los “sanitiza” y válida para evitar que puedan llegar a ejecutar acciones maliciosas (ejecutar código no deseado - XSS<sup>33</sup>).

---

<sup>30</sup> OAuth2.0: <https://oauth.net/2/>

<sup>31</sup> New Relic: <https://newrelic.com/>

<sup>32</sup> Log: [https://es.wikipedia.org/wiki/Log\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Log_(inform%C3%A1tica))

<sup>33</sup> XSS - Cross-site scripting: [https://es.wikipedia.org/wiki/Cross-site\\_scripting](https://es.wikipedia.org/wiki/Cross-site_scripting)

## 7. Funcionalidades y Manual de usuario

Para detallar las capacidades expuestas por el sistema y cómo funciona el mismo, debemos centrarnos en los 2 posibles roles de usuarios que existen.

Por un lado, los administradores de los restaurantes, que están a cargo de la gestión del comercio haciendo uso directamente de la aplicación desarrollada como objetivo de este proyecto y, por el otro, los comensales, quienes interactúan con la aplicación móvil antes nombrada.

En las siguientes imágenes se puede apreciar de forma gráfica, cómo sería el sistema para cada uno de los roles anteriormente nombrados y cómo los mismos harían uso de éste.

### 7.1. Perspectiva del administrador de restaurante



Imagen 2: Perspectiva del administrador

## 7.2. Perspectiva del comensal



Imagen 3: Perspectiva del comensal

A continuación, se listan las historias de usuarios que fueron detectadas en el transcurso del proyecto a modo de resumen:

- Historia 1: Autenticación
- Historia 2: ABM categorías
- Historia 3: Visualización de categorías
- Historia 4: Búsqueda de categorías
- Historia 5: ABM productos
- Historia 6: Búsqueda de productos
- Historia 7: Vista previa de la carta
- Historia 8: Descarga de la carta
- Historia 9: ABM mesas

- [Historia 10](#): Reserva de mesas
- [Historia 11](#): Abrir cuenta - por usuario administrador
- [Historia 12](#): Abrir cuenta - por aplicación móvil
- [Historia 13](#): Mesa/Pedidos - acciones
- [Historia 14](#): Nuevo pedido - por usuario administrador
- [Historia 15](#): Nuevo pedido - por aplicación móvil
- [Historia 16](#): Cierre de cuenta
- [Historia 17](#): ABM cupones
- [Historia 18](#): Canjeo de cupones
- [Historia 19](#): Gestión de publicidad
- [Historia 20](#): Perfil del restó
- [Historia 21](#): Reportes estadísticos.
- [Historia 22](#): Caja y contabilidad
- [Historia 23](#): Caja - Búsqueda sobre movimientos
- [Historia 24](#): Gestión de turnos
- [Historia 25](#): Pantalla de pedidos y comandas
- [Historia 26](#): Historial de cuentas por mesa
- [Historia 27](#): Impresión de comandas
- [Historia 28](#): ABM Mozos
- [Historia 29](#): Gestión de cubiertos por mesa
- [Historia 30](#): Mesas Mostrador/Barra
- [Historia 31](#): Agregar descuento sobre mesa

En el “[Anexo IV](#)” se puede encontrar el detalle y estado de estas historias de usuario.

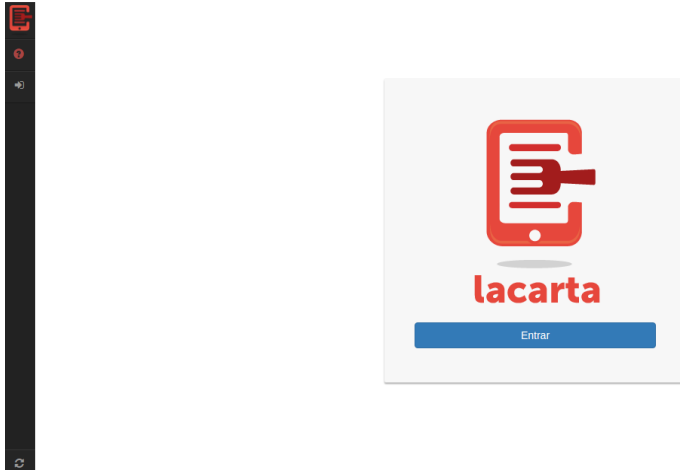
Como el objetivo principal de este proyecto es el diseño y desarrollo de la aplicación para los administradores de restaurantes, creamos el “[Anexo V](#)”, donde se encuentra un Manual de Usuario que detalla las distintas funcionalidades, dando una breve descripción de las mismas junto con imágenes para facilitar su comprensión.

De todas formas, en las siguientes secciones se detallan y describen brevemente las interfaces gráficas de usuario más importantes.

### *7.3. Página de Inicio*

La siguiente imagen describe la pantalla que se ve al acceder al sistema antes de autenticarse mediante usuario y contraseña.





*Imagen 4: Acceso al sistema*

#### 7.4. Menú de accesos lateral izquierdo

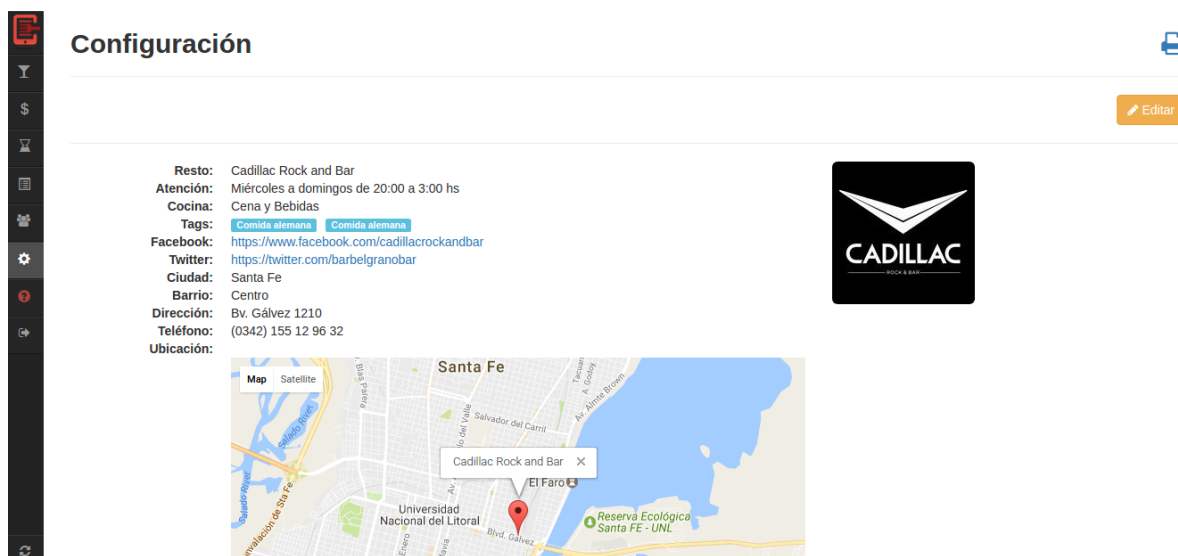
Una vez autenticado en el sistema, el usuario contará con un menú en el lado izquierdo visible en todo momento, que le permitirá acceder a todas las funcionalidades del mismo.



## 7.5. Perfil del restó

La siguiente imagen representa el perfil de administrador del restó, donde puede ver la información cargada para su cuenta (nombre, ubicación, facebook, etc.) y acceder a la edición de la misma.

La información cargada en esta sección es la que verán los usuarios de la aplicación móvil al acceder a dicho comercio.



**Configuración**

**Resto:** Cadillac Rock and Bar  
**Atención:** Miércoles a domingos de 20:00 a 3:00 hs  
**Cocina:** Cena y Bebidas  
**Tags:** Comida alemana Comida alemana  
**Facebook:** <https://www.facebook.com/cadillacrockandbar>  
**Twitter:** <https://twitter.com/barbelgranbar>  
**Ciudad:** Santa Fe  
**Barrio:** Centro  
**Dirección:** Bv. Gálvez 1210  
**Teléfono:** (0342) 155 12 96 32  
**Ubicación:**

**CADILLAC**  
ROCK & BAR

Map Satellite  
Santa Fe  
Cadillac Rock and Bar  
El Faro  
Reserva Ecológica Santa FE - UNL  
Universidad Nacional del Litoral  
Bv. Gálvez  
Av. Rincón Brown  
Av. San Martín  
Av. 14 de Mayo  
Av. 25 de Mayo  
Av. 30 de Mayo  
Av. 31 de Mayo  
Av. 32 de Mayo  
Av. 33 de Mayo  
Av. 34 de Mayo  
Av. 35 de Mayo  
Av. 36 de Mayo  
Av. 37 de Mayo  
Av. 38 de Mayo  
Av. 39 de Mayo  
Av. 40 de Mayo  
Av. 41 de Mayo  
Av. 42 de Mayo  
Av. 43 de Mayo  
Av. 44 de Mayo  
Av. 45 de Mayo  
Av. 46 de Mayo  
Av. 47 de Mayo  
Av. 48 de Mayo  
Av. 49 de Mayo  
Av. 50 de Mayo  
Av. 51 de Mayo  
Av. 52 de Mayo  
Av. 53 de Mayo  
Av. 54 de Mayo  
Av. 55 de Mayo  
Av. 56 de Mayo  
Av. 57 de Mayo  
Av. 58 de Mayo  
Av. 59 de Mayo  
Av. 60 de Mayo  
Av. 61 de Mayo  
Av. 62 de Mayo  
Av. 63 de Mayo  
Av. 64 de Mayo  
Av. 65 de Mayo  
Av. 66 de Mayo  
Av. 67 de Mayo  
Av. 68 de Mayo  
Av. 69 de Mayo  
Av. 70 de Mayo  
Av. 71 de Mayo  
Av. 72 de Mayo  
Av. 73 de Mayo  
Av. 74 de Mayo  
Av. 75 de Mayo  
Av. 76 de Mayo  
Av. 77 de Mayo  
Av. 78 de Mayo  
Av. 79 de Mayo  
Av. 80 de Mayo  
Av. 81 de Mayo  
Av. 82 de Mayo  
Av. 83 de Mayo  
Av. 84 de Mayo  
Av. 85 de Mayo  
Av. 86 de Mayo  
Av. 87 de Mayo  
Av. 88 de Mayo  
Av. 89 de Mayo  
Av. 90 de Mayo  
Av. 91 de Mayo  
Av. 92 de Mayo  
Av. 93 de Mayo  
Av. 94 de Mayo  
Av. 95 de Mayo  
Av. 96 de Mayo  
Av. 97 de Mayo  
Av. 98 de Mayo  
Av. 99 de Mayo  
Av. 100 de Mayo

Imagen 7: Perfil del restó

## 7.6. Administración de la carta

En esta imagen se ve la pantalla que permite llevar a cabo la administración de la carta que expone el restó, junto con sus categorías y productos. En el lado izquierdo, un árbol jerárquico de las diferentes categorías y sobre el derecho los productos y la descripción de los mismos.



Imagen 8: Administración de la carta

## 7.7. Administración de mesas y pedidos

En la siguiente captura se visualiza la funcionalidad que permite administrar mesas, pedidos y cuentas.

Sobre lado izquierdo tenemos el listado de mesas ordenadas por un algoritmo definido que da un peso a estas en base a los pedidos pendientes, la cantidad de los mismos y los comensales sin atender (detallado más adelante en las historias de usuario).

En el centro tenemos el listado de productos ordenados junto con la cantidad y detalle del estado de cada uno de estos.

Finalmente, sobre el lado derecho tenemos el menú de adición rápida de productos (parte superior derecha) que permite realizar nuevos pedidos sobre la mesa, junto con un grupo de acciones (parte inferior derecha) asociadas a la cuenta actual.

The screenshot displays a software interface for managing tables and orders. On the left, a sidebar titled 'Mesas' (Tables) shows a list of tables numbered 1 to 13. Table 1 is highlighted in green and has a '4' next to it. The main area is titled 'Mesa 1 : en uso Mozo : Waters, Roger'. It features a table with the following data:

Producto	Cantidad	Precio	Total
5566 - Papas fritas 1'	2	\$55,00	\$110,00
291 - Lomo a las dos pimientos 1'	4	\$150,00	\$600,00
292 - Pinta Kolsch 2'	2	\$50,00	\$100,00
294 - Pinta Indian Pale Ale 3'	2	\$50,00	\$100,00
<b>Total</b>			<b>\$910,00</b>

At the bottom of the main area, there are buttons for 'Comandar', 'Entregar', 'Editar', and 'Cancelar'. On the right side, there is a 'Cuenta' (Account) section with fields for 'Producto', 'Precio unitario', and 'Cantidad', along with an 'Observaciones' field. Below these are buttons for '+ Adicionar', '+ Cerrar', '- Descuento', '+ Detalle', and '+ Imprimir ticket'. At the very bottom right, there is a counter showing '3' desde inicio de Cuenta and '1' de inventario.

Imagen 9: Administración de mesas y pedido

## 7.8. Listado de turnos y movimientos

La siguiente imagen expone la pantalla donde se listan los turnos que fueron aconteciendo en el restaurante junto con los movimientos ejecutados sobre estos (apertura y cierre de caja, cobro sobre mesas, retiro de dineros, etc.)

**Turnos**

Responsable...   + Opciones

Hoy

18:07 (Steve Jobs)		Efectivo	Efectivo B	Crédito	Débito	Total
18:35:26	Cobro de cuenta en mesa 3	—	—	\$70,00	—	\$70,00
18:34:55	Cobro de cuenta en mesa 5	\$160,00	—	—	—	\$160,00
18:33:32	Cobro de cuenta en mesa 1	\$700,00	\$60,00	—	—	\$760,00
18:15:11	Cobro de cuenta en mesa 3	\$125,00	—	—	—	\$125,00
18:07:51	Apertura de caja	\$558,60	—	—	—	\$558,60
<b>Total</b>		<b>\$1.543,60</b>	<b>\$60,00</b>	<b>\$70,00</b>	<b>\$0,00</b>	<b>\$1.673,60</b>

17:53 / 17:53 (Walter White)

17:50 / 17:50 (Joe Cocker)

Imagen 10: Listado de turnos y movimientos

## 7.9. Cajas y movimientos

La siguiente pantalla representa la visión contable del restaurante, detallando los movimientos sobre las distintas cajas (pago a mozos, cobro sobre mesas, retiro de efectivo, etc...), llevando un resumen totalizador de los mismos y permitiendo generar nuevos.

**Caja**

+ Opciones

\$ \$1.514,57	\$ \$60,00	\$ \$70,00	\$ \$0,00
EFFECTIVO	EFFECTIVO B	CRÉDITO	DÉBITO

Hora	Concepto	Efectivo	Efectivo B	Crédito	Débito	Total
07/06/2017						
18:40	Pago a Mozo Waters, Roger	-\$29,03	—	—	—	-\$29,03
18:35	Cobro de cuenta en mesa 3	—	—	\$70,00	—	\$70,00
18:34	Cobro de cuenta en mesa 5	\$160,00	—	—	—	\$160,00
18:33	Cobro de cuenta en mesa 1	\$700,00	\$60,00	—	—	\$760,00
18:15	Cobro de cuenta en mesa 3	\$125,00	—	—	—	\$125,00
18:07	Apertura de caja	\$558,60	—	—	—	\$558,60

Turno \$1.115,00      Dia \$1.415,00      Mes \$1.415,00

Imagen 11: Cajas y movimientos

## 8. Gestión de riesgos

Para el manejo de los riesgos utilizamos un plan basado en medidas de contención y contingencia que se detallaron oportunamente en el plan del proyecto.

En el mismo establecimos la prioridad de los riesgos y calculamos la exposición (ER) a los mismos utilizando la ecuación:

$$ER = \text{Probabilidad} * \text{Impacto}$$

Durante el desarrollo nos vimos expuestos a varios de estos riesgos por lo cual destacamos la importancia de haber planteado el plan de contingencia que nos permitió, no solo afrontarlos de la mejor manera posible, sino estar preparados al momento que sucedieron.

Uno de los primeros riesgos que nos tocó enfrentar fue el hurto de los recursos utilizados para el desarrollo del proyecto, principalmente la notebook que se estaba utilizando junto con documentación impresa y escrita. Para afrontar este riesgo habíamos tomado las medidas de contingencia de realizar las tareas utilizando recursos compartidos en la web, razón por lo cual se pudo minimizar el impacto del suceso.

Si bien el incidente mencionado anteriormente no fue el de mayor impacto en el desarrollo del proyecto, se detalló el mismo para denotar aún más lo fundamental de la implementación de una adecuada gestión de riesgos.

Los que riesgos que mayormente impactaron tanto al desarrollo de la implementación como la construcción de este informe fueron:

- **Estimación del proyecto errónea/desacertada:** esto estuvo fuertemente ligado al poco conocimiento sobre el negocio/rubro, lo que nos llevó a la necesidad de incorporar funcionalidades no planificadas que eran de carácter indispensable (por ejemplo, manejo de turnos y gestión de caja) que lógicamente postergaron los tiempos del proyecto tal como se detalló en el plan de gestión de riesgos.
- **Baja en productividad/rendimiento de recursos humanos:** este factor se produjo principalmente por la disminución de los tiempos disponibles y dedicados al proyecto luego de cumplir las jornadas laborales. Además, se produjo una pérdida de constancia que acrecentó el impacto. Para dar solución a esto se trabajó en mejorar la comunicación y realizar avances pequeños que sirvieron para tratar el inconveniente sin producir jornadas exhaustivas.

- **Efecto “cascada” de retraso en tareas:** esto se produjo como consecuencia de los riesgos anteriormente expuestos. Para hacer frente a esto recurrimos a la replanificación de los sprints y a postergar fechas de entregas, como también así la salida al mercado.

Además, cabe destacar que la naturaleza de la tecnología y metodología utilizada para el proyecto, nos llevó a exponernos a otros riesgos que detallamos en el plan y que nos parece oportuno destacar:

- Falta de experiencia en testing automático (Protractor).
- Falta de experiencia en tareas de planificación.
- Utilización de nuevas herramientas y/o tecnologías.
- Requisitos poco claros y alto nivel de cambio en los mismos.

Durante el desarrollo del proyecto no nos fue fácil manejar la gestión de los riesgos ni responder al pie de la letra con las acciones que habíamos planteado. Los riesgos que más nos afectaron tuvieron que ver con la falta de tiempo para dedicarle al proyecto debido a cuestiones laborales, familiares y/o de salud. Estos los sufrimos más en la etapa de confección del informe final que en el desarrollo y como conclusión podemos decir que aprendimos a darle más importancia a los riesgos ajenos a cuestiones tecnológicas, de conocimiento o experiencia y tener más en cuenta cuestiones externas al proyecto que terminan afectando muchísimo en los tiempos del mismo.

## 9. Estimación vs. Realidad

Como se puede apreciar al principio de este informe, al momento de plantearnos un objetivo, proyectamos una cobertura de un 20% (20 comercios) del total de comercios gastronómicos detectados en dicho momento (100 comercios). Hoy en día, luego del desarrollo y primer intento de lanzamiento de la aplicación, nos encontramos con que dicha meta fue demasiado ambiciosa y por el momento, no pudo ser cumplida. La principal razón a la que le atribuimos este inconveniente es nuestra inexperiencia en el mercado y en esta sección pretendemos detallar nuestras vivencias haciendo hincapié en los aprendizajes obtenidos como resultados de los distintos errores cometidos durante el transcurso del proyecto.

### 9.1. *Comprensión del negocio*

Nuestros primeros enfoques fueron dirigidos a la construcción de una aplicación gastronómica basándonos en lo que creíamos importante y definiéndolo como nos parecía más lógico y razonable.

Luego de diversas interacciones con los demás sistemas y reuniones con usuarios de estos, descubrimos que la terminología era muy específica y común entre todos, y que muchas de las funcionalidades importantes eran llevadas a cabo de una misma forma. Por esto comprendimos que debíamos respetar una estructura y ciertos lineamientos que ofrecían los sistemas de la competencia.

Esto nos llevó a realizar cambios sobre lo que habíamos comenzado y descartar o volver a desarrollar algunas funcionalidades. Lo positivo fue que se descubrió en una temprana etapa del proyecto, y a causa de esto, el costo fue menor.

### 9.2. *Competencia instalada y nicho de mercado*

Al momento de indagar en el mercado sobre las aplicaciones de la competencia, verificamos que en la gran mayoría de los diferentes comercios gastronómicos importantes se utilizaba MaxiREST. Varias eran las razones de su liderazgo pero podemos destacar que su llegada a los clientes es masiva y también que en mucho de los casos los propietarios de los emprendimientos o empleados claves de estos contaban con experiencias previas sobre el mismo.



Esto nos hizo dar cuenta de lo complicado que sería introducir el producto y nos llevó a enfocar nuestro nicho de consumidores hacia nuevos emprendedores en situaciones más precarias: locales gastronómicos que no tengan software de gestión o no estén en condiciones de afrontar el pago de uno.

### *9.3. Inestabilidad del negocio*

Una de las cuestiones que pudimos detectar en cuanto al negocio al que apuntábamos, era la inestabilidad del mismo en la zona. Es de público conocimiento la rotación de comercios gastronómicos en la ciudad de Santa Fe, y en muchas ocasiones la precariedad de estos. Pudimos evidenciar que algunos ni siquiera contaban con PCs o conexiones a internet y que tampoco tomaban a esto como algo prioritario o primordial para la continuidad del negocio.

Debido a las realidades detalladas anteriormente, nuestra llegada a los usuarios finales de la aplicación se tornaba cada vez más compleja y por ende nuestro nicho de mercado se hacía más acotado.

### *9.4. Comprensión de los clientes*

Al momento de comenzar con el relevamiento del proyecto, asumimos de forma errónea tal vez, que la apertura al uso de aplicaciones nuevas para facilitar la gestión de los comercios gastronómicos por parte de nuestros usuarios directos iba a ser mayor, o tal vez sin tantas barreras.

Inicialmente tuvimos una percepción de que al ser una aplicación gratuita y de alta accesibilidad (sólo se requiere de una PC y conexión a internet), la aceptación de la misma iba a ser considerablemente alta. Pasado el tiempo, terminamos confirmando que esto no era tan así al momento de contactar con algunos de estos y corroborar su reticencia y/o desconocimiento en cuanto a los beneficios del uso de sistemas de gestión.

Por un lado, tenemos el tipo de cliente profesional, con experiencia en el rubro y cuyos negocios inician con grandes inversiones, sin escatimar en costos. Estos tendían a utilizar las herramientas más populares, como lo es MaxiRest.

Luego, tenemos los emprendedores más precarios, los cuales suelen centrar sus atenciones en la continuidad del negocio en el corto plazo y no creen primordial la gestión de esto mediante software específico, o dilatan esto lo más que puedan.

Sumado a esto, debemos mencionar la realidad de nuestro país, en el que la actividad privada se encuentra atada a un régimen tributario agresivo y en el que las diferentes crisis terminan impactando en la baja del consumo de la actividad gastronómica, razón que incrementa la rotación de comercios y lleva a estos a ser extremadamente precavidos. Esto hacía que en ciertos casos se postergue el uso de sistemas en la actividad hasta tanto se logre la estabilidad y se garantice la continuidad, si es que se llegaba a esto.

Las cuestiones mencionadas nos hicieron ser selectivos en cuanto a los posibles usuarios de la aplicación y amoldarnos a las necesidades de estos ofreciendo flexibilidades y, en ciertos casos, un acercamiento más informal con el fin de llegar a un acuerdo.

### *9.5. Extensibilidad de la aplicación*

Durante el desarrollo del proyecto se detectaron posibles extensiones de la aplicación que permitirían brindar funcionalidades extras y una mayor flexibilidad en su uso, cuestión que nos pareció importante mencionar en esta sección.

Una de estas extensiones corresponde a la posibilidad de agregar el perfil de usuarios del tipo “mozo”. Estos contarían con una visión acotada de la aplicación y se encargarían (mediante el uso de tablets y/o dispositivos móviles) de ir tomando pedidos en las diferentes mesas, evitando así, la necesidad de descarga de la aplicación en los teléfonos celulares por parte de los comensales. Básicamente, este perfil tendría habilitadas las funcionalidades de Gestión de mesas y cuentas.

Otro posible agregado al sistema sería el desarrollo de un módulo de “delivery”, el cual le permitiría al restaurant poder llevar una gestión de los pedidos realizados mediante la modalidad de envíos a domicilio, contemplando las diversas características de la misma.

Por lo detallado anteriormente y lo indagado en el mercado, vimos que existen una variedad de posibles capacidades a agregar al sistema con el objetivo de exponer mayores funcionalidades y permitir cubrir distintas necesidades comunes.

## 10. Conclusiones

El diseño y desarrollo de esta aplicación nos aportó principalmente la oportunidad de aplicar la mayor parte de los conocimientos adquiridos académicamente en un contexto muy complejo, real y sobre todo para un mercado masivo, como lo es el rubro gastronómico. Si bien contamos con experiencia laboral en el área de sistemas de información, el hecho de afrontar una construcción por nuestra cuenta, con un conjunto de interesados y product owner que aprendían del negocio conforme lo hacíamos nosotros, hizo que las adversidades sean mayores, y sobre todo, que las decisiones tomadas sean determinantes.

En cuanto a la parte técnica, podemos mencionar que este trabajo nos permitió perfeccionar los conocimientos acerca de la tecnología y las metodologías utilizadas, como así también conocer un amplio espectro de nuevas herramientas. Si bien esta cuestión no es lo más relevante de este proyecto, es oportuno destacar que pudimos ganar experiencia en muchos aspectos, los cuales resultan útiles para un mejor desenvolvimiento en cualquier ámbito laboral.

Asimismo, consideramos positivo hacer hincapié en los inconvenientes tenidos durante este proyecto, sobre todo aquellos relacionados con la interpretación del negocio y como pudimos sortearlos aplicando los distintos enfoques obtenidos en nuestra formación.

Otra de las cuestiones que no podemos dejar de lado es el aprendizaje obtenido en el intento de la salida al mercado del producto, las estrategias de marketing (en la que solíamos participar parcialmente) y la interacción con los distintos usuarios finales que llegaron a realizar pruebas sobre el sistema. Si bien todavía no se ha llegado a tener un caso de uso estable por parte de algún comercio gastronómico, la búsqueda de esto persiste, las esperanzas no se agotan y debido a la calidad del producto final, nos hace pensar que todavía tiene buenas posibilidades de ser aceptado por el mercado.

Para nosotros este proyecto no solo nos representó la significativa tarea de concluir nuestra carrera académica, sino que también fue nuestra primera incursión en el emprendedurismo. Como todo desafío en el cual el rédito puede ser muy grande, entonces también lo son, en igual o mayor magnitud, los riesgos. El aprendizaje adquirido y el resultado de este proyecto, sin duda alguna, nos brinda un aporte muy valioso como profesionales.

Por último, es importante mencionar también algunos aprendizajes o concepciones en relación a la tecnología y a los negocios.

Que la gente pueda pedir lo que desea comer (y elija hacerlo) desde su celular o un dispositivo facilitado por la empresa no resulta descabellado. Ahora, que esta se convierta en la nueva forma de pedir, ya es algo más difícil. Lo cierto es que la tecnología está cambiando la forma en que vivimos, y lo está haciendo de una manera cada vez más agresiva, poniendo en jaque nuestros hábitos y forma de ver el mundo. Empresas como Rappi o Glovo han llevado esto al límite, generando la correspondiente polémica. Muy probablemente, si no es este emprendimiento quien acerca esta posibilidad al comensal, será quien hoy domina el mercado en los salones gastronómicos: MaxiRest. No sin incursionar en un conflicto con aquellos restaurantes que se niegan a ofrecer esta posibilidad a los usuarios o aquellos terceros que rechazan la remoción de intermediarios en cualquier cadena de valor.

Otra vez, Uber, Rappi, Glovo o Airbnb prosperaron aún con inconvenientes y claro está, empoderando al consumidor final. A pesar de los usos y costumbres de la sociedad, a pesar de dilemas éticos e, incluso, de la ley.

La tecnología no llegó sólo para hacer nuestras vidas más sencillas, sino también para cambiarlas.

## 11. Bibliografía

- [1] Material de cátedra: Lineamientos para el Informe del PFC – (Anexo\_B\_-\_Lineamientos\_para\_el\_informe\_de\_Proyectos.doc)
- [2] Material de cátedra: Proyecto Final – Tema 1 (PRESENTACION\_TEMA\_1\_-\_PROYECTO\_FINAL\_2014.pdf)
- [3] Material de cátedra: Proyecto Final – Tema 2 (PROYECTO\_-\_TEMA\_2\_(2013).pdf)
- [4] Material de cátedra: Proyecto Final – Tema 2 – Técnicas de estimación (ROYECTO - TEMA 2 Estimacion por PCU (2015).pdf)
- [5] Gastronomía de la región: Municipalidad de Santa Fe, 2019.  
<http://www.santafeturismo.gov.ar/web/info-util/gastronomia/restoranes.php>
- [6] Sistema gastronómico MaxiREST: MaxiREST, 2019.  
<https://www.maxisistemas.com.ar/>
- [7] Extreme Programming: Wikipedia, 2019.  
[https://es.wikipedia.org/wiki/Programaci%C3%B3n\\_extrema](https://es.wikipedia.org/wiki/Programaci%C3%B3n_extrema)
- [8] Scrum: Wikipedia, 2019:  
[https://es.wikipedia.org/wiki/Scrum\\_\(desarrollo\\_de\\_software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software))
- [9] Waffle.IO, 2017: <https://waffle.io/>
- [10] GIT Software Control Management: GIT, 2019: <https://git-scm.com/>
- [11] Single Page Application: Wikipedia, 2019 [http://en.wikipedia.org/wiki/Single-page\\_application](http://en.wikipedia.org/wiki/Single-page_application)
- [12] Single Page Application: GitBook @juanda, 2014.  
[https://juanda.gitbooks.io/webapps/content/spa/arquitectura\\_de\\_un\\_spa.html](https://juanda.gitbooks.io/webapps/content/spa/arquitectura_de_un_spa.html)
- [13] Pusher services: Pushear team, 2019. <https://pusher.com/>
- [14] Vagrant Boxes: Vagrant, 2019. <https://www.vagrantup.com/>
- [15] Angular Framework: AngularJS, 2017. <https://angularjs.org/>
- [16] Transferencia de Estado Representacional: REST Services, 2019.  
[https://es.wikipedia.org/wiki/Transferencia\\_de\\_Estado\\_Representacional](https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional)

- [17] API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos: BBVA Open4U, 2016. <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>
- [18] Virtual private services, VPS: Wikipedia 2019 [https://es.wikipedia.org/wiki/Servidor\\_virtual\\_privado](https://es.wikipedia.org/wiki/Servidor_virtual_privado)
- [19] Secure Shell – SSH: Wikipedia, 2019. [https://es.wikipedia.org/wiki/Secure\\_Shell](https://es.wikipedia.org/wiki/Secure_Shell)
- [20] DigitalOcean cloud services, 2019. <https://www.digitalocean.com/>
- [21] Uncomplicated Firewall - UFW: Wikipedia, 2019. [https://es.wikipedia.org/wiki/Uncomplicated\\_Firewall](https://es.wikipedia.org/wiki/Uncomplicated_Firewall)
- [22] Protocolo de autenticación OAuth 2.0, 2019. <https://oauth.net/2/>
- [23] Laravel PHP framework – V5.1: <https://laravel.com/>
- [24] New Relic monitoring: <https://newrelic.com/>
- [25] Electron-Build, generador aplicaciones cross-plataforma: <https://www.electron.build/>

# Anexos

## 12. Anexo I - Glosario

- **MaxiREST:** es el sistema gastronómico más utilizado y representa un competidor directo de nuestra aplicación.
- **Extreme Programming / XP:** metodología de gestión y desarrollo de proyectos ágiles.
- **Scrum Programming:** metodología de gestión y desarrollo de proyectos ágiles.
- **Git:** herramienta para el control de versiones de código.
- **GitHub:** gestor de repositorios Git, herramienta para gestión de proyectos, seguimiento de cambios, comunicación entre desarrolladores, etc.
- **Waffle.IO:** herramienta de gestión de proyectos ágiles integrada a GitHub.
- **Slack:** herramienta de comunicación que permite gestionar grupos y canales de chat.
- **Appear.In:** herramienta de video llamadas y comunicación.
- **MySQL:** motor de base de datos.
- **SPA / Single Page Application:** tipo de aplicación cuyo desarrollo se basa en páginas parciales cuya carga se da dinámicamente a demanda en contenidos parciales.
- **Pusher:** sistema de notificaciones, suscripciones y canales de comunicación que permite la gestión de información en tiempo real.
- **Ubuntu:** sistema operativo basado en UNIX.
- **Vagrant:** sistema de virtualización y administración de ambientes de desarrollo.
- **AngularJS:** framework JavaScript utilizado para la construcción de single page Application.
- **REST:** estilo de arquitectura de software basado en el protocolo HTTP.

- **VPS (Virtual Private Server):** servidor virtual privado en donde se alojan las aplicaciones web. En este proyecto se alojaron la REST API y el Backend Administrator.
- **SSH:** protocolo cuya principal función es el acceso remoto a un servidor por medio de un canal seguro en el que toda la información está cifrada.
- **DigitalOcean:** proveedor estadounidense de servidores virtuales privados.
- **UFW (Uncomplicated Firewall):** firewall de fácil implementación y uso.
- **E2E / End-to-end:** metodología de pruebas que simulan el uso de un usuario sobre la aplicación probando flujos de inicio a fin y validando resultados esperados.
- **OAuth 2.0:** es un estándar abierto que permite flujos simples de autorización para sitios web o aplicaciones informáticas.
- **New Relic:** herramienta destinada a la monitorización de aplicaciones y a la elaboración de métricas como también configuración de alarmas y notificaciones.
- **Código QR:** Un código QR es un código de barras bidimensional cuadrado que puede almacenar los datos codificados. En la aplicación mobile se los utiliza para identificar mesas.



## 13. Anexo II: Product Backlog

En este anexo se podrán encontrar los diferentes requerimientos hallados a lo largo del proyecto y que fueron añadidos al Product Backlog para su posterior desarrollo.

A su vez los mismos fueron priorizados y cuentan con el estado actual en torno a su desarrollo o no del mismo

Roles	Administrador: Adm Sistema: Sis
Estados	Pendiente: Pen Desarrollo: Des Hecho: Hec Rechazado: Rec
Prioridades	Baja: B Media: M Alta: A

*Tabla 2: Referencias de roles, estados y prioridades*

Rol	Título	Descripción	Estado	Prioridad
Adm	Autenticación	Podré iniciar sesión utilizando mis credenciales. Además podré finalizar dicha sesión	Hec	A
Adm	ABM categorías	Podré administrar las categorías de ítems de menú	Hec	A
Adm	Visualización de categorías	Podré navegar y visualizar las categorías jerarquizadas en forma de árbol.	Hec	M
Adm	Búsqueda de categorías	Podré buscar categorías ingresando texto de forma sencilla	Hec	B
Adm	ABM Productos	Podré administrar los productos de mi menú junto con sus características como precio, nombre, categoría, etc.	Hec	A
Adm	Búsqueda de productos	Podré buscar productos ingresando texto de forma sencilla	Hec	B
Adm	Vista previa de la carta	Podré hacer uso de una vista previa del menú confeccionado tal cual será desplegada a los comensales	Pen	M

Adm	Descargar la carta	Podré descargar en PDF el menú confeccionado para los comensales	Pen	B
Adm	ABM mesas	Podré administrar las mesas de mi local	Hec	A
Adm	Reservar Mesas	Podré administrar reservas disponiendo por cantidad de comensales y horario	Pen	B
Adm	Iniciar cuenta/pedidos sobre mesa de forma MANUAL	Podré abrir una cuenta sobre una mesa indicando que comensales están ordenando en ella y adicionar pedidos sobre la misma de forma manual	Hec	A
Sis	Iniciar cuenta/pedidos sobre mesa de forma AUTOMÁTICA	El sistema deberá ser capaz de interpretar y dar gestión al inicio de cuentas y pedidos realizados por comensales desde la aplicación mobile.	Hec	A
Adm	Mesa/Pedidos - Acciones	Podré realizar acciones sobre las mesas que incluyan, ver detalle, destacar, actualizar pedidos, entre otras	Hec	A
Adm	Nuevo pedido de forma MANUAL	Podré adicionar pedidos de forma manual sobre mesas con cuentas ya iniciadas	Hec	A
Sis	Nuevo pedido de forma AUTOMÁTICA	El sistema deberá ser capaz de interpretar y dar gestión a pedidos realizados por comensales desde la aplicación mobile sobre sus mesas.	Hec	A
Adm	Cierre de cuenta	Podré gestionar un cierre de cuenta sobre una mesa que se encuentra siendo atendida. Puede existir una previa solicitud de cierre de cuenta por parte de la aplicación mobile de los comensales como no.	Hec	A
Adm	ABM de cupones	Podré gestionar cupones para descuentos y promociones que beneficiarán a los comensales	Des	M
Adm	Canjeo de cupones	Podré gestionar el canjeo de cupones por parte de comensales que solicitaron el uso directamente desde la aplicación mobile	Pen	M
Sis	Gestión de publicidad	El sistema hará visible la publicidad configurada para cada usuario en particular	Hec	M
Adm	Perfil del Restó	Podré modificar mi perfil configurando valores como nombre, ubicación, imagen, tipo de cocina, etc.	Hec	A

Adm	Reportes estadísticos	Podré visualizar reportes estadísticos sobre distintos datos recolectados	Pen	B
Adm	Visualizar pedidos pendientes de comandar	Podré ver en todo momento la cantidad de pedidos pendientes de comandar	Hec	A
Adm	Buscar mesas	Podré buscar sobre las mesas ingresando texto de forma sencilla	Hec	M
Adm	Asistente de creación de mesas	Podré hacer uso de un asistente que me facilite la creación de mesas en masa	Hec	M
Adm	Renovar e imprimir código QR	Podré renovar e imprimir el código QR en cada cierre de cuenta	Hec	A
Adm	Detalle e historial de cuenta	Podré ver por cada mesa el detalle de la cuenta actual con sus pedidos y totales como así también un historial de cuentas abiertas sobre dicha mesa.	Hec	A
Adm	Gestionar e imprimir comandas	Podré seleccionar grupo de pedidos sobre una mesa e imprimir/realizar la comanda	Hec	A
Adm	Agregar descuento sobre cuenta	Podré agregar un descuento sobre una mesa del tipo fijo o porcentual	Hec	M
Adm	Modulo contable	Podré hacer uso de un módulo contable donde se podrá visualizar los distintos movimientos y administrar distintas cajas configurables. Debo además poder agregar movimientos contables de forma manual	Hec	A
Adm	Gestionar cubiertos por mesa	Podré designar cantidad de cubiertos por mesa en cada cuenta	Hec	B
Adm	Adición rápida de productos	Podré adicionar productos sobre una mesa de forma rápida y sencilla con solo escribir el nombre o código de un producto	Hec	A
Adm	ABM de mozos	Podré administrar los mozos con los que trabajo como definir porcentaje de comisión por cada uno	Hec	A
Adm	Gestión de turnos	Podré gestionar turnos dentro de mi comercio asignando un mozo como encargado y realizando cierres y aperturas de los mismos. Se debe poder pagar a los mozos	Hec	A
Adm	Asignar mozo a mesa	Podré asignar un mozo por defecto a una mesa para que todas las cuentas sobre dicha mesa le sean derivados	Hec	M

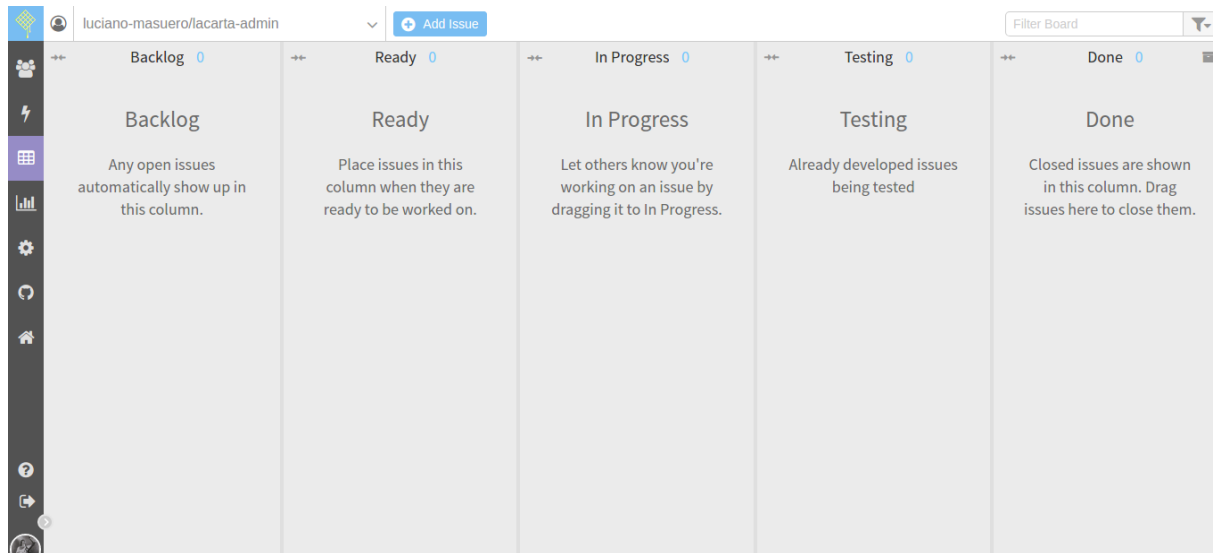
Adm	Mesas como mostrador/barra	Podré gestionar mesas que funcionen como mostrador o barras y cobrar un precio diferenciado sobre los productos que se ordenan en las mismas	Hec	A
Adm	Inactividad sobre mesa	Podré visualizar el tiempo de inactividad sobre una mesa que posee una cuenta abierta	Hec	M
Adm	Configurar impresora comandera	Podré configurar la impresora destinada a las comandas, ticket y QRs.	Hec	A

*Tabla 3: Product Backlog*

## 14. Anexo III: Gestión del desarrollo

### 14.1. Waffle.IO

Waffle.IO es una herramienta gratuita integrada con GitHub que ofrece los servicios y funcionalidades de un **ScrumBoard** típico en el cual se puede hacer un seguimiento de tareas, tener una visibilidad sobre los Sprints como así también servir de gestor de Product Backlog entre otras cosas.



*Imagen 12: ScrumBoard de Waffle.IO*

Como se puede ver en el gráfico anterior, la pizarra fue configurada en 5 columnas:

- **Backlog:** contenía todos los Issues<sup>34</sup> detectados y que debían ser llevados a cabo en algún momento del proyecto. A los Issues se les asignaba un Sprint una vez incluidos en el algún Sprint planning.
- **Ready:** una vez relevado todos los issues que se llevaban a cabo dentro de un Sprint en particular, se le daba comienzo al mismo, y asignaban bajo esta columna.
- **In Progress:** una vez asignado el Issue a un desarrollador en particular y comenzado el trabajo sobre el mismo, pasaban a esta columna, donde permanecían hasta tanto el trabajo sea realizado y llevado a una etapa de testing.

---

<sup>34</sup> Waffle.io se encuentra integrado con GitHub. La unidad atómica de trabajo en GitHub es el Issue. Un Issue no contiene ningún tipo de connotación positiva o negativa, sino que indica alguna especie de trabajo necesario de llevar a cabo. Toda tarea a realizar, ya sea un nuevo feature, un cambio o algún BUG era reflejada en un Issue y llevada a cabo según el caso en particular.

- **Testing:** cuando el desarrollador pasaba su trabajo a la etapa de pruebas los Issues caían bajo esta columna a la espera de ser probados.
- **Done:** una vez realizada las pruebas se generaba una versión y desplegaba un release para la demostración de lo desarrollado durante el sprint. Los Issues terminan cayendo en esta columna donde aparecen durante un tiempo configurable y luego se archivan.

Una de las ventajas más grandes de esta herramienta es que permitía que los issues se trasladan entre las distintas columnas de manera automática por distintos eventos que se lanzan mediante **GitHub** o mediante la interacción con el repositorio de GIT. Algunas de estos son:

- **Comienzo de sprint:** al comenzar un Sprint desde GitHub los Issues bajo el mismo pasaban a la columna **Ready**.
- **Asignación y primer push:** al asignarse un Issue a un developer y crear una rama de trabajo en GIT el mismo pasaba bajo la columna **In Progress**.
- **Solicitud de Merge:** al llevarse a cabo una solicitud de merge sobre la rama de trabajo de un Issue, él mismo pasaba a la columna **Testing**.
- **Merge:** una vez atendida la solicitud de Merge sobre un Issue y llevado a cabo dicho merge el Issue pasaba a la columna **Done** donde luego de un tiempo era archivada.

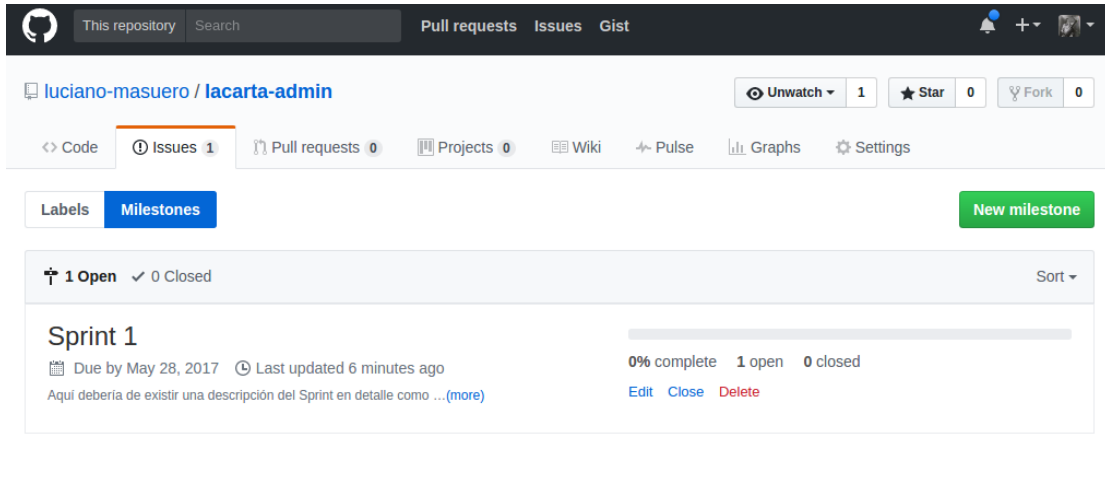
## 14.2. GitHub

### 14.2.1. Sprint

En la siguiente imagen se puede ver como luce un Sprint en **GitHub**<sup>35</sup>. Un detalle a destacar es que sobre el mismo se puede ver la fecha de inicio y fin (opcional) como así también el porcentaje de Issues cerrados bajo dicho sprint para determinar así el nivel de progreso sobre el mismo.

---

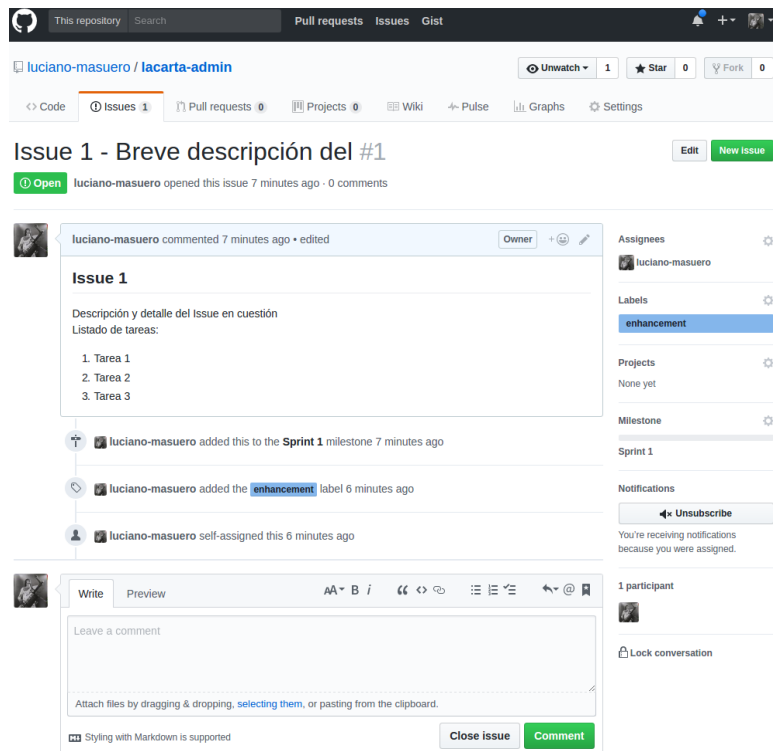
<sup>35</sup> En GitHub el concepto que nosotros decidimos llamar Sprint se lo conoce como Milestone



*Imagen 13: Sprint en GitHub*

### 14.2.2. Issues

En la siguiente imagen se puede ver como luce un Issue en GitHub. También es posible ver datos como el Sprint al cual está asignado, el desarrollador, labels y los comentarios sobre el mismo.



*Imagen 14: Issue en GitHub*

## 15. Anexo IV: Historias de usuario

A continuación, se detallan las historias de usuarios que se definieron en la etapa de planificación del proyecto, con una breve descripción de las mismas más un detalle tanto de las dificultades encontradas como los cambios necesarios que fueron surgiendo a la hora de implementarlas.

Es preciso mencionar que estas fueron planificadas y entregadas junto con el Plan Formal de Proyecto Final y que muchas de ellas luego fueron sufriendo modificaciones o incluso se agregaron otras nuevas que se encuentran detalladas aquí al final de la sección.

### 15.1. Historia 1: Autenticación

El usuario administrador deberá contar con la funcionalidad para acceder al sistema mediante la autenticación utilizando credenciales provistas. De la misma forma deberá poder finalizar dicha sesión.

**NOTA:**

Cualquier acceso sin credenciales válidas debe ser denegado.

---

Esta historia de usuario no tuvo mayores dificultades a la hora de ser llevada a cabo, ya que se trató de un requerimiento básico y muy común de toda aplicación. Solo se tuvieron que tener en consideración los distintos roles necesarios junto con sus respectivas funcionalidades.

En cuanto a tecnología se utilizó lo propuesto por el estándar de autenticación OAuth 2.0 (<https://oauth.net/2/>).

### 15.2. Historia 2: ABM categorías

El usuario administrador deberá poder gestionar las categorías de productos mediante el alta, baja y edición de las mismas. Los datos a definir por categoría serán: nombre y categoría estándar (lista provista por el sistema).

**NOTA:**

- **Categoría:** Forma más amplia de agrupar los productos (Ej.: Bebidas, Postres, Entradas).



- *Categoría estándar*: categoría provista por el sistema para ser asociada a la categoría a crear. Ejemplo: “Bebidas”, “Platos calientes”, “Pastas”, etc.
  - **Subcategoría**: Forma secundaria de agrupar los productos dentro de una categoría (Ej. bebidas sin alcohol y con alcohol).
  - Las categorías y subcategorías deberán permitir al usuario administrador agrupar los productos ofrecidos por el restó en el menú, es decir, deberán poder facilitar la confección del menú digital de la misma forma en que se realiza en forma física (carta del restó).
  - El usuario administrador podrá además ocultar la categoría, y la eliminación de la misma será solamente “lógica”.
  - Se permitirá hasta un nivel de categoría y subcategoría.
- 

Para la implementación de esta historia se hizo un énfasis en la usabilidad que se le iba a brindar a los administradores de la carta para que puedan confeccionar de manera correcta su menú ya que esto es nada más y nada menos la forma de presentación del restaurante y de la oferta hacia los clientes.

Se trabajó mucho en desarrollar una interfaz que permita realizar el alta de la carta de forma rápida e intuitiva, como así también permitir modificaciones minimizando en una cantidad reducida de pasos.

### *15.3. Historia 3: Visualización de categorías*

El usuario administrador deberá poder navegar las categorías en un menú desplegable donde se encontrarán las mismas jerarquizadas en forma de árbol. En este mismo menú, el usuario administrador deberá tener la capacidad de arrastrar categorías definiendo jerarquías.

Al posicionarse sobre cada una de las categorías del árbol, el usuario administrador verá un detalle de la misma, sus subcategorías y los productos asociados.

---

Esta historia de usuario fue particularmente difícil de implementar debido a la funcionalidad de arrastrar categorías dentro de otras para definir jerarquías. Para esto se evaluaron distintos componentes existentes para no incurrir en desarrollo propios que derivan

en una extensión de los tiempos del proyecto. Para la evaluación de las distintas opciones se tuvieron en cuenta factores como facilidad de interacción, usabilidad, performance, soporte, documentación, estética, etc.

Durante el desarrollo y por motivos de criticidad se decidió posponer el desarrollo de la funcionalidad de “Drag and Drop” dado que se había invertido mucho tiempo en ella y los resultados no eran los deseados.

#### **15.4. Historia 4: Búsqueda de categorías**

El usuario administrador deberá tener la posibilidad de buscar fácilmente una categoría entre todas las categorías disponibles.

---

La implementación de esta historia no llevo mayores dificultades. Como guía para el desarrollo se tuvo en cuenta el comportamiento de distintos motores de búsqueda existentes

#### **15.5. Historia 5: ABM productos**

El usuario administrador deberá poder gestionar los productos mediante el alta, baja y edición de los mismos. Los datos a definir por producto serán: nombre, descripción, imagen, precio, código y categoría.

##### **NOTA:**

- Producto: Producto ofrecido por un Restó en el Menú.
- Mediante el ABM Producto el usuario administrador gestionará los productos ofrecidos por el restó, según las necesidades del mismo, agrupándolos en las categorías creadas.
- Son ejemplos de producto (con sus respectivas categorías y/o subcategorías):
  - Bebidas sin alcohol (categoría)
  - Coca-cola 1 lt (producto)
  - Bebidas con alcohol (categoría):
  - Vinos (subcategoría)

- Trumpeter Malbec 750 ml (producto)
  - El usuario administrador podrá además ocultar el producto, y la eliminación del mismo será solamente “lógica”.
- 

Para el desarrollo de esta historia se hizo mucho énfasis en la usabilidad. Se buscó desarrollar un ABM simple e intuitivo mediante una clara distribución de la información a agregar/editar y de las acciones disponibles.

### *15.6. Historia 6: Búsqueda de productos*

El usuario administrador deberá tener la posibilidad de buscar fácilmente un producto entre todos los productos disponibles.

---

El desarrollo de la búsqueda de productos siguió los mismos principios y fue similar la búsqueda de categorías, con el agregado de que ya se contaba con la experiencia en el desarrollo de la primera lo que lo hizo más fácil.

### *15.7. Historia 7: Vista previa de la carta*

El usuario administrador deberá tener la posibilidad de obtener una vista previa de la carta confeccionada con sus categorías, productos y demás información detallada.

**NOTA:**

- Carta: listado de productos con sus precios ordenados por categoría (lo que usualmente se llama “carta” en un Restó).
  - Esta vista previa deberá reflejar cómo los comensales visualizarán la carta en la aplicación móvil.
- 

Se decidió dejar pendiente el desarrollo de esta Historia de usuario dado que se llegó a la conclusión de que la confección de una carta lleva un trabajo de diseño y estilo bastante importante y por ende la funcionalidad ofrecida no llegaría a satisfacer las necesidades o requerimientos mínimos de los usuarios (sin un esfuerzo muy importante) y caería desuso.

La idea es recolectar propuestas luego de usuarios del sistema para desarrollar algo más cercano a lo deseado en promedio.

### **15.8. Historia 8: Descarga de la carta**

El usuario administrador deberá tener la posibilidad de descargar la carta en formato pdf.

---

De la misma forma y por las mismas razones que la Historia 7 fue pospuesta esta también.

### **15.9. Historia 9: ABM mesas**

El usuario administrador deberá poder realizar una gestión de las mesas de su restó mediante el alta, baja y edición de las mismas. Los datos a definir por mesa serán: nombre (para identificar las mismas de forma unívoca y nemotécnica) y código QR (generable, que será utilizado por los usuarios de la aplicación móvil para abrir cuentas y realizar pedidos en la mesa).

#### **NOTA:**

- El usuario podrá darle cualquier nombre a las mesas, tanto numérico como alfanumérico, de forma que pueda identificar las mismas a su conveniencia. Por ejemplo:
    - Con números: 1, 2, 3, 4, ..., 40.
    - Utilizando nombres para identificar lugar donde se encuentra: 1-patio, 2-patio, 1-piso uno, 1-planta baja, etcétera.
    - Las mesas generadas por el ABM mesas permitirán a los usuarios de la aplicación móvil abrir una cuenta en la misma y realizar pedidos.
- 

Además de lo especificado en la historia, se agregó una funcionalidad para agregar mesas de forma masiva y agilizar así la carga, brindando distintos mecanismos de nombrado de mesas.

### **15.10. Historia 10: Reserva de mesas**

El usuario administrador deberá poder gestionar las reservas de mesas realizadas por medios tradicionales (vía telefónica, personalmente, etc.), como así también aquellas originadas desde la aplicación móvil.

Las reservas contarán con los siguientes datos: horario de comienzo, nombre a cargo de la reserva y cantidad de personas.

**NOTA:**

- Las reservas generadas mediante la aplicación móvil ingresarán al sistema en un estado “pendiente”. Las mismas serán luego confirmadas por el usuario administrador generando también una notificación al usuario móvil.
- 

Por motivos de prioridades decidimos postergar esta Historia de usuario dado que la importancia nos pareció menor luego de ver cómo funcionaban los demás sistemas y tener reuniones con administradores de resto.

### **15.11. Historia 11: Abrir cuenta - por usuario administrador**

El usuario administrador deberá poder iniciar pedidos sobre una mesa ejecutando la acción de abrir cuenta sobre la misma. Esta acción servirá para gestionar una mesa en donde no se está utilizando la aplicación móvil para la realización de pedidos y permitirá administrar la cuenta de dicha mesa a través del sistema.

**NOTA:**

- Cuenta: conjunto de pedidos asociados a una mesa.
  - Pedido: conjunto de productos asociados a una cuenta.
  - Esta historia contempla la ejecución manual de la acción de abrir una cuenta en la mesa de un restó lo que permitirá al usuario administrador poder gestionar el mismo sin la necesidad de interactuar con la aplicación móvil.
- 

Para desarrollar esta historia nos enfocamos en que el proceso sea lo más rápido e intuitivo posible para facilitar la administración de las distintas cuentas.

Técnicamente no presentó mayores dificultades y se pudo desarrollar la historia tal cual se planificó.

La solución consistió en una lista de mesas ordenadas por un algoritmo de prioridades en donde el usuario podía seleccionar una y agregar una comanda a la misma. En ese instante se abría una cuenta sobre la misma.

### **15.12. Historia 12: Abrir cuenta - por aplicación móvil**

El sistema deberá poder gestionar las cuentas que son iniciadas por comensales sentados en las mesas desde la aplicación móvil (ver nota debajo).

El usuario administrador deberá poder visualizar y gestionar estas cuentas iniciadas desde la aplicación móvil y ejecutar cualquiera de las acciones disponibles.

#### **NOTA:**

La forma en que los comensales inician la cuenta desde su dispositivo móvil consta de los siguientes pasos o acciones que deberían ejecutar:

- Comensal/es llegan al restó y eligen mesa disponible.
- Comensal/es inician aplicación móvil.
- Comensal/es utilizan el código QR ubicado en la mesa (física) para abrir una cuenta.
- El sistema abre una cuenta y se encuentra disponible para aceptar pedidos.
- Los comensales comienzan a ordenar sus pedidos desde la aplicación móvil.

A estos comensales se los identifica como “Sentados”.

---

La mayor dificultad en la implementación de esta historia radicó en el manejo de eventos para que la acción de apertura de una cuenta desde la app se vea reflejada en el sistema web de los administradores y viceversa. Para esto tuvimos que adquirir los conocimientos en el manejo de eventos a través de websockets y aprender a utilizar el paquete que elegimos utilizar para la implementación ([Pusher](https://pusher.com/), <https://pusher.com/>).

### 15.13. Historia 13: Mesa/Pedidos - acciones

El usuario administrador podrá realizar varias acciones sobre las mesas. Dichas acciones son:

- Ver detalle: al posicionarse sobre la mesa se verá información de la misma tal como:
    - Lista de comensales conectados a la mesa (vía código QR).
    - Lista de pedidos: indicando productos, precios y estado de los pedidos, como así también el comensal que generó el mismo.
    - Tiempo de inicio de la cuenta.
  - Marcar como destacada: el usuario administrador podrá destacar la mesa como importante para agilizar su visualización y facilitar su seguimiento.
  - Cambiar estado de pedidos (cancelado, entregado, etc.).
  - Editar pedidos.
  - Renovar código QR: esta acción le permite regenerar el código QR por motivos de seguridad.
  - Descargar código QR: la acción descarga un archivo PDF que será impreso y colocado en la mesa correspondiente.
- 

La realización de esta historia fue compleja ya que involucró la sincronización entre las acciones realizadas desde la app que debían ser reflejadas en el sistema web y viceversa.

Técnicamente requirió profundizar todavía más en el manejo de eventos y en el framework Angular para mantener la interfaz con la información actualizada constantemente.

Además de la complejidad en el manejo de la información, también fue complejo llegar a una interfaz intuitiva y que presente la gran cantidad de datos de manera elegante y legible. Para esto se manejaron varias alternativas de diseño de la pantalla y se realizaron constantes cambios para pulir la misma.

#### *15.14. Historia 14: Nuevo pedido - por usuario administrador*

El usuario administrador podrá gestionar nuevos pedidos sobre mesas que cuenten con cuentas abiertas indicando los productos que incluyen los mismos. Esta acción le permite al usuario administrador poder llevar a cabo la gestión de pedidos realizados en una mesa sin necesidad de interactuar con la aplicación móvil.

---

Esta historia no llevo mayor dificultad. Se trabajó en que el agregado de pedidos para los administradores sea una tarea rápida agregando funcionalidades para que la misma pueda ser realizada con atajos de teclados y búsquedas para agilizar.

Otra cosa a destacar es que se cambió el hecho de que para hacer el pedido debía de existir una cuenta abierta sobre la mesa. La solución final llevó a que si se realiza un pedido sobre una mesa que no poseía una cuenta abierta, creaba una.

#### *15.15. Historia 15: Nuevo pedido - por aplicación móvil*

El sistema deberá permitir gestionar el evento de un nuevo pedido realizado por un comensal vía aplicación móvil. Esta acción hará visible mediante una notificación la mesa en la cual se haya realizado el pedido alertando al usuario administrador de esta situación y permitiéndole gestionar el mismo.

---

Tal como se describió en historias anteriores la mayor labor de este tipo de historias radicó en el manejo de eventos. Para esta historia en particular se trabajó en reflejar las notificaciones de pedidos en una forma que permita ser captada lo más rápido posible por los administradores para optimizar la gestión y además que sea visible en cualquiera de las interfaces del sistema relacionadas con pedidos.

El mecanismo de alerta se basa en efectos visuales sobre la lista de las mesas y componentes que permiten visualizar rápido la cantidad de pedidos nuevos realizados sobre una mesa como así también notificaciones sobre estos eventos.



### 15.16. *Historia 16: Cierre de cuenta*

El usuario administrador deberá poder gestionar el cierre de cuenta luego de una petición de cierre de cuenta (desde aplicación móvil) o por decisión propia. Esta acción tendrá como resultado:

- Generación de detalle de cuenta.
  - Los usuarios de aplicación móvil que se encuentren con una sesión abierta en la mesa quedarán desvinculados de la misma.
  - La mesa quedará disponible y se finalizará el proceso de pedido sobre la mesa.
- 

Se trabajó mucho en la presentación del detalle de la cuenta para mostrar el contenido de la información de forma prolija y fácil de leer.

Técnicamente el mayor desafío fue mantener los estados de las cuentas según las acciones realizadas desde la app y la notificación instantánea de esta acción para que el administrador pueda gestionar la misma.

### 15.17. *Historia 17: ABM cupones*

El usuario administrador deberá poder realizar una gestión de los cupones que ofrecerá su restó mediante la alta, baja y edición de los mismos.

Se podrán gestionar dos tipos de cupones: cupones que se aplican a un producto y cupones que se aplican a una cuenta. Para ambos tipos de cupones se deberán definir los siguientes datos: nombre, descripción, imagen, fecha de inicio, fecha de vencimiento y descuento. Los cupones que se aplican a productos deberán definir adicionalmente el producto al cual aplican. Los cupones que se aplican a cuentas deberán definir dos parámetros adicionales (a los comunes para ambos tipos de cupones): mínimo y máximo, que establecen respectivamente el total mínimo que debe exceder la cuenta para poder emplear el cupón, y el monto máximo de descuento que cubre el cupón.

Los comensales podrán hacer uso de estos cupones a la hora de realizar pedidos desde la aplicación móvil.

---

Si bien se comenzó con el desarrollo de esta y se realizó gran parte de la misma por motivos de tiempo y prioridades como así también complejidad decidimos excluirla de la versión inicial e incluir otras más relevantes para el negocio.

### **15.18. Historia 18: Canjeo de cupones**

El sistema deberá permitir el canje de cupones de los dos tipos existentes.

Para el caso de canjear cupones que aplican a un producto, esta acción implica:

- Descuento sobre el precio del producto asociado al cupón.
- Marcar el cupón como “utilizado”.

Para el caso de canjear cupones que aplican a una cuenta, esta acción implica:

- Descuento sobre el total de la cuenta a la que se le aplica el cupón.
- Marcar el cupón como “utilizado”.

---

Debido a la decisión tomada en la historia anterior (17) esta queda postergada automáticamente.

### **15.19. Historia 19: Gestión de publicidad**

El sistema deberá hacer visible en distintas secciones diversos tipos de avisos publicitarios configurados en la aplicación.

#### **NOTA:**

La aplicación consultará una API y ésta le retornará las imágenes publicitarias que deberá colocar en las secciones definidas para la publicidad.

---

La complejidad de esta historia residió en la ubicación de la publicidad sin que llegue a ser invasiva como que vaya en contra de la estética de la aplicación.

En definitiva, la solución consiste en la carga de una imagen de un tamaño determinado en una sección especial de la aplicación mobile.

### 15.20. *Historia 20: Perfil del restó*

El usuario administrador podrá modificar/agregar la información general de su restó, que podrá ser consultada en todo momento por los usuarios de la aplicación móvil. Los datos que podrá modificar/agregar serán: nombre del restó, dirección, teléfono, rubro/s al que pertenece e imagen.

---

No existió complejidad en esta historia y la misma fue desarrollada sin inconvenientes

### 15.21. *Historia 21: Reportes estadísticos.*

El usuario administrador podrá visualizar una interfaz de reportes donde encontrará distintos gráficos estadísticos de datos recolectados por la aplicación. Los mismos podrán ser filtrados por día, semana, mes, año o totales.

---

Se decidió postergar esta historia hasta que se tenga un uso de la aplicación y poder así determinar los datos relevantes o significativos para el negocio y de ahí poder elaborar estadísticas que aporten valor a los administradores.

Las historias anteriormente nombradas corresponden a las relevadas en las etapas iniciales del proyecto. A medida que avanzábamos en el mismo fuimos descubriendo que había funcionalidades importantísimas para el negocio que habían quedado fuera. Es por esto que Sprint a Sprint y reunión a reunión íbamos modificando el backlog y agregando/postergando historias para llegar a un producto mínimo y viable (MVP) que nos permita satisfacer las necesidades primordiales de los usuarios finales.

La siguiente lista de historias corresponde a las halladas durante el desarrollo del proyecto.

### **15.22. Historia 22: Caja y contabilidad**

El sistema debe permitirle al usuario administrador ver un historial de los movimientos realizados en su comercio como así también gestionar nuevos movimientos cuando lo desee.

Acciones que generarán movimientos son:

- Movimiento manual
- Cierre de cuenta
- Cierre y apertura de caja/turno
- Pago de mozos

El usuario administrador además tendrá la posibilidad de configurar hasta 4 cajas donde imputar los movimientos.

La forma de visualizar los movimientos será en una lista paginada y separada por días (usando el mismo criterio que utiliza WhatsApp con la mensajería) donde se deberá visualizar:

- Hora
- Concepto
- Monto en cada caja (Determinando si corresponde a egreso o ingreso)
- Total imputado

A su vez deberá existir al final de la lista una forma clara de visualizar el total imputado por turno, día y mes.

### **15.23. Historia 23: Caja - Búsqueda sobre movimientos**

Dentro de la sección caja de la aplicación donde pueden visualizarse todos los movimientos, el usuario administrador deberá tener la posibilidad de realizar una búsqueda ingresando texto. Además, podrá filtrar entre fechas de movimiento.

#### **15.24. Historia 24: Gestión de turnos**

El administrador deberá tener la posibilidad de llevar una gestión de los turnos sobre la aplicación. Un turno consiste en una fecha y hora de comienzo, un responsable y una fecha y hora de fin.

El administrador deberá tener la posibilidad de iniciar y finalizar turnos. Todos los pedidos realizados en un turno estarán asociados al mismo y no se podrá realizar ninguno hasta tanto no se abra un turno.

En la vista de turnos se tiene la posibilidad de realizar un pago a los mozos donde al hacer click se abre una ventana que desplegará:

- Lista de mozos
- Monto total de cuentas cerradas por mozo
- Porcentaje de comisión configurado para el mozo
- Caja desde donde se realizará el pago
- Botón para realizar el pago: se encarga de pagar el monto y registrar el movimiento en la caja indicada

#### **NOTA:**

Esto permite asociar todos los movimientos contables durante un turno específico y un encargado en caso de tener múltiples.

#### **15.25. Historia 25: Pantalla de pedidos y comandas**

El administrador debe contar con una pantalla donde pueda llevar a cabo la gestión diaria del comercio sin necesidad de salir de la misma.

Desde dicha pantalla deberá poder ver:

- Lista de mesas junto con sus pedidos pendientes y estado
- Lista de pedidos sobre la mesa seleccionada
- Mozo asociado a la mesa seleccionada (si es que tiene)
- Menú de comanda para realizar una adición sobre dicha mesa de forma rápida y sencilla

- Tiempo desde apertura de cuenta e inactividad sobre la mesa
- Detalle de la cuenta
- Cantidad de cubiertos
- Historial de cuentas sobre la mesa

A su vez deberá poder:

- Realizar pedidos de forma rápida mediante la selección (autocompletable) de productos, ingresar la cantidad y presionar la tecla “Enter”.
- Editar pedidos
- Comandar pedidos
- Marcar como entregados pedidos que fueron comandados
- Ver el detalle de la cuenta
- Imprimir ticket
- Actualizar cantidad de cubiertos
- Cerrar la cuenta
- Destacar la mesa
- Asignar mozo encargado de la mesa

### ***15.26. Historia 26: Historial de cuentas por mesa***

El usuario administrador deberá poder visualizar el historial de cuentas sobre la mesa. Se deberá desplegar una lista con las cuentas mostrando el estado de las mismas, el monto consumido, cubiertos, descuentos realizados y hora del cierre.

Tiene que existir un botón que permite ver el detalle de cada cuenta que deberá ser el mismo que se muestra desde la pantalla descrita en la historia 25.

### **15.27. Historia 27: Impresión de comandas**

El usuario administrador podrá realizar la comanda de uno o varios pedidos desde la pantalla descrita en la historia 25 y generar así una impresión con la comandera que tendrá el detalle de los productos comandados.

### **15.28. Historia 28: ABM Mozos**

El usuario administrador deberá poder llevar a cabo la gestión de mozos si es que lo desea dando de alta los mismos desde la sección de Mozos.

Los datos que podrá ingresar son:

- Nombre
- Apellido
- Celular
- Porcentaje comisión

En la misma pantalla deberá existir la posibilidad de buscar mozos ingresando texto.

### **15.29. Historia 29: Gestión de cubiertos por mesa**

El usuario administrador podrá indicar la cantidad de cubiertos por mesa desde la pantalla descrita en la historia 25 mediante un botón contador que le permitirá incrementar/decrementar la misma.

### **15.30. Historia 30: Mesas Mostrador/Barra**

El usuario administrador deberá tener la posibilidad de configurar las mesas como Mostrador. Los mostradores son barras o exhibidores en los que el precio de venta de los productos es el indicado como 'Precio mostrador' configurado desde el ABM de productos.

### **15.31. Historia 31: Agregar descuento sobre mesa**

El usuario administrador deberá de tener la posibilidad de agregar descuentos sobre las cuentas por medio de la pantalla descrita en la historia 25.

Los descuentos pueden ser porcentuales sobre la cuenta o un monto fijo y se le puede añadir el motivo de descuento para mayor información o auditoría futura.

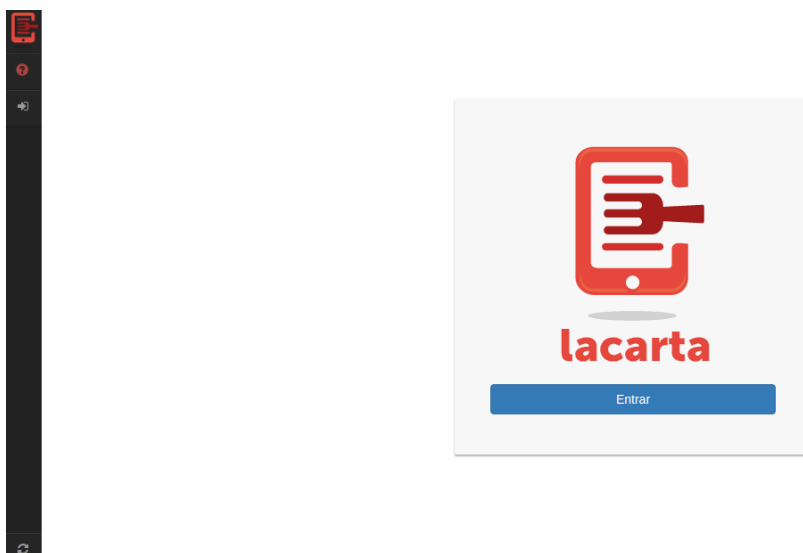
## 16. Anexo V: Manual de usuario

En el siguiente anexo se detallan las funcionalidades más importantes de la aplicación junto con imágenes de las pantallas para dar una descripción que brinde al usuario una visión gráfica de las posibilidades que brinda el sistema y cómo utilizarlas.

Se asume que el usuario ya se contactó con el proveedor de la aplicación y posee una cuenta activa junto con su usuario y contraseña correspondiente.

### 16.1. Pantalla de Inicio

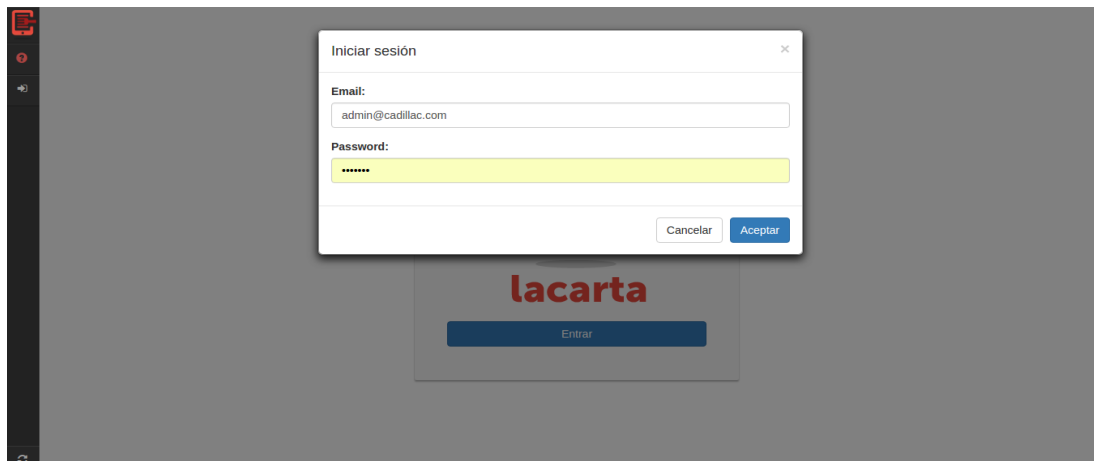
La siguiente imagen pertenece a la pantalla de inicio de la aplicación. La misma posee una interfaz simple y minimalista mediante la cual un usuario no autenticado podrá ingresar al sistema presionando el botón con la etiqueta “Entrar”.



*Imagen 15: Pantalla de inicio*



Al presionar dicho botón se le presentará una ventana tipo modal donde deberá ingresar sus credenciales y presionar “Aceptar” para posteriormente acceder a “lacarta”.



*Imagen 16: Inicio de sesión*

Desde ahora en adelante, todas las pantallas que serán detalladas requerirán de previa autenticación del usuario para su acceso.

## 16.2. Menú de accesos lateral izquierdo

En la siguiente imagen vemos el menú de opciones de acceso que posee un usuario luego de autenticarse en el sistema.



*Imagen 17: Menú colapsado*

*Imagen 18: Menú expandido*

En la imagen del lado izquierdo se ve el menú en su forma colapsado junto con los íconos y/o accesos de pantallas. En la imagen del lado derecho se puede ver el menú en su forma expandida con texto descriptivo por acceso.

A continuación, procedemos a describir cada uno de los accesos:

- **Colapsar/Expandir Menú:** permite al usuario poder colapsar o expandir el menú a su gusto como se muestra en las imágenes anteriores.
- **Administrar mesas y pedidos:** mediante este acceso el usuario tiene la posibilidad realizar la gestión de mesas y administrar los pedidos sobre las mismas.
- **Administración de cajas:** mediante este acceso el usuario tiene la posibilidad de llevar la gestión de la caja y realizar distintos movimientos contables sobre las mismas.
- **Administración de turnos:** mediante este acceso el usuario tiene la posibilidad de gestionar turnos y visualizar el historial de los mismos junto con sus movimientos registrados.
- **Administración de carta:** mediante este acceso el usuario tiene la posibilidad de gestionar la carta de su restó creando la jerarquía de categorizaciones como así también los productos bajo las mismas.
- **Administración de mozos:** mediante este acceso el usuario tiene la posibilidad de gestionar el personal de mozos de su resto.
- **Perfil de restó:** mediante este acceso el usuario tiene la posibilidad de modificar el perfil y datos de su restó a gusto
- **Manual de ayuda:** este acceso representa un link hacia un archivo pdf con el manual de ayuda y preguntas frecuentes sobre la aplicación
- **Cerrar sesión:** mediante este acceso el usuario finaliza su sesión en el sistema.
- **Recargar:** este acceso permite al usuario realizar una recarga de la sección en la que se encuentra refrescando los datos.

### 16.3. Perfil del restó

En la pantalla del perfil de restó el usuario tendrá la posibilidad de visualizar cuestiones básicas de su perfil.

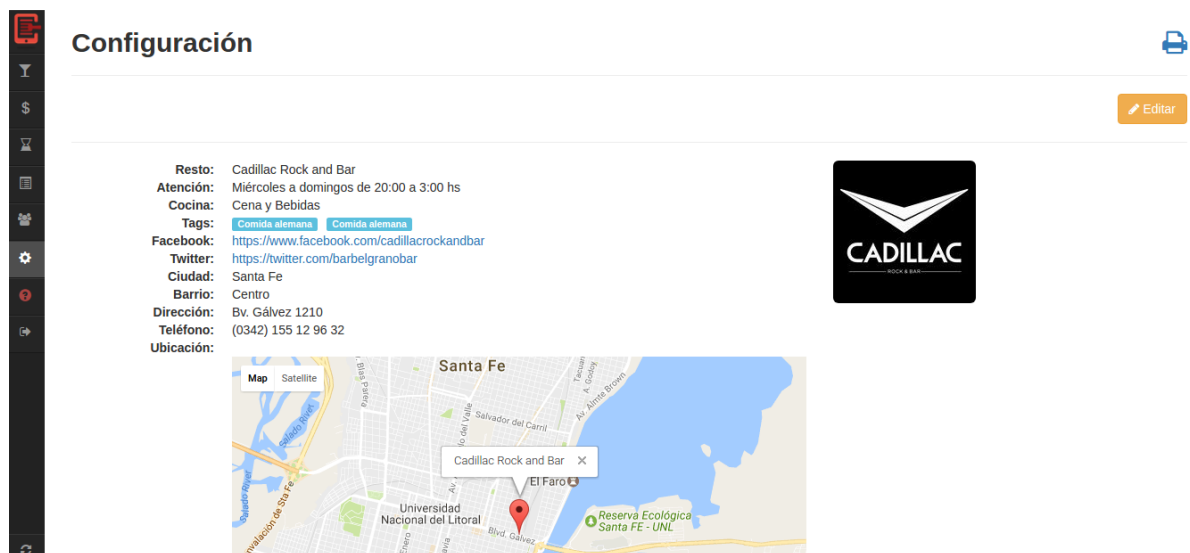


Imagen 19: Perfil del restó

En el margen superior derecho de la pantalla de perfil el usuario cuenta con un botón para editar el mismo. Una vez presionado dicho botón se muestra la siguiente pantalla.

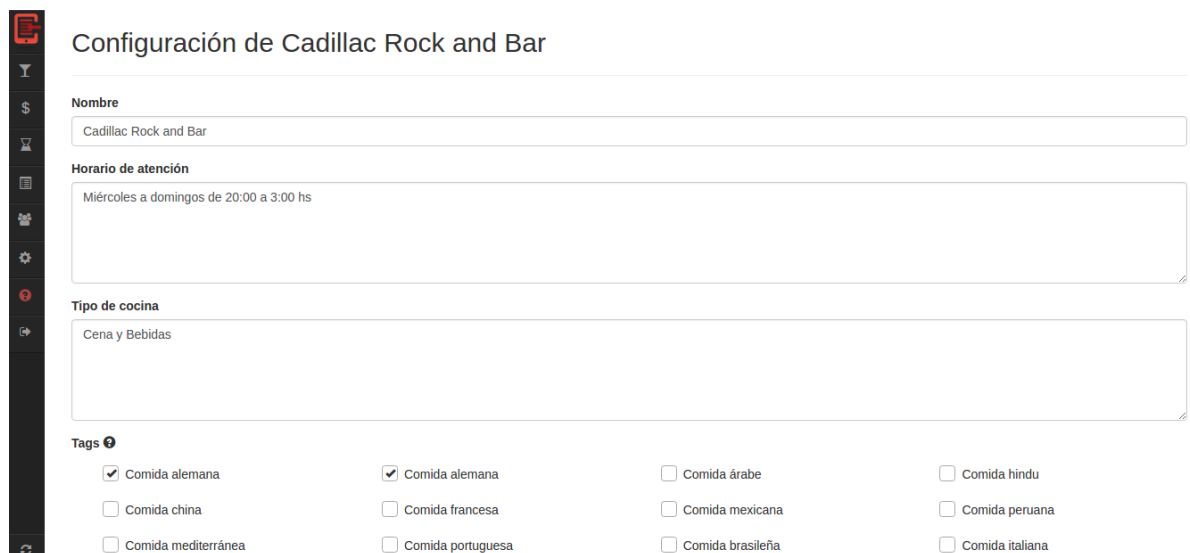


Imagen 20: Detalles del perfil del restó

En esta pantalla el usuario tendrá la posibilidad de configurar los siguientes datos relacionados a su perfil:

- Nombre
- Horario de atención
- Tipo de cocina
- Etiquetas que lo identifiquen
- Sitio web
- Imagen
- Facebook
- Twitter
- Ubicación
- Ciudad
- Barrio
- Teléfono

Esta información es la que visualizan los usuarios de la aplicación mobile cuando acceden a perfil público del restó.

#### 16.4. Administración de la Carta

En esta interfaz el usuario tiene la posibilidad de definir la jerarquización de categorías y subcategorías como así también los productos que se encuentran bajo las mismas.

Al posicionarse sobre una categoría o crear una, el usuario ve en su pantalla la siguiente imagen:



Imagen 21: Configuración de la carta

Dentro de una categoría el usuario tiene la posibilidad de definir nuevas subcategorías o agregar productos bajo la misma (solamente uno de las dos es posible).

Si decide optar por agregar subcategorías dentro de una categoría luego al seleccionar dicha esta verá la siguiente pantalla:



The screenshot displays a web interface for managing products. On the left is a sidebar menu titled 'Categorías' with a list of categories: 'Bebidas con alcohol' (3 items), 'Aperitivos', 'Cervezas' (selected), 'Vinos', 'Bebidas sin alcohol', 'Cafetería', 'Carnes', 'Ensaladas', 'Entradas', 'Pastas', 'Pizzas', 'Postres' (1 item), and 'Tablas'. The main content area is titled 'Cervezas'. At the top right of this area are buttons for '+ Subcategoría', '+ Producto', 'Editar', and 'Eliminar'. Below these buttons, it states 'Subcategoría perteneciente a Bebidas con alcohol'. A table titled 'Productos' lists three items:

Código	Nombre	\$ Mesa	\$ Mostrador
292	Pinta Kölsch	\$50,00	\$50,00
293	Pinta Imperial Stout	\$50,00	\$50,00
294	Pinta Indian Pale Ale	\$50,00	\$50,00

Imagen 22: Productos de una categoría

Dentro de una subcategoría el usuario podrá ver la lista de productos ubicados bajo la misma como así también agregar nuevos productos.

Si opta por agregar nuevo producto presionando el botón que indica ello se le presentará la siguiente pantalla con el formulario de carga de producto

**Categorías**

- ▼ Bebidas con alcohol 3
- Aperitivos
- Cervezas**
- Vinos
- Bebidas sin alcohol
- Cafetería
- Carnes
- Ensaladas
- Entradas
- Pastas
- Pizzas
- Postres 1
- Tablas

**Nuevo producto:**

Nombre

Categoría al que pertenece

Código

Precio

Precio Mostrador

Descripción

*Imagen 23: Agregar nuevo producto*

El usuario podrá configurar los siguientes datos del producto:

- Nombre
- Precio regular
- Código (Atajo para ser usado en la adición)
- Precio de mostrador
- Imagen
- Descripción

La siguiente pantalla muestra el formulario que le permite al usuario agregar una categoría luego de presionar el botón con signo “+” que se encuentra en el margen inferior derecho del listado de categorías



Imagen 24: Agregar nueva categoría

## 16.5. Administración de Mozos

En la siguiente pantalla el usuario tiene la posibilidad de llevar a cabo la gestión de mozos de su restó.

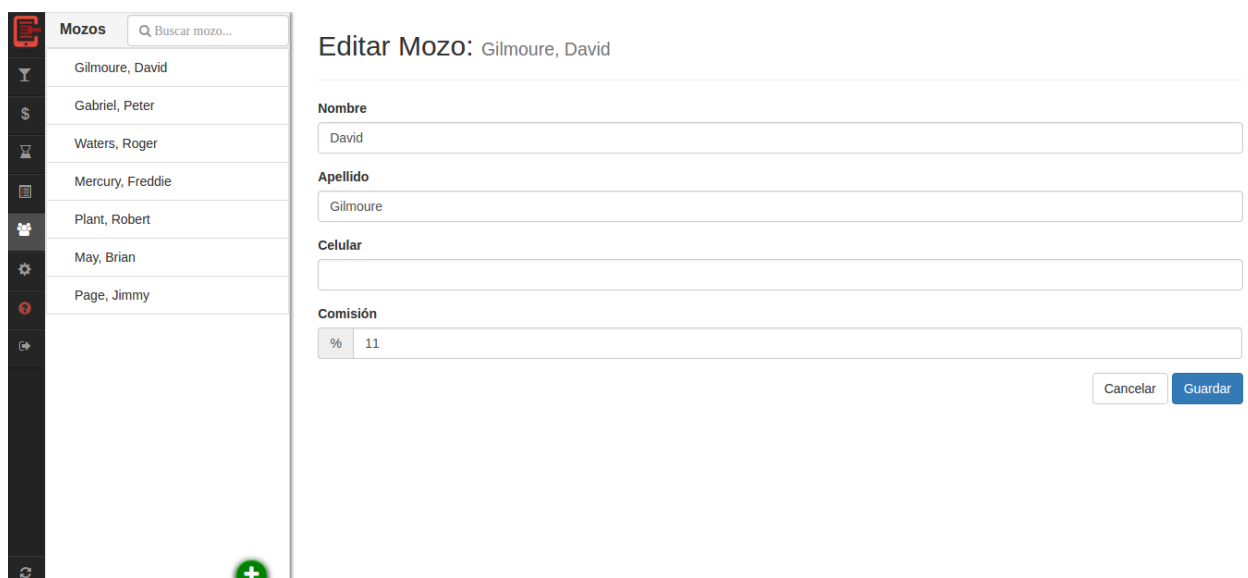


Imagen 25: Administración de mozos

El formulario de alta y modificación es bastante sencillo y permite al usuario cargar los siguientes datos a un mozo:

- Nombre
- Apellido

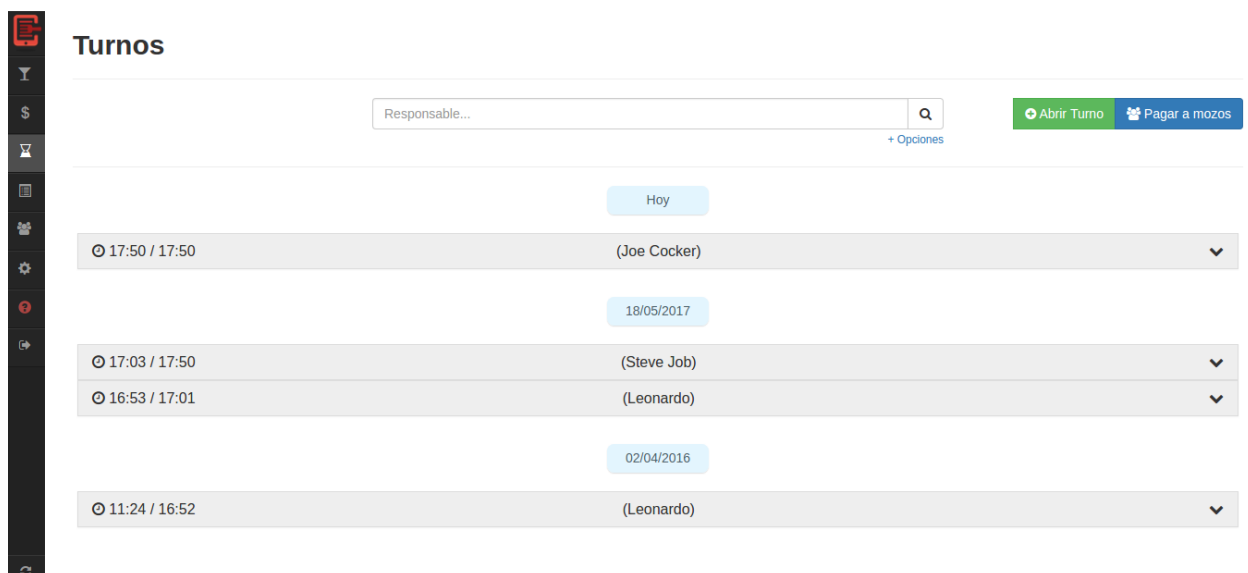
- Celular
- Comisión (porcentaje a percibir según sus atenciones)

Luego de realizar su carga, estos mozos se encuentran disponibles para ser asignados a las distintas mesas y reflejar así la cobertura de los mismos en el espacio físico del resto. Esto permitirá además conocer el monto de comisión para cada mozo según el porcentaje configurado y las ventas que haya realizado el mismo durante el plazo de un turno.

### 16.6. Administración de turnos

Desde esta interfaz el usuario tiene la posibilidad de gestionar la apertura y cierre de los turnos como así también asignar el encargado sobre los mismos.

La idea principal es tener a un encargado por turno quien será el responsable de los movimientos que se realicen durante el mismo y poder así llevar un control más detallado de lo que sucede en el resto y bajo la supervisión de quien.



*Imagen 26: Administración de turnos*

A la hora de abrir un turno se le muestra al usuario una ventana tipo modal como la que se puede ver en la siguiente imagen:



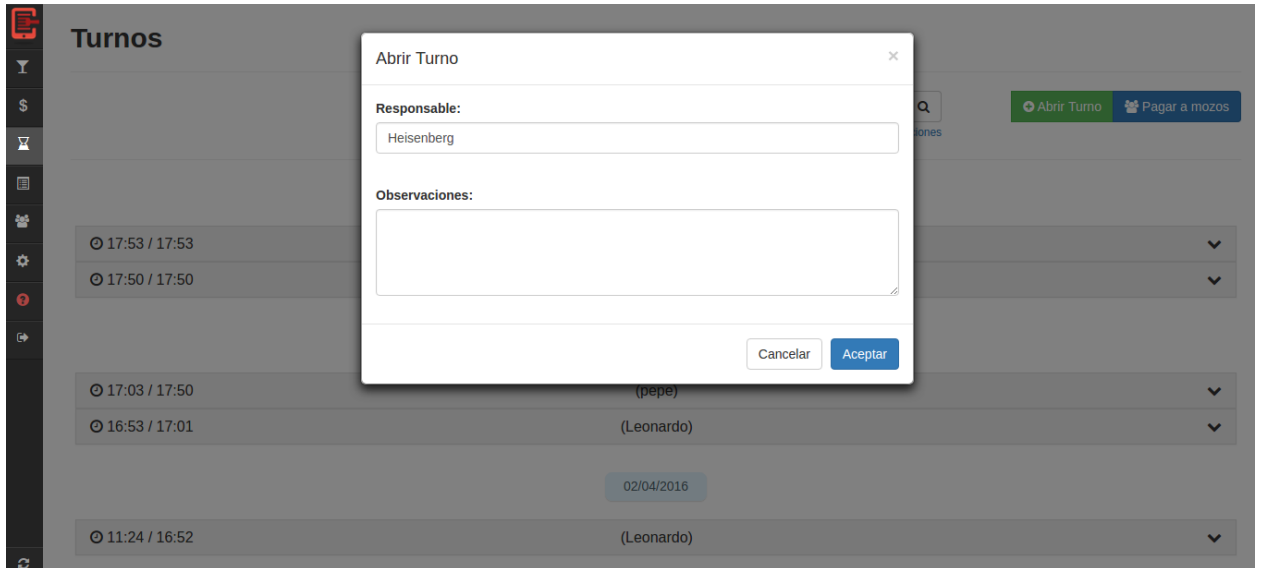


Imagen 27: Abrir nuevo turno

En ella configura el nombre del responsable (puede elegir alguno previamente cargado o agregar nuevos) y en caso de ser necesario agrega las observaciones pertinentes.

Al abrir un turno todos los movimientos realizados durante dicho turno (no se pueden realizar movimientos si no existe un turno abierto) quedan asociados al mismo, y al cerrarlo es posible visualizarlos agrupadamente como se muestra en esta interfaz.

Hora	Concepto	Efectivo	Efectivo B	Crédito	Débito	Total
18:46:06	Retiro de dinero / Cierre de lote	—	—	-\$70,00	—	-\$70,00
18:43:56	Pago limpieza	-\$300,00	—	—	—	-\$300,00
18:42:15	Proveedores	-\$200,00	—	—	—	-\$200,00
18:40:13	Pago a Mozo Waters, Roger	-\$29,03	—	—	—	-\$29,03
18:35:26	Cobro de cuenta en mesa 3	—	—	\$70,00	—	\$70,00
18:34:55	Cobro de cuenta en mesa 5	\$160,00	—	—	—	\$160,00
18:33:32	Cobro de cuenta en mesa 1	\$700,00	\$60,00	—	—	\$760,00
18:15:11	Cobro de cuenta en mesa 3	\$125,00	—	—	—	\$125,00

Imagen 28: Movimientos de turnos

## 16.7. Administración de mesas y pedidos

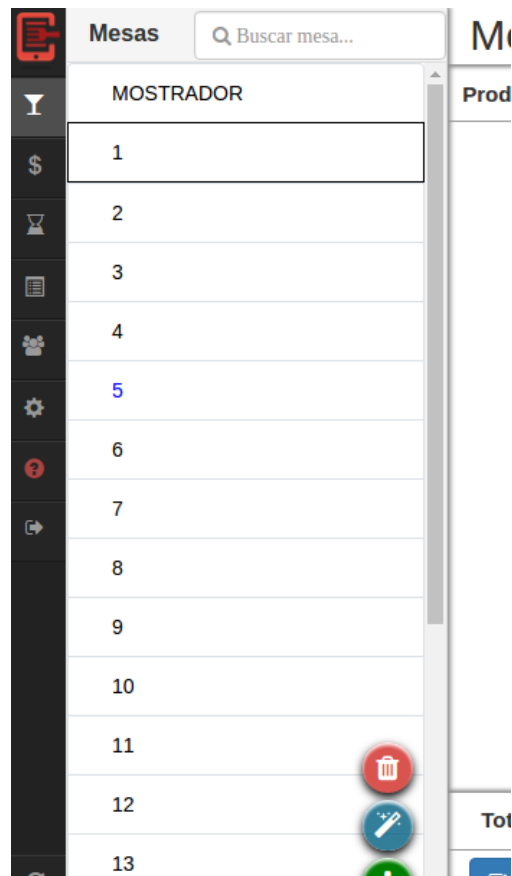
La siguiente pantalla representa la principal interfaz del sistema dado que en la misma es donde se realizan las gestiones más importantes en lo que respecta al funcionamiento de un comercio gastronómico.



Imagen 29: Administración de mesas y pedidos

Comenzaremos dando una descripción de cada sector en la pantalla, teniendo en cuenta que las funcionalidades disponibles en la misma, se irán detallando en interfaces posteriores para ser más precisos en su descripción.

### 16.7.1. Listado de Mesas



*Imagen 30: Listado de mesas*

Se encuentra en el lateral izquierdo de la pantalla y en él se listan las mesas configuradas para el resto. Posee un campo de texto en la parte superior que permite filtrar las mismas.

Al realizar click sobre una de las mesas se carga la información relevante de la misma en la sección lateral derecha, que se irá detallando a continuación.

### 16.7.2. Mesa y Mozo



Imagen 31: Mesa y mozo

Se encuentra en la parte superior de la pantalla y en esta se ve el nombre configurado para la mesa junto con su estado y el mozo asignado.

Haciendo click sobre el Mozo se despliega el listado de los mismos desde donde se pueden cambiar la asignación.

### 16.7.3. Pedidos y Adición

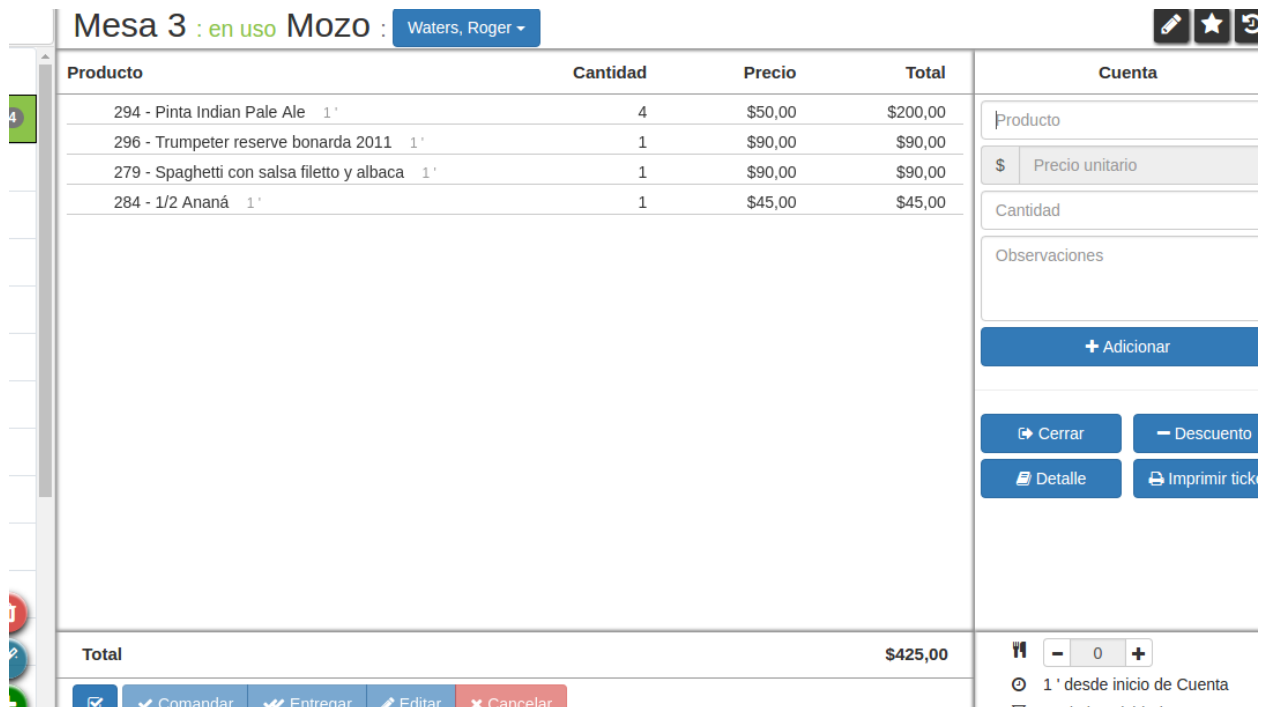


Imagen 32: Pedido y adición

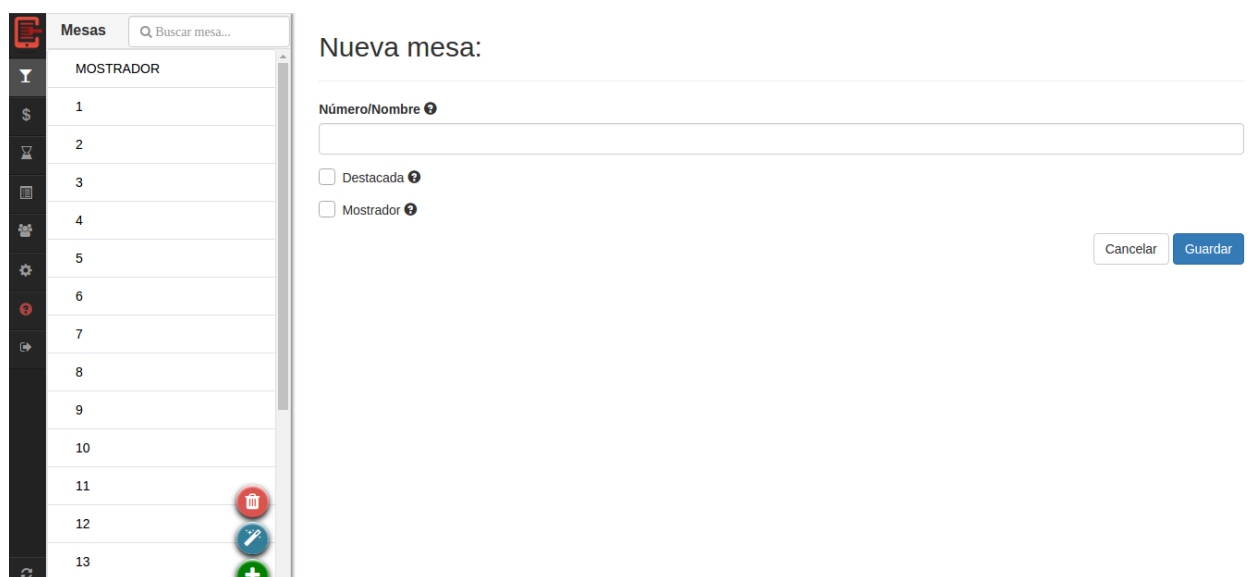
En esta sección es donde se puede ver la información más relevante de la mesa como son los pedidos asociados a la misma junto con sus estados, cantidades, montos y tiempo desde que se realizó el pedido. Además de esto se puede visualizar un total consumido hasta el momento, el tiempo de inactividad sobre la misma y la cantidad de cubiertos configurados.

En la parte lateral derecha se encuentra el menú de acciones de la mesa donde se puede adicionar pedidos, cerrar la cuenta, agregar descuentos, ver el detalle sobre la cuenta e imprimir ticket (estas acciones se describen más adelante).

En la parte inferior se ven las acciones disponibles sobre las adiciones. Al seleccionar una o varias de estas se vuelven disponibles las mismas, entre las cuales se puede: comandar, marcar como entregado, editar los datos o cancelarla.

## 16.8. Crear Mesa

En la siguiente interfaz se ve el formulario de creación de mesa donde simplemente se carga el nombre configurado para la mesa en cuestión y si se quiere marcar la misma como destacada o como tipo Mostrador.



The image shows a software interface for creating a new table. On the left, there is a sidebar with a list of tables numbered 1 to 13. The list is titled 'Mostrador' and has a search bar 'Buscar mesa...'. Below the list, there are three icons: a red trash can, a blue pencil, and a green plus sign. The main area is titled 'Nueva mesa:' and contains a text input field for 'Número/Nombre', two checkboxes for 'Destacada' and 'Mostrador', and 'Cancelar' and 'Guardar' buttons.

*Imagen 34: Nueva mesa*

Uno puede acceder a este formulario de carga de mesa presionando el botón de color verde situado en la parte inferior derecha del listado de mesas.

A continuación damos detalles acerca del tipo de mesa y su posibilidad de destacarla:

- Mesa destacada: al destacar una mesa esto hará que la misma se visualice de un color distinto a las demás y que tenga mayor prioridad que otras en el algoritmo de orden de listado. La idea es que cuando alguien importante se sienta sobre una

mesa marcarla como destacada para no perderla de vista y estar pendiente de su atención.

- Mesa mostrador: las mesas tipo mostrador son aquellas donde se realizan adiciones pasajeras sin necesidad de sentarse en la misma dado que simulan una compra para llevar o algo similar como delivery. Los precios que se utilizan en los productos son los cargados diferenciados bajo “Precio mostrador” en el formulario de producto.

### 16.9. Asistente de creación de mesas

La siguiente pantalla describe un asistente de creación de mesas. En la misma se le permite crear una determinada cantidad de mesas utilizando un prefijo y configurando los números desde y hasta cómo se desee.

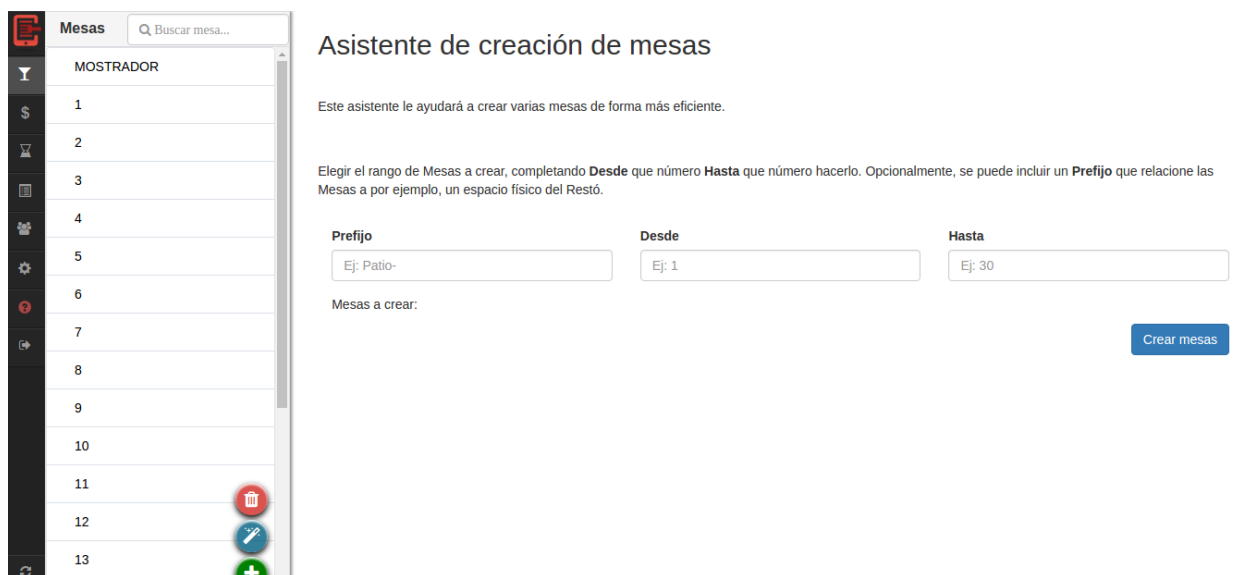


Imagen 35: Asistente de creación de mesas

Un ejemplo útil sería cuando en temporada estival se decide armar mesas sobre algún descubierto o patio y se despliegan X cantidad de mesas con el prefijo “Patio-”. Como resultado se tendrán mesas bajo el nombre Patio-1, Patio-2, Patio-3... etc.

## 16.10. Asignación de mozos

En la siguiente pantalla se ve cómo el usuario puede, opcionalmente, asignar un mozo a una mesa. Para ello, posicionado en la misma debe presionar sobre el botón que se ve en la imagen y elegir de un listado que se despliega el mozo que cree correspondiente.



Imagen 36: Asignación de mozos

Como mencionamos con anterioridad, una mesa con mozo asignado asociará todos los pedidos al mismo permitiéndole al encargado del turno tener una idea del desempeño de cada mozo y si se desea pagar en torno a una comisión según lo facturado.

## 16.11. Abrir cuenta sobre una Mesa / Adición

En la siguiente pantalla se describe como un usuario puede abrir una cuenta sobre una mesa. Para esto lo único que se debe hacer es añadir un pedido utilizando el menú de adición que se encuentra en la derecha.

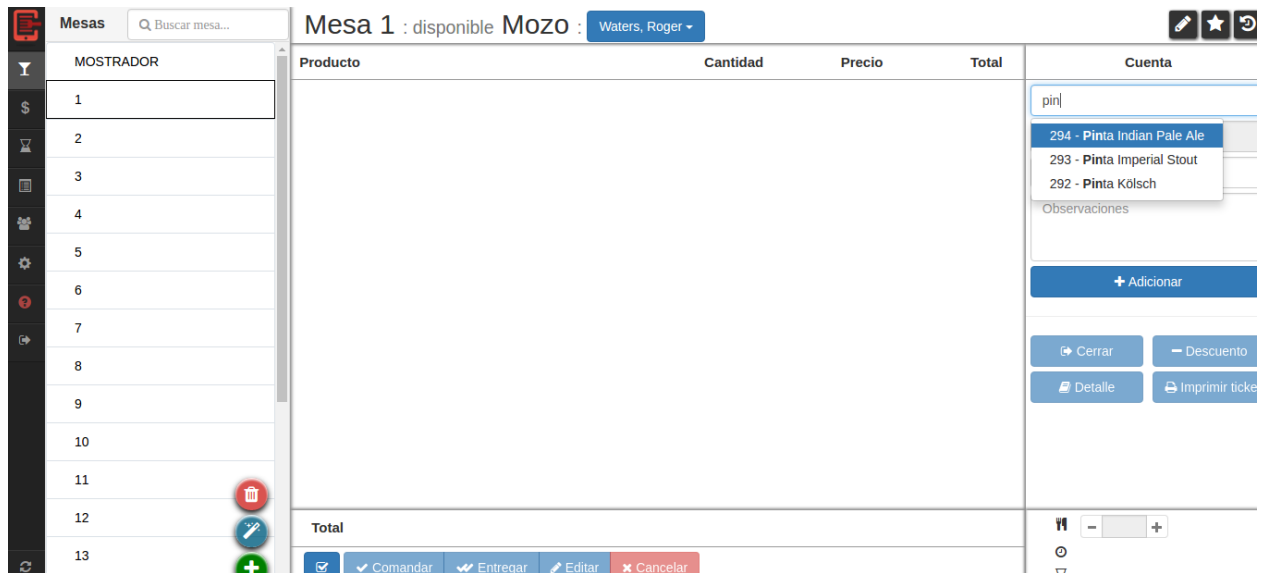


Imagen 37: Abrir cuenta sobre mesa

Los pasos para añadir un producto son:

- Posicionarse sobre el campo de texto del menú de adición
- Escribir las letras para indicar la búsqueda del producto y seleccionar el mismo cuando aparezca.
- Indicar la cantidad de unidades ordenadas sobre dicho producto
- Agregar opcionalmente observaciones (por ej.: “Sin mayonesa”)
- Presionar Adicionar

En el ejemplo de la pantalla anterior se ve cómo el usuario realiza una adición de una Pinta de cerveza.

Una vez adicionados los productos, los mismos se cargan en el listado de productos sobre la cuenta abierta en la mesa (sección central) como se ve en la siguiente pantalla:





Imagen 38: Listado de productos

A su vez en la barra lateral izquierda donde se listan las mesas disponibles del resto se refleja la cantidad de pedidos sin atender que posee la mesa y se las ordenas según una preferencia de prioridades que se explicará más adelante.

### 16.12. Comandar adición

Una de las primeras acciones a realizar luego de la adición de un producto es la “Comanda” de los mismos. Para llevarla a cabo se seleccionan los deseados y se presiona el botón que se encuentra debajo con la etiqueta “Comandar”.

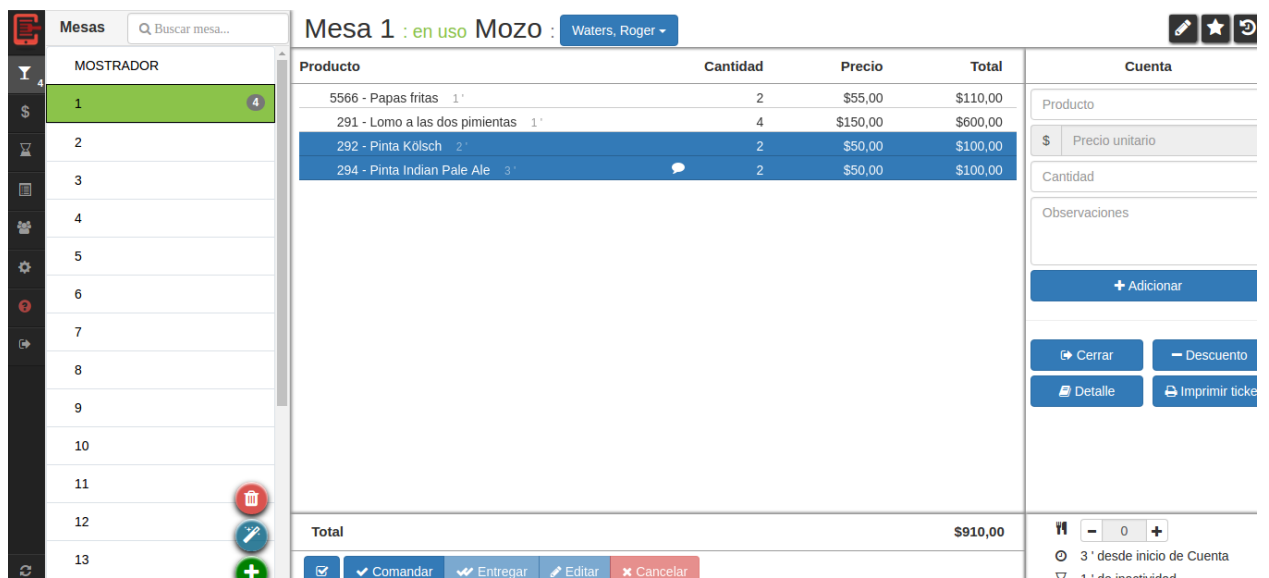


Imagen 39: Comandar adición

Al comandar una o varias adiciones, se imprime utilizando la “Comandera” una comanda (papel pequeño con detalle de los productos) que luego es enviada a la cocina para indicar lo que se necesita preparar. Posterior a esto se cambia el estado de los productos agregándole un “check” que evidencia la acción como se ve en la siguiente imagen.

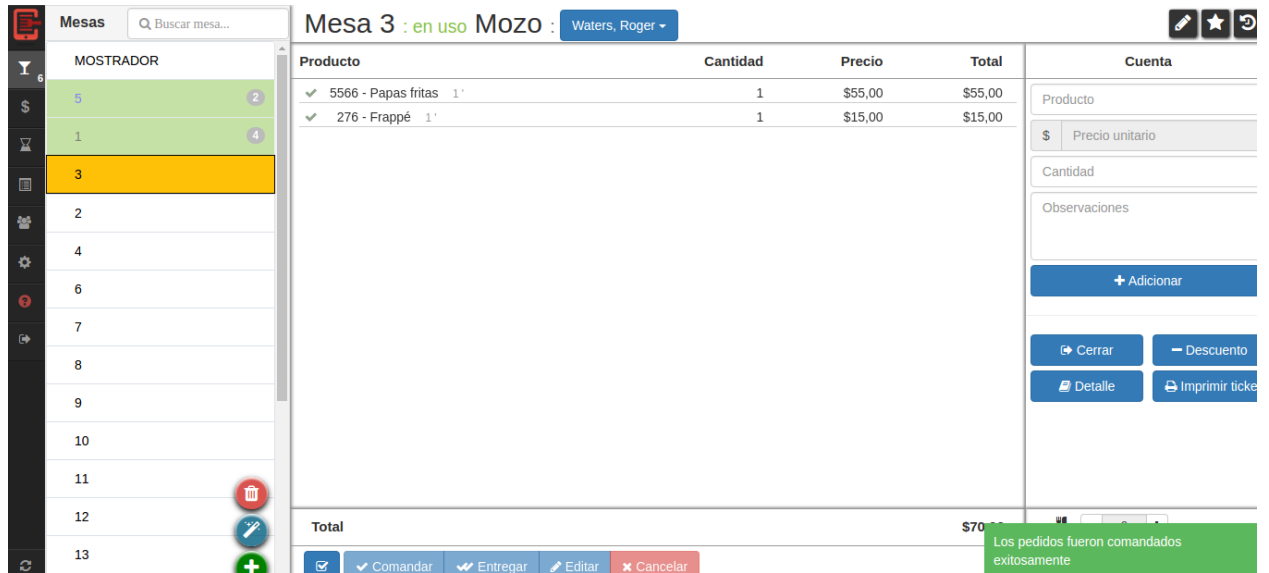


Imagen 40: Pedidos comandados

### 16.13. Entregar pedidos sobre mesa

Una vez finalizada la preparación de los productos adicionados sobre la mesa se puede asentar la entrega de los mismos seleccionándolos y presionando el botón que se encuentra debajo con la etiqueta “Entregar”.

Al ser entregados los productos se les marca con un doble check que manifiesta esto cómo es posible verlo en la siguiente imagen.

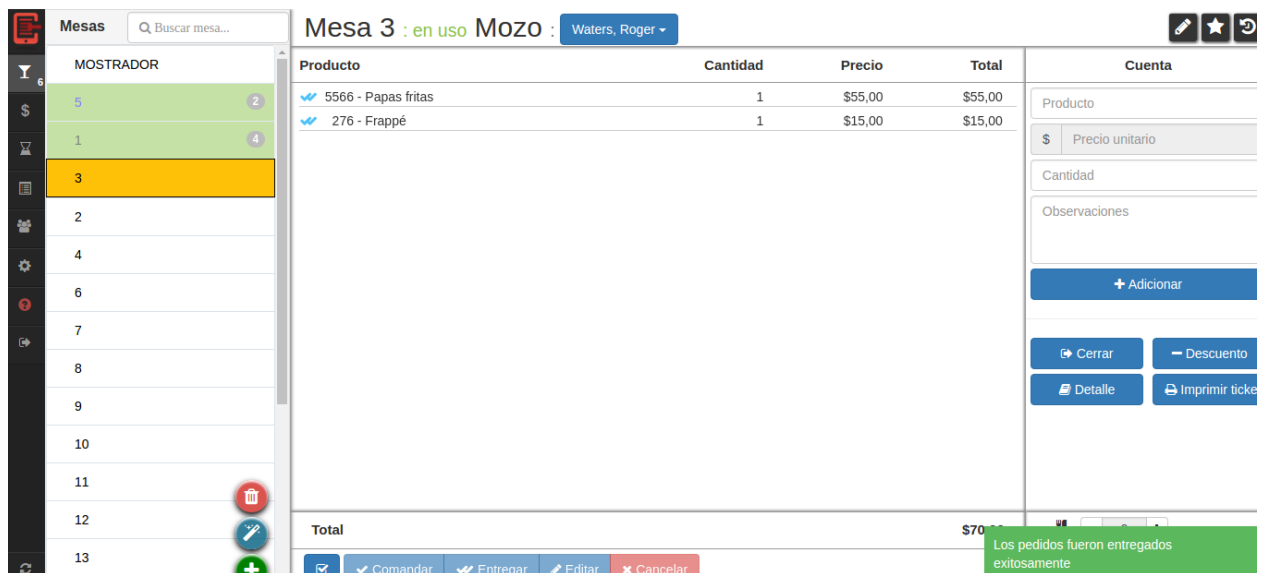


Imagen 41: Pedidos entregados

#### 16.14. Editar adición sobre Mesa

Una de las posibles acciones a realizar sobre una cuenta es la edición de las adiciones que se llevaron a cabo en la misma ya sea por un error al crearlas o una posterior modificación sobre las mismas.

Para esto el usuario debe seleccionar la adición deseada y presionar el botón que se encuentra debajo bajo la etiqueta de “Editar”. Esto abrirá una ventana tipo modal donde el usuario podrá cambiar el producto, las cantidades y la observación sobre la adición.

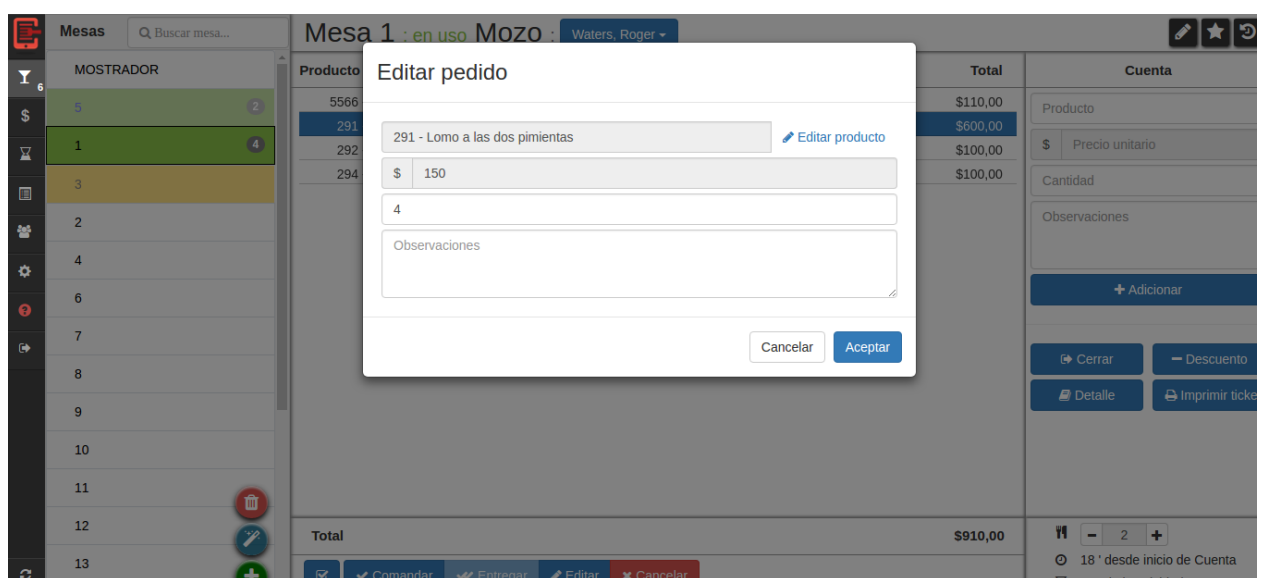


Imagen 42: Editar pedido

## 16.15. Cancelar adición sobre Mesa

Existe la posibilidad también en la que el usuario desee cancelar la adición para que no sea tenida en cuenta. Para ello selecciona la deseada desde la lista y presiona el botón que se encuentra debajo con la etiqueta “Cancelar”. De esta forma se abrirá una ventana tipo modal que pedirá confirmación de cancelación al usuario.

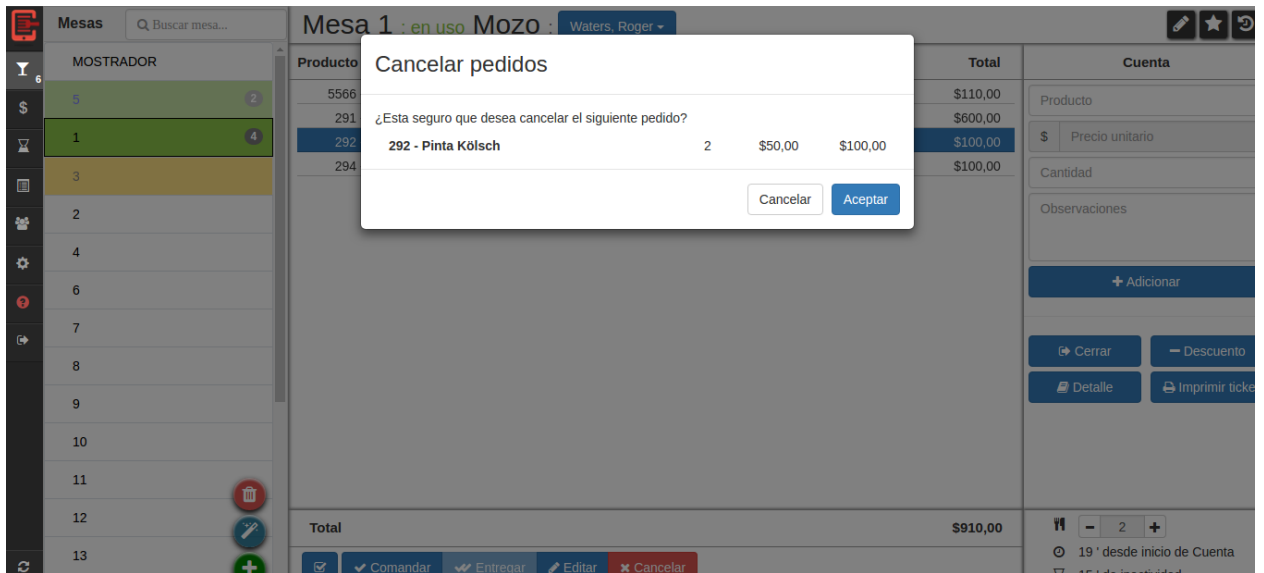


Imagen 43: Cancelar pedidos

Los pedidos cancelados quedan listados y sobrescritos por una línea horizontal como se ve en la siguiente imagen:



Imagen 44: Pedidos cancelados

### 16.16. Configurar cantidad de cubiertos

Otra de las acciones comunes al abrir una cuenta sobre una mesa es configurar (opcionalmente) la cantidad de cubiertos sobre la misma (con el fin de representar la cantidad de personas físicas sobre la mesa). Para esto el usuario tiene que presionar el botón con el signo “+” que se encuentra en la sección inferior derecha, tantas veces como sea necesario.

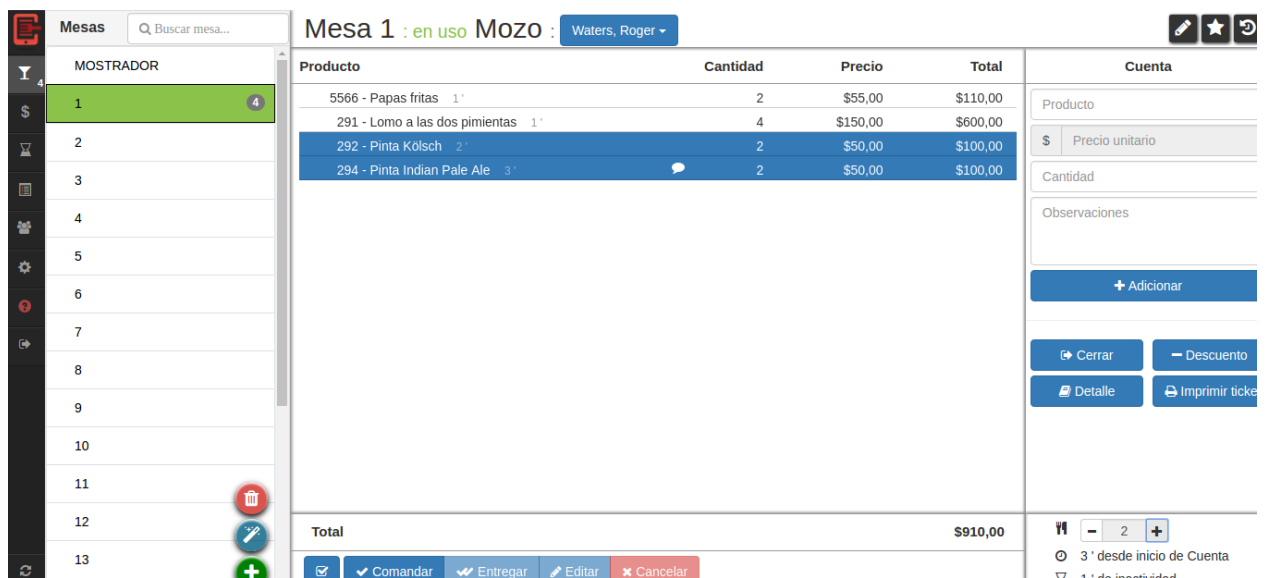


Imagen 45: Configurar cubiertos

## 16.17. Algoritmo de orden de mesas

Una de las características importantes del listado de mesas es el orden en el cual se encuentran las mismas ubicadas en la barra lateral izquierda. Para esto es que se diseñó un algoritmo cuyo fin es ponderar cada mesa y permitirle al usuario tener visibles en la parte superior las mesas más importantes o que precisan ser atendidas de manera urgente.

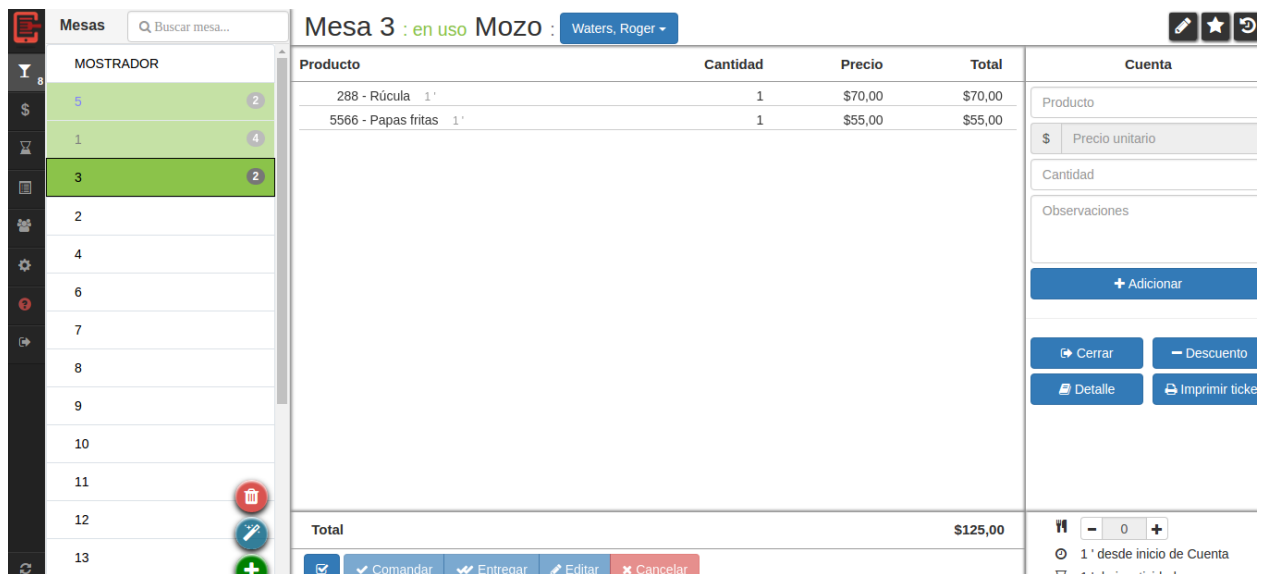


Imagen 46: Vista de mesas ponderadas

El algoritmo tiene en cuenta el siguiente nivel de ponderaciones:

- **Mesa mostrador:** las mesas de tipo mostrador reciben la mayor ponderación por lo que se ubican en la parte superior. La idea es que las mismas se usan para pedidos rápidos y/o delivery. En la imagen anterior se puede ver la mesa “MOSTRADOR” como ejemplo
- **Cuenta Activa:** el segundo nivel de ponderación tiene en cuenta a aquellas mesas que poseen una cuenta activa. En la imagen anterior tenemos a las mesas de un color verde como aquellas que tienen cuentas activas.
- **Mesa Destacada:** el tercer nivel de ponderación tiene que ver con aquellas mesas que han sido marcadas como destacadas. Gráficamente esto se distingue con una visión tipo link sobre la mesa destacada.
- **Pedidos pendientes:** el último nivel de ponderación tiene que ver con la cantidad de pedidos pendientes que tiene la mesa y agrega una peso a la misma cada uno de estos. Los pedidos pendientes son aquellos adicionados que todavía no han

sido entregados a los comensales. Por mesa se puede ver la cantidad de los mismos.

### 16.18. Detalle de cuenta sobre mesa

Posicionado sobre la vista de una mesa el usuario tiene la posibilidad de ver el detalle de la misma con los datos relevantes a la cuenta que permanece abierta. Para ello debe presionar el botón que aparece en la sección de acciones sobre la cuenta y que tiene la etiqueta “Detalle”. Una vez que presione, se abre una ventana de tipo modal donde se listan los productos, sus cantidades, precio unitario, precio por cantidades, descuentos y monto consumido total sobre la mesa.

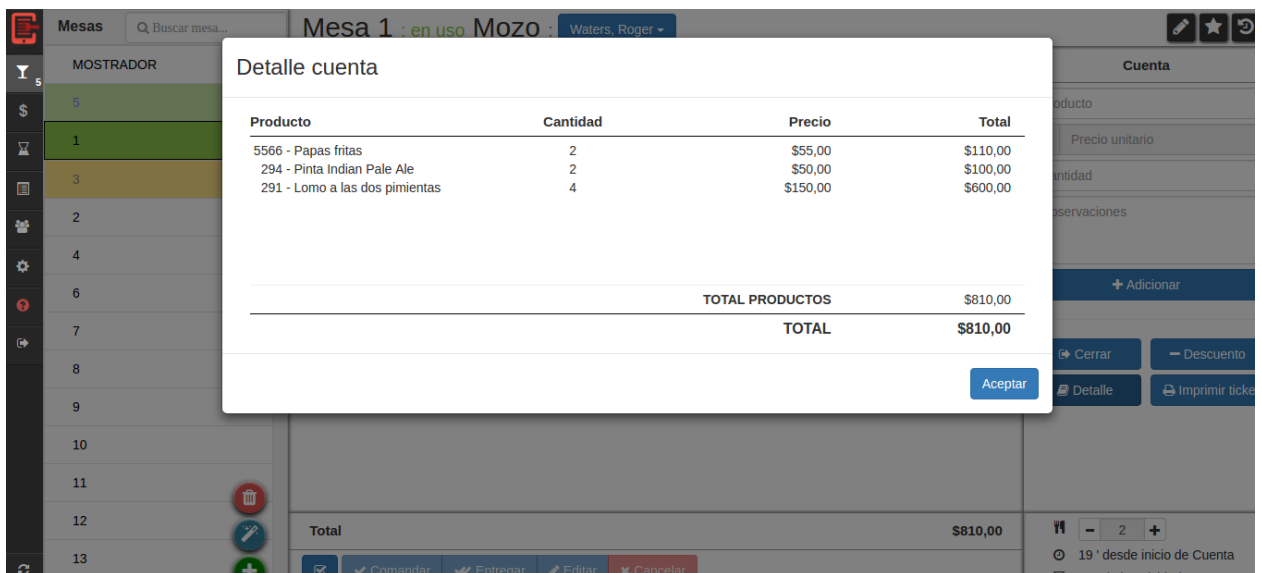


Imagen 47: Detalle de una cuenta

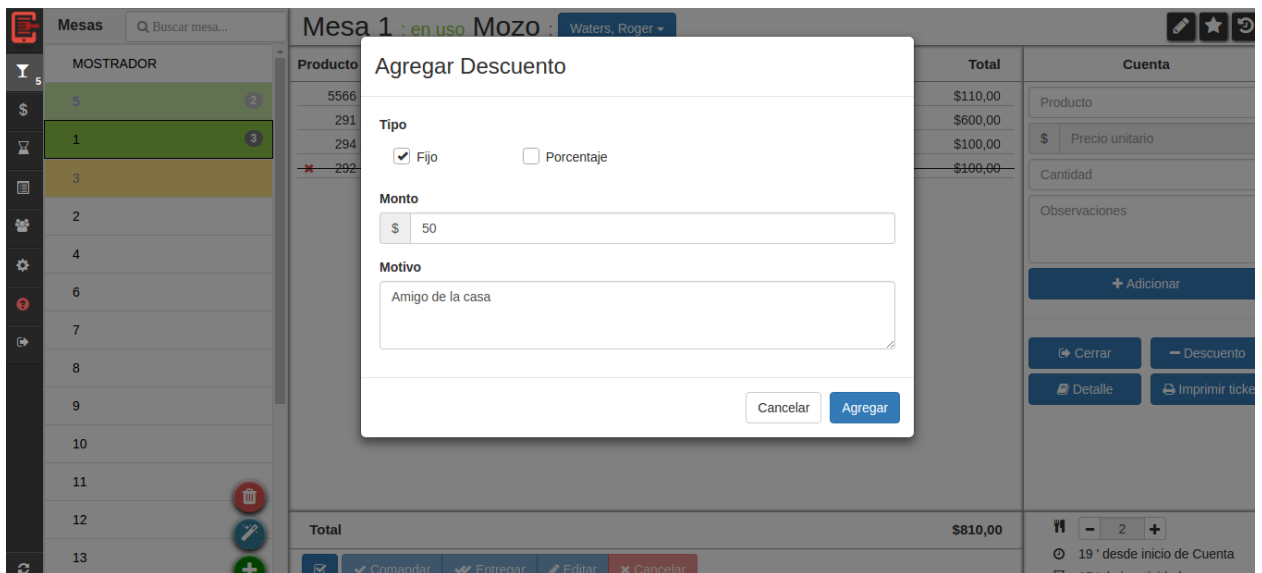
### 16.19. Agregar descuento sobre cuenta en una mesa

El usuario cuenta con la posibilidad de añadir un descuento sobre una cuenta abierta en una mesa para reflejar el uso de alguna promoción o por el simple hecho de bajar el costo de la misma.

Para ello, debe presionar sobre el botón con la etiqueta “Descuento” y se abrirá una ventana tipo modal. Dentro de esa ventana deberá configurar el tipo de descuento y el motivo del mismo. Existen 2 tipos de descuentos:

- Fijo: monto fijo sobre la cuenta

- Porcentaje: un porcentaje del total consumido en la cuenta



*Imagen 48: Agregar descuento en una cuenta*

## 16.20. Historial de cuentas sobre mesa

El usuario tiene la posibilidad de ver el historial de todas las cuentas cerradas sobre una mesa determinada posicionándose sobre la misma y presionando el botón superior derecho con una imagen de un reloj.

Al acceder al historial de la mesa se podrá ver la lista de cuentas sobre dicha mesa con una breve descripción de los montos, cantidad de cubiertos, descuentos y hora de cierre.

Además, cada historia de cuenta posee un botón con la etiqueta “Ver detalle” que al ser presionado abrirá una ventana de tipo modal con el detalle de la cuenta (similar al detalle anteriormente visto).



Fecha	Estado	Montos	Cubiertos	Descuentos	Cierre
Hoy	Cobrada	\$760,00 total \$760,00 pagado	2	\$50,00	18:33
18/05/2017	Cobrada	\$150,00 total \$150,00 pagado	0	\$0,00	17:35
18/04/2017	Cobrada	\$30,00 total \$30,00 pagado	0	\$10,00	16:52
18/04/2017	Cobrada	\$160,00 total \$160,00 pagado	0	\$0,00	16:51
18/04/2017	Cobrada	\$50,00 total \$50,00 pagado	0	\$0,00	18:04

Imagen 49: Historial de cuentas de una mesa

### 16.21. Listado de turnos y movimientos

El usuario podrá realizar la gestión de los turnos sobre su resto ingresando al menú de los Turnos (Menú número 4 detallado anteriormente) como se ve en la siguiente imagen.

Hora	Concepto	Efectivo	Efectivo B	Crédito	Débito	Total
18:07 (Steve Jobs)						
18:35:26	Cobro de cuenta en mesa 3	---	---	\$70,00	---	\$70,00
18:34:55	Cobro de cuenta en mesa 5	\$160,00	---	---	---	\$160,00
18:33:32	Cobro de cuenta en mesa 1	\$700,00	\$60,00	---	---	\$760,00
18:15:11	Cobro de cuenta en mesa 3	\$125,00	---	---	---	\$125,00
18:07:51	Apertura de caja	\$558,60	---	---	---	\$558,60
<b>Total</b>		<b>\$1.543,60</b>	<b>\$60,00</b>	<b>\$70,00</b>	<b>\$0,00</b>	<b>\$1.673,60</b>
17:53 / 17:53 (Walter White)						
17:50 / 17:50 (Line Craker)						

Imagen 51: Listado de turnos y movimientos

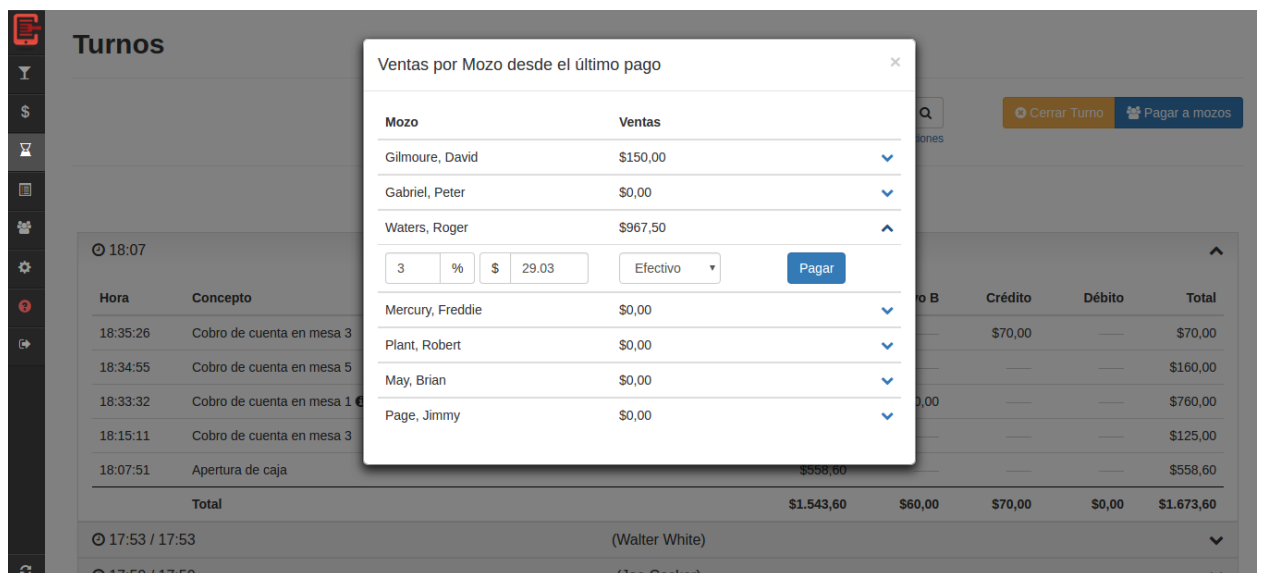
En dicha interfaz tendrá un listado de turnos colapsados con el nombre de su encargado. Al hacer click sobre un turno se desplegará un listado de movimientos realizados en dicho turno con el detalle del mismo como hora, concepto, tipo de caja a la que imputa y total impactado.

Los movimientos listados sobre un turno son todos aquellos que tuvieron lugar luego de la apertura del mismo y pueden ser tanto movimientos del tipo manual como aquellos generados por el sistema a la hora de realizar un cierre y cobro de cuenta como el pago a los mozos.

## 16.22. Realizar pago a mozos

Desde la pantalla de gestión de turnos el usuario tiene la posibilidad de realizar pago a los mozos que ha configurado sobre su restó. Para esto debe presionar el botón con la etiqueta “Pagar a mozos” y una ventana tipo modal se abrirá.

Dentro de dicha ventana se listaran los mozos y un total de lo facturado durante el turno. Al hacer click se despliega un formulario en línea donde se podrá configurar el pago hacia el mismo utilizando el porcentaje pre-configurado o sobrescribiendo con uno deseado.



The screenshot shows a modal window titled "Ventas por Mozo desde el último pago" (Sales by Staff since the last payment). The modal contains a table with the following data:

Mozo	Ventas
Gilmoure, David	\$150,00
Gabriel, Peter	\$0,00
Waters, Roger	\$967,50
Mercury, Freddie	\$0,00
Plant, Robert	\$0,00
May, Brian	\$0,00
Page, Jimmy	\$0,00

Below the table, there is a form with the following fields:

- A percentage input field containing "3" and a dropdown menu set to "%".
- A currency input field containing "\$" and a value of "29.03".
- A payment method dropdown menu set to "Efectivo".
- A blue "Pagar" button.

The background shows a "Turnos" (Shifts) screen with a table of transactions and a "Pagar a mozos" button.

Imagen 52: Pago a mozos

## 16.23. Cajas y movimientos

El usuario puede ingresar a la gestión de Caja accediendo al menú (Menú número 3 detallado anteriormente) como se muestra a continuación.

Hora	Concepto	Efectivo	Efectivo B	Crédito	Débito	Total
07/06/2017						
18:40	Pago a Mozo Waters, Roger	-\$29,03	---	---	---	-\$29,03
18:35	Cobro de cuenta en mesa 3	---	---	\$70,00	---	\$70,00
18:34	Cobro de cuenta en mesa 5	\$160,00	---	---	---	\$160,00
18:33	Cobro de cuenta en mesa 1	\$700,00	\$60,00	---	---	\$760,00
18:15	Cobro de cuenta en mesa 3	\$125,00	---	---	---	\$125,00
18:07	Apertura de caja	\$558,60	---	---	---	\$558,60
Turno		\$1.115,00				
Dia		\$1.415,00				
Mes		\$1.415,00				

Imagen 53: Cajas y movimientos

Como se ve en la pantalla el usuario tiene la posibilidad de visualizar los movimientos realizados segregados por fecha con una breve descripción de los mismos.

Además posee una visión de lo que cada una de las cajas tiene registrado hasta el momento en relación a los movimientos realizados sobre las mismas.

Si lo desea, el usuario puede realizar una búsqueda sobre los movimientos mediante el ingreso de texto o seleccionando las fechas en las opciones avanzadas.

## 16.24. Nuevo movimiento sobre caja

Desde la gestión de cajas tiene la posibilidad de agregar un movimiento presionando el botón con la etiqueta “Nuevo movimiento”. Al hacerlo se le abrirá una ventana tipo modal donde podrá ingresar los detalles del movimiento a realizar indicando:

- Tipo (Ingreso, Egreso, Retiro de dinero)
- Concepto
- Detalle
- Monto por caja que implica el movimiento

Debito	Total
	-\$200,00
	-\$29,03
	\$70,00
	\$160,00
	\$760,00
	\$125,00
Turno	\$1.115,00
Dia	\$1.415,00
Mes	\$1.415,00

Imagen 54: Nuevo movimientos sobre caja

Al presionar el botón “Aceptar” se ejecutará el movimiento y se impactarán en las cajas especificadas según el tipo configurado.

## 16.25. Configuración de las cajas

El usuario tiene la posibilidad de realizar una breve configuración sobre cada caja que posee en su resto (Por defecto se tienen 4 cajas).

Las posibilidades a configurar son:

- Nombre de la caja

- Imprimir ticket fiscal: Si la caja está configurada con la opción de imprimir ticket fiscal, a la hora de cerrar la cuenta y asignarle un monto a dicha caja se realiza la impresión un ticket fiscal por el monto deseado.
- Vuelta a \$0: Si la caja se encuentra configurada con la opción vuelta a 0 al realizar un cierre de turno la misma vuelve a cero automáticamente generando un movimiento de cierre de caja.

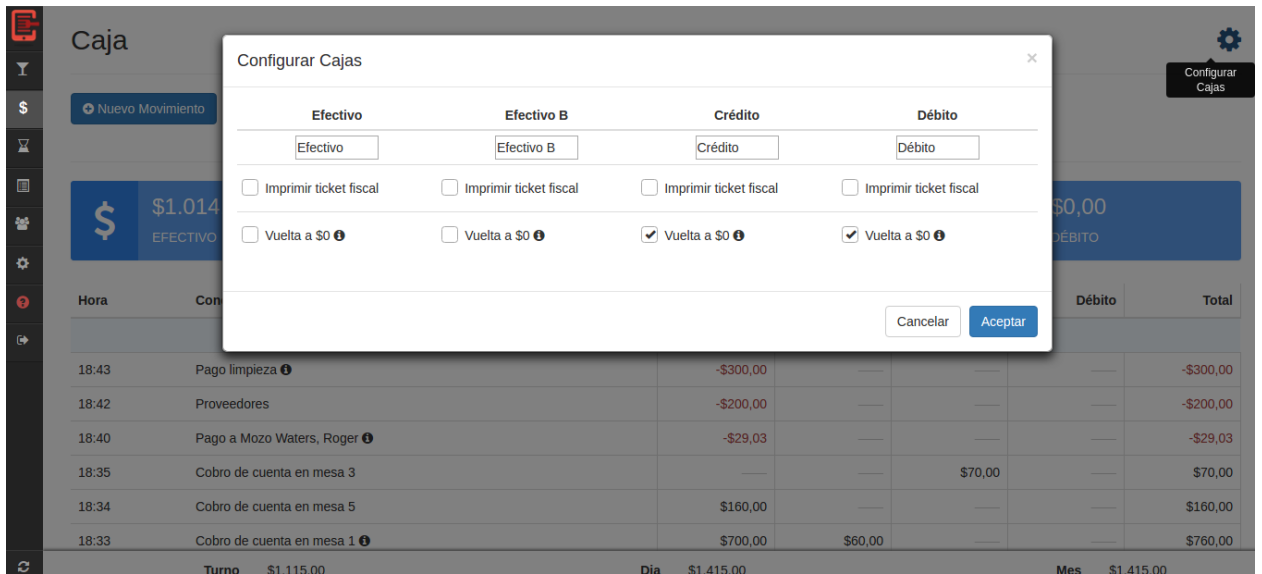


Imagen 55: Configurar cajas

## 16.26. Cerrar Turno

El usuario tiene la posibilidad de realizar el cierre del turno accediendo a la opción de menú “Administración de turnos” (Opción 4) y presionando el botón con la etiqueta “Cerrar Turno”.

Al presionar se abrirá una ventana tipo modal donde el usuario podrá visualizar el responsable del mismo y los totales en cada una de las cajas configuradas en el resto. Si lo desea también podrá dejar un comentario en forma de observación para que quede grabado al momento de cerrar el Turno. Finalmente presionando el botón Aceptar que se ve en la siguiente imagen se procede con el cierre del turno.

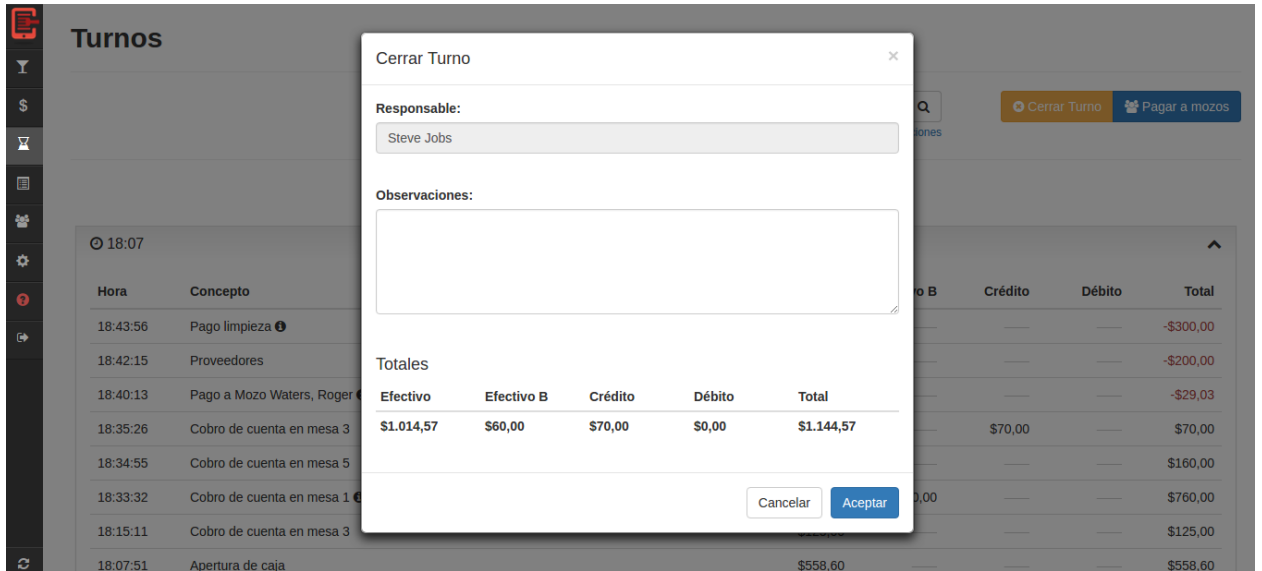


Imagen 56: Cerrar un turno

Ni bien es ejecutada la acción aparece un nuevo item de turno en la lista que se encuentra en la misma página como se muestra en la siguiente imagen y el usuario ya se encuentra habilitado a comenzar uno nuevo.

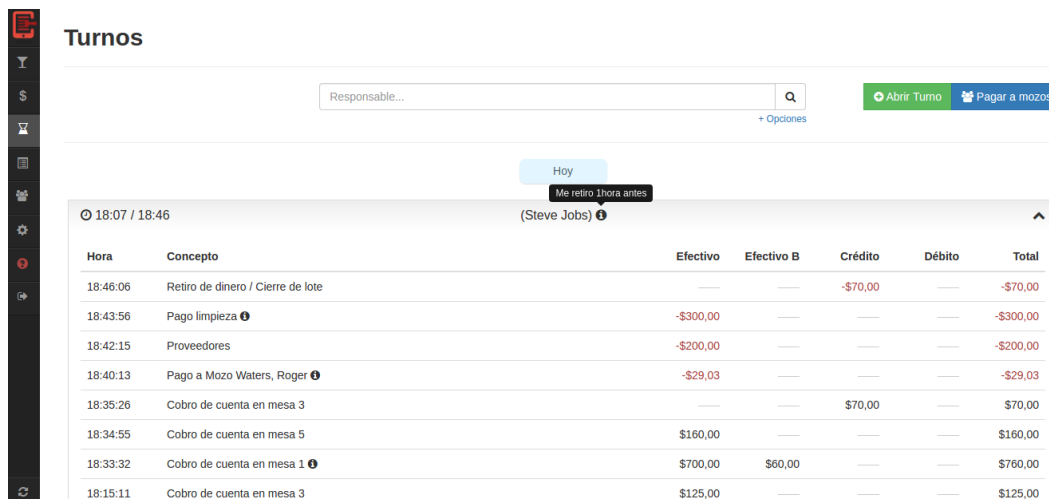


Imagen 57: Vista nuevo turno en listado