

|  |  |  |
| --- | --- | --- |
| Proyecto Final de Carrera |  | |
| CAMPAGNA, Daniela  YUNGES, Bárbara | | Ingeniería en Sistemas de Información |

*“Diseño e implementación de sistema de E-Commerce, facturación y gestión de stock para empresa de reparación, venta de repuestos y accesorios para motos”*

Director: Ing. David Campo

– 2018 –

Contenido

[1. Introducción 4](#_Toc533797259)

[*1.1. Delimitación del tema* 4](#_Toc533797260)

[*1.2. Objetivo general* 4](#_Toc533797261)

[*1.3. Objetivos específicos* 5](#_Toc533797262)

[*1.4. Ámbito del sistema* 5](#_Toc533797263)

[*1.5. Antecedentes a la solución* 5](#_Toc533797264)

[*1.6. Aportes que se espera realizar* 6](#_Toc533797265)

[*1.7. Organización del informe* 6](#_Toc533797266)

[2. Tecnologías utilizadas 7](#_Toc533797267)

[*2.1. Aplicación Web* 7](#_Toc533797268)

[2.1.1. Capa del navegador 8](#_Toc533797269)

[2.1.2. Capa del Servidor 8](#_Toc533797270)

[2.1.3. Persistencia de datos 9](#_Toc533797271)

[2.1.4. Versionado 9](#_Toc533797272)

[3. Metodología de resolución 13](#_Toc533797273)

[*3.1. Etapa 1: Revisión bibliográfica e investigación preliminar* 13](#_Toc533797274)

[*3.2. Etapa 2: Desarrollo del sistema utilizando la metodología de desarrollo Incremental* 14](#_Toc533797275)

[3.2.1. Etapa Inicial 14](#_Toc533797276)

[3.2.2. Etapa de Análisis 14](#_Toc533797277)

[3.2.3. Etapa de Diseño 15](#_Toc533797278)

[3.2.4. Etapa de Codificación 15](#_Toc533797279)

[3.2.5. Etapa de Prueba 15](#_Toc533797280)

[3.2.6. Etapa de Instalación y Capacitación 15](#_Toc533797281)

[*3.3. Actividades predecesoras y sucesoras* 15](#_Toc533797282)

[*3.4. Actores* 16](#_Toc533797283)

[*3.5. Plan de Incrementos* 17](#_Toc533797284)

[*3.6. Requerimientos del sistema* 17](#_Toc533797285)

[3.6.1. Requerimientos Funcionales 17](#_Toc533797286)

[3.6.2. Requerimientos no Funcionales 18](#_Toc533797287)

[*3.7. Diagrama de Casos de Uso* 20](#_Toc533797288)

[3.7.1. Relación de herencia entre los actores 20](#_Toc533797289)

[3.7.2. Casos de uso del actor Usuario 21](#_Toc533797290)

[3.7.3. Casos de uso del actor Cliente Web 21](#_Toc533797291)

[3.7.4. Casos de uso del actor Empleado 21](#_Toc533797292)

[3.7.5. Casos de uso del actor Administrador 23](#_Toc533797293)

[*3.8. Diagrama de Clases* 23](#_Toc533797294)

[*3.9. Diagrama de Actividades* 25](#_Toc533797295)

[4. Diseño del sistema 27](#_Toc533797296)

[*4.1. Diseño preliminar de interfaces gráficas* 27](#_Toc533797297)

[*4.2. Modelo de entidad-relación* 61](#_Toc533797298)

[5. Arquitectura del sistema 63](#_Toc533797299)

[6. Construcción 63](#_Toc533797300)

[*6.1. Iteración 0* 66](#_Toc533797301)

[6.1.1. IDE para el desarrollo de la aplicación 66](#_Toc533797302)

[6.1.2. Elección del repositorio remoto 70](#_Toc533797303)

[6.1.3. Cierre de la iteración 0 71](#_Toc533797304)

[*6.2. Iteración 1* 71](#_Toc533797305)

[6.2.1. Cierre de la iteración 1 72](#_Toc533797306)

[*6.3. Iteración 2* 72](#_Toc533797307)

[6.3.1. Cierre de la iteración 2 72](#_Toc533797308)

[*6.4. Iteración 3* 73](#_Toc533797309)

[6.4.1. Cierre de la iteración 3 73](#_Toc533797310)

[7. Conclusiones 74](#_Toc533797311)

[*7.1. Conclusiones finales* 74](#_Toc533797312)

[*7.2. Trabajos futuros* 75](#_Toc533797313)

[8. Bibliografía 76](#_Toc533797314)

[*8.1. Referencias bibliográficas* 76](#_Toc533797315)

[*8.2. Herramientas y programas utilizados* 76](#_Toc533797316)

[*8.3. Bibliografía y sitios consultados* 76](#_Toc533797317)

# 1. Introducción

## *1.1. Delimitación del tema*

Hoy en día, la competencia en el mercado de bienes y servicios ha llegado a límites en los que temas como la facilidad de compra de un producto desde el hogar y la promoción online son delimitantes en el éxito de un negocio.

Los comerciantes que triunfan con sus emprendimientos son, en su mayoría, aquellos que supieron apostar por la venta y promoción de sus productos a través de Internet. Por otro lado, desde el punto de vista de un consumidor tipo, cada vez se visualiza más una tendencia a la compra de necesidades a través de una computadora en su hogar y recibiendo la mercadería a través del correo en su mismo domicilio.

En esta tendencia, se agrega el hecho de que, los clientes exigen conocer de antemano el precio de un producto promocionado en Internet, para decidir si dirigirse al negocio a comprarlo o no. Esto posibilita y genera una gran ventaja para el consumidor, el cual está en la posición de poder comparar precios, calidades y ventajas del mismo producto o similar en algún otro comercio. En contraparte, implica una desventaja para el comerciante, el cual en años anteriores habría tenido al cliente ya en su negocio generándole un costo de decisión: “Si tener que gastar tiempo en recorrer otros negocios preguntando precios y observando calidades o comprar el producto en el negocio en el que se encuentra.”

Es por esto que planteamos el desarrollo de un sistema que permita utilizar las facilidades que ofrece el uso de la tecnología de la información. Este sistema brindará la flexibilidad necesaria al comercio para poder disminuir costos, obtener un alcance mayor y continuo de promociones de sus productos y adecuarse a las necesidades actuales del cliente.

## *1.2. Objetivo general*

Desarrollar un sistema que permita a los potenciales consumidores acceder rápidamente y de manera sencilla al catálogo de productos del comercio y poder visualizar precios, información e imágenes del producto que desea adquirir para luego agregarlos a un carrito virtual y poder pagarlos y recibirlos en su hogar si así lo quisiese. Además, el sistema contemplará la administración del stock en el comercio y admitirá tanto ventas presenciales como la gestión de las reparaciones de los vehículos recibidos en el taller. Esto permitirá, por un lado, mantener un seguimiento personalizado de cada cliente, y por otro, le brindará al usuario un historial detallado de los productos y reparaciones de cada vehículo que posea.

## *1.3. Objetivos específicos*

Para alcanzar lo descrito anteriormente, se proponen los siguientes objetivos específicos:

* Analizar estructuras para el acceso interactivo a catálogos de productos web, precios y folletos publicitarios de la empresa.
* Analizar e implementar el tipo de comercio B2C (Business to Consumer) como forma de comercio electrónico para operaciones entre la empresa y usuario final.
* Realizar el análisis, diseño e implementación del rol para gestión de productos y clientes (ABMC Clientes, ABMC Productos, y relación entre ambos)
* Implementar un historial de compra en el que se genere una asociación entre el cliente, su vehículo y el repuesto adquirido mediante la web, personalmente en el local y los utilizados en alguna reparación.
* Implementar venta online de productos mediante Mercado Pago y brindar la posibilidad de envíos a domicilio mediante Mercado Envíos.
* Diseñar y desarrollar una interfaz web amigable que funcione correctamente en dispositivos móviles.

## *1.4. Ámbito del sistema*

La aplicación será utilizada tanto por personal del local como por clientes que tengan el interés de realizar compras por internet. El personal del local está bien definido, por lo tanto serán tres personas involucradas, el dueño del local y dos empleados que utilizarán el sistema. Por otro lado, no es posible saber de antemano la cantidad de usuarios que accederán desde la web para realizar consultas o compras, pero en base a estimaciones del mismo dueño, se puede concluir que la utilizarían entre cien y doscientas personas a la semana.

## *1.5. Antecedentes a la solución*

La gran y mayor importancia del sistema radica en que, en este momento, el cliente no posee ninguna solución informática para realizar sus ventas. Recientemente adquirió un nuevo negocio, el cual trajo consigo gran cantidad de productos y repuestos para motos de competición. En la actualidad solamente se dedica a vender presencial en su local con algún tipo de publicidad en las redes sociales.

Debido a la problemática planteada anteriormente, esta metodología de ventas le implica una desventaja muy grande por sobre sus competidores, con lo cual el sistema de comercio electrónico planteado le sería de gran ayuda para poder posicionarse entre los pioneros del rubro en la venta online.

Hoy en día, solamente es posible comprar este tipo de productos para motos por Mercado Libre o en escasos negocios online de tiendas del rubro, las cuales carecen de tecnologías novedosas y no permiten visualizar de forma amigable los productos y su información, sumado a que no implementan métodos de seguridad para la parte de pago online.

## *1.6. Aportes que se espera realizar*

Esperamos que este proyecto contribuya a mejorar la relación oferta-demanda entre los clientes y el negocio. Además, creemos que le brindará al cliente una herramienta competitiva y novedosa para el ámbito del mercado automotriz.

Desde el punto de vista de nuestra formación profesional, sin dudas el desarrollo de este sistema nos permitirá afianzar, aplicar y llevar a la práctica los conocimientos adquiridos en la carrera y los incorporados durante el desarrollo del proyecto.

## *1.7. Organización del informe*

El informe se estructurará en varios capítulos. Comenzaremos describiendo las tecnologías utilizadas en el desarrollo del sistema, explicando el porqué de la elección junto con un breve desarrollo de cada una. Para ello se dividieron las tecnologías en sub-grupos bien definidos y se las separó en ítems dentro de ellos.

En el próximo capítulo se ubica la metodología de desarrollo que adoptamos, la cual decidimos estructurar en etapas, la etapa de investigación y conocimiento tanto de las tecnologías a utilizar como de los requerimientos definidos por el usuario y la mejor forma de llevarlos a cabo, y la etapa del desarrollo del sistema utilizando la metodología elegida para hacerlo. En este capítulo se comenta también el orden en el cual se llevarán a cabo los requerimientos del usuario y se describirán brevemente algunos diagramas significativos del análisis del sistema.

A continuación se sitúa el modelo de diseño del sistema, en el cual se muestra y se explica con detalles cada interfaz del sistema junto con el modelo de entidad-relación para comprender la estructura que se utiliza para persistir los datos.

En el siguiente capítulo se muestra la arquitectura global del sistema y se comenta brevemente lo utilizado. Luego, en el capítulo 6 nos centramos en la construcción del sistema con más detalle explicando además lo que incluyó cada iteración.

Por último, se encuentra en el capítulo 7 las conclusiones que sacamos tanto del proyecto final como del sistema en sí, y terminando con la bibliografía consultada para la elaboración de este informe en el capítulo 8.

Cabe destacar, que se incluyen tres anexos, ya que consideramos que el material era de importancia suficiente pero no primordial de agregar en este documento. En el Anexo A se incluyeron la descripción de los casos de uso, para poder comprender los flujos normales y alternativos de cada uno. Por otro lado, en el Anexo B, se agregó un documento de casos de pruebas realizados durante el desarrollo del sistema. Y, por último, en el Anexo C se agregó información sobre la gestión de riesgos y planes de contingencia que utilizamos durante el transcurso del desarrollo del sistema.

# 2. Tecnologías utilizadas

Las tecnologías que se explican a continuación son las elegidas para llevar adelante el desarrollo del proyecto. Se explican todas las necesarias para la implementación del “front-end” y el “back-end” del sitio respectivamente.

## *2.1. Aplicación Web*

Aunque existen muchas variaciones posibles, una aplicación web está normalmente estructurada como una aplicación de tres capas. En su forma más común, el navegador web ofrece la primera capa, interpretando el código. El servidor que ofrece este código y toda la información es la segunda capa. Por último, una base de datos constituye la tercera y última capa. Esta es la arquitectura que adquirimos para realizar la aplicación.

El navegador web manda peticiones a la capa intermedia, la cual ofrece servicios valiéndose de consultas y actualizaciones a la base de datos, y, a su vez, proporciona una interfaz de usuario. [1]

Para cada una de estas capas, se utilizarán las siguientes tecnologías:

### 2.1.1. Capa del navegador

Son muy utilizados lenguajes o arquitecturas que no son propiamente lenguajes de programación, como HTML o XML. Se utilizan para servir los datos adecuados a las necesidades del usuario, en función de cómo hayan sido definidos por el dueño de la aplicación.

Los desarrolladores web generalmente utilizan lenguajes interpretados (scripts) en el lado del cliente para añadir más funcionalidades, especialmente para ofrecer una experiencia interactiva que no requiera recargar la página cada vez (lo que suele resultar molesto a los usuarios).

Los lenguajes y herramientas que se utilizarán son los siguientes:

1. HTTP,
2. HTML,
3. CSS,
4. JavaScript,
5. JQuery,
6. Ajax,
7. Bootstrap.

### 2.1.2. Capa del Servidor

Se investigó entre los posibles lenguajes a utilizar en esta etapa importante y se llegó a la conclusión de utilizar C#, ya que se cuenta con una experiencia previa de utilización.

Para la parte de pruebas de código, se utilizarán herramientas brindadas por el mismo entorno de desarrollo. Pero, al emplear la versión gratuita del mismo, no trae incorporado ninguna herramienta útil para saber el porcentaje de cobertura de código de los tests. Para ello, investigando, se llegó a la conclusión de utilizar la extensión libre y gratuita AxoCover, disponible en GitHub para su descarga y acople con Visual Studio. Esta herramienta indica mediante distintos colores en el código aquél que fue alcanzado por las pruebas y aquél que no.

### 2.1.3. Persistencia de datos

Debido a su excelente integración con el IDE (entorno de programación), para almacenar los datos se eligió SQL Server, la cual se implementará localmente en una computadora dedicada a tal trabajo por el cliente en su local. Posteriormente, y ya que no se cuenta aún con datos certeros sobre la cantidad de personas que ingresarán al sitio web, se analizará la performance en las consultas y se tomará la decisión de adquirir almacenamiento en la nube. Esto se tendrá en cuenta como trabajo a futuro y en base al análisis del funcionamiento en el tiempo del sistema. Una de las tecnologías sugeridas es Azure.

Además, se utilizará Entity Framework para el acceso a datos.

### 2.1.4. Versionado

Para el control de versiones de código se decidió utilizar Git, ya que tiene una excelente integración con el IDE, lo cual permite un uso rápido y fácil.

Cada tecnología nombrada se explicará a continuación:

a) HTTP

*Hypertext Transfer Protocol (HTTP)*  es un protocolo de la capa de aplicación para la transmisión de documentos hipermedia, como HTML. Fue diseñado para la comunicación entre los navegadores y servidores web, aunque puede ser utilizado para otros propósitos también. Sigue el clásico modelo cliente-servidor, en el que un cliente establece una conexión, realizando una petición a un servidor y espera una respuesta del mismo. Se trata de un protocolo sin estado, lo que significa que el servidor no guarda ningún dato (estado) entre dos peticiones. [2]

b) HTML

HyperText Markup Language (HTML) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página. Determina el contenido de la página web, pero no su funcionalidad. Otras tecnologías distintas de HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript). [3]

c) CSS

Cascading Style Sheets (CSS) es el lenguaje utilizado para describir la presentación de documentos HTML o XML. CSS describe cómo debe ser renderizado el elemento estructurado en pantalla, en papel, hablado o en otros medios.

Desarrollado en niveles, CSS1 es ahora obsoleto, CSS2.1 es una recomendación y CSS3, ahora dividido en módulos más pequeños, está progresando en camino al estándar. [4]

d) JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. [5]

e) JQuery

JQuery es una biblioteca de JavaScript que permite simplificar el desarrollo de páginas web. Es considerado un framework de JavaScript debido a que contiene un conjunto de funciones que ya fueron desarrolladas y probadas, estando listas para ser utilizadas de una manera muy simple. Esto permite lograr disminuir los tiempos y desarrollar aplicaciones más eficientes.

JQuery es software libre y de código abierto. A su vez permite programar sin preocuparse por la compatibilidad con el navegador que  utilice el usuario ya que es compatible con varios navegadores y es ampliamente utilizado por los desarrolladores. [6]

f) AJAX

Asynchronous JavaScript And XML (Ajax) es una técnica mediante la cual las aplicaciones se ejecutan en el cliente, es decir, en el navegador, y se mantiene una comunicación asíncrona con el servidor en segundo plano. Esto permite realizar cambios sobre las páginas web, comunicándose con el servidor, sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones. Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores gracias a que está basado en estándares abiertos como JavaScript. [7]

g) Bootstrap

Bootstrap es un framework de Twitter que simplifica el proceso de creación de sitios y aplicaciones web, conteniendo una colección de herramientas de software libre. A partir de la versión 2.0 soporta diseños sensibles, que significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (ordenador, tablets o celular).

Bootstrap cuenta con una serie de plantillas CSS y ficheros JavaScript que permiten integrar el framework de forma sencilla en los proyectos webs y ahorran mucho tiempo y esfuerzo. Ofrece distintos elementos de diseño basado en HTML y CSS como botones, menú desplegable, formularios, barras de herramientas, así como, extensiones de JavaScript opcionales adicionales para mejorar el diseño del sitio web.

Bootstrap es de código abierto y está disponible en GitHub, donde los desarrolladores pueden participar en el proyecto y hacer sus propias contribuciones a la plataforma. [8]

h) C#

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes. [9]

Para el desarrollo del proyecto se utilizará el IDE Microsoft Visual Studio versión 2015 y, además se investigará e implementará el uso de tecnologías de programación como LINQ (Language-Integrated Query) el cual consiste en la integración directa de funciones de consulta en el lenguaje C#. Con LINQ, una consulta es una construcción de lenguaje de primera clase, igual que las clases, los métodos, los eventos, etc.

i) SQL Server

Microsoft SQL Server es un sistema de manejo de bases de datos del modelo relacional, desarrollado por la empresa Microsoft.

El lenguaje de desarrollo utilizado es Transact-SQL, una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL). [10]

Además, se utilizará la interfaz de usuario Microsoft SQL Server Management con la cual se puede manejar, configurar, desplegar, actualizar y administrar una instancia SQL Server. La herramienta es bastante amigable para el usuario y también es exhaustiva. Es incluida en cada versión de SQL Server y es regularmente actualizada y mejorada.

j) Entity Framework

Es un conjunto de API de acceso a datos para el Microsoft .NET Framework. Una entidad del Entity Framework es un objeto que tiene una clave representando la clave primaria de una entidad lógica de datastore. Un modelo conceptual Entity Data Model (modelo Entidad-Relación) es mapeado a un modelo de esquema de datastore. Usando el Entity Data Model, el Framework permite que los datos sean tratados como entidades independientemente de sus representaciones del datastore subyacente. [11]

k) Visual Studio Community

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación y permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET.  [12]

Se determinó utilizarlo debido a su excelente acople con las demás herramientas de Microsoft, como SQL Server y Entity Framework, y la facilidad de uso con herramientas externas como Git. Además, como se comentó previamente, se decidió utilizar la versión gratuita del mismo, ya que ofrece todo lo necesario para desarrollar el sistema.

l) Git

Git es un sistema de control de versiones, el cual en este caso, será muy útil ya que el equipo de desarrollo será compuesto por dos personas que realizarán cambios al código desde computadoras diferentes.

Con Git se podrá controlar todos los cambios que se hacen en la aplicación y en el código, y se tendrá control absoluto de todo lo que pasa en el mismo, pudiendo volver atrás en el tiempo, o abrir diferentes ramas de desarrollo, entre otras acciones.

Se trabajará en equipo de una manera muy sencilla y optimizada, de forma que al ser dos personas trabajando en ciertas funcionalidades del proyecto, podremos estar editando cada una por separado una parte diferente del código al mismo tiempo. Cuando algún integrante acabe de desarrollar el código, se utilizará Git para mezclar los cambios con la otra compañera, de forma que el código se mezcla de manera perfecta rápidamente y sin generar ningún tipo de fallo.

También proporciona un listado de los cambios (commits) y se permite volver atrás en el tiempo a cualquiera de esos cambios. Además, se tendrá la posibilidad de trabajar con ramas de desarrollo, que permitirán desarrollar cosas nuevas sin afectar al código principal.

# 3. Metodología de resolución

Como metodología de trabajo se propone un proceso que consta de las siguientes etapas:

## *3.1. Etapa 1: Revisión bibliográfica e investigación preliminar*

Principalmente, en esta etapa, deberemos definir qué metodología de desarrollo se adapta más a los requerimientos que brinde el usuario y, a su vez, investigar la forma correcta de llevar a cabo tal metodología.

Además, tendremos que familiarizarnos con las herramientas y lenguajes de programación a utilizar. Esto requiere una previa investigación para analizar los pros y contras de cada herramienta conocida por el equipo, tanto por la facilidad de uso, la experiencia con la que cuente alguno de los miembros y la integración de la misma con las demás herramientas.

Se pretende, por otro lado, adquirir los conocimientos necesarios sobre bases de datos para definir cuál será el motor que se utilizará en el sistema de modo de agilizar las transacciones necesarias. Y, por último, se deberán rever los conceptos adquiridos durante la carrera sobre las redes de datos para utilizarlos al momento de decidir cómo integrarlos con el sistema.

Se definirán, además, tareas de evaluación de requerimientos funcionales y no funcionales que serán necesarios para el correcto y eficaz funcionamiento final del sistema.

## *3.2. Etapa 2: Desarrollo del sistema utilizando la metodología de desarrollo Incremental*

El proceso de desarrollo de software que más se adaptaba a la necesidad del cliente es el Modelo Incremental. En una primera etapa, se definen los requerimientos funcionales y no funcionales, con lo cual posteriormente se diseña la arquitectura  del sistema. En este mismo momento se define el modelo de datos y se crea la base de datos.

Una vez terminada la etapa de requerimientos, se dividen en grupos de prioridad y, aquellos que se consideren más importantes, se desarrollarán en el primer incremento, con lo cual se le entrega al cliente un producto con el que puede interactuar y así retroalimentar el proceso.

Cada incremento posterior de la Etapa de Inicial se compondrá de los siguientes requerimientos funcionales en la lista de prioridad, junto con la retroalimentación del cliente. Por lo tanto, todos los incrementos estarán compuestos por las siguientes etapas: Análisis, Diseño, Codificación, Prueba, Instalación y Capacitación.

### 3.2.1. Etapa Inicial

Se realiza el diseño de la arquitectura del sistema. Además, se diseña e implementa la base de datos. Esta etapa se realiza sólo en la primera iteración.

### 3.2.2. Etapa de Análisis

Se realiza un relevamiento de los requerimientos funcionales, no funcionales y la prioridad de ejecución de los mismos. Además, se acuerdan los roles y controles de acceso a los diferentes usuarios del sistema.

### 3.2.3. Etapa de Diseño

Se realiza el modelo de diseño de un incremento. Luego del análisis, se consultan los requisitos del incremento definidos en la Especificación de Requerimientos. Se determina la estructura requerida para el incremento y se detallan los componentes que se utilizarán.

### 3.2.4. Etapa de Codificación

Se realiza la codificación correspondiente a un incremento, siempre teniendo en cuenta el modelo de diseño definido previamente.

### 3.2.5. Etapa de Prueba

Se realiza la evaluación del funcionamiento de la iteración completa luego de terminar la etapa de codificación. Deberá realizarse por el desarrollador, el cual mediante pruebas unitarias determinará si el incremento es correcto y puede integrarse a la rama principal, y también por el cliente, el cual validará el cumplimiento de los requisitos pactados.

### 3.2.6. Etapa de Instalación y Capacitación

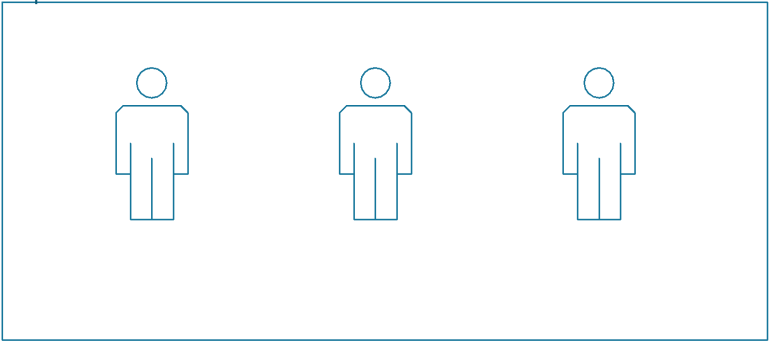
Se realiza la entrega del incremento al cliente. Se instala para que pueda ser probado por el mismo y a su vez se lo capacita. Si el producto cumple con todos los requisitos esperados por el cliente, entonces el incremento pasa a ser un incremento validado.

## *3.3. Actividades predecesoras y sucesoras*

|  |  |  |
| --- | --- | --- |
| ETAPA | ENTRADAS | SALIDAS |
| Inicial | * Boceto para definir la arquitectura del sistema. * Boceto de entidades y relaciones para base de datos. * Formato de Excel para la carga masiva de datos. | * Documento de Arquitectura del sistema. * Modelo de Datos y scripts de creación y carga de datos. |
| Análisis | * Documento de Arquitectura. * Listado de requisitos del sistema. * Definición de Roles y controles de acceso. | * Documento de Requerimientos Funcionales y No Funcionales. |
| Diseño | * Modelo de Datos. * Documento de Requerimientos Funcionales y No Funcionales. * Documento de Arquitectura. | * Diagrama de Casos de uso. * Diagrama de Entidad-Relación. * Diagrama de Clases. |
| Codificación | * Diagrama de Casos de uso. * Diagrama de Entidad-Relación. * Diagrama de Clases. | * Código Fuente |
| Prueba | * Documento de Requerimientos Funcionales y No Funcionales. * Documento de Arquitectura. * Diagrama de Casos de uso. * Diagrama de Entidad-Relación. * Código Fuente. | * Resultados de Pruebas Unitarias. * Resultados de Pruebas de Integración. * Resultados de Pruebas de Estrés. * Resultados de Pruebas de Seguridad. |
| Instalación y Capacitación | * Documento de Requerimientos Funcionales y No Funcionales. * Código Fuente. | * Programa Ejecutable. * Manual de Usuario. |

## *3.4. Actores*

Se pueden identificar diferentes actores dentro del sistema, los cuales interactúan de manera diferente con el mismo. A continuación se detallan los actores y sus roles:



Cliente Web

Utiliza el sistema desde su computadora personal con acceso a Internet. Puede navegar la página, buscar productos y armar carritos web para la compra de los mismos.

Además, posee dentro de su cuenta personal, información detallada de su historial de compra de productos y reparaciones de sus vehículos.

Empleado

Utiliza el sistema desde el negocio. Puede realizar la búsqueda, el alta, baja y modificación de productos, clientes, ventas y reparaciones. El sistema, además, le permite seleccionar los productos pedidos por un cliente personalmente en el local y generar una venta.

Administrador

Utiliza el sistema como un Empleado. Este actor hereda el comportamiento de su padre (actor Empleado). Además, tiene la característica especial de poder gestionar los empleados (agregar, modificar o quitar un empleado), realizar la gestión de precios, es decir, generar promociones o incrementos por inflación o suba de precios de un proveedor; gestionar los rubros, sub-rubros y productos incluidos en cada uno de ellos; y, por último, podrá visualizar las métricas de facturación del período.

## *3.5. Plan de Incrementos*

Según lo acordado con el cliente, se determinó que en un primer incremento se entreguen aquellos casos de uso que se relacionen directamente con el actor “Cliente”, los cuales cumplen la función de visualización y búsqueda de productos a través de la web, así como también el diseño general de la página. Luego se entregarán los casos de uso relacionados al actor “Empleado”, los cuales se encargan de la administración del stock y las gestiones de clientes, ventas y reparaciones. Y, por último, los casos de uso relacionados al rol de “Administrador” el cual será capaz de actuar como súper usuario.

## *3.6. Requerimientos del sistema*

### 3.6.1. Requerimientos Funcionales

Se definieron un conjunto de requisitos funcionales que cada uno de estos actores deberá poder llevar a cabo en el sistema.

Cliente

* Identificarse y autenticarse con el sistema cada vez que desee iniciar sesión en el mismo.
* Modificar la contraseña y sus datos personales por medio del sistema cuando lo desee.
* Visualizar el catálogo de productos ofrecidos ordenados por rubros e indicando precio, descripción, stock y descuento -si correspondiese- en cada caso.
* Agregar productos a un carrito web que tenga la posibilidad de calcular subtotales y totales del pedido.
* Realizar la compra de los productos elegidos.
* Decidir medio de pago y forma de envío de los productos elegidos.
* Realizar una compra presencialmente en el negocio y poder visualizarla en el historial de su cuenta.
* Realizar una reparación de un vehículo de su propiedad en el local y poder ver el detalle y materiales utilizados en el historial de su cuenta.

Empleado

* Gestionar los productos a mostrar en el catálogo.
* Crear clientes con sus datos y administrarlos en el sistema.
* Crear venta en el local en el caso en que el cliente se encuentre personalmente en el mismo.
* Dar de alta vehículos a los clientes del local junto con los datos de cada vehículo.
* Crear reparaciones a los vehículos de los clientes para que posteriormente puedan visualizarlos en su cuenta personal.
* Visualizar ventas y reparaciones por cliente.

Administrador

* Dar de alta y baja a los empleados del local.
* Modificar los rubros del sistema y también agregar o dar de baja a alguno de ellos.
* Gestionar los precios de los productos, ya sea incrementando los valores de los mismos o generando promociones a determinados productos por el tiempo que desee.
* Visualizar métricas y resultados de ventas del período.

### 3.6.2. Requerimientos no Funcionales

Mantenibilidad

Todo el sistema deberá estar completamente documentado. El sistema debe contar con interfaces de administración de todo lo que resulte cambiante en el tiempo, como los productos, sus imágenes y los empleados, entre otros, para que el usuario pueda gestionar todo desde el mismo sistema sin depender de los desarrolladores. Además, el código fuente del sistema deberá tener comentarios para guiar a los programadores en caso de futuras correcciones o agregados.

Seguridad

El sistema mantendrá un registro de usuarios habilitados para el ingreso al Sistema, mediante la implementación de usuario y contraseña cifrada almacenada en la base de datos. Los usuarios estarán relacionados a un rol y a un perfil particular; estos roles y perfiles determinarán qué acciones tiene permitido realizar un usuario sobre un tipo de gestión particular. Así mismo los roles y usuarios determinarán qué datos serán visibles y que datos se mantendrán reservados para cada usuario del sistema.

Además, el sistema contará con creación y vencimientos de sesión para los casos en que el usuario permanezca inactivo durante demasiado tiempo.

Disponibilidad

La disponibilidad del sistema debe ser continua, garantizando un esquema adecuado que permita ante una posible falla en cualquiera de sus componentes, contar con un plan de contingencia.

Al brindar servicio a los usuarios por medio de internet, el servidor y la base de datos deberán estar disponibles a toda hora y todos los días del año, ya que una compra por medio de carrito web puede gestionarse en cualquier momento del día.

Fiabilidad

Es uno de los factores que dará confianza al cliente, para lo cual el sistema estará controlando todo tipo de transacción. Debe contemplar requerimientos de confiabilidad y consistencia de los componentes de negocio ante recuperaciones. En caso de fallas de algún componente, no debe haber pérdida de información. Debe contemplar requerimientos de consistencia transaccional y brindarle al usuario la confianza necesaria para realizar compras electrónicas por medio del mismo.

Portabilidad

La portabilidad del sitio web será para todos aquellos navegadores que soporten lenguaje HTML5.  Entre ellos: Internet Explorer 9 (mejorada en IE10), Firefox 9, Chrome 16, Safari 5.1 y Opera 11.60.

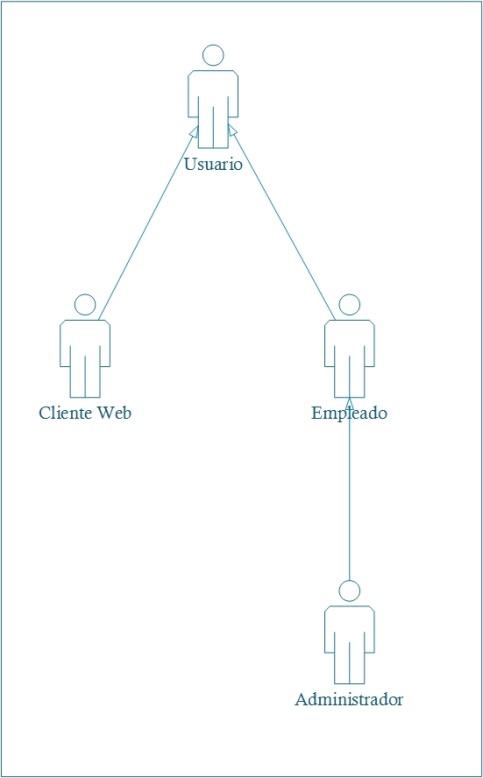
La portabilidad en los teléfonos celulares y tablets será similar a la de los sitios web de escritorio ya que la aplicación será diseñada utilizando Bootstrap, lo cual permite que se adapte responsivamente dependiendo el tamaño de la pantalla.

## *3.7. Diagrama de Casos de Uso*

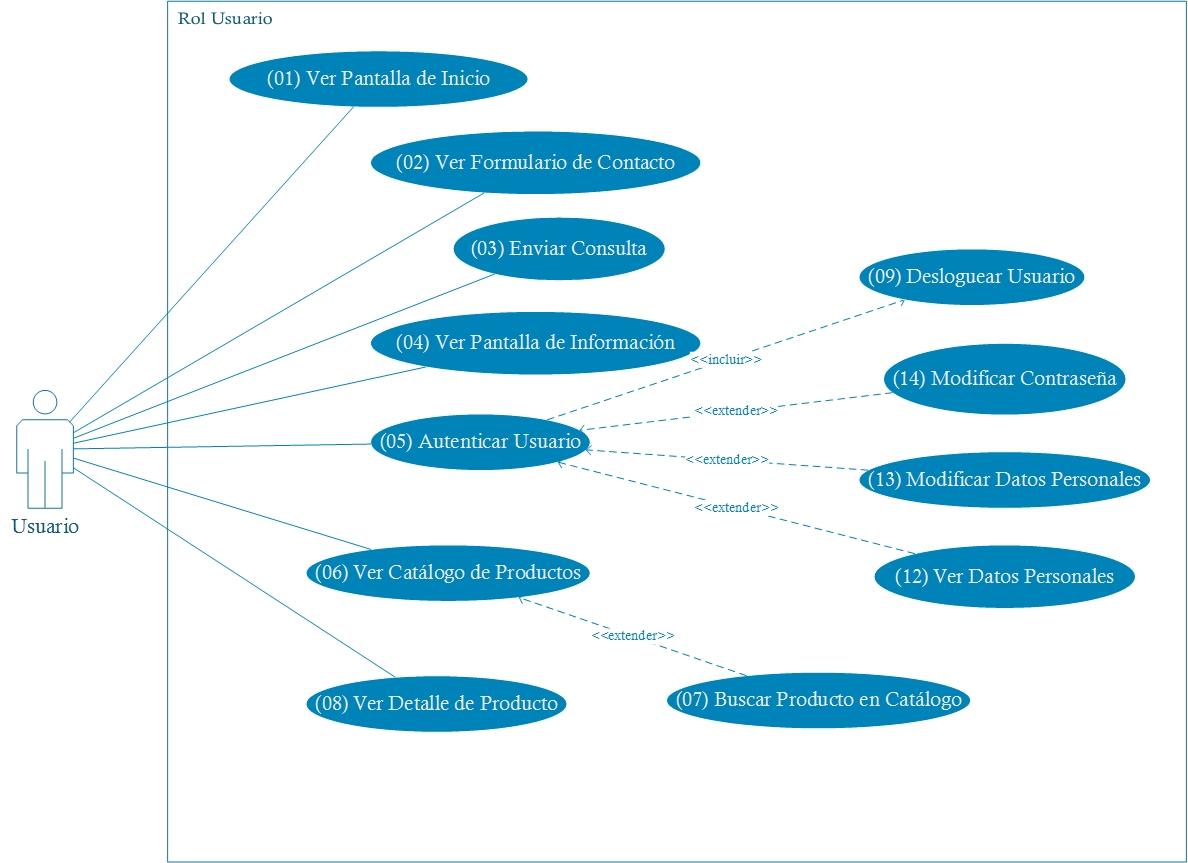
En los siguientes diagramas se pueden observar todos los casos de usos que se implementarán en el sistema y que cubren los requerimientos funcionales planteados en la sección anterior.

En primer lugar, se muestra la herencia que posee cada actor. A fines prácticos, se separó al actor Usuario (diferente del Cliente Web) para reunir en él todos los casos de uso comunes a los tres actores.

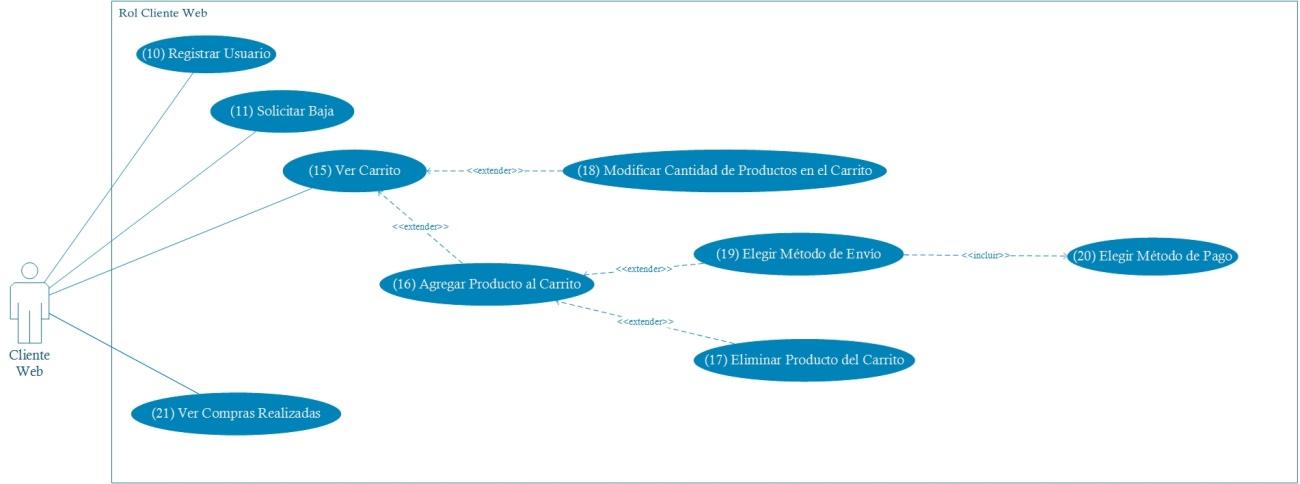
### 3.7.1. Relación de herencia entre los actores



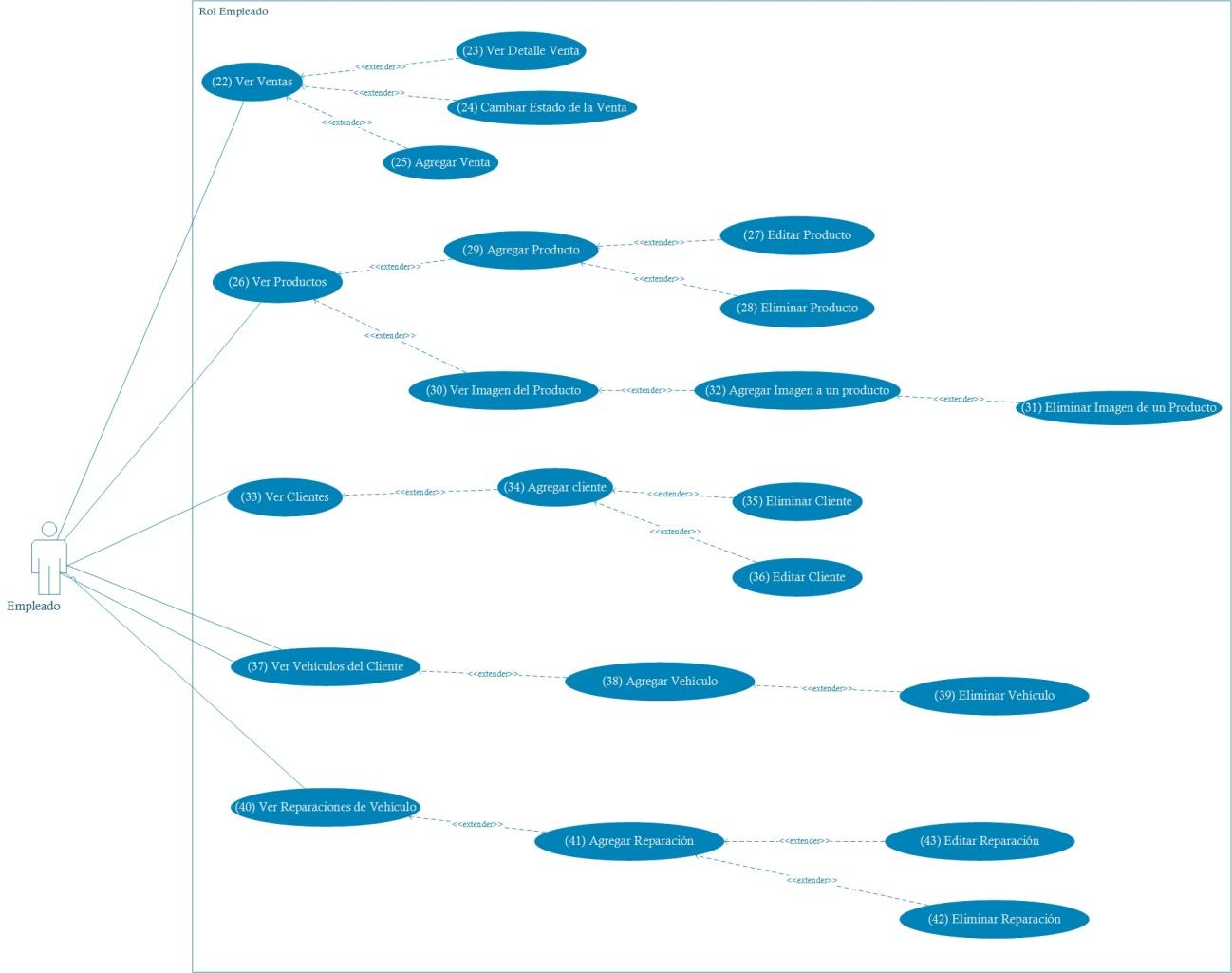
### 3.7.2. Casos de uso del actor Usuario

****

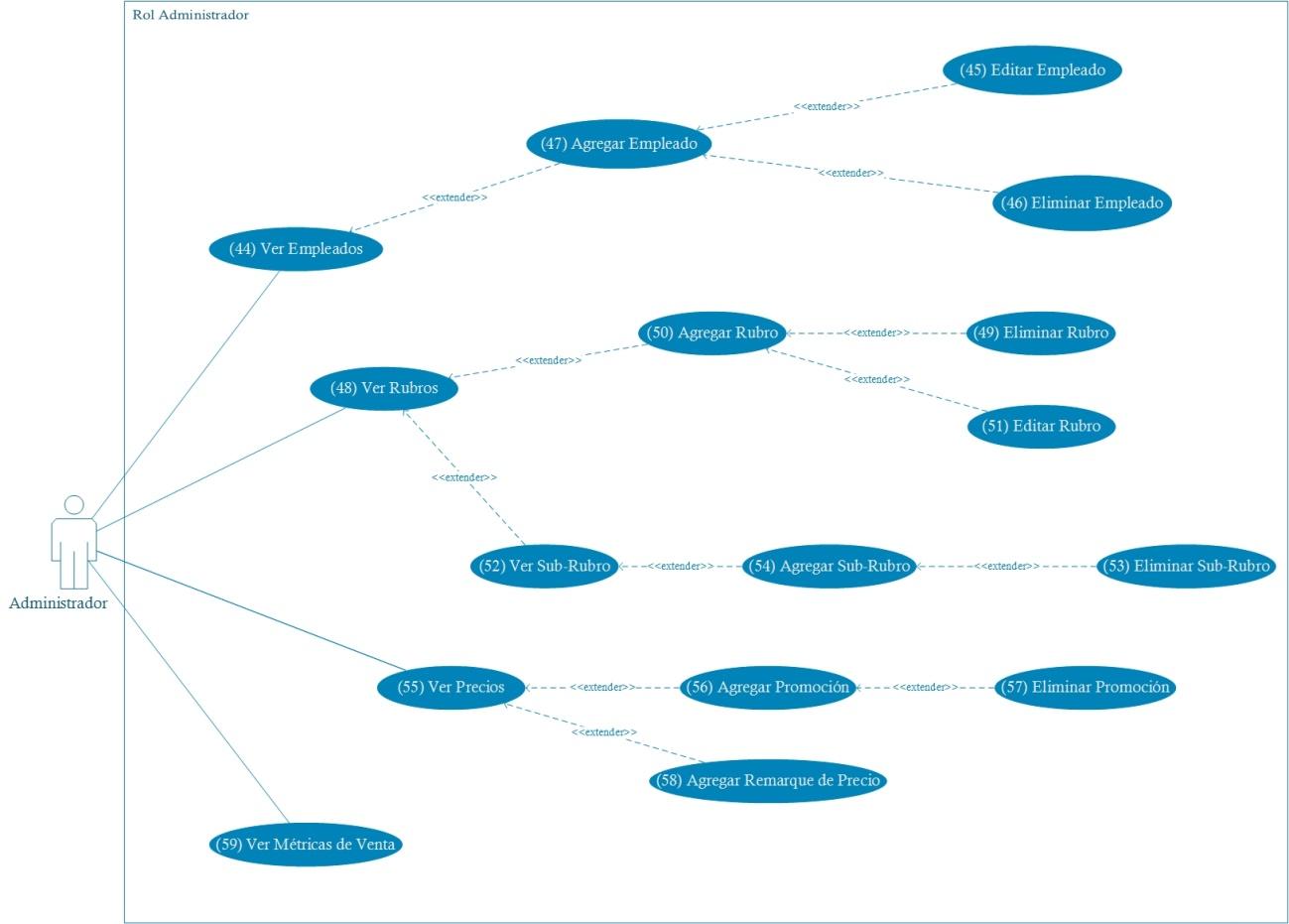
### 3.7.3. Casos de uso del actor Cliente Web

****

### 3.7.4. Casos de uso del actor Empleado

****

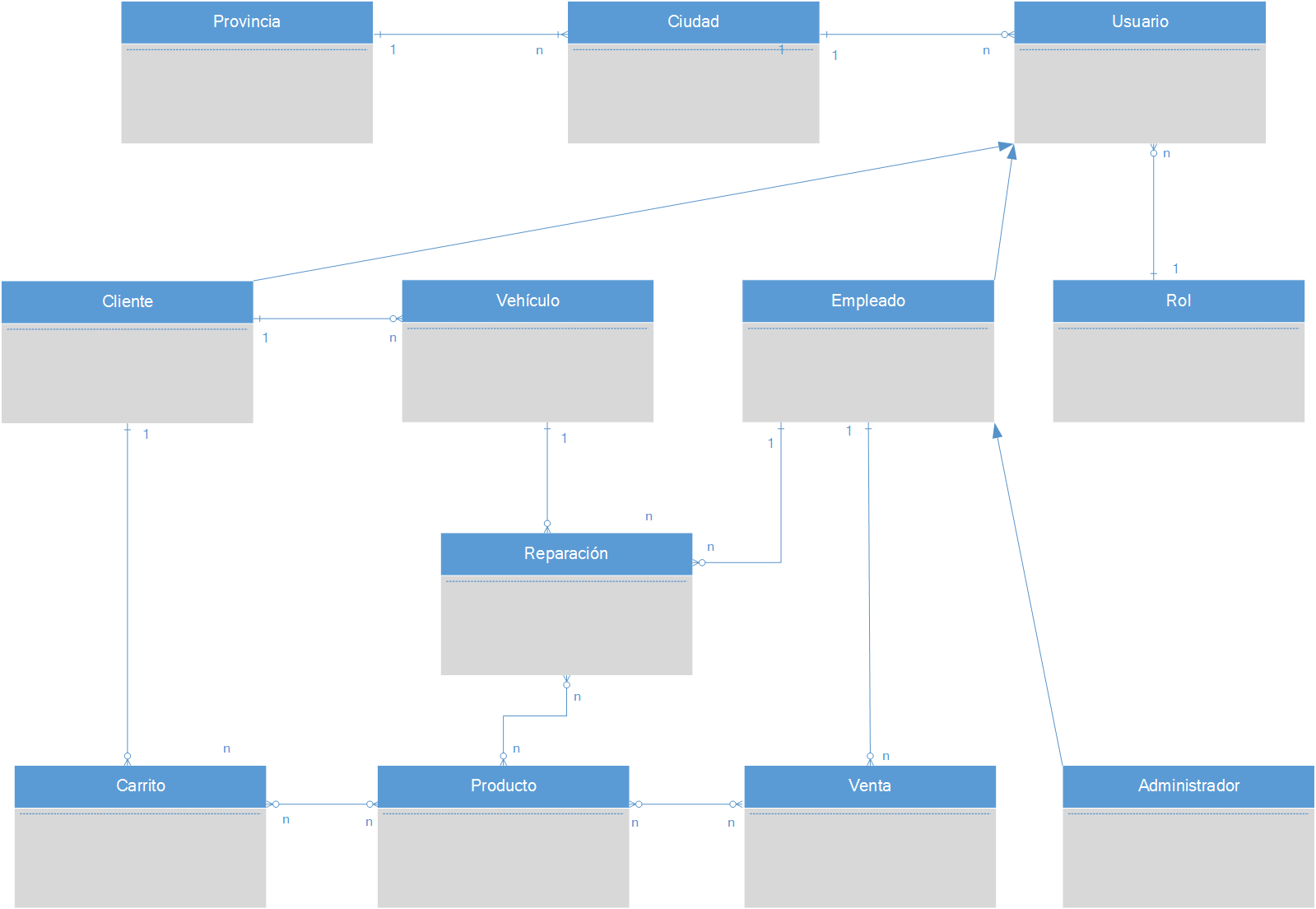
### 3.7.5. Casos de uso del actor Administrador

****

La descripción detallada de los casos de uso del sistema, es decir, la secuencia de pasos realizada en la ejecución de cada uno, se pueden encontrar en el Anexo A.

## *3.8. Diagrama de Clases*

A continuación se presenta el Diagrama de Clases del sistema representando las principales entidades y las relaciones entre ellas.

****

Este modelo explica el dominio de nuestro sistema y las clases que forman parte del mismo.

Como se puede apreciar, se modeló la herencia comentada previamente entre el Usuario, Cliente Web, Empleado y Administrador. Por su parte, un Cliente puede tener de cero a muchos vehículos registrados en el sistema, los cuales serán los que podrán registrar reparaciones. Estas reparaciones deberán ser generadas por el Empleado dentro del sistema, por lo tanto, cada empleado puede tener cero o más reparaciones realizadas. Lo mismo aplica para las ventas, realizadas a clientes presentes en el local.

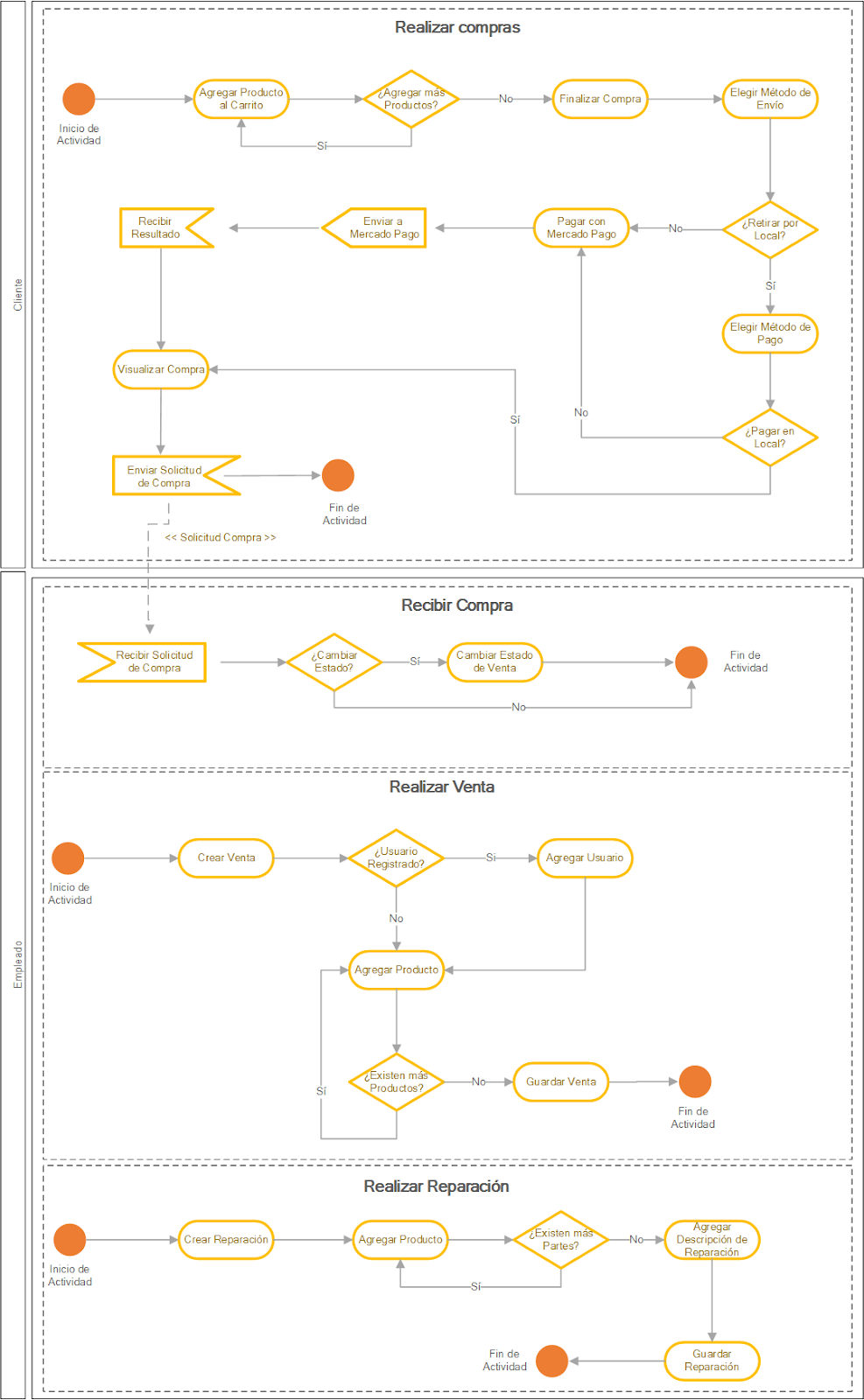
Por otro lado, un cliente puede generar un carrito a través de la web, en vez de dirigirse al local a realizar la compra, por lo tanto, un cliente puede tener cero o más carritos generados a su nombre y el carrito es único para cada cliente.

Tanto el carrito web, como las ventas y las reparaciones tienen o pueden tener cero a muchos productos agregados.

Por último, un usuario cualquiera tiene asociado un solo rol con el cual ingresa a la aplicación. Y, se separó en distintas clases la Provincia y Ciudad que puede tener un usuario.

## *3.9. Diagrama de Actividades*

En el diagrama, se muestran los flujos de acciones para los subprocesos más importantes:

****

Subproceso Realizar compras

Este subproceso lo inicia el Cliente desde la web, agregando a su carrito personal la cantidad de productos que desee comprar. Una vez terminada la carga de los mismos, finaliza la compra y elige el método de envío. Aquí es donde el flujo se separa, ya que, sólo si elige retirar los productos por el local tiene la opción de pagar en el local. Sino, tanto retirando en el local como enviando a su domicilio, el cliente puede elegir la opción de Mercado Pago. Esto es así, debido a que no vimos el sentido en dejar al cliente pagar en el local pero recibir los productos en su domicilio.

Si se elige la opción de pago mediante Mercado Pago, se redirige a la página para que el cliente complete los datos de su tarjeta, y se espera el retorno para conocer el resultado de la operación. Tanto luego de pagar online como eligiendo pagar en el local, se visualiza la venta al finalizar el proceso, lo cual dispara un evento al empleado para informarle la actividad.

Subproceso Recibir compra

Al dispararse el evento de una compra por internet, el empleado la podrá visualizar entre sus ventas dentro de su perfil. Podrá ver los productos pedidos por el cliente, la cantidad, el precio final que se le cobró y, sobre todo, la forma de envío y pago seleccionado. Con estos datos tendrá la posibilidad de cambiar de estado a la venta para informarle al cliente. Por ejemplo, si los productos ya estuvieran disponibles para retirar en el local, el empleado podría cambiarle el estado a “A retirar” y así el cliente sabría que puede pasar a buscar el pedido inmediatamente.

Este cambio de estado no es necesario hacerlo apenas llega la solicitud, por lo que el empleado puede hacerlo más tarde si así lo requiere.

Subproceso Realizar venta

El empleado puede generar la venta a un cliente que se presenta en el local. Esta opción está disponible tanto para clientes ya registrados en el sistema, los cuales al finalizar la venta verán lo comprado en su perfil personal y, para clientes que se presenten sin ánimos de registrarse en el sistema. Por lo tanto, si el usuario se encuentra registrado, es posible agregarlo a la venta buscándolo por nombre, apellido o DNI. Luego de esto, se agregan todos los productos requeridos y se finaliza la venta.

Subproceso Realizar reparación

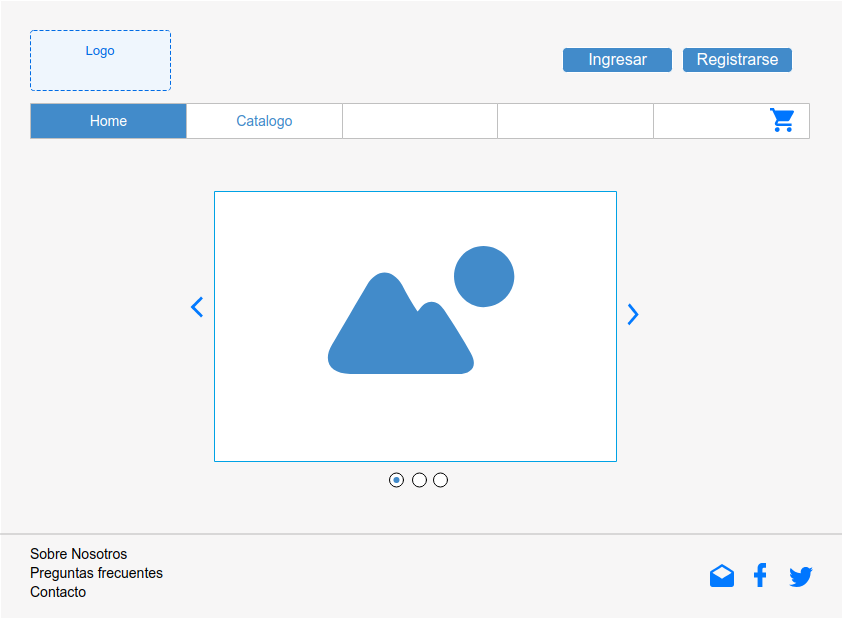
Este subproceso comienza en los datos del vehículo de un usuario, al cual se le desea registrar una reparación realizada para tener un seguimiento del vehículo. Para ello, se agregan los productos utilizados en la misma, se incluye una descripción de lo realizado y se finaliza la reparación.

# 4. Diseño del sistema

## *4.1. Diseño preliminar de interfaces gráficas*

A continuación se presentan las interfaces de usuario que representan los principales casos de uso del sistema.

CDU01 – Ver pantalla de inicio

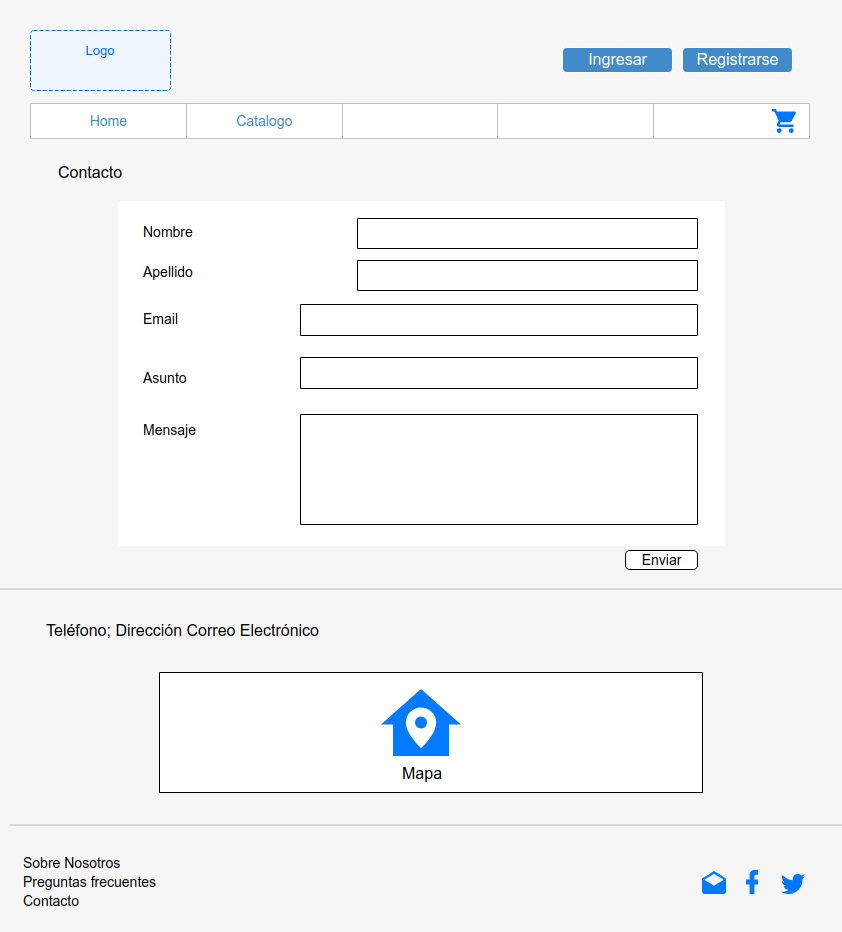


Esta vista será la pantalla que se visualizará al ingresar al sistema por medio de la URL en el navegador. No será requisito estar autenticado en el sistema para poder visualizarla y contendrá un banner de imágenes del local que irán cambiando a través del tiempo.

Además presentará una barra de menú de opciones por la cual se accederá al catálogo de productos,  al login del usuario (o deslogueo si correspondiese) y a la registración de nuevos usuarios.

Por otra parte, se mostrará publicidad del local junto con un menú inferior, el cual contendrá opciones informativas para el usuario, como por ejemplo, los términos y condiciones, métodos de pago, dirección y datos del local, formulario de contacto, entre otros.

CDU02 – Ver formulario de contacto



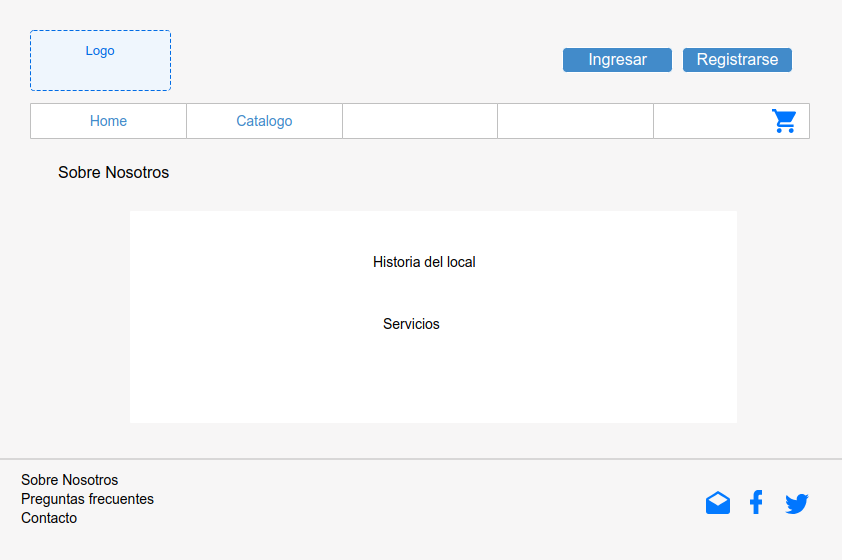
Este formulario no requiere la autenticación del usuario en el sistema y puede ser accedida a través de cualquier otra pantalla, ya que su acceso se encuentra en el menú general perteneciente a toda la aplicación.

En este formulario, el usuario podrá realizar consultas  al negocio, completando los datos requeridos, es decir, nombre y apellido del interesado y un correo electrónico para que, posteriormente, alguno de los miembros del equipo del local pueda responder la solicitud.

Por otro lado, al enviar el formulario (*CDU03 – Enviar consulta*), el negocio recibirá en su casilla de correo un mail con todos los datos ingresados por el usuario, en el cual el asunto del mismo y el cuerpo del mensaje serán los que se hayan ingresado en el formulario.

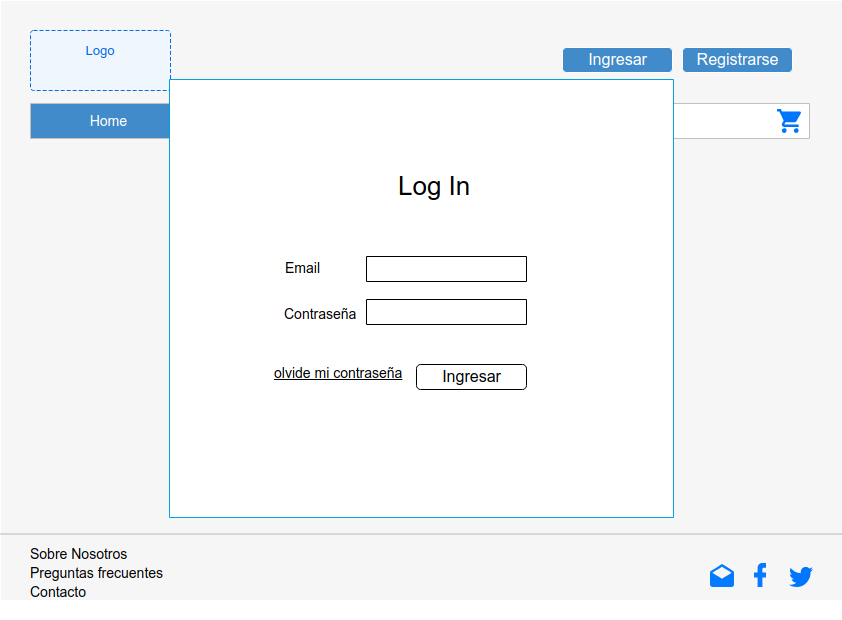
Como datos adicionales, en esta pantalla se podrá visualizar un mapa con la ubicación física del local, junto con datos como número de teléfono, horarios de atención y correo electrónico.

CDU04 – Ver pantalla de información

**

En esta vista se mostrarán datos informativos del negocio, como por ejemplo, una breve historia, el logo, sus integrantes, entre otros datos de valor que la empresa pretenda brindarle a sus clientes.

CDU05 – Autenticar usuario

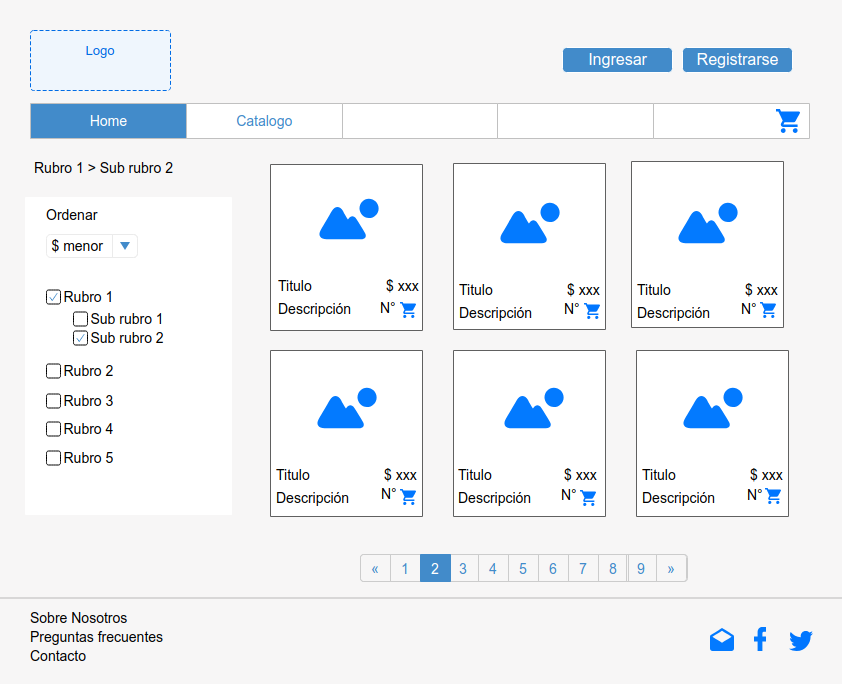


El objetivo principal de esta interfaz es realizar la autenticación correspondiente del usuario al sistema. Cuando el mismo complete el nombre de usuario y su contraseña, si éstos son correctos, se le dará acceso y se le asignarán los permisos correspondientes. Si no se logra una correcta autenticación, se mostrará un mensaje de error.

Será una pantalla común para todos los tipos de usuarios. El sistema validará el rol que cumpla cada uno al momento de ingresar y, dependiendo cuál sea mostrará o no cierta información. Además, una vez ingresado, el usuario contará con un tiempo de sesión, el cual luego de mantenerse inactivo durante determinado tiempo se vencerá y deberá autenticarse nuevamente.

En el caso en que el usuario no posea aún una cuenta creada en el sistema, podrá hacerlo mediante el *CDU10– Registrar usuario*.

CDU06 – Ver catálogo de productos



Esta interfaz será la principal del sistema, ya que permite visualizar todos los artículos ofrecidos por el negocio y que están disponibles para una compra inmediata. Se puede ingresar a verla sin la necesidad de estar registrado en el sistema.

Los productos ofrecerán una o más imágenes, junto con el nombre del artículo, su precio y oferta –si correspondiera-. Además, cada uno contará con un botón que permitirá ver el detalle de cada producto con más información del mismo, abarcado por *el CDU08 – Ver detalle de producto*. A su vez, el usuario puede elegir agregar un producto a su carrito personal, lo cual da lugar al *CDU16 – Agregar producto a carrito*, y para el cual es requisito necesario estar autenticado en el sistema. Si esta condición no se cumple se redirige al usuario al *CDU05 – Autenticar usuario*.

Como un menú lateral se visualizarán los rubros y sub-rubros existentes, que contarán con la posibilidad de filtrar los productos mostrados en pantalla, para facilitar la búsqueda al usuario. El cual, además, tendrá la posibilidad de buscar entre ellos por nombre de producto (*CDU07 – Buscar productos en catálogo*).

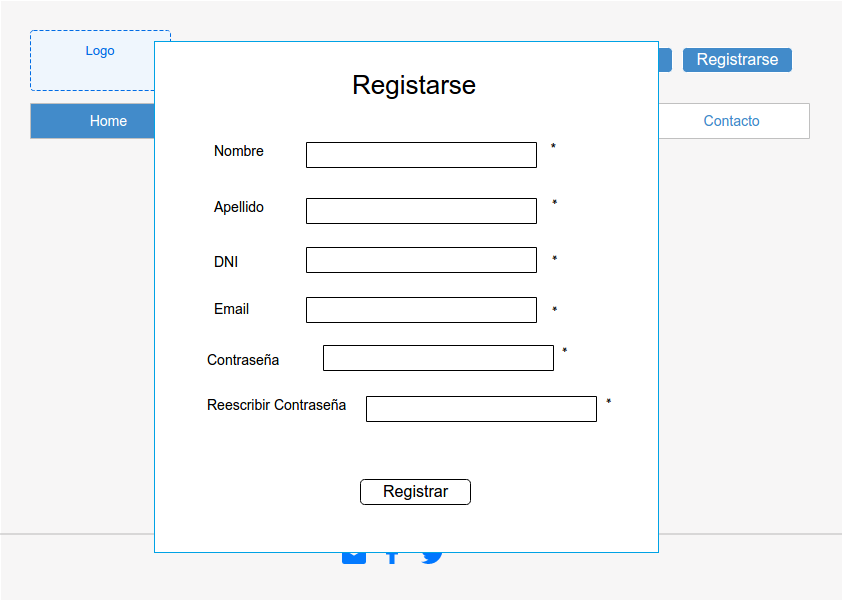
CDU08 – Ver detalle de producto

**

Para acceder a esta interfaz el usuario no tendrá la obligación de autenticarse. Este caso es una extensión del *CDU06 – Ver catálogo de productos*. Permite visualizar detalles del producto elegido en el catálogo. Así, es posible ver más imágenes del producto (si existieran), una descripción más extensa del mismo y el stock disponible que posee el local.

El usuario tendrá la posibilidad de agregar el producto al carrito desde aquí, haciendo uso del *CDU16 – Agregar producto a carrito,* o podrá regresar al catálogo para seguir explorando productos.

CDU10 – Registrar usuario



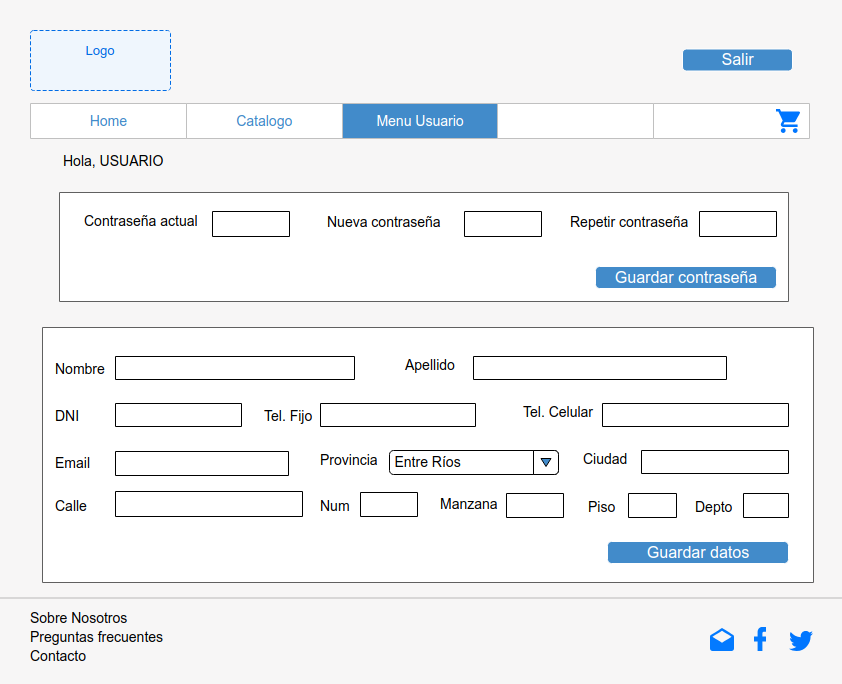
El objetivo de esta interfaz es crear un nuevo usuario del sistema. Para ello se le pedirán datos personales para completar su perfil. Deberá ser obligatoria la carga de su nombre y apellido, correo electrónico y provincia. Además, tendrá que elegir una contraseña que cumpla con los requisitos del sistema.

Cada campo será validado antes de enviar la solicitud de registración, tanto para corroborar el ingreso obligatorio de los datos, como para brindar seguridad e integridad al sistema en caso del ingreso de datos corruptos o malintencionados. En el caso en que algún dato falte o no cumpla con las reglas establecidas –el formato del correo electrónico, o los caracteres necesarios para la contraseña, por ejemplo- se informará mediante un mensaje de error.

Una vez completo correctamente, el sistema validará que el usuario no exista previamente en el sistema. Esto es, que el correo electrónico ingresado no se encuentre ya registrado en otra ocasión. En ese caso, deberá devolver un mensaje al usuario para que cambie su correo, o inicie sesión con la cuenta ya creada.

Una vez terminada sin errores la registración, el usuario será autenticado inmediatamente y podrá operar con su cuenta en el sistema.

CDU12 – Ver datos personales

**

En este caso de uso, un usuario autenticado en el sistema es capaz de visualizar, modificar, agregar o eliminar sus datos personales, así como también, tiene la posibilidad de cambiar su contraseña de acceso al sistema (*CDU14 – Modificar contraseña*).

Entre los datos que no podrá cambiar luego de haberse registrado se encuentra el correo electrónico, el cual representa el nombre de usuario único para el ingreso. El resto de los datos son modificables mediante el *CDU13 – Modificar datos personales*, aunque algunos son obligatorios de completar. Estos datos le ofrecen una atención más personalizada ya que permiten brindarle al local más información sobre el usuario.

Este caso es común para todos los roles del sistema, ya que un Empleado o el mismo Administrador pueden modificar sus datos personales. Con la particularidad de que los Clientes del sistema tendrán la posibilidad de dar de baja su cuenta desde esta pantalla (*CDU11 – Solicitar baja*).

CDU15 – Ver carrito

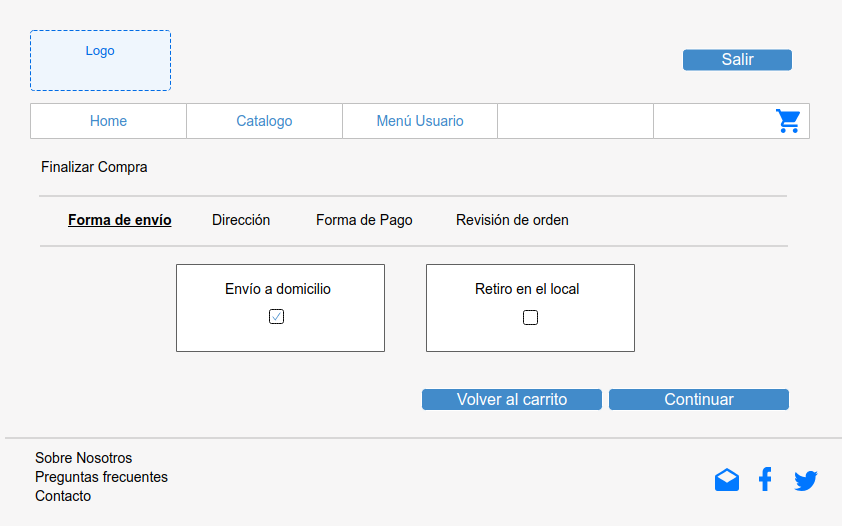


Para poder ingresar a consultar los productos del carrito, el usuario deberá estar autenticado en el sistema. Una vez en la interfaz, podrá visualizar los productos que haya agregado, allí mismo tendrá la posibilidad de elegir la cantidad del producto que requiere (*CDU18 – Modificar cantidad de producto en carrito*), y el sistema le mostrará tanto los subtotales de cada producto (precio unitario menos descuento -si correspondiere- multiplicado por la cantidad elegida) como el total general del carrito.

Cada producto tendrá un link el cual al ser presionado dirigirá al usuario al detalle del producto, el cual está abarcado por el *CDU08 – Ver detalle de producto*. Además, cada producto podrá ser eliminado del carrito si así lo decidiese el usuario (*CDU17 – Eliminar producto de carrito*), y se podrá agregar nuevos desde el catálogo mediante el *CDU16 – Agregar producto al carrito*.

Cabe destacar que el carrito permanecerá creado con los productos hasta que el usuario decida terminar la compra o elimine los productos del mismo. Es decir, no se eliminará con la terminación de la sesión del usuario.

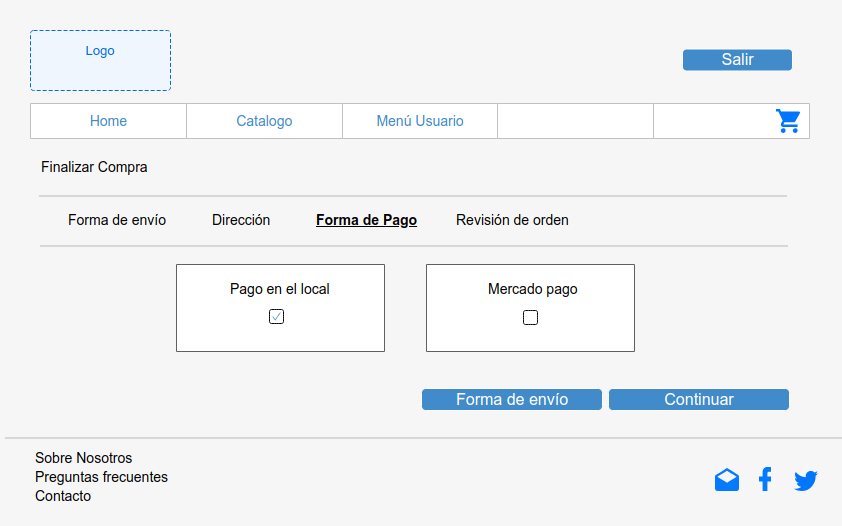
CDU19 – Elegir método de envío

**

Una vez que el cliente decida la compra de los productos en su carrito, tendrá la posibilidad de elegir el medio de pago que desea utilizar para adquirirlos. El cliente cuenta con las siguientes opciones:

* Retiro por local: esta opción le brinda al usuario la posibilidad de generar la compra y dirigirse personalmente al local a retirar los productos. En este caso, el sistema le mostrará un mapa con la ubicación física del local y los horarios de atención.
* Envío a domicilio: el sistema le brinda la opción de, mediante la plataforma de Mercado Envíos, recibir los productos en su domicilio. Dependiendo de la ubicación del cliente, el negocio podrá entregarle los productos por cuenta propia.

CDU20 – Elegir método de pago

**

Este caso de uso incluye al caso de uso *CDU19 – Elegir método de envío*, debido a que es condición necesaria su ejecución para este caso. Si el cliente eligió previamente el retiro de los productos por el local, se habilitará la opción de pago en el local para que pueda asistir personalmente a realizar el pago junto con el retiro de la mercadería, y la opción de pago online.

Por otro lado, si se eligió la opción de envío a domicilio, se habilitará la opción de pago online, la cual le brinda al cliente la posibilidad de utilizar la plataforma de Mercado Pago, para poder acceder a la compra mediante tarjetas de crédito con sus financiaciones y promociones, tarjetas de débito o realizar transacciones mediante negocios del estilo “Rapipago”. Para ello, al finalizar el proceso de compra, el sistema lo redirigirá a la página de Mercado Pago con todos los datos de la compra ya completos, para que el cliente ingrese los datos de su tarjeta de manera segura. Una vez terminado el pago, se redirigirá nuevamente a la página del local y se mostrará un mensaje informativo, de éxito en el caso de que la compra haya resultado exitosa o de error en el caso en que se haya cancelado o hubiere habido algún tipo de fallo en el proceso.

CDU21 – Ver compras realizadas

**

Este caso de uso requiere de la autenticación del usuario, el cual podrá consultar todas las compras que haya realizado en el local, ya sea de manera presencial como online, y un detalle de las mismas.

A su vez, podrá visualizar e informarse sobre el estado de aquellas compras realizadas por internet, ya que, una vez realizado el pago de uno o varios productos, la compra queda en estado Pendiente, hasta el momento en que la misma esté lista para ser retirada por el local, o bien enviada a domicilio, dependiendo la opción elegida por el usuario al momento de la compra. Por lo tanto, podrá observar este cambio de estado en cada compra particular que haya realizado.

Además, en esta pantalla verá las reparaciones de los vehículos que haya hecho en el local. Cada una tendrá un detalle de lo realizado, los repuestos incluidos, la fecha y precio total de la operación, para que tenga la posibilidad de hacerle un seguimiento a su vehículo y recordar en todo momento las reparaciones realizadas.

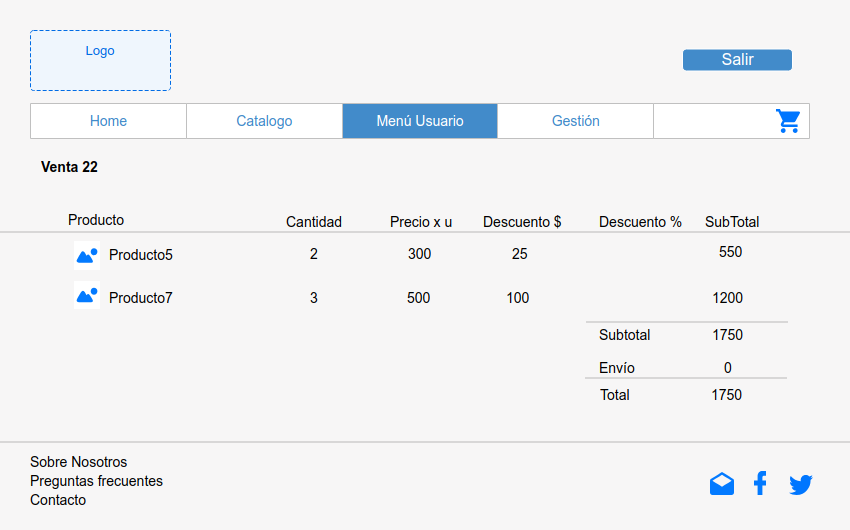
CDU22 – Ver ventas

**

Así como el cliente puede, desde su cuenta autenticada, ver las compras y reparaciones hechas, el empleado del local puede visualizar la totalidad de las ventas y reparaciones que se hayan creado. Se le permitirá buscar por cualquiera de los campos de las ventas para encontrar alguna particular.

Además, cada venta tendrá opciones, como ver el detalle y cambiar de estado. Cada una de estas acciones se detallarán en los casos de uso redactados a continuación. El empleado no podrá eliminar las ventas que se hayan realizado, ya que se considera un dato que debería quedar permanentemente en el sistema, para luego poder representar las métricas de venta correctamente.

CDU23 – Ver detalle de venta

**

El empleado autenticado podrá ingresar al detalle de una determinada venta para observar los productos, precios y descuentos incluidos en la misma.

Por otro lado, también podrá visualizar los detalles de alguna reparación realizada, así como el usuario y el vehículo reparado, los repuestos utilizados y la cantidad, el precio de cada uno en ese momento y una descripción hecha por el encargado de la reparación detallando los cambios y reparaciones realizados al vehículo en determinada fecha.

CDU24 – Cambiar el estado de la venta

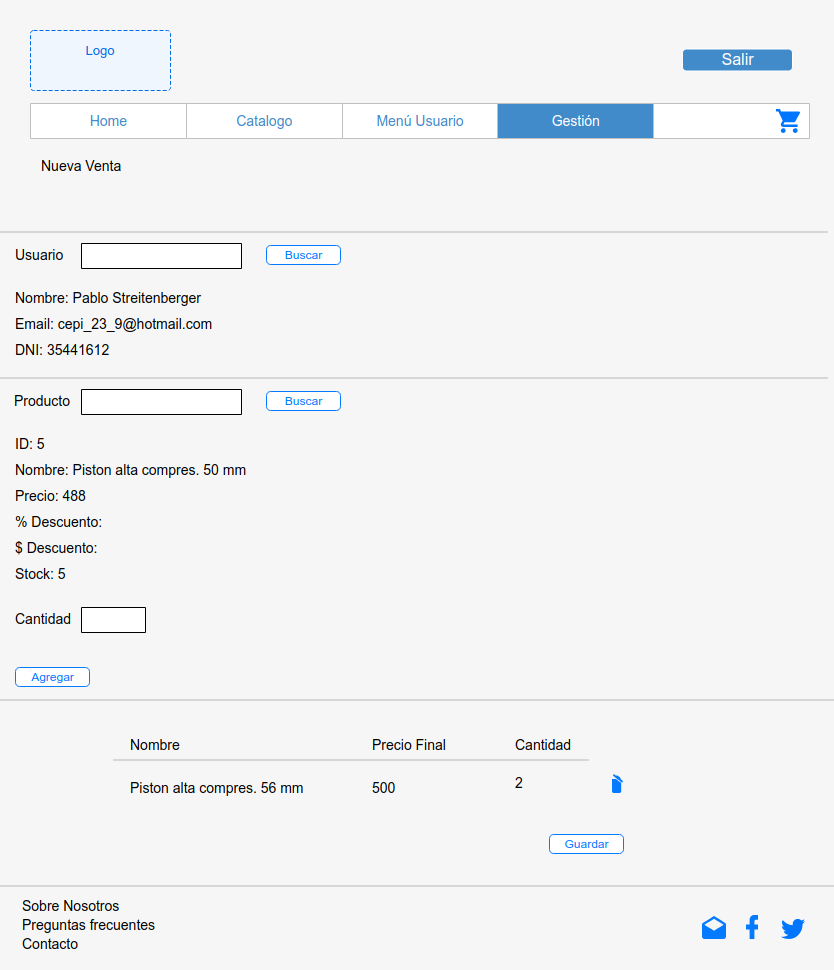
**

El empleado o administrador del sistema indistintamente, podrá cambiar de estado una venta para informarle al cliente la etapa en la que se encuentra el pedido. Para esto, la venta no puede estar previamente en estado Terminada, ya que se considera que una vez que se pasó a ese estado, los productos fueron entregados al cliente.

Los estados por los que puede pasar la venta son los siguientes:

* Pendiente: se genera la venta automáticamente en este estado luego de que el cliente realiza la compra online. Este estado indica al cliente que debe quedar a la espera de la confirmación del local y, por otro lado, le da la oportunidad al empleado de corroborar el pago realizado.
* Confirmado: una vez corroborado el pago, el empleado puede cambiar el estado a Confirmado para darle a conocer al cliente que el pedido fue procesado correctamente. Este estado le da el tiempo al negocio de preparar el pedido antes de ser retirado o enviado al cliente.
* A Retirar: en el caso en que el cliente hubiera elegido la opción de retiro del producto por el local, el empleado puede avisarle por este medio que el pedido está listo para ser retirado.
* Terminado: este estado indica, por un lado, en el caso en que el pedido sea enviado a domicilio, una vez entregado, se deberá pasar a estado Terminado para dar por finalizada la venta. Por otro lado, si el cliente escogió el retiro por el local, luego de ser entregado el producto personalmente, se deberá cambiar la venta a estado Terminado.
* Cancelado: si por alguna razón, el cliente o el local decidiesen cancelar la transacción, la venta puede pasarse a este estado. Si en un futuro se quisiese retomar la venta con los valores pactados a la fecha podría nuevamente pasarse a alguno del resto de los estados.
* Reparación: este estado es exclusivo para aquellas ventas que hayan sido una reparación, para poder identificarlas del resto. Este tipo de estado se genera en la venta luego de ingresar la reparación de un vehículo de algún usuario. Si el usuario se encuentra registrado en el sistema, en la pantalla en la que se informa sus compras, podrá observar una con este estado.

CDU25 – Agregar venta

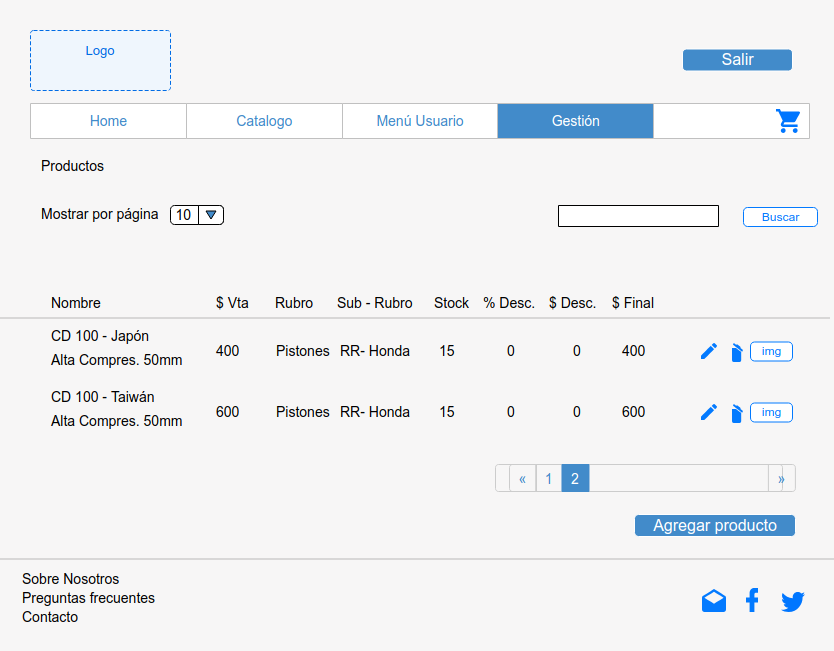


Para el caso en que un cliente se presente personalmente en el local a realizar una compra, el empleado o administrador pueden generarle una venta desde el sistema, para que el usuario luego pueda visualizarla entre sus consumos personales.

Para ello, se deberá buscar por nombre, apellido o documento al usuario, y se deberá ir agregando los productos comprados junto con la cantidad ordenada de cada uno. Una vez guardado, se generará la venta y quedará en el historial.

Por otro lado, puede ocurrir que el usuario no se encuentre registrado en el sistema y que tampoco desee darse de alta al momento de la compra, por lo cual, el usuario quedará vacío y se generará la venta para dejar un registro en el sistema y que luego impacte en las métricas del local.

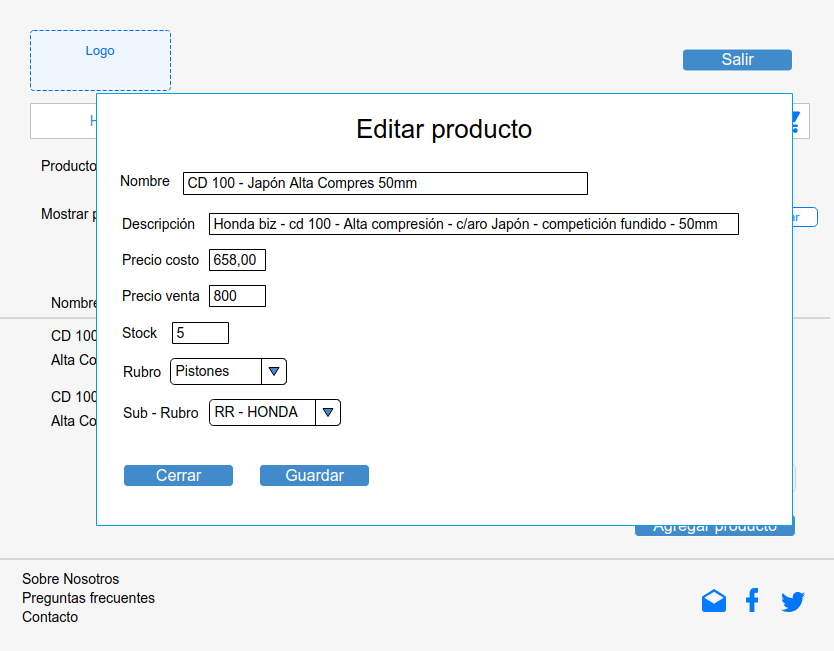
CDU26 – Ver productos

**

Para ingresar a esta pantalla es necesario que el usuario tenga rol de empleado o administrador. En esta, se podrá visualizar el listado de los productos disponibles en el catálogo mostrado a los clientes. Cada producto mostrará información como su nombre, descripción, precios de compra y venta, rubro y sub-rubro al cual pertenece, el stock disponible, los descuentos que tuviere, entre otros.

El empleado podrá realizar acciones sobre estos productos, las cuales se detallarán en los casos de uso siguientes.

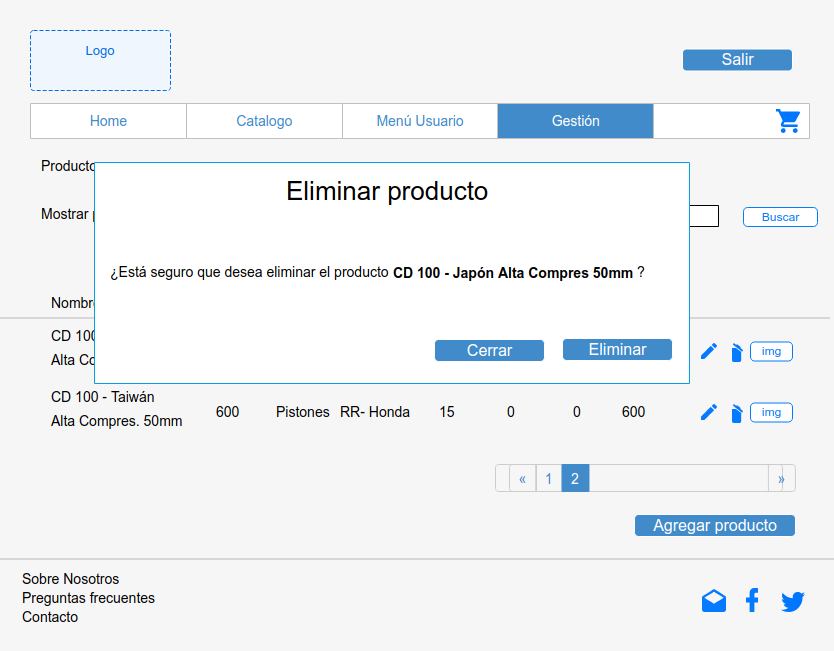
CDU27 – Editar producto

**

En el caso en que se desee editar algún dato de un producto, el empleado o administrador podrá acceder a los datos en forma de edición y modificarlos. En este caso es posible editar tanto el nombre, como la descripción, los precios, el stock (sólo por el usuario administrador) y el rubro o sub-rubro. Para el caso de querer editar o agregar alguna promoción o descuento se deberá remitir al *CDU55 – Ver precios*.

Luego de guardar los nuevos valores, estos serán reflejados automáticamente en el catálogo mostrado a los usuarios.

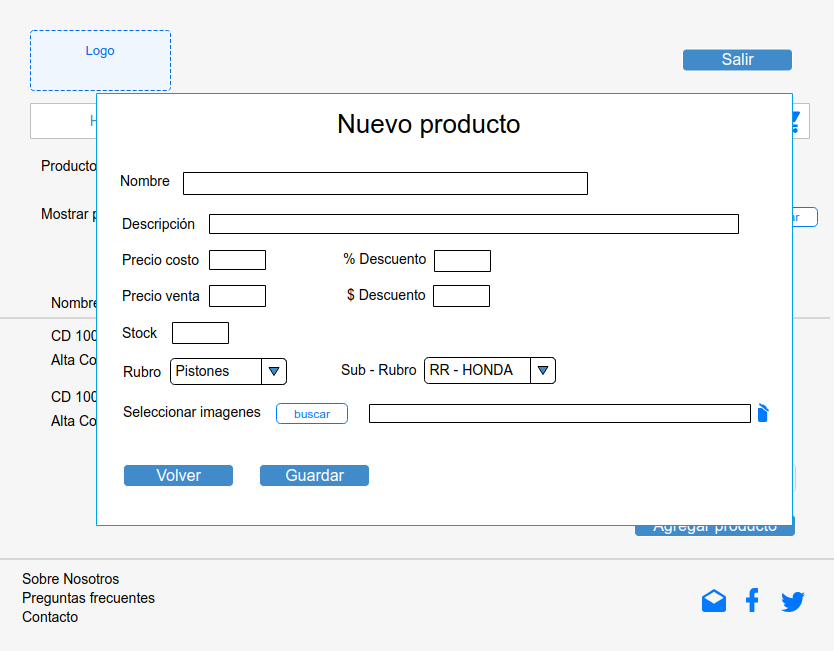
CDU28 – Eliminar producto

**

En el caso en que el negocio decida no vender o mostrar más un determinado producto, se tendrá la opción de eliminarlo del listado. Cabe aclarar que también se tiene la posibilidad de mantenerlo entre los productos y modificarle el stock a cero, para el caso en que el producto se siga vendiendo pero no se consiga momentáneamente.

Para esto, se recibirá un mensaje de confirmación para eliminar el producto y corroborar la decisión del usuario. Posteriormente el producto ya no estará disponible para la venta.

CDU29 – Agregar producto



Cuando ingrese un artículo nuevo al stock del negocio, el empleado o administrador pueden agregarlo desde esta pantalla. Para ello, deberán completar los datos del producto, como su nombre, descripción o datos importantes, precio de compra y precio de venta, rubro y sub-rubro al cual pertenece y stock disponible. Además, en este momento podrá agregarle una imagen al producto. Posteriormente, podrá agregarle más o eliminarla haciendo uso *del CDU30 – Ver imágenes del producto*.

Una vez ingresados los datos del producto, el mismo se da de alta y ya está disponible en el catálogo.

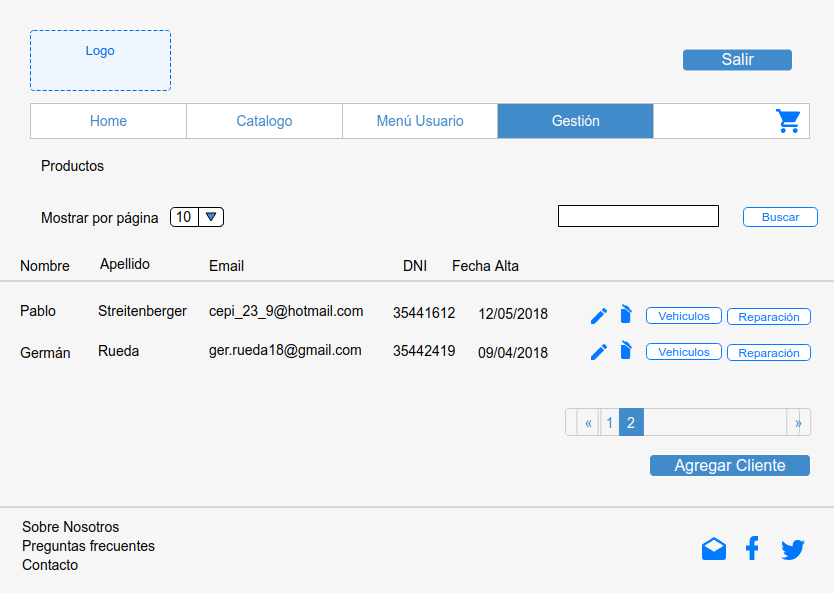
CDU30 – Ver imágenes del producto

**

Cada producto tendrá la posibilidad de tener varias imágenes ilustrativas, para ello, el empleado o administrador deberá gestionarlas particularmente para cada uno. Al seleccionar un producto e ingresar a la pantalla de imágenes, podrá ver las que tenga cargadas (si tuviera). Podrá eliminar (*CDU31 – Eliminar imagen de un producto*) o agregar nuevas desde su computadora (*CDU32 – Agregar imagen a un producto*).

Las imágenes que aparezcan primera y segunda en el listado serán las que se mostrarán en la pre-visualización de los productos en el catálogo. El resto de las imágenes podrán ser vistas cuando el cliente ingrese a mirar el producto con más detalle.

CDU33 – Ver clientes

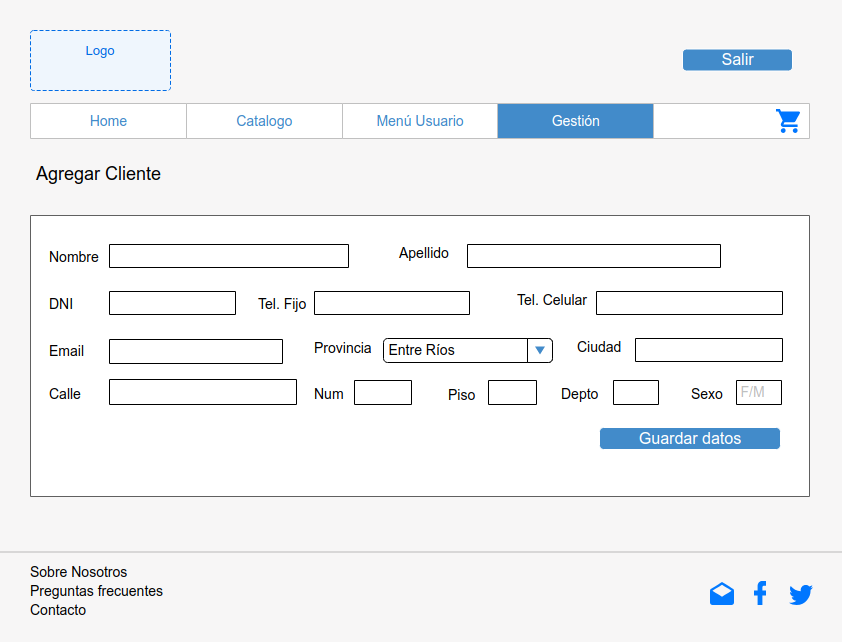
**

Tanto el empleado como el administrador pueden ingresar a esta pantalla a consultar todos los clientes registrados. Para ser cliente, un usuario debe haberse previamente registrado en la página mediante su computadora, o bien, algún empleado del local lo puede haber dado de alta en el sistema, como abarca el *CDU34 – Agregar cliente*.

En la pantalla se podrán visualizar los principales datos del cliente como su nombre y apellido, correo electrónico, documento, fecha de alta en el sistema, ciudad, provincia y calle de residencia y un número de teléfono de contacto. Además, se podrá buscar alguno particularmente ingresando su nombre, apellido o documento.

A continuación se detallarán los siguientes casos de uso asociados a esta pantalla.

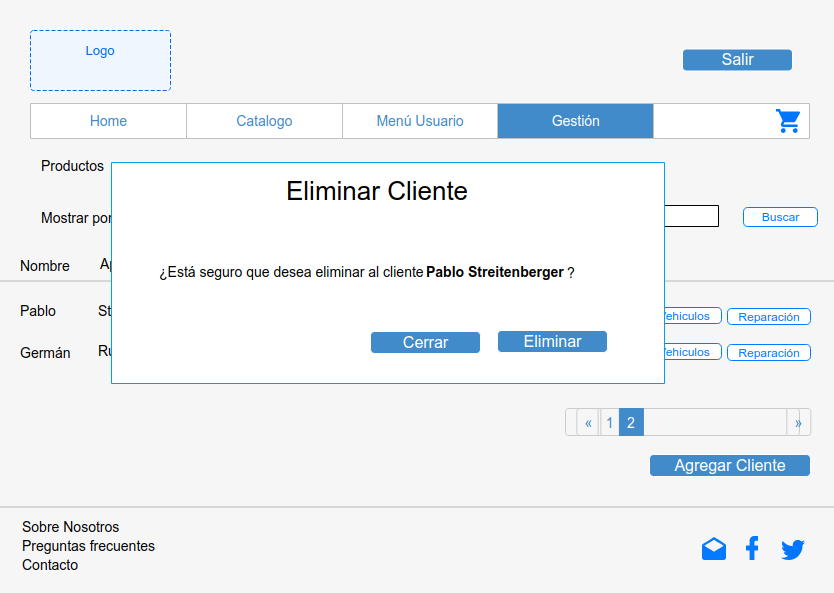
CDU34 – Agregar cliente

**

Para agregar un cliente, el empleado o administrador deberá completar todos los datos del cliente. Esto es útil para los casos en que el cliente se haya presentado a comprar personalmente en el local y aún no posea usuario registrado en el sistema. Por tal razón, el empleado puede darlo de alta para registrarle la compra o reparación de su vehículo y que el cliente luego pueda visualizarla desde su computadora y tenerla en su historial.

Luego de completar los datos, los cuales serán de carácter obligatorio el nombre y apellido, el email (que luego será el nombre de usuario del cliente y con el cual accederá al sistema), el documento (que representará la contraseña de ingreso inicial, que luego deberá cambiar una vez ingresado), y la provincia de residencia, el cliente ya será parte del sistema y podrá registrar compras y reparaciones de vehículos.

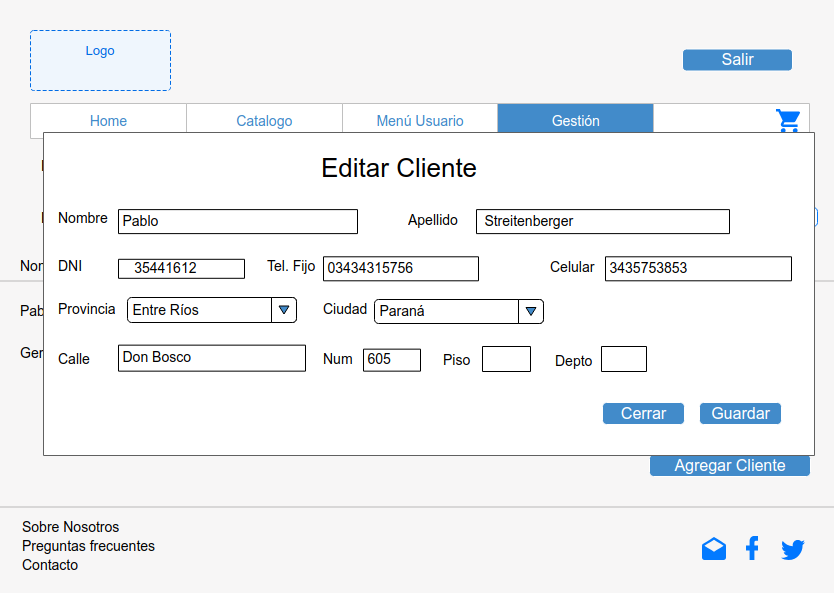
CDU35 – Eliminar cliente

**

Para el caso en que un cliente necesite ser dado de baja por algún motivo en especial, el empleado o administrador podrá eliminarlo del listado correspondiente. Para ello, se mostrará un mensaje de confirmación de eliminación para evitar cualquier tipo de error.

Una vez desligado el cliente del sistema, se podrá utilizar nuevamente su nombre de usuario, es decir, el correo electrónico.

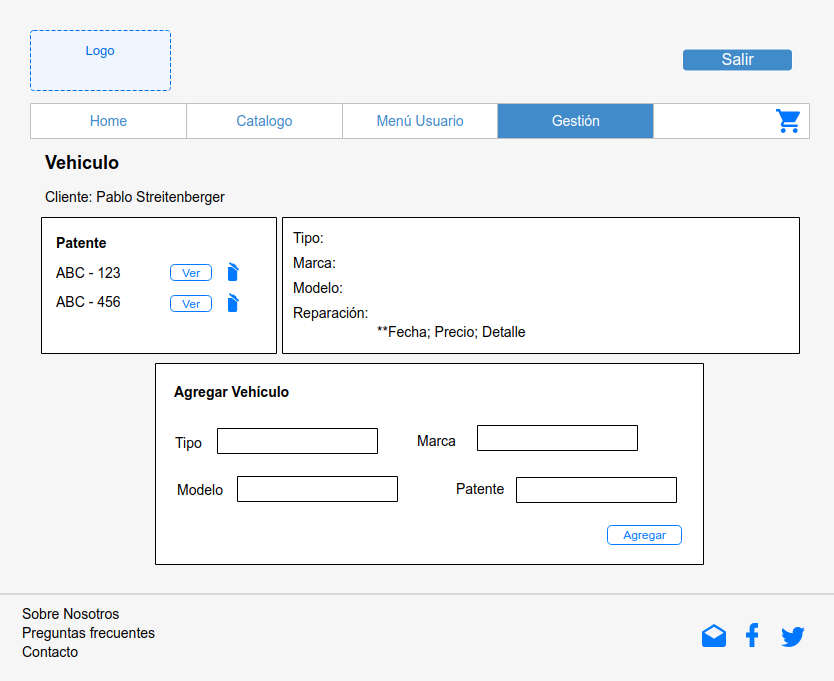
CDU36 – Editar cliente

**

De forma similar al alta de clientes, el sistema desplegará una pantalla con los datos completos del cliente en forma de edición. Cabe recordar, que todos menos el correo electrónico pueden ser modificados, pero siempre y cuando se respete la completitud de los mismos.

Al guardar los nuevos datos, el cliente habrá sido modificado dentro del sistema y verá los cambios en sus opciones personales al ingresar en su cuenta.

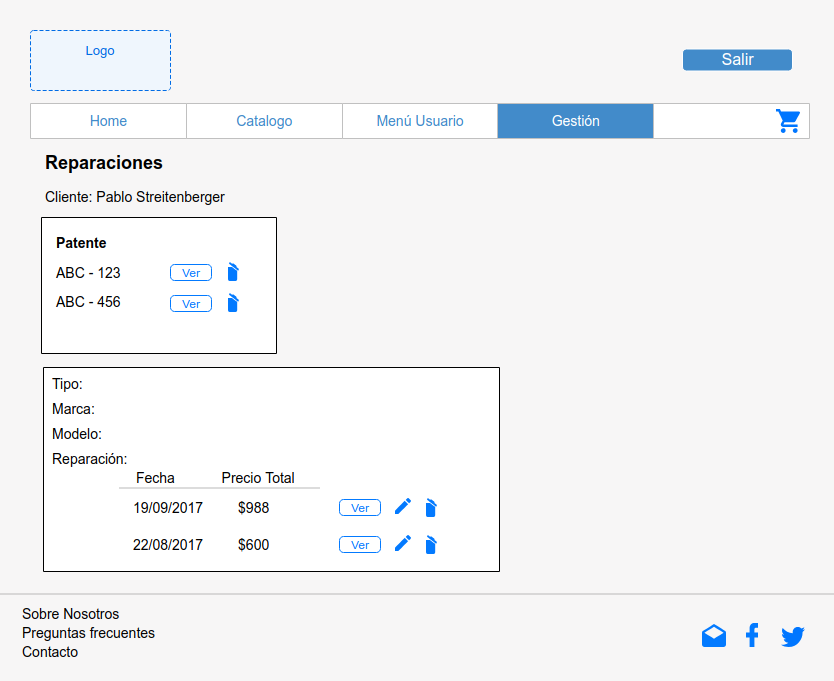
CDU37 – Ver vehículos del cliente



Al ingresar en esta vista, el empleado o administrador podrá visualizar un listado de los vehículos que tenga dados de alta el usuario, estos serán identificados por medio de su patente. Al hacer clic sobre alguno de ellos, se deberá desplegar la información que se posea sobre el mismo, es decir, el tipo de vehículo, la marca, el modelo y un historial en forma de listado de las reparaciones que haya tenido previamente ese vehículo.

Por otro lado, podrán ser dados de alta nuevos vehículos (*CDU38 – Agregar vehículo*). Para ello, deberá completarse los datos del vehículo y guardarlo. Esto lo agregará a la lista de vehículos del usuario. Y podrán ser, además, eliminados aquellos que ya no pertenezcan al cliente (*CDU39 – Eliminar vehículo*).

CDU40 – Ver reparaciones de vehículo

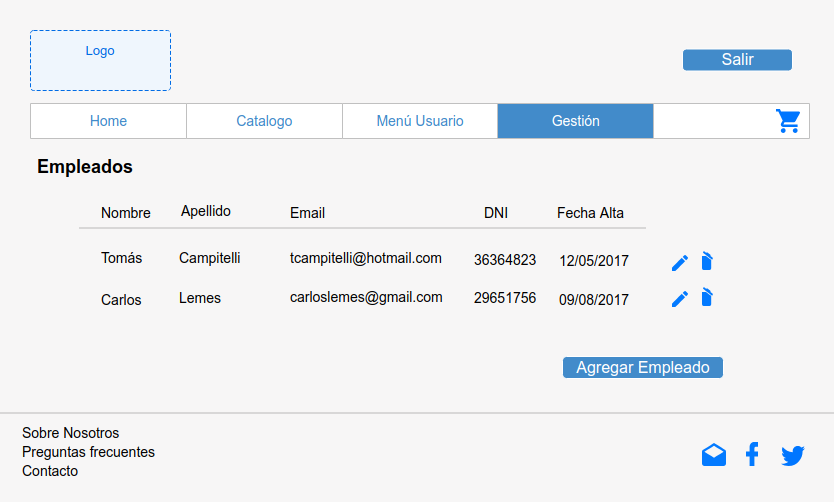
**

A esta vista se podrá acceder desde la gestión de usuario, del *CDU33 – Ver clientes*. Al ingresar, se podrá ver el listado de vehículos que el usuario posea, aquellos que hayan sido agregados previamente en el *CDU37 – Ver vehículos del cliente*.

Al hacer clic sobre alguno particular, se desplegará información del vehículo. Si el mismo posee reparaciones previas en su historial, las mismas podrán ser editadas (*CDU43 – Editar reparación*) o eliminadas (*CDU42 – Eliminar reparación*).

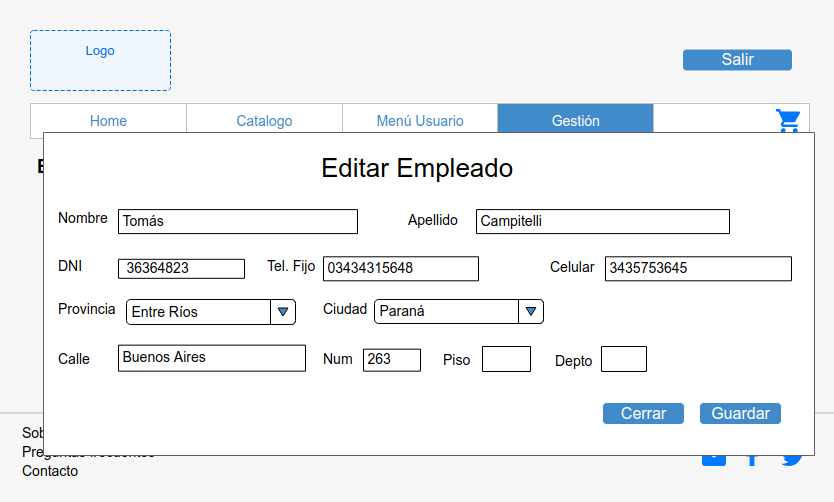
Por otro lado, se podrá agregar una nueva reparación (*CDU41 – Agregar reparación*). Para esto, deberá seleccionarse un vehículo del listado y completar los datos de la reparación, es decir, los repuestos y la cantidad utilizada y una descripción comentando brevemente los detalles. Una vez guardada, el cliente la podrá visualizar luego en su cuenta personal como se detalló en casos de uso anteriores.

CDU44 – Ver empleados

**

Este caso de uso podrá ser llevado a cabo sólo por el usuario administrador. Al ingresar verá un listado con todos aquellos empleados dados de alta en el sistema. Esto significa que estos usuarios podrán ingresar al sistema con rol de empleado. Entre los datos a visualizar se encuentran el nombre y apellido, el correo electrónico, el documento y la fecha en la que se dio de alta en el sistema.

CDU45 – Editar empleado

**

Similar a la edición de un cliente, este caso de uso puede ser sólo accedido por el administrador. En él, se le podrán modificar al empleado los datos personales, con excepción del correo electrónico utilizado como nombre de usuario para el ingreso al sistema. Una vez guardados los nuevos datos, el empleado los podrá visualizar desde su cuenta personal.

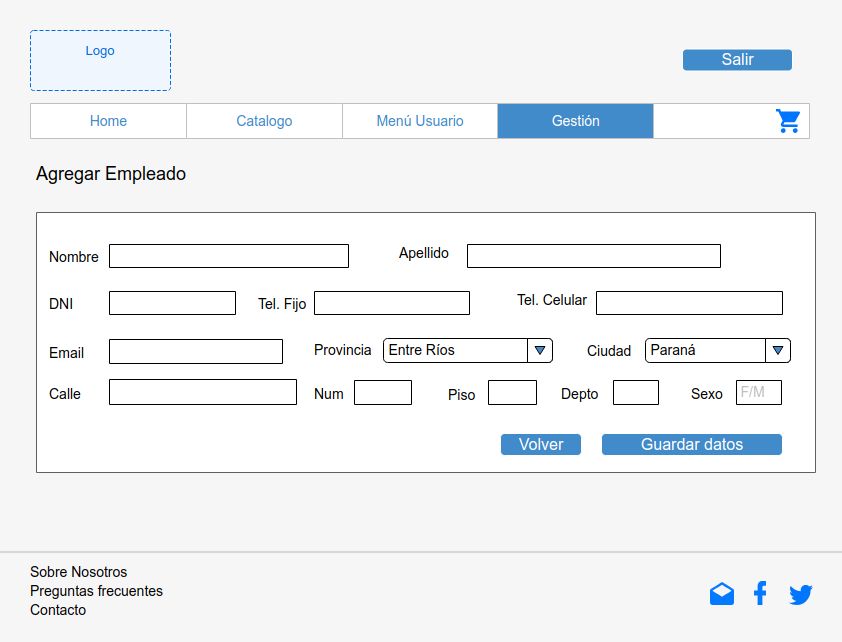
CDU46 – Eliminar empleado

**

Para el caso en que un empleado sea desvinculado del local, se podrá eliminar el usuario para que no sea posible su ingreso al sistema. Para esto, el administrador deberá eliminarlo, previamente aceptando el mensaje de confirmación.

Una vez realizado esto, el empleado no podrá ingresar más al sistema con ese rol.

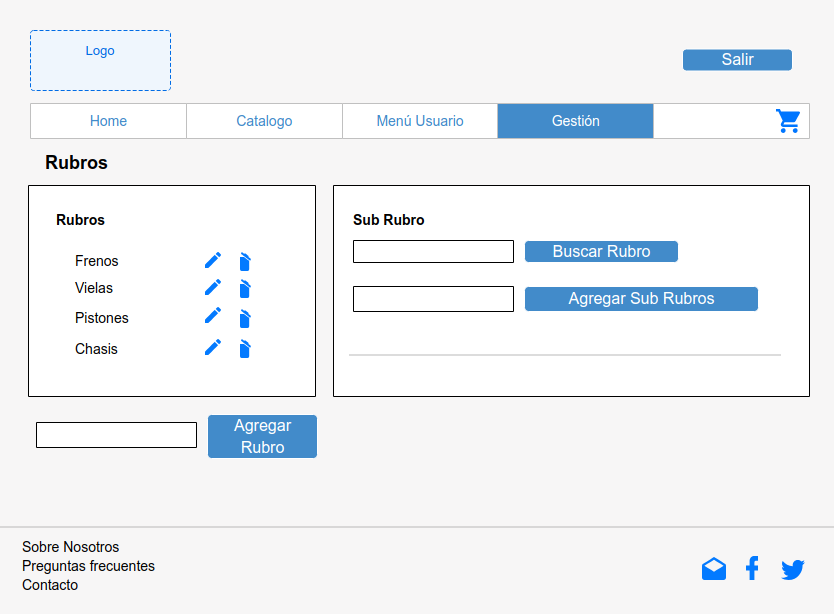
CDU47 – Agregar empleado

**

El administrador podrá dar de alta un nuevo empleado en el sistema para que tenga los privilegios y pueda generar ventas, gestionar productos y clientes. Para esto, el administrador deberá completar los datos del empleado en el sistema, teniendo en cuenta de que tanto el correo electrónico ingresado como el documento serán tomados como usuario y contraseña respectivamente para el posterior ingreso al sistema.

Una vez corroborados y guardados los datos, el empleado ya podrá ingresar al sistema como tal y gozar de las funcionalidades de empleado. La primera vez que ingrese al sistema, se le pedirá que cambie la contraseña generada por el administrador.

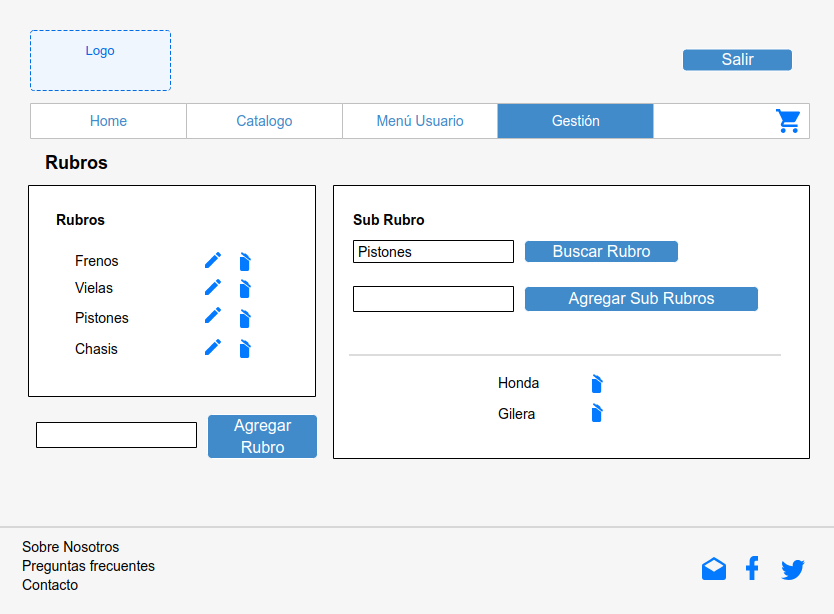
CDU48 – Ver rubros

**

El administrador puede gestionar los rubros y sub-rubros a los cuales pertenecerán los productos. Para esto, al ingresar a la pantalla, podrá ver un listado de todos los rubros que se encuentren cargados. Allí, tendrá la posibilidad de eliminarlos (*CDU49 – Eliminar rubro*), con lo cual se eliminarán también todos los sub-rubros y productos asociados, o agregar nuevos rubros (*CDU50 – Agregar rubro*). Para esto último, deberá escribir el nombre del rubro y presionar Agregar, lo cual ingresará el nuevo rubro en el sistema y se visualizará automáticamente en el listado.

Además se tendrá la posibilidad de editar el nombre de un rubro en particular, para el caso en que se desee cambiar alguno sin tener que eliminar todos sus sub-rubros asociados. Esto se realiza con el *CDU51 – Editar rubro*.

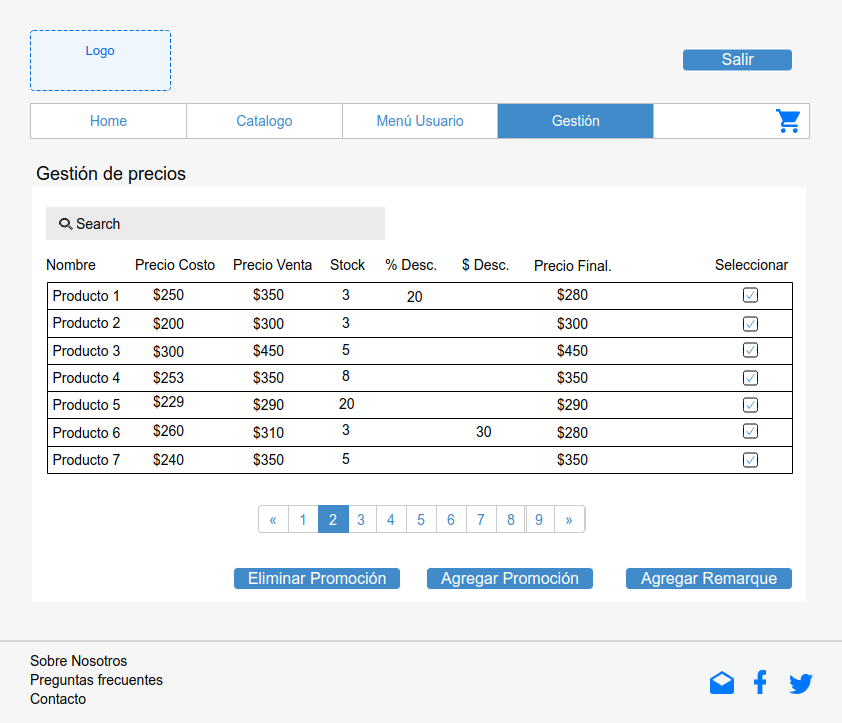
CDU52 – Ver sub-rubros

**

Similar a la visualización de los rubros, el administrador puede elegir un rubro particular, lo cual hará que se despliegue un listado con todos los sub-rubros asociados al rubro elegido.

Allí mismo, y de manera similar al caso de uso anterior, se podrá eliminar un sub-rubro (*CDU53 – Eliminar sub-rubro*), lo cual hará que todos los productos pertenecientes a ese sub-rubro se den de baja automáticamente; o podrá también agregar nuevos sub-rubros al rubro elegido (*CDU54 – Agregar sub-rubro*) escribiendo el nombre del mismo y presionando Agregar, que hará que se genere en el listado el nuevo sub-rubro.

CDU55 – Ver precios

**

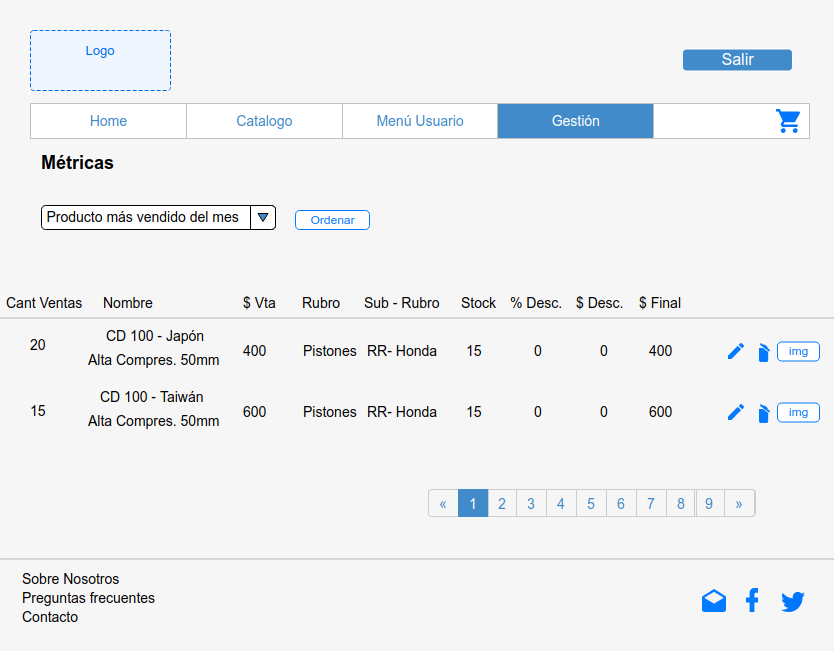
Sólo el administrador del sistema tiene los privilegios necesarios para realizar esta actividad. Al ingresar a la pantalla, se podrán observar todos los productos dados de alta en el sistema, y también se tendrá la posibilidad de buscar alguno en particular.

Podrán realizarse tres acciones en esta misma pantalla: agregar una promoción (*CDU56 – Agregar promoción*), eliminar una promoción existente (*CDU57 – Eliminar promoción*) y agregar remarque de precios a los productos (*CDU58 – Agregar remarque de precio*).

Para cualquiera de las opciones, será necesario tildar o seleccionar los productos que se verán afectados por la acción. Para el caso de agregar una promoción, se deberá elegir si se realizará un descuento fijo en pesos o si se deberá calcular el precio final luego de aplicarle un porcentaje de descuento. Luego de agregarlo, se podrán visualizar en la grilla de productos los descuentos agregados. De igual forma funcionará el remarque de precios, el cual tendrá el comportamiento contrario, ya que en vez de generar descuentos en pesos o porcentaje, generará aumento de los precios en pesos o algún porcentaje ingresado.

Por otro lado, aquellos productos a los que se les quiera eliminar una promoción agregada, deberán ser seleccionados en la grilla y luego de ello se podrán eliminar los descuentos presionando el botón dedicado a ello.

CDU59 – Ver métricas de venta

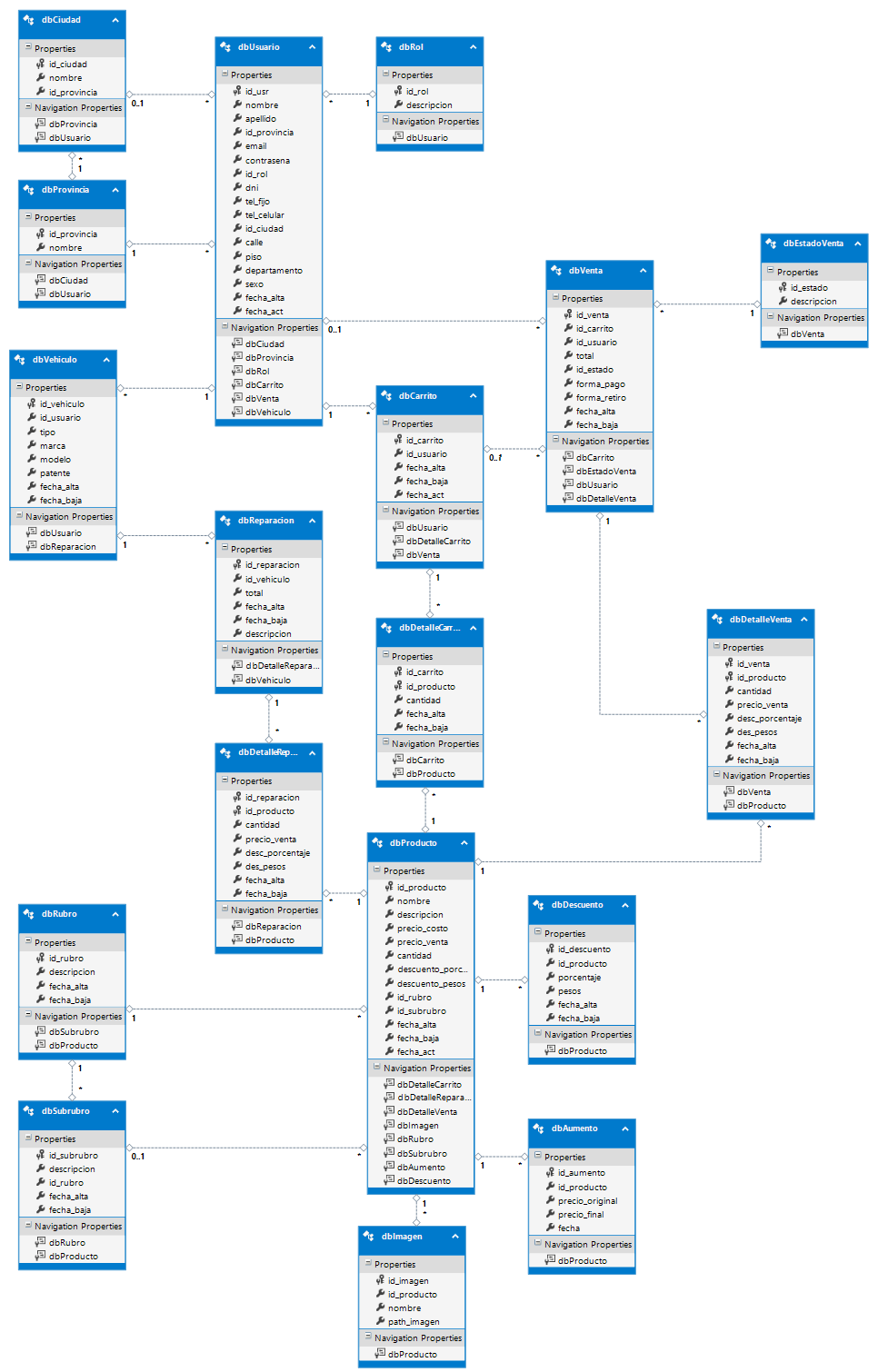
**

Sólo el usuario administrador podrá acceder a esta pantalla. En ella se muestran las métricas de ventas del último tiempo para que el mismo pueda tener una idea general del estado de las ventas del negocio.

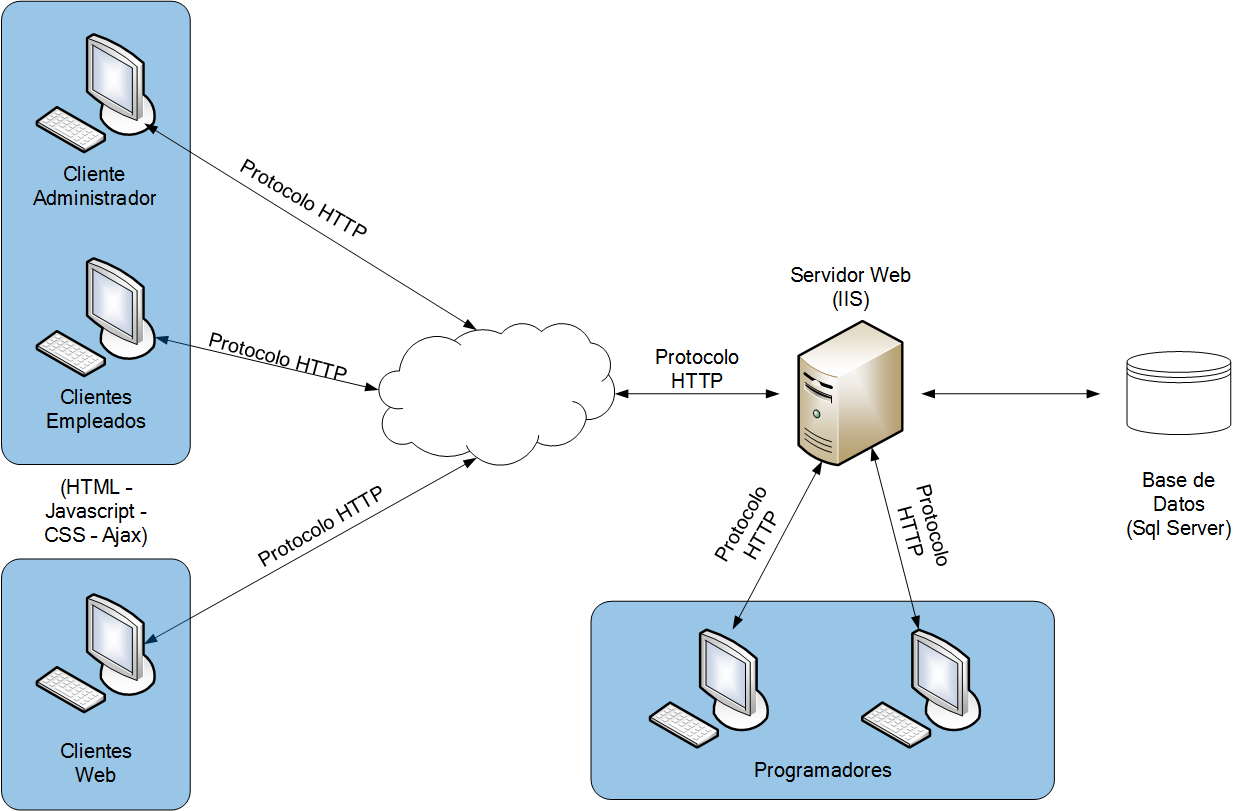
Podrá completar el filtro de fechas y con ello la tabla irá cambiando a los valores correspondientes para ese intervalo. En esta se verán datos como la cantidad de ventas del intervalo, total facturado, ganancia neta, entre otros datos de valor para el dueño del negocio.

## 

## *4.2. Modelo de entidad-relación*



# 5. Arquitectura del sistema



En el diagrama observamos la distribución física de los dispositivos de hardware que componen el sistema.

El Administrador, los Empleados y los Clientes del local pueden acceder al sistema a través de su navegador web desde una PC.

Los Empleados y el Administrador entre otras cosas pueden realizar la búsqueda, el alta, baja y modificación de productos, clientes, ventas y reparaciones. Además, el administrador puede gestionar a los empleados, generar promociones o incrementos por inflación o suba de precios de un proveedor, etc.

Por otro lado, los Clientes pueden buscar productos, agregarlos al carrito web para la compra de los mismos; además de visualizar su historial de compra y reparaciones

En todos los casos los dispositivos, incluidos los de los programadores, se conectan directamente con el servidor IIS que contiene el sitio web. A su vez, se conecta con la base de datos para responder a las solicitudes recibidas.

# 6. Construcción

En esta etapa se lleva adelante la construcción del sistema por medio de las iteraciones anteriormente planteadas. Esta construcción está guiada por el trabajo planteado en la etapa de análisis y diseño, lo que permite realizar el trabajo ordenadamente. Para lograrlo, se separaron algunas tareas entre los miembros del equipo para que cada uno pueda avanzar en sus tiempos libres, mientras que otras se hicieron en conjunto ya que era necesaria la presencia de ambos para decidir qué pasos seguir. También hubo reuniones con el cliente del sistema, el cual luego de cada iteración se encargaba de probar y familiarizarse con el sistema, lo cual sirvió de ayuda para seguir refinando los requerimientos de las etapas posteriores.

La división de responsabilidades planteó la necesidad de integrar continuamente el software desarrollado y evaluar rápidamente si existían fallos en los cambios realizados. En consecuencia, al cierre de cada iteración ya se tenían realizadas las pruebas correspondientes.

En la siguiente tabla se muestran los plazos propuestos y funcionalidades a realizar dentro de los mismos en la fase de construcción.

|  |  |  |  |
| --- | --- | --- | --- |
| **Incremento** | **Funcionalidad** | **Fecha inicio** | **Fecha fin** |
| **Inicial** | Se plantearán en esta iteración todas aquellas funcionalidades básicas principales que se deben implementar para que el sistema comience a funcionar. Los casos de uso que incluye son, entre otros:   * CDU05 – Autenticar usuario * CDU09 – Desloguear usuario * CDU01 – Ver pantalla de inicio * CDU02 – Ver formulario de contacto * CDU10 – Registrar usuario * CDU06 – Ver catálogo de productos * CDU08 – Ver detalle de producto * CDU15 – Ver carrito   Estos casos de uso implican la creación de la arquitectura general del sistema, ya que implican (dejando de lado la posibilidad de compra de productos) el circuito completo para el usuario. | 06/11/2017 | 23/02/2018 |
| **1** | En esta iteración, el usuario ya tendrá la posibilidad de generar una compra. Por lo tanto, se contemplan los casos de uso necesarios para realizarlo que, entre otros se encuentran:   * CDU16 – Agregar producto al carrito * CDU17 – Eliminar producto del carrito * CDU18 – Modificar cantidad de producto * CDU19 – Elegir método de envío * CDU20 – Elegir método de pago * CDU21 – Ver compras realizadas | 26/02/2018 | 25/07/2018 |
| **2** | En la segunda iteración, se realizarán los casos de uso requeridos tanto por el empleado como por el administrador para gestionar el sistema, por ejemplo:   * CDU26 – Ver productos * CDU29 – Agregar producto * CDU28 – Eliminar producto * CDU32 – Agregar imagen a un producto * CDU22 – Ver ventas * CDU23 – Ver detalle venta * CDU25 – Agregar venta * CDU24 – Cambiar estado de la venta * CDU41 – Agregar reparación * CDU38 – Agregar vehículo | 26/07/2018 | 28/09/2018 |
| **3** | Por último se realizarán aquellos casos de uso que sólo son gestionables por el administrador del sistema y que, si bien son una parte importante del mismo, no son exclusivamente necesarios para el funcionamiento en general:   * CDU50 – Agregar rubro * CDU54 – Agregar sub-rubro * CDU44 – Ver empleados * CDU47 – Agregar empleado * CDU46 – Eliminar empleado * CDU56 – Agregar promoción * CDU58 – Agregar remarque de precios * CDU59 – Ver métricas de venta | 01/10/2018 | 16/11/2018 |

Cabe aclarar que desde el 19/11/2018 al 28/12/2018 el equipo se dedicó exclusivamente al desarrollo de este informe.

A continuación se describe el trabajo realizado y los criterios tenidos en cuenta durante el desarrollo de las iteraciones propuestas en el plan de Iteraciones de la fase de construcción. También se describen las elecciones que se tomaron al momento de resolver cómo se implementarían las funcionalidades:

## *6.1. Iteración 0*

La idea principal de esta primera iteración, fue determinar la estructura general del sistema, para poder generar la conectividad necesaria y que los usuarios pudieran interactuar desde el primer momento. Si bien en esta etapa no se habilita el ingreso de usuarios externos, para que el cliente pudiera realizar pruebas resultaba necesario.

Por lo tanto, se realizaron los casos de uso que permiten que un usuario se registre e ingrese al sistema, vea sus datos personales, recorra el catálogo permitiéndole ver tanto los productos como el detalle de cada uno y finalmente se desloguee del sistema.

Para que esto fuera posible, se cargaron en primera instancia datos de prueba directamente desde la base de datos, ya que se deja para las próximas iteraciones la gestión de los productos.

Durante esta etapa, se decidió utilizar un estándar de codificación para el desarrollo de la aplicación web y, a continuación, se explica el mismo:

### 6.1.1. IDE para el desarrollo de la aplicación

Se decidió la utilización de un Entorno de Desarrollo para la programación del sistema debido a que provee un marco de trabajo amigable para la mayoría de los lenguajes de programación y puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto. A continuación se detallan las ventajas que se tuvieron en cuenta a la hora de elegir trabajar con un IDE. Un IDE permite:

* Realizar debug del código.
* Encontrar fácilmente el código donde se declara cierta función o variable.
* Validar el código en cuanto a errores de sintaxis.
* Realizar refactorización de código.
* Utilizar diferentes tipos de lenguaje (C#, JavaScript, CSS, etc.).
* Realizar test unitarios.

Se determinó utilizar el IDE Visual Studio, que además aporta las siguientes ventajas:

* Permite vincular fácilmente la base de datos con el código y visualizar el diagrama de tablas resultante.
* Permite realizar el versionado de código con Git desde su interfaz.
* Su versión Community es gratuita y ofrece la mayoría de las herramientas necesarias.
* Permite instalar paquetes externos que brinden herramientas extras, como por ejemplo para el análisis de la cobertura de código alcanzada por los test unitarios.

Por otro lado, se deben también, indicar las desventajas de utilizarlo, entre las que se encuentran las siguientes:

* Es necesaria una gran cantidad de recursos del sistema, por lo que se requieren computadoras con cierta capacidad de procesamiento (1GB de RAM para la versión 2015 que finalmente utilizamos, contra 4GB de RAM y disco de estado sólido SSD para la versión 2017 considerada en un principio).
* Las versiones no básicas son pagas, las cuales aportan más herramientas de programación.
* Se necesita considerable cantidad de tiempo para configurar el entorno y poder ejecutar el código.

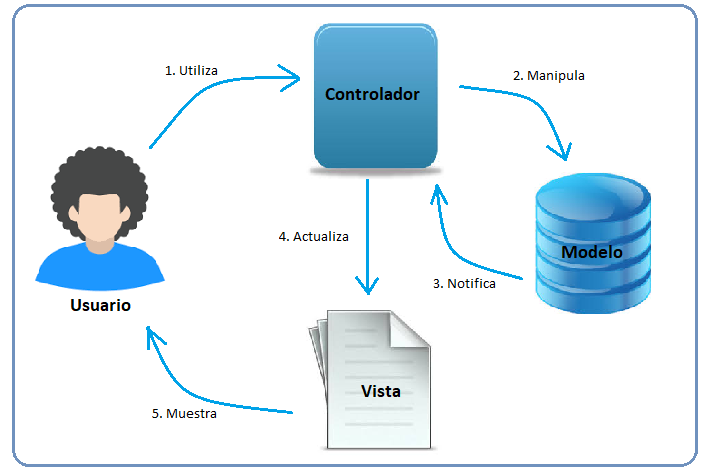
Más allá de las desventajas citadas, se llegó a la conclusión que las ventajas las superan y permiten un mejor desarrollo del trabajo.

Por otro lado, se tuvo que definir el patrón de arquitectura del software a utilizar. Se optó por el modelo MVC (Modelo-Vista-Controlador) que separa los datos y la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones [13]. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

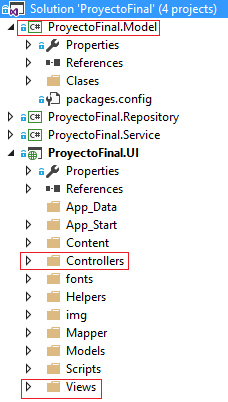
**Modelo:**Es la lógica de negocio del software, comúnmente hace referencia, pero no está limitado a, la base de datos, entidades, repositorios y consultas. Todo lo referente a manipulación de datos en el nivel más bajo, se encuentra en el modelo, lo que se denomina el CRUD del sistema (Create, Read, Update, Delete), las cuatro operaciones esenciales que se pueden realizar con los datos, además de la validación de los mismos.

**Vista:**Es la interfaz de usuario, todo lo referente a la presentación de la aplicación estará de forma organizada en múltiples vistas, en las cuales no habrá ningún tipo de lógica compleja. Las vistas deben estar limitadas a mostrar información únicamente, con lógica sencilla como: impresión de variables, condicionales y ciclos para repetir un elemento. Toda esta información debe ser procesada y validada previamente por el controlador y el modelo, de tal forma que se reduzca al mínimo la programación en la vista. En ellas también se asignan formatos especiales a los datos mostrados, como formato de monedas, fechas, entre otros; ya que estos datos vienen del modelo guardados en un formato estándar, queda de parte de la vista mostrarlos de forma amigable al usuario.

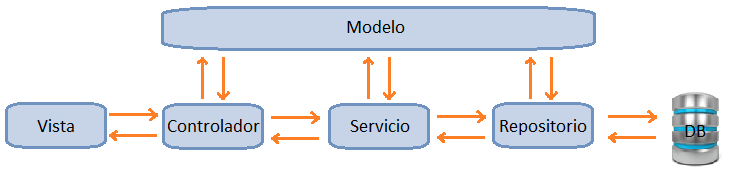
**Controlador:**Es el intermediario entre la vista y el modelo, el controlador se encarga de atender y procesar las acciones que realice el usuario a través de la interfaz, enviando dicha información al modelo (de ser necesario) y regresando una respuesta a la vista. Los controladores deben mantenerse también con un mínimo de lógica y ser lo más modulares posible. En el controlador también existe una capa de validación, la cual debe controlar los datos y organizarlos de tal forma que al momento de ser enviados al modelo, se realice la validación final para ejecutar la operación deseada por el usuario.



A continuación se puede observar el resultado de aplicar MVC a la arquitectura del sistema:



Por medio de esta arquitectura se intenta representar lo siguiente:



Como se mencionó anteriormente, la Vista muestra los datos requeridos por el usuario de una forma amigable. Estos datos son enviados por el Controlador, el cual se encarga de la recepción de peticiones y envío de los resultados de las mismas. Para poder determinar estos resultados, necesita comunicarse por medio del Servicio, el cual funciona como interfaz pública. Esto quiere decir que, sólo los métodos expuestos en el Servicio pueden ser accedidos por el Controlador, para preservar la seguridad de las capas del sistema.

El Servicio es el encargado de ejecutar las peticiones al Repositorio, el cual es la última capa y representa lo más bajo del sistema, es decir, es el que se comunica directamente con la base de datos, ejecuta las consultas, y devuelve los resultados de las mismas al Servicio.

Por otro lado, se encuentra transversalmente el Modelo, el cual sirve de apoyo tanto al Controlador, como al Servicio y Repositorio. Este último, es el encargado de realizar la transformación de los datos resultantes de la base de datos en entidades del Modelo.

Se puede observar en la imagen que no se contempló que la Vista se comunique con el Modelo. Esto se planificó así ya que se utilizan View Models,  los cuales representan los datos que se desean mostrar en la Vista. Sirven tanto como un método más de seguridad como para, por ejemplo, mostrar en la Vista datos de dos o más entidades del modelo. A modo de ejemplo, si se tiene la clase Provincia y Usuario en el Modelo, y se desearan mostrar en la Vista los datos de la provincia en la que reside el usuario, se necesitarían enviar a la Vista dos clases del Modelo. Con la ayuda de un View Model es posible crear un nuevo objeto que integre estas dos en el Controlador, y se envíe sin problemas a la Vista para ser mostrado.

### 6.1.2. Elección del repositorio remoto

Como se detalló anteriormente, se decidió utilizar Git para el control del versionado del código. Con cada commit realizado, el código debía guardarse en algún repositorio remoto para luego poder recuperarlos en caso de errores, y realizar ramas luego de cada iteración terminada.

Para esto se decidió utilizar el alojamiento de Visual Studio. Estos repositorios permiten alojar código y tienen compatibilidad con cualquier cliente GIT. Además se integran a la perfección con Visual Studio y tiene herramientas con las que es posible comparar código, útil para comparar dos o más commits, y tableros en los cuales se pueden asignar requerimientos a cualquier miembro del equipo de desarrollo.

Por otro lado, se decidió la utilización ya que, en un futuro la idea del cliente es llevar la base de datos a la nube para no depender de los problemas que conllevan las bases de datos locales. Por lo tanto, es posible luego utilizar el almacenamiento que brinda Azure.

### 6.1.3. Cierre de la iteración 0

Al finalizar se obtuvo una versión funcional del sistema que cumplía con los requisitos y metas propuestas al inicio del mismo. Sin embargo, se tuvo un retraso considerable con respecto a la fecha de finalización supuesta en el plan. Esto se debió a que esta etapa incluyó el aprendizaje y configuración de las herramientas y tecnologías necesarias para la construcción del sistema.

## *6.2. Iteración 1*

En esta iteración, el objetivo general era que el usuario ya registrado y logueado en el sistema pueda armar una compra. Para ello, el desarrollo se centró en los casos de uso de gestión del carrito web.

En una primera etapa, se creó la posibilidad de agregar productos al carrito, tanto desde el catálogo como desde el detalle de un producto en particular. Posteriormente, se logró que el usuario pudiera elegir la cantidad deseada de cada producto agregado al carrito para luego proceder a la compra.

Se le brinda la posibilidad al usuario de pagar y retirar por el local o pagar mediante alguno de los medios de pago que ofrece Mercado Pago, y recibir el pedido en su domicilio mediante Mercado Envíos. Luego de proceder con la compra (fallida o exitosa), el cliente es redirigido automáticamente al sistema, el cual le indica un mensaje informativo sobre la compra.

Por último, y para completar el circuito del cliente, se agregó la página donde él mismo pudiera ver las compras realizadas, reparaciones (opción a desarrollar en las próximas iteraciones) y el estado de cada una.

### 6.2.1. Cierre de la iteración 1

Luego de terminar con la iteración, se logró terminar el circuito del cliente, ya que son posibles todas las acciones que puede realizar un usuario en el sistema.

Si bien, al no estar implementada la gestión de ventas por parte de los empleados, aún no es posible visualizar el estado de los pedidos realizados a través de la web, ni tampoco se contemplan aún las ventas y reparaciones presenciales, por lo que todavía no se permite la utilización masiva para todos los usuarios.

En cuanto a las estimaciones, se terminó necesitando más tiempo del planeado inicialmente ya que, se tuvo que aprender y familiarizar con las herramientas de Mercado Pago y Mercado Envíos, y la forma de utilizar cada una con sus particularidades.

## *6.3. Iteración 2*

En la segunda iteración, se planeó junto con el cliente, realizar aquellos casos de uso que impliquen tanto a los empleados del local como al propio administrador, que realizará, a su vez, el mismo trabajo que los empleados inicialmente.

Para lograr esto, se comenzó con la gestión de productos, la cual resultaba ser la más importante, ya que teniendo esto implementado, los usuarios podrían ir cargando los productos a mostrar en el catálogo junto con sus imágenes.

A continuación, se prosiguió con la implementación de la gestión de ventas, lo cual permitiría realizar ventas a clientes que se presentaran presencialmente en el local y, además, observar las ventas realizadas a través de la web y cambiarles su estado para alertar al cliente sobre su pedido.

Luego, siguió el desarrollo de la gestión de clientes. Útil, no sólo para dar de alta nuevos usuarios y para observar aquellos registrados a través de la web, sino también porque esta parte incluye la gestión de los vehículos y reparaciones  de cada usuario. Así, un cliente podría presentarse a reparar alguno de sus vehículos en el local e inmediatamente observaría el detalle en su cuenta personal.

### 6.3.1. Cierre de la iteración 2

Al término de la iteración, se pudieron completar todos los casos de uso planteados  en un principio.

Cabe destacar que las iteraciones anteriores fueron siendo probadas a medida que se le entregaron al cliente. Con lo cual, resultaba importante no cambiar el funcionamiento de lo entregado culpa de las nuevos desarrollos. Esto retrasó por unos días el proyecto por sobre lo estimado previamente. Retraso que no habría ocurrido probablemente si el equipo de desarrollo hubiera tenido más experiencia en el análisis de los requisitos y, por sobre todo, la relación necesaria entre algunos casos de uso.

## *6.4. Iteración 3*

Por último, en la iteración final se decidió realizar aquellos casos de uso orientados estrictamente al usuario administrador, ya que, si bien son muy importantes para el sistema, no imposibilitaban la utilización del mismo mientras se realizaba esta última etapa.

Para esto, se comenzó por la gestión de los empleados, la cual resulta útil para dar de alta o baja nuevos usuarios que puedan ingresar y realizar las funciones de empleado.

Luego se optó por continuar con la gestión de rubros y sub-rubros. Aunque previamente se realizó una carga mínima de rubros y productos asociados para que el cliente pudiera ir probando la aplicación, este gestor le brinda la posibilidad al administrador de agregar y quitar rubros del listado, junto con los productos asociados a ellos.

A continuación, se realizó la gestión de precios. Se determinó con el cliente que los precios podrían tener que aumentarse masivamente por incrementos de inflación o variación del dólar, y, a su vez, resultaría necesario incluir promociones y descuentos sobre determinados productos por algún tiempo determinado. Por lo tanto, se realizaron los casos de uso relacionados a los pedidos y se corroboraron ingresando al catálogo de productos para observar los cambios de precio.

Por último, pero no menos importante, se creó la vista de métricas de ventas. Por medio de esta, el usuario administrador puede consultar y realizar un seguimiento del estado de las ventas del negocio y, por este medio, realizar acciones de gestión de los precios que ayuden a incrementar los ingresos.

### 6.4.1. Cierre de la iteración 3

Al término de la última iteración, se le entrega al cliente el sistema funcionando completamente. Si bien se fueron realizando pruebas al fin de cada una de las etapas, se planificó con el usuario que, mientras el equipo de desarrollo completara la documentación que aquí se detalla, él iría probando el sistema, para luego realizar las últimas mejoras y corrección de errores.

Con respecto al tiempo consumido por esta iteración, se pudo terminar con todas las funcionalidades días antes de lo estimado, por lo que se llevó a cabo un nuevo caso de uso agregado: la gestión de métricas de venta. Cabe indicar que, esta solicitud, no fue inicialmente pedida por el usuario ya que al no contar previamente con un sistema de tal magnitud no visualizó la posibilidad de tener indicadores de ventas. Luego de reuniones, y debido a la utilización del sistema mientras se desarrollaba, surgió la idea conjunta de agregar este requerimiento.

# 7. Conclusiones

## *7.1. Conclusiones finales*

En la propuesta de este proyecto de tesis, se hizo énfasis en la explotación de las ventajas que brinda el comercio electrónico en el éxito de un negocio, en este tiempo en donde las tecnologías de la información forman parte de nuestra vida diaria. Se propuso realizar un sistema que permitiera conectar al comprador y al vendedor de una manera más personalizada y oportuna.

El producto creado resultó de gran importancia para nuestro cliente, el cual se encontraba operando su negocio sin ningún tipo de tecnología, lo que le resultaba en pérdidas de dinero y muy pocas ventas. Además, consideramos que es un producto original desde el punto de vista de la personalización que se le brinda al cliente en cuanto a sus vehículos y las reparaciones realizadas en el local. Esto, le da la oportunidad de realizar un seguimiento puntual de los problemas que haya tenido su vehículo en el tiempo e, inclusive, le sirve como herramienta a la hora de venderlo, ya que puede demostrarle al potencial comprador todas las reparaciones, productos que se le cambiaron o arreglaron y la fecha de realización.

Las experiencias vividas durante el transcurso del proyecto nos demostraron que, la metodología utilizada como guía para el desarrollo, nos permitió refinar las funcionalidades del sistema desde un comienzo. A pesar de no obtener un sistema completo funcionando en etapas tempranas, la metodología nos sirvió para tener pequeños módulos del sistema que el cliente pudo ir probando mientras seguíamos el desarrollo siguiente. Esto, permitió que al entregar el último incremento, el usuario estuviera familiarizado con el sistema y listo para utilizarlo.

Podemos concluir que el producto desarrollado en este proyecto responde a la demanda necesitada por el cliente e inclusive le brinda más herramientas que las que se pensó en un principio, como por ejemplo, la métrica de ventas.

## *7.2. Trabajos futuros*

Durante el desarrollo del sistema, fuimos descubriendo posibilidades para expandir las funcionalidades que el mismo soporta. A continuación detallamos algunas de ellas que nos proponemos desarrollar para mejorar el sistema:

* Utilizar Azure como base de datos en la nube. Al incrementar los usuarios del sistema, será necesario contar con un servicio en la nube que brinde escalabilidad, seguridad y, sobre todo, disponibilidad en todo momento. Al mantener la base de datos local estas tres cosas se vuelven complejas con el aumento de los usuarios. Además, SQL Server no es libre y sólo posee gratuita su versión Express que ofrece 10 GB de almacenamiento.
* Agregar el módulo de facturación electrónica de AFIP para las ventas. Si bien es algo que se había acordado en un principio con el cliente, se decidió en conjunto dejarlo para futuras implementaciones, ya que aún no se encuentra empadronado en el régimen monotributista. Además, cuando esto suceda, ingresaría en categorías exentas aún de emitir obligatoriamente las facturas electrónicamente. Por lo tanto, sólo necesitaría el talonario de papel cuando decida empadronarse. Pero, cuando las ventas comiencen a incrementarse y su categoría supere la letra F, necesitará el módulo para que la facturación se genere automáticamente luego de las ventas.
* Implementar como mejora la posibilidad de agregar los precios de los productos en dólares, pero que al usuario le aparezcan en pesos, transformados con la cotización del momento. Este cambio surgió en etapas finales, cuando el cliente observó que muchos de sus productos son importados y deberían variar su precio constantemente, debido a los cambios en la cotización del dólar.

# 8. Bibliografía

## *8.1. Referencias bibliográficas*

[1] <https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web>

[2] <https://developer.mozilla.org/es/docs/Web/HTTP>

[3] <https://developer.mozilla.org/es/docs/Web/HTML>

[4] <https://developer.mozilla.org/es/docs/Web/CSS>

[5] <https://librosweb.es/libro/javascript/capitulo-1.html>

[6] <https://desarrolloweb.com/articulos/introduccion-jquery.html>

[7] <http://www.ibrugor.com/blog/que-es-ajax-para-que-sirve/>

[8] <https://es.wikipedia.org/wiki/Bootstrap_(framework)>

[9] <https://es.wikipedia.org/wiki/C_Sharp>

[10] <https://es.wikipedia.org/wiki/Microsoft_SQL_Server>

[11] <https://es.wikipedia.org/wiki/ADO.NET_Entity_Framework>

[12] <https://es.wikipedia.org/wiki/Microsoft_Visual_Studio>

[13] <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>

## *8.2. Herramientas y programas utilizados*

* Visual Studio Community 2015: <https://visualstudio.microsoft.com/es/vs/older-downloads/>
* Axocover: <https://marketplace.visualstudio.com/items?itemName=axodox1.AxoCover>
* Bootstrap: <https://getbootstrap.com/>
* JQuery: <https://jquery.com/download/>
* SQL Server Express 2008: <https://www.microsoft.com/es-ar/download/details.aspx?id=1695>
* Entity Framework 6: <https://docs.microsoft.com/en-us/ef/ef6/fundamentals/install>
* Versionado de código: <https://proyectocampagnayunges.visualstudio.com/_git/ProyectoDaniBarby>
* Web para mock-ups: <https://app.moqups.com/>

## *8.3. Bibliografía y sitios consultados*

* Definiciones: <https://developer.mozilla.org/es/docs/Web>
* Manuales de programación y buenas prácticas: <https://librosweb.es/libros/>
* Métodos para generar casos de prueba: <https://www.redalyc.org/articulo.oa?id=75017199005>
* Comunidad de programadores: <https://stackoverflow.com/>
* Ayuda de programación: <http://www.w3schools.com/>
* Api de Mercado Pago: <https://www.mercadopago.com.ar/developers/es/>
* Tutorial de manejo de Sesiones: <https://www.guru99.com/asp-net-session-management.html>
* Apis de AFIP para facturación electrónica: <https://www.afip.gob.ar/ws/>
* Tutorial de instalación en servidor IIS: <https://www.guru99.com/deploying-website-iis.html>
* Pressman, Roger S. “Ingeniería de Software: Un enfoque práctico” – 7ma Ed. McGraw-Hill. 2010.
* Sommerville, Ian. “Ingeniería de Software”– 9na Edición. Addison-Wesley. 2011.
* Bracalenti, Claudio. “Gestión de Riesgo en los Proyectos de Software” – 2009.
* Apuntes y diapositivas de la Cátedra de Proyecto Final.