

# **ESPECIFICACIÓN DE REQUISITOS SOFTWARE**

Empresa: Joni Nueva Visión y Monaguillos

# ÍNDICE

<b>ÍNDICE</b>	<b>1</b>
<b>EL ÁMBITO DEL PROYECTO</b>	<b>3</b>
<b>EL ALCANCE DEL PROYECTO</b>	<b>3</b>
<b>MODELO DEL DOMINIO</b>	<b>4</b>
<b>ATRIBUTOS DEL SISTEMA SOFTWARE</b>	<b>5</b>
<b>FUNCIONES DEL PRODUCTO</b>	<b>7</b>
PRODUCTOS	7
VENTAS	7
PÁGINA WEB	7
CLIENTES	7
ADMINISTRADORES	7
PRODUCTOS	8
VENTAS	10
PÁGINA WEB	13
CLIENTES	14
ADMINISTRADORES	15
<b>HARDWARE NECESARIO</b>	<b>16</b>
On Premise:	16
Pequeño:	16
Mediano:	16
Grande:	17
Cloud Services:	17
Google Cloud Platform:	17
Amazon web services:	18
<b>TECNOLOGÍA DE SOFTWARE</b>	<b>18</b>
Front-End:	18
AngularJS:	18
React:	18
Vue.js:	18
Back-End:	19
Laravel:	19
Django:	19
Flask:	19
NodeJS:	19
Bases de Datos:	19
MySQL:	19
PostgreSQL:	19

MongoDB:	20
Redis:	20
Servidores Web:	20
Apache:	20
Nginx:	20

## EL ÁMBITO DEL PROYECTO

El proyecto puede analizar datos de plataformas de redes sociales para identificar usuarios con tendencias o síntomas depresivos y proporcionar enlace al sitio. Los usuarios con problemas psicológicos pueden realizar asesoramiento psicológico y comprar medicamentos a través de esta plataforma.

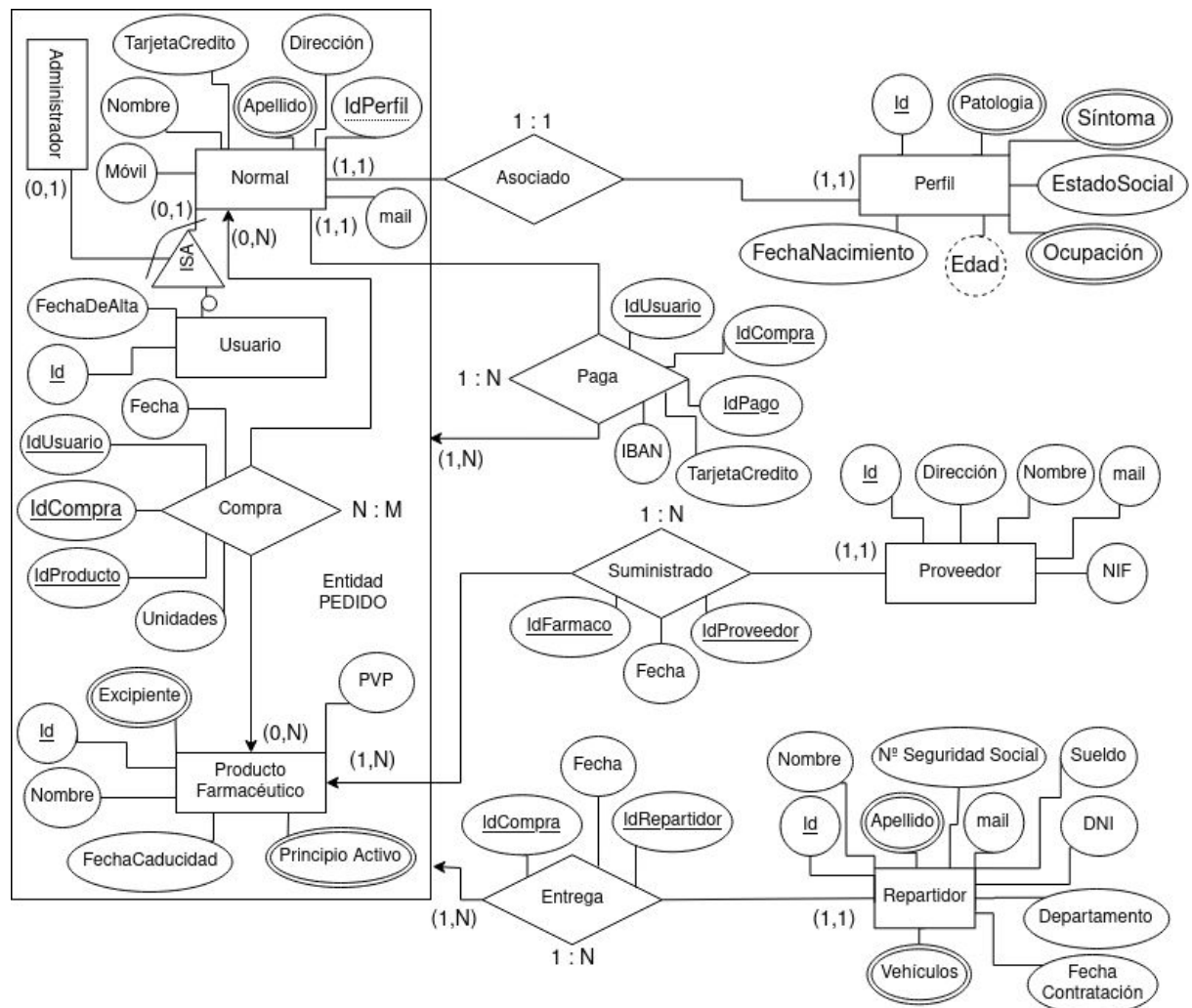
Solo el administrador puede ver y gestionar estos pedidos y datos específicos de los usuarios. Por otro lado, las instituciones médicas o clínicas pueden brindar consultas y respuestas a los usuarios que necesitan ayuda o tratamiento a través de la Web.

## EL ALCANCE DEL PROYECTO

El objetivo de este proyecto es facilitar que las personas con enfermedades mentales reciban tratamiento. Deberá cumplir con las siguientes especificaciones:

- Gestionar fuentes de información de usuarios potenciales: crear y eliminar información de usuarios. Los datos requeridos son: la fuente de información del usuario, los síntomas del usuario y alguna información básica del usuario.
- Gestionar información de usuarios: crear y eliminar registros de usuarios. Los datos necesarios para el registro son: nombre de usuario y clave aunque también se puede almacenar información extra como son: correo electrónico y móvil.
- Gestionar equipo médico: guardar información de médicos profesionales y tiempo de consulta disponible.
- Gestionar información de medicamentos: crear y eliminar medicamentos. Se necesita cierta información del proveedor, nombre del medicamento, precio, etc.
- Gestionar información de compra: crear pedido de usuario. Los datos necesarios son: Nombre de usuario, dirección, lista de medicamentos, precio total, etc.

# MODELO DEL DOMINIO



# REQUISITOS NO FUNCIONALES

- Rendimiento:
  - En el caso de que se adjunte una foto para el perfil del usuario, la velocidad del sistema oscila entre 2 y 5 segundos.
  - La velocidad del sistema para cargar una página de la aplicación web ha de durar menos de 3 segundos.
  - El proyecto debe tener una cierta capacidad de análisis y filtrado de datos para identificar a los usuarios con tendencias depresivas. La velocidad del análisis debe ser de 10.000 registros cada 10 ms.
- Concurrencia:
  - El sistema debe funcionar correctamente con al menos 150 personas al mismo tiempo, entre consultas, carros de la compra y transacciones. Por debajo de esta cantidad los tiempos de respuesta no deben verse alterados significativamente. En el caso de que el número de usuarios sea superior al máximo estipulado, el sistema tendrá un retardo de 0.2 segundos por acción siempre que el número de usuarios no sea superior al doble del máximo ya descrito.
- Fiabilidad:
  - La información de la BDD permanecerá consistente en caso de que una operación contra esta termine de forma abrupta. El objetivo será evitar la corrupción de los datos y la generación de redundancias.
  - Si se da una avería hardware en un nodo del sistema, debe existir un computador que actúe a modo de respaldo, de esta forma, el servicio perdido se recuperará lo más rápido posible.
- Seguridad:
  - La realización de un back-up tendrá lugar cada 24h.
  - Cualquier petición del lado cliente al lado del servidor estará cifrada con el protocolo https.
  - Toda comunicación del servidor con servicios externos estarán cifradas mediante criptografía SHA-512.
  - No se guardarán las contraseñas de los usuarios, si no un hash asociado a estas.
  - Las contraseñas de usuario han de tener una longitud mínima de 8 caracteres gráficos, debe existir al menos 1 carácter en mayúscula y 2 caracteres gráficos no alfanuméricos.
  - Si el cliente está inactivo durante 10 minutos, se le pedirá que se autentique de nuevo.

- Se utilizará un sistema de verificación en dos pasos para verificar la identidad de un cliente antes de cada pago.
- Para realizar un pago de forma segura, se utilizará la pasarela de pagos de paypal.
- Confidencialidad:
  - Los datos de los clientes no serán cedidos a terceros.
  - Solo se tratará información obtenida a partir de fuentes públicas de datos o ofrecida de buena voluntad por los usuarios.
  - La información recolectada se usará únicamente con la finalidad de detectar usuarios potenciales y hacerles publicidad dirigida a través de banners publicitarios o correos electrónicos.
  - La información sólo será empleada por algoritmos de inteligencia artificial y profesionales médicos.
- Disponibilidad:
  - Se puede contactar con un profesional médico de Lunes a Viernes de 09:00 a 14:00 y de 16:00 a 21:00, los Sábados de 09:00 a 14:00.
  - El sistema de ventas y la ayuda al cliente estarán accesibles 24h.
- Usabilidad:
  - Las interfaces de la aplicación serán intuitivas.
  - La curva de aprendizaje de los usuarios será baja, unos 10 minutos.
  - Los usuarios no han de recibir mensajes de error técnicos.
  - Se proporcionarán manuales de uso básico, medio y avanzado de la aplicación a los usuarios finales.
  - Existirá un sistema de ayuda online que resuelvan las dudas de los usuarios que experimenten cualquier tipo de problema o dificultad.
  - El servicio dispondrá de una versión móvil para que el usuario pueda comprar un medicamento cuando quiera.
  - Se pondrá a disposición del usuario un sistema de restablecimiento de contraseña en caso de extravío o de olvido.
- Mantenibilidad:
  - Se utilizará una arquitectura multicapa que permita gestionar el desarrollo de los diferentes módulos del producto.
  - Se harán uso de tecnologías de programación orientada a objetos, ya que estas:
    - Hacen que los programas sean más legibles y posean una mejor documentación.
    - Facilitan la reusabilidad de los componentes software.
    - Permiten construir componentes con un alto nivel de cohesión funcional y un bajo nivel acoplamiento entre estos.

- Simulan mejor los modelos de datos con los que representar el mundo real.

## REQUISITOS FUNCIONALES

En primer lugar se enumeran las funciones con una breve descripción, y en segundo lugar se explicará cada una de ellas en detalle.

### PRODUCTOS

**Alta Producto:** Añade un producto al catálogo de productos disponibles.

**Baja Producto:** Elimina un producto del catálogo de productos disponibles.

**Actualizar Producto:** Actualiza la información asociada con un producto.

**Lista Productos:** Muestra los productos disponibles en la tienda.

**Buscar Producto:** Permite realizar búsquedas de productos.

### VENTAS

**Añadir Producto a Cesta:** Añade un producto a la cesta de compra.

**Eliminar Producto de Cesta:** Elimina un producto de la cesta de compra.

**Modificar Cantidad de Producto en Cesta:** Modifica la cantidad de un producto en la cesta de compra.

**Realizar Compra:** Permite comprar los productos de la cesta de compra.

**Ver Historial de Compras:** Muestra los pedidos previamente realizados por un cliente.

**Cancelar Pedido:** Permite a un cliente cancelar un pedido previamente realizado.

### PÁGINA WEB

**Inicio Sesión:** Permite al usuario entrar con su cuenta en el sistema.

**Cerrar Sesión:** Permite al usuario cerrar su sesión del sistema.

**Algoritmo de búsqueda de personas:** Se trata de un programa independiente considerado como una única función cuyo propósito es buscar constantemente a personas adecuadas para anunciarles la tienda online.

### CLIENTES

**Registro Clientes:** Añade un nuevo usuario al sistema como cliente.

**Baja Cliente:** Elimina un usuario del sistema.

### ADMINISTRADORES

**Listado Clientes:** Muestra una lista con los clientes.



**Ver Información Cliente:** Permite a los administradores ver la información de un cliente.

**Listado Pedidos:** Muestra una lista con los pedidos realizados.

**Ver Información Pedido:** Permite a los administradores ver la información de un pedido.

A continuación se detallarán todas las funciones.

## **PRODUCTOS**

### **Alta Producto**

- Prioridad: Media.
- Estabilidad: Alta.
- Descripción: Permitirá añadir un producto nuevo al catálogo.
- Entrada: La "ficha" del producto que se desee añadir.
- Salida: Confirmación de que se ha añadido correctamente.
- Origen: Vista con listado de productos en el portal de administradores.
- Destino: Vista con listado de productos en el portal de administradores.
- Acciones: El administrador desplazará el cursor sobre la "ficha" del producto que desea dar de alta, y pulsará sobre el icono de añadir.
- Precondición: Haber iniciado sesión como administrador del sistema.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Administradores.

### **Baja Producto**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permitirá eliminar un producto del catálogo cuando se agote el inventario, o se tenga que descatalogar el producto por completo.
- Entrada: La "ficha" del producto que se desea dar de baja.
- Salida: Eliminación de la información asociada al producto en la base de datos.
- Origen: Vista con listado de productos en el portal de administradores.
- Destino: Vista con listado de productos en el portal de administradores.
- Acciones: El administrador desplazará el cursor sobre la "ficha" del producto que desea dar de baja, y pulsará sobre el icono de eliminar.
- Precondición: Haber iniciado sesión como administrador del sistema.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Administradores.

### **Actualizar Producto**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: La función permitirá a un administrador cambiar la información asociada a un determinado producto.
- Entrada: La "ficha" del producto que se desea modificar.
- Salida: La nueva "ficha" modificada del producto. El producto es modificado en la base de datos.
- Origen: Vista con listado de productos en el portal de administradores.
- Destino: Vista con listado de productos en el portal de administradores.
- Acciones: El administrador seleccionará la "ficha" de un producto, lo cual le permitirá modificar su información.
- Precondición: Haber iniciado sesión como administrador del sistema.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Administradores.

### **Lista Productos**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: La función mostrará una lista con las "fichas" de los productos disponibles en la base de datos.
- Entrada: Ninguna.
- Salida: Muestra "fichas" de productos disponibles en la base de datos.
- Origen:
  - ◆ Cliente: La página de inicio de la aplicación.
  - ◆ Administrador: El portal de administradores.
- Destino:
  - ◆ Cliente: La página de inicio de la aplicación.
  - ◆ Administrador: El portal de administradores.
- Acciones:
  - ◆ Cliente: Conectarse a la página web.
  - ◆ Administrador: Seleccionar la opción de "listado de productos" en el portal de administradores.
- Precondición:
  - ◆ Cliente: Ninguna.
  - ◆ Administrador: Haber iniciado sesión.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Clientes y Administradores.

### **Buscar Producto**

- Prioridad: Alta.

- Estabilidad: Alta.
- Descripción:
  - ◆ Administradores: La función permitirá buscar un producto por su identificador o nombre.
  - ◆ Cliente: La función permitirá buscar un producto por su nombre.
- Entrada:
  - ◆ Administradores: El id o el nombre del producto.
  - ◆ Cliente: El nombre del producto.
- Salida: Muestra las "fichas" de los productos que encajen con la búsqueda.
- Origen:
  - ◆ Cliente: Página web.
  - ◆ Administrador: El portal de administradores.
- Destino:
  - ◆ Cliente: Página web con los resultados de la búsqueda.
  - ◆ Administrador: El portal de administradores con los resultados de la búsqueda.
- Acciones: Escribir en el campo de búsqueda y pulsar el botón de buscar (o intro)
- Precondición:
  - ◆ Cliente: Ninguna.
  - ◆ Administrador: Haber iniciado sesión como administrador.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Clientes y Administradores.

## VENTAS

### **Añadir Producto a Cesta**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permitirá a un cliente añadir un producto a su cesta de compra.
- Entrada: El producto que se desea añadir a la cesta.
- Salida: Se añade el producto a la cesta del cliente.
- Origen: Pantallas donde se muestran productos.
- Destino: La misma pantalla desde donde se añadió el producto.
- Acciones: Mover el cursor sobre la "ficha" de un producto, y seleccionar el icono de "añadir a cesta".
- Precondición: Iniciar sesión como cliente.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.

- Usuarios: Clientes.

### **Eliminar Producto de Cesta**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permitirá a un cliente eliminar un producto de su cesta de compra.
- Entrada: El producto que se desea eliminar de la cesta.
- Salida: Se elimina el producto de la cesta del cliente.
- Origen: Pantalla de la cesta.
- Destino: Pantalla de la cesta.
- Acciones: Mover el cursor sobre la "ficha" de un producto, y seleccionar el icono de "eliminar de la cesta".
- Precondición: Iniciar sesión como cliente.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Clientes.

### **Modificar Cantidad de Producto en Cesta**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permitirá a un cliente modificar la cantidad de un producto en su cesta.
- Entrada: El producto que se desea modificar en la cesta, junto con la nueva cantidad.
- Salida: Se modifica la cantidad del producto en la cesta del cliente.
- Origen: Pantalla de la cesta.
- Destino: Pantalla de la cesta.
- Acciones: Mover el cursor sobre la "ficha" de un producto, y seleccionar la nueva cantidad.
- Precondición: Iniciar sesión como cliente. Solamente mostrar cantidades del producto disponibles en el inventario.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Clientes.

### **Realizar Compra**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permitirá a un cliente realizar una compra con los productos de su cesta.
- Entrada: La cesta de la compra con todos los productos, y los datos del cliente.

- Salida: Se realiza la compra y se le indica al cliente un identificador para realizar la transferencia bancaria y finalizar el pago.
- Origen: Pantalla de la caja o de fin de compra.
- Destino: Pantalla de confirmación de la compra, o de error de compra.
- Acciones: Pulsar el botón de "finalizar compra".
- Precondición: Iniciar sesión como cliente y tener productos en la cesta.
- Postcondición: Ejecutar la función en menos de 10 segundos.
- Efectos laterales: Si ocurre un error en el servidor, se llevará al cliente a la pantalla de error de compra.
- Usuarios: Clientes.

### **Ver Historial de Compras**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permitirá a un cliente ver el historial con todas las compras realizadas.
- Entrada: Los datos del cliente.
- Salida: Lista con todas las compras e información de las mismas.
- Origen: Perfil del usuario.
- Destino: Pantalla del "Historial de Compras".
- Acciones: Pulsar el botón/enlace de "ver historial de compras".
- Precondición: Iniciar sesión como cliente.
- Postcondición: Ejecutar la función en menos de 1 segundos.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Clientes.

### **Cancelar Pedido**

- Prioridad: Media.
- Estabilidad: Alta.
- Descripción: Permitirá a un cliente cancelar uno de los pedidos todavía no entregados por el proveedor.
- Entrada: Los datos del cliente y del pedido.
- Salida: Se cancela el pedido indicado.
- Origen: Un pedido en concreto dentro del historial de compras.
- Destino: Pantalla del "Historial de Compras".
- Acciones: Pulsar el botón/icono de "cancelar pedido" sobre el pedido deseado.
- Precondición: Iniciar sesión como cliente. El pedido no ha sido marcado como "enviado" por parte del vendedor.
- Postcondición: Ejecutar la función en menos de 1 segundos.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Clientes.

# **PÁGINA WEB**

## **Inicio Sesión**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permite al usuario entrar en el sistema.
- Entrada: Usuario y contraseña.
- Salida: Inicio de sesión y mensaje informando si se ha iniciado sesión correctamente o ha ocurrido algún problema.
- Origen: Ventana inicial de la página web para iniciar sesión.
- Destino: En función del usuario:
  - ◆ Cliente: ventana principal.
  - ◆ Administrador: portal de administradores.
- Acciones: El usuario tendrá que introducir sus credenciales y pulsar iniciar sesión.
- Precondición: Estar en la página de inicio de sesión.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Clientes y administradores.

## **Cerrar Sesión**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permite al usuario salir del sistema.
- Entrada: Botón para cerrar la sesión actual.
- Salida: Cierre de la sesión y mensaje informando si se ha cerrado sesión correctamente o ha ocurrido algún problema.
- Origen: Ventana de usuario.
- Destino: Ventana principal.
- Acciones: El usuario tendrá que pulsar el botón de cierre de sesión y confirmar la acción.
- Precondición: Haber iniciado sesión en el sistema.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Clientes y administradores.

## **Algoritmo de búsqueda de personas**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Programa independiente de la página web considerado como una única función con el propósito de realizar "web scraping" de personas que encajen a perfiles adecuados para anunciarles la tienda online.

- Entrada: No aplica.
- Salida: Ninguna desde el punto de vista de un usuario. Anuncia la tienda online a determinadas personas que se determinen adecuadas. Guarda logs de información, y acciones realizadas.
- Origen: No aplica.
- Destino: No aplica.
- Acciones: No aplica.
- Precondición: No aplica.
- Postcondición: No aplica.
- Efectos laterales: Debido a la complejidad de buscar personas que encajen a perfiles concretos, será posible que el algoritmo anuncie a personas no adecuadas.
- Usuarios: No aplica.

## CLIENTES

### **Registro**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permitirá que un usuario cree una cuenta en el sistema.
- Entrada: Datos del usuario.
- Salida: Creación de la cuenta.
- Origen: Ventana inicial de la página web para iniciar sesión.
- Destino: Ventana principal.
- Acciones: El cliente pulsará en registrarse e introducirá sus datos.
- Precondición: Estar en la página de inicio de sesión.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Clientes.

### **Baja Cliente**

- Prioridad: Baja.
- Estabilidad: Alta.
- Descripción: Permitirá que un usuario borre sus datos del sistema.
- Entrada: Botón de borrar la cuenta.
- Salida: Eliminación de la información asociada al producto en la base de datos.
- Origen: Ventana de usuario.
- Destino: Ventana principal.
- Acciones: El usuario pulsará en el botón de darse de baja.
- Precondición: Haber iniciado sesión como cliente del sistema.
- Postcondición: Ejecutar la función en menos de 1 segundo.

- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Administradores.

## **ADMINISTRADORES**

### **Listado Clientes**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Muestra una lista con los clientes del sistema.
- Entrada: Ninguna.
- Salida: Lista de clientes.
- Origen: Portal de administradores.
- Destino: Portal de administradores.
- Acciones: El administrador selecciona la opción de listado de clientes.
- Precondición: Haber iniciado sesión como administrador del sistema.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Administradores.

### **Ver Información Cliente**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permitirá a los administradores ver la información de un cliente.
- Entrada: Cliente deseado.
- Salida: Información del cliente.
- Origen: Vista con listado de clientes en el portal de administradores.
- Destino: Ventana con la información del cliente.
- Acciones: El administrador elegirá de que cliente quiere ver la información.
- Precondición: Haber iniciado sesión como administrador del sistema.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Administradores.

### **Listado Pedidos**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Muestra la lista con los pedidos realizados.
- Entrada: Ninguna.
- Salida: Lista de pedidos.



- Origen: Portal de administradores.
- Destino: Portal de administradores.
- Acciones: El administrador selecciona la opción de listado de pedidos.
- Precondición: Haber iniciado sesión como administrador del sistema.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Administradores.

### **Ver Información Pedido**

- Prioridad: Alta.
- Estabilidad: Alta.
- Descripción: Permitirá a los administradores ver la información de un pedido.
- Entrada: Pedido deseado.
- Salida: Información del pedido.
- Origen: Vista con listado de pedidos en el portal de administradores.
- Destino: Ventana con la información del pedido.
- Acciones: El administrador elegirá el pedido del cual quiere ver la información.
- Precondición: Haber iniciado sesión como administrador del sistema.
- Postcondición: Ejecutar la función en menos de 1 segundo.
- Efectos laterales: Si ocurre un error en el servidor, se mostrará un mensaje indicándolo.
- Usuarios: Administradores.

## **HARDWARE NECESARIO**

A continuación listamos el Hardware necesario mínimo para el montaje del sistema:

### ***On Premise:***

Pequeño:

CPU	1 Core 2.0 GHz
RAM	512 MB
Disco Duro	10 GB

Mediano:

CPU	4 Cores 2.0 GHz
RAM	8 GB
Disco Duro	256 GB

Grande:

CPU	16 Cores 2.0 GHz
RAM	32 - 64 GB
Disco Duro	512 - 1024 GB

Los sistemas operativos en las instancias de servidores están basados en distribuciones Linux.

## **Cloud Services:**

### Google Cloud Platform:

Es una plataforma que ha reunido todas las aplicaciones de desarrollo web que Google estaba ofreciendo por separado. Es utilizada para crear ciertos tipos de soluciones a través de la tecnología almacenada en la nube y permite por ejemplo destacar la rapidez y la escalabilidad de su infraestructura en las aplicaciones del buscador.

Google Cloud se refiere al espacio virtual a través del cual se puede realizar una serie de tareas que antes requerían de hardware o software y que ahora utilizan la nube de Google como única forma de acceso, almacenamiento y gestión de datos.

- App Engine - Plataforma como servicio para colocar en producción aplicaciones Java, PHP, Node.js, Python, C#, .Net, Ruby y Go.
- Compute Engine - Infraestructura como servicio para correr Microsoft Windows y Linux virtual machines.
- Kubernetes Engine (GKE) or GKE on-prem ofrecida como parte de Anthos platform[6][7] - Containers como servicio basado en Kubernetes.

## Amazon web services:

Es una colección de servicios de computación en la nube pública (también llamados servicios web) que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon.com. Es una de las ofertas internacionales más importantes de la computación en la nube y compite directamente contra servicios como Microsoft Azure y Google Cloud Platform. Es considerado como un pionero en este campo.

- Cloud computing: todo lo necesario para la creación de instancias y el mantenimiento o el escalado de las mismas. Amazon EC2 es el rey indiscutible dentro de los servicios de computación en la nube de Amazon.
- Bases de datos: distintos tipos de bases de datos pueden permanecer en la nube mediante el servicio Amazon RDS, que incluye distintos tipos a elegir como MySQL, PostgreSQL, Oracle, SQL Server y Amazon Aurora, o Amazon DynamoDB para NoSQL.

## TECNOLOGÍA DE SOFTWARE

En este apartado describiremos las tecnologías de software más significativas para el desarrollo del proyecto:

### *Front-End:*

#### AngularJS:

Es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

#### React:

Es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre. Han participado en el proyecto más de mil desarrolladores diferentes.

#### Vue.js:

Es un framework de JavaScript de código abierto para la construcción de interfaces de usuario y aplicaciones de una sola página. Vue.js cuenta con una arquitectura de adaptación gradual que se centra en la representación declarativa y la composición de componentes. La biblioteca central se centra sólo en la capa de vista. Las características avanzadas necesarias para aplicaciones

complejas como el enrutamiento, la gestión de estados y las herramientas de construcción se ofrecen a través de librerías y paquetes de apoyo mantenidos oficialmente, con Nuxt.js como una de las soluciones más populares.

## **Back-End:**

### **Laravel:**

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5 y PHP 7. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti".

### **Django:**

Es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como MVC (Modelo-Vista-Controlador).

### **Flask:**

Es un framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD.

### **NodeJS:**

Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

## **Bases de Datos:**

### **MySQL:**

Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo.

### **PostgreSQL:**

También llamado Postgres, es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL, similar a la BSD o la MIT.

## MongoDB:

Es un sistema de base de datos NoSQL, orientado a documentos y de código abierto. En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

## Redis:

Es un motor de base de datos en memoria, basado en el almacenamiento en tablas de hashes (clave/valor) pero que opcionalmente puede ser usada como una base de datos durable o persistente.

## **Servidores Web:**

### Apache:

Es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual según la normativa RFC 2616. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido.

### Nginx:

Es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3).

Sugerencia de desarrollo inicial para el cliente:

- Front-End: Angular
- Back-End: Django
- Base de Datos: PostgreSQL / MongoDB
- Servidor Web: Nginx

En principio se comienza a desarrollar el proyecto con las herramientas de software sugeridas, pero pueden cambiar durante el desarrollo del proyecto. Cualquier cambio será debidamente informado y aprobado en conjunto con el cliente.