# JUSTIFICACIÓN DEL PROYECTO

Shop’s Admin permite la gestión interna de una tienda de productos tecnológicos.

En ella trataremos con dos tipos de clientes: Minoristas y Mayoristas, los mayoristas tendrán privilegios como elegir si paga la compra en el acto o poder aplicarle un descuento a las facturas.

La aplicación tendrá 2 módulos: Venta y Administración. A la parte de administración solo accederán los usuarios administradores, por lo que tendremos usuarios administradores y empleados.

Se tienen disponibles varias plantillas para cada módulo para facilitar el uso de la aplicación.

# PLANIFICACIÓN

Para la planificación del proyecto se ha definido 5 tareas, Análisis, Diseño, Implementación, Pruebas y Entrega del proyecto:

1. En el **Análisis** (8 abril – 15 abril): donde se realizan los estudios previos, se establecen los requisitos tanto físicos como funcionales y el modelado de datos.
2. En el **Diseño** (16 abril – 20 abril): se elabora los prototipos de baja fidelidad y se seleccionan las tecnologías a utilizar.
3. La fase de **Implementación** (21 abril – 25 mayo): la más larga y complicada, ya que se lleva a cabo todo el desarrollo del software de la aplicación.
4. En la etapa de **Pruebas** (26 mayo – 31 mayo): se elabora una lista con todas las funcionalidades a comprobar, se realizan y si surge algún fallo (que surgieron muchos) se le aplican cambios y mejoras que se vayan encontrando.
5. En la última tarea **Entrega del proyecto** (1 junio – 17 junio): se completa la documentación, se documenta el código y se realiza la presentación de la defensa del proyecto.

## Estimación de Costes

En la estimación de costes los únicos costes que tenemos que soportar son los costes de los RRHH, ya que el software utilizado será libre.

Se ha estimado unos costes teniendo en cuenta que se dedicaría unas 3-4h de Lunes a Viernes y 6-7h Sabados y Domingos, para escoger un valor medio se estiman unas 5 horas por día (35h semanales) y cada hora tendrá un precio de 10€.

El coste total económico del proyecto será de 3400€.

# ANÁLISIS

En la fase de análisis vamos a definir el comportamiento del sistema.

## Requisitos

### Tipos de Usuarios

Para empezar tendremos dos tipos de usuarios:

* **Administrador**: Usuario con todos los privilegios y el único con acceso al Módulo de Administración
* **Empleado**: Usuario que sólo tiene acceso al Módulo de Venta

### Módulo de Administración

#### Estadísticas

Será la **página principal** del módulo de Administración y mostrará diversos datos, como el número de ventas por el mes actual, mes anterior, semana actual y semana anterior, gráficos por tipo de cliente y, facturas pagadas y no pagadas. También, se mostrará los productos más y menos vendidos.

#### Menú agregar

El usuario podrá agregar proveedores, categorías, productos, clientes y usuarios.

El proceso de agregación es similar, primero se validan los datos y si son estos son correctos, se guardan y estarían visibles desde el Menú de Listar que veremos a continuación.

Destacando:

1. Que al añadir un usuario se le envía un correo con la contraseña
2. Y que al añadir el cliente, se elegirá el tipo: Mayorista o Minorista

#### Menú Listas

Se podrán ver todos los elementos teniendo la opción e ver cada elemento en detalle, modificarlo o, arlo de baja de alta, según su estado.

Destacando:

1. Que se podrá **buscar** por cualquier campo
2. Al **usuario** solo se podrá cambiarle su estado, si se le da de baja se le denegaría el acceso a la aplicación. Sus datos solo los podrá modificar el usuario cuando acceda a su perfil.
3. En la **lista de facturas** se podrá ver las facturas pendientes y pagadas.

En las **Pendientes** se podrá ver en PDF, marcarla como pagada o cambiarle un descuento, estas facturas pertenecerán a clientes mayoristas y en las **Pagadas**: se podrá ver o descargar en PDF

#### Plantillas

Se podrá cambiar la plantilla de cada módulo, cambiando totalmente su aspecto.

#### Avisos Stocks

Aparecerán avisos cuando un producto tenga un stock de 50 artículos o menos, esto se comprobará cada minuto.

### Módulo de Venta

#### Tienda

Se implementa la típica **tienda online** como la que hicimos en servidor, donde podemos ver los productos y añadirlos al **carrito**

En el **carrito** podemos cambiar la cantidad del producto que se va a comprar, validándola automáticamente, quitar un producto del carrito o vaciarlo completamente.

#### Proceso de venta

Cuando se finaliza la venta se elige el tipo de cliente que realiza la compra de una lista desplegable. Dándose un escenario para cada tipo de cliente:

* **Minorista**: se realiza la compra, se genera su albarán y su factura correspondiente, pudiendo verlas en PDF.
* **Mayorista**: se realiza la compra y si se ha elegido pagar en el acto, se generaría su albarán y su factura, como el cliente minorista. Si no se elige pagar en el acto, se guardaría el albarán en la última factura *no pagada* del cliente y se podrá mostrar el albarán en PDF.

## Casos de Usos

A continuación, podremos ver los diagramas de los casos de usos de las principales funcionalidades de la aplicación.

Los casos de usos sirven para especificar la **comunicación** y el **comportamiento** entre un sistema y los usuarios.

Como explicamos antes a la funciones del Módulo de administración solo accederán los administradores.

Y los primeros casos pertenecen al Módulo de Venta los dos usuarios tendrán acceso, y, por último, el perfil de usuario estará disponible en ambas partes para que el usuario cambie su perfil.

## Prototipos

Aquí podemos ver los prototipos de baja fidelidad de la aplicación, estos prototipos son una representación limitada del software, que permite explorar su uso en situaciones reales y son útiles para definir ideas.

Estos 4 prototipos son: las estadísticas, un ejemplo de agregar un elemento, un ejemplo de una lista y el listado con los productos con stocks bajos.

En estas diapositivas vemos ejemplos de cómo dar de baja o de alta a un elemento, de ver en detalle un elemento y la configuración de plantillas.

Y en estas diapositivas vemos pantallas del Módulo de Venta, como los productos, el carrito o los principales pasos del proceso de venta.

## Requisitos no funcionales de la aplicación

* Debe estar disponibles desde los navegadores más modernos.
* Se deben ver desde cualquier dispositivo para eso contamos con un diseño web responsive gracias a Boostrap, lo que también ayudará a la navegación por la aplicación.
* Y, por último, todas las tecnologías y herramientas utilizadas son gratuitas.

# MODELO DE DATOS

Aquí podemos observar el diagrama entidad-relación de la aplicación. Los aspectos a destacar son:

Por una parte, es como guardamos las *Ventas.* Cada venta se guarda en un “Albarán” y a su vez en una “Factura” que es donde tendrá la información más de detallada. Si la *Venta* es realizada por un “Cliente Mayorista” y elige no pagarla, se irá acumulando el “Albarán” en la última factura no pagada del cliente.

Por otra parte, también hay que mencionar los campos *referencia*  de las tablas “Producto”y *Categoría* y el campo *numalbaran* de la tabla “Albarán” que son campos o códigos autonuméricos que identifican a cada elemento. Además, el campo *numfactura* es el código que identifica a cada factura, y no puede existir dos facturas del mismo año con el mismo *número de factura*.

Estos campos son creados mediantes triggers o disparadores en la base de datos, cada vez que se realiza una inserción en ella.

# TECNOLOGÍAS

Todas las tecnologías utilizadas son libres.

En la parte del **cliente**, utilizaremos las ya conocidas HTML, CSS, JavaScript, Bootstrap como framework de diseño web y jQuery. En jQuery he usado 3 plugins:

* Highcharts: para los gráficos en la página principal
* Select2 para generar listas desplegables escribibles, para elegir el cliente en la venta
* DataTables plugin para generar tablas HTML dinámicas, usado en las facturas

En la parte del **servidor**, se utiliza el servidor apache, el lenguaje PHP y la base de datos MySQL.

Y para la comunicación asíncrona entre ambos, utilizamos AJAX.

La herramientas de desarrollo son: NetBeans como IDE(entorno de desarrollo), CodeIgniter como framework, MySQL WorkBench para el modelado de datos, GitHub como control de versiones, FileZilla para subir la aplicación a gerión, Balsamiq Mockups para el diseño de los prototipos de baja fidelidad y Creately para el diseño de los distintos diagramas.

# ARQUITECTURA MVC

El modelo-vista-controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario.

Ventajas que presenta:

* Reutilización del código
* Separación entre datos, lógica de negocio y presentación
* Facilita el desarrollo y el mantenimiento

Desventajas:

* Hay que ceñirse a la estructura del patrón
* Es más difícil de adaptarse
* Y obliga a mantener un gran número de ficheros por la distribución componentes

# AMPLIACIONES

1. Quitar el *index.php*  de las URLs generadas por CodeIgniter, lo que permitiría acceder más fácilmente al módulo de Administración además de tener URLs más amigables.
2. Constructor de búsqueda para los distintos elementos, pudiendo elegir el campo por el que buscar.
3. Pasarela de pago que permita pagar por tarjeta de crédito.
4. Implementar el plugin de jQuery [DataTables](#_DataTables) en todas las listas.
5. Instalador que permita disponer de la aplicación fácilmente en cualquier servidor.
6. Añadir fotos de los usuarios y clientes.
7. Poder realizar pedidos a los proveedores.
8. Avisos cuando las facturas lleven mucho tiempo sin ser pagadas.
9. Añadir más plantillas con diferentes diseños a cada módulo.
10. Permitir configurar cada plantilla manualmente, por ejemplo, eligiendo su color.
11. Permitir configurar distintos parámetros como el número de elementos a mostrar por página en cada lista o en el módulo de venta, el número de minutos que tiene que tardar en comprobar el stock de los productos, etc.