

# Documentación de Análisis y Diseño de Aplicaciones Web



**Solicitante:**

**I.T.S. – Instituto Tecnológico Superior Arias - Balparda**

**Nombre de Fantasía del Proyecto: Lonig System**

**Grupo de Clase: 3ºIE**

**Turno: Vespertino**

**Materia: Análisis y Diseño de Aplicaciones WEB**

**Nombre de los Integrantes del Grupo:**

**Bruno Acosta, Kevin Alvarez, Mauricio Brun**

**Fecha de entrega: 7/11/2020**

**Instituto Tecnológico Superior Arias Balparda**

**Gral. Flores 3591 esq. Bvar. José Batlle y Ordoñez - Montevideo**

## Objetivo

Este documento tiene como objetivo guardar la información y servir como guía para el cliente sobre el avance del proyecto con los siguientes requerimientos:

Proveer de una herramienta:

que permita presentar los productos que se tienen a la venta,  
realizar la compra/ventas y gestión las entregas de nuestras ventas.  
También realiza el registro de clientes y nuevos productos.  
Generar un remito de compra de mercadería para nuestros clientes  
que se enviará por mail luego de autorizar la venta.

## Alcance

La herramienta de gestión de venta de productos en el sector de seguridad ocupacional, en el rubro de la construcción, a desarrollarse; proveerá la función de administrar las diversas operaciones, de compra y venta, de los productos existentes para tal fin. Así como brindar un seguimiento del estado del envío de dichos productos al consumidor por ambas partes. Sin dejar de promocionar, y difundir, las ventajas y la necesidad del uso de los mismos en el desarrollo de la actividad anteriormente mencionada. Además, deberá administrar los datos involucrados (productos, clientes, proveedores. etc.).

# Índice

<b>1. Relevamiento de datos</b>	<b>4</b>
1.1. Entrevista	5
1.2. Observación	6
1.3. Cuestionario	6
<b>2. Modelo de formulario.</b>	<b>7</b>
<b>3. Documento ESRE</b>	<b>10</b>
3.1. Propósito	10
3.2. Alcance	10
3.3. Definiciones, acrónimos y abreviaciones	10
3.4. Requerimientos funcionales	10
3.5. Requerimientos no funcionales	12
3.6. Limitaciones	13
<b>4. Roles y perfiles de usuarios</b>	<b>14</b>
Product Owner:	16
Developer:	16
Tester:	16
Product Manager:	16
System Architect:	16
<b>5. Diagrama de casos de uso</b>	<b>17</b>
<b>6. Diagrama de estados</b>	<b>23</b>
<b>7. Diagrama de clases</b>	<b>25</b>
<b>8. Factibilidad</b>	<b>26</b>
<b>9. Calidad de Software de Páginas Web.</b>	<b>28</b>
<b>Bibliografía.</b>	<b>40</b>

# **1. Relevamiento de datos**

## **1.1. Entrevista**

La entrevista es genuinamente necesaria a la hora de recabar información más detallada y completa, así como también poder reconocer el modo de reaccionar de la persona y gracias a la amplia comunicación, qué es lo que opina y piensa, hecho que nos ayuda a reconocer con quién y qué trabajaremos, a su vez, su aplicación otorga al entrevistador la posibilidad de reconocer las necesidades del cliente e identificar verdaderamente lo que debemos producir para él.

Hemos decidido aplicarla debido a que necesitamos averiguar muchas cosas con respecto a la página web que vamos a realizar, tales sean las funcionalidades que desea que tenga, el tema del producto junto con el público al que apunta, aspectos importantes a la hora del desarrollo y más.

Gracias a la aplicación de este método de relevamiento podemos lograr recabar datos específicos y que correspondan directamente a nuestros intereses.

Más allá de la cantidad adaptable de información específica que logramos recopilar, también la versatilidad de adaptación de este método nos beneficia enormemente, pudiendo no sólo conseguir información directa, sino que también acceder en cierto sentido a información que el entrevistado no es capaz de expresar, pero que el entrevistador puede intuir o deducir, permitiendo ponerlo en duda durante el desarrollo de la entrevista y llegar a entendernos en un lenguaje en común.

## **1.2. Observación**

La observación nos brinda información sobre las características de los empleados y el cliente junto con las actividades que realizan, consiguiendo datos de tipo cualitativos y cuantitativos de primera mano sin la necesidad de interrumpir el entorno y modo de trabajo. Su aplicación en este contexto es capaz de otorgarnos datos sobre ciertos aspectos que los sujetos observados no pueden ni son capaces de demostrar en palabras. Podemos conseguir informarnos con respecto a la página de Facebook del cliente y el contenido de la que esta dispone actualmente, su información en cuanto a ubicación, público, mercado y proveedores.

## **1.3. Cuestionario**

El cuestionario es capaz de obtener información de una manera rápida y eficiente, contando con un costo bajo y capaz de filtrar también a los participantes de este para realizar una entrevista de manera posterior. Los participantes tienen la libertad necesaria en los temas y, al mismo tiempo, no genera información que carezca de uso.

Utilizamos el cuestionario porque este nos permite resolver las dudas que tenemos sobre muchos temas, ya sea el hardware que proporcionaremos, software incluyendo en esto las licencias pertinentes y recopilar datos estrechamente relacionados a los requerimientos funcionales y no funcionales. En cuanto a la recopilación de datos, la cantidad de preguntas específicas que podemos idear en este método de relevamiento logra facilitar la obtención de dicha información, generando a su vez que el planeamiento del trabajo y la identificación de los requerimientos funcionales y no funcionales se facilite exponencialmente.

Las distintas maneras que tenemos de plantear los distintos cuestionamientos en un cuestionario amplía la capacidad del encuestado de responder certeramente.

## **2. Modelo de formulario.**

### **Hardware:**

#### **Abiertas:**

- ¿Cuáles son los componentes que desearía tener en el servidor?
- ¿Cuáles son los componentes que desearía tener en las terminales?
- ¿Tiene algún tipo de preferencias en cuanto al que deberíamos usar?

#### **Cerradas:**

- ¿Desea que le proporcionemos hardware?
- ¿Cuánto dinero estaría dispuesto a gastar en equipamiento?
- ¿De qué empresas desearía que proviniera el hardware?

#### **Mixtas:**

- ¿Tiene equipamiento informático actualmente? En caso de tenerlo, ¿cuales son los componentes de los cuales dispone?
- ¿Cuántas terminales quisiera tener para sus empleados y por qué?
- ¿Quisiera algún tipo de garantía con respecto al hardware? ¿Cuál?

### **Software:**

#### **Abiertas:**

- ¿Qué es lo que le resulta más cómodo en un sistema operativo?
- ¿Qué tipo de aplicaciones y/o programas le gustaría tener en sus terminales?
- ¿Qué sistema operativo le parecería más útil para sus empleados?

#### **Cerradas:**

- ¿Qué antivirus propone para sus computadoras?
- ¿Qué tipo de licencia de sistema operativo le gustaría tener en su servidor?
- ¿Qué tipo de licencia de sistema operativo le gustaría tener en sus equipos?

## **Mixtas:**

- ¿Qué sistema operativo le parece más cómodo? Fundamente.
- ¿Qué sistemas operativos cree que son los más adecuados y por qué?
- ¿Quisiera contratar nuestro servicio técnico? ¿Para qué cosas?

## **Usuarios:**

### **Abiertas:**

- ¿Qué acciones adicionales quiere que pueda realizar un cliente?
- ¿Qué otros datos quisiera solicitar al usuario?
- ¿Qué tipos de datos les gustaría ver si fueran un cliente?

### **Cerradas:**

- ¿Cuántos empleados tendrán el rol “Comprador”?
- ¿Desea que sea obligatorio solicitar un método de pago al crear una cuenta?
- ¿Cuántos usuarios utilizando el cliente estima?

### **Mixtas:**

- ¿Puede un usuario tener múltiples perfiles? ¿Bajo qué circunstancias?
- ¿Quisiera un usuario como invitado? ¿A qué tendría acceso?
- ¿Quisiera que los clientes registrados sean obligatoriamente mayores de 18 años de edad? ¿Por qué?

## **Página web:**

### **Abiertas:**

- ¿Qué tipo de impresión quiere dar con la página?
- ¿Cómo quiere que se distribuyan los elementos de la página?
- ¿Qué clase de datos u opciones extra desea que muestre el pie de página?

### **Cerradas:**

- ¿Tiene planes sobre la modificación del logo de la empresa?
- ¿Desea que la página web tenga la barra de navegación fija en la parte superior de la pantalla?

- ¿Quiere que los productos sean mostrados por cantidad en stock?

### **Mixtas:**

- ¿Tiene algún tipo de paleta de colores que proponer? En caso de tenerla, ¿cuál sería?
- ¿Quisiera que la barra de navegación tenga alguna/s imagen/es extra además del isotipo de la empresa? ¿Cuál/es sería/n?
- ¿Tiene algún ejemplo que desee presentar para el diseño de la página? ¿Cuál?

### **Temática del programa:**

#### **Abiertas:**

- ¿Qué tipo de ofertas aplican?
- ¿Cómo guardan la información de las compras y las ventas?
- ¿En base a qué criterios deciden comprar a las distintas distribuidoras?

#### **Cerradas:**

- ¿Quisiera que la presentación del producto en la página pueda tener mucha o poca información?
- ¿Existe una cantidad de compras límite que un usuario puede hacer de un producto?
- ¿Se priorizará compras al por mayor para las empresas?

#### **Mixtas:**

- ¿Qué aspecto merece mayor prioridad de trabajo en cuanto a la interfaz de los empleados? Fundamente.
- ¿Cuándo se solicitaría el código postal al usuario? ¿Por qué?
- ¿Quisiera tener la posibilidad de otorgar más permisos a un usuario? Justifique.



## 3. Documento ESRE

### 3.1. Propósito

Este archivo se hizo con el motivo de definir las especificaciones funcionales y no funcionales para el desarrollo de un sistema de información web que permitirá la compra de indumentaria para la construcción.

### 3.2. Alcance

Crear una página de venta de indumentaria para la construcción, que permita la compra para distintos tipos de clientes, la adquisición de productos para su catálogo por parte de distintos distribuidores y la venta del equipamiento y su correspondiente entrega.

### 3.3. Definiciones, acrónimos y abreviaciones

Nombre	Descripción
usuario	Persona que usará el sistema para gestionar procesos.
RF	Requerimiento Funcional.
RNF	Requerimiento no Funcional.

### 3.4. Requerimientos funcionales

RF1	Login	Para la utilización de las funciones que contiene la página web, se requerirá un inicio de sesión.
-----	-------	--

		En este se pedirá el nombre de usuario y contraseña, los cuales se comparan con los datos guardados en la base de datos.
RF2	Alta de usuarios	Existen distintos tipos de usuarios, los cuales son los siguientes: Jefe, Vendedor, Comprador, Cliente. El usuario Jefe será capaz de realizar todas las funciones de los demás usuarios.
RF 3	Registrar Cliente	Las personas serán capaces de registrarse, creando así un usuario Cliente que tendrá que ser verificado por un Jefe o un Vendedor posteriormente. Se les pedirá lo siguiente: nombre completo, contraseña (que se deberá repetir para verificarla), mail, teléfono, fecha de nacimiento.
RF4	Alta Usuario	El Jefe tendrá la opción de agregar cualquier tipo de usuario. Se le pedirá nombre completo, contraseña (que se deberá repetir para verificarla), correo, fecha de nacimiento, dirección, tipo de usuario (Vendedor, Comprador o Cliente) y año de ingreso (este último solamente para el personal).
RF5	Verificar Cliente	El usuario Jefe y Vendedor tendrán que verificar a los clientes luego de registrarse para que estos puedan realizar acciones.
RF6	Carrito de compras	Los clientes serán capaces de guardar de forma temporal, aquellos productos que les interese comprar en el carrito de compras.  En este serán capaces de aumentar y disminuir la cantidad de productos de cierto tipo que desee, o eliminarlos del carrito. Se podrá ver el precio total.

RF7	Crear pedido	<p>Los clientes serán capaces de crear pedidos al realizar una compra.</p> <p>Se le pedirá elegir un método de pago, El pedido tendrá distintos estados, siendo estos “cancelado”, “pago pendiente”, “pago realizado”, “en transporte” y “entregado”.</p>
RF8	Enviar remito	<p>Se les enviará un remito por mail luego de que un pedido se encuentre en estado “entregado”.</p> <p>El remito contendrá los siguientes datos: dirección, nombre del cliente, fecha de entrega, producto entregado, cantidad de los productos, teléfono</p>
RF9	Gestionar pedidos	<p>El Jefe y el Vendedor serán capaces de gestionar los pedidos.</p> <p>Serán capaces de cambiar el estado de los pedidos.</p>
RF10	Paquetes	<p>Existirán conjuntos de productos similares denominados paquetes, que serán más baratos que el comprar los productos por separado.</p>

### 3.5. Requerimientos no funcionales

RNF 1	Apariencia	Utilizar colores corporativos.
RNF 2	Usabilidad y Humanidad	Información clara, concisa y visible, para mejorar la calidad del usuario.
RNF 3	Rendimiento	Página programada eficientemente para reducir tiempos de carga.

RNF 1	Apariencia	Utilizar colores corporativos.
RNF 4	Operacionales	Mostrar mensajes breves que sirvan de guía para la utilización del producto.
RNF 5	Mantenimiento y soporte	Realizar un backup de manera diaria.
RNF 6	Seguridad	Elaboración de restricciones detalladas para cada usuario con el objetivo de asignar los permisos adecuados.
RNF 7	Legales	Actualizar al cliente para que cumpla los requerimientos legales en medida del cambio.
RNF8	Culturales	Adecuarse al idioma con jerga fuertemente instaurada al nivel social en el contexto del cliente.

### 3.6. Limitaciones

El siguiente sistema se desarrollará con la aplicación de los siguientes ambientes de trabajo, los cuales condicionarán el desarrollo y calidad del producto en base a las peticiones de la letra.

Las limitaciones son las siguientes:

- Centos 7 para los servidores.
- MariaDB ver. 5.5.68 como motor de base de datos.
- PHP ver. 7.4.22.
- Workbench 6.3.4

## 4. Roles y perfiles de usuarios

### Product Owner:

- Manejar los KPIs.
- Elaborar reportes.
- Contabilizar los resultados y la responsabilidad sobre el resultado final.
- Determinar los requisitos generales y actividades iniciales del proyecto.
- Representar a los usuarios del producto.
- Buscar y asegurar los recursos financieros que requiere el proyecto para iniciarse y desarrollarse.
- Analizar la viabilidad del emprendimiento.
- Garantizar que el producto se entregue.
- Desarrollar y establecer los criterios para aceptar las historias de los usuarios.
- Aprobar o negar los productos entregables.

### Product Manager:

- Encargarse del Product Backlog, qué contiene y el desarrollo del producto.
- Gestión del presupuesto.
- Medición de resultados del producto.
- Es el encargado de gestionar el negocio de un producto.
- Define la visión y estrategia del producto.
- Prioriza las características y capacidades del producto.
- Crea el plan de programación.
- Identificar qué problemas y necesidades tiene el consumidor para saber qué productos crear.
- Garantizar que los productos se creen y entreguen correctamente teniendo en cuenta la propuesta de valor.
- Analizar cómo responden los usuarios a dichos productos y obtener insights para mejorarlos constantemente.

## Developers:

- Crear y desarrollar nuevos programas o sistemas.
- Investigar las necesidades de los usuarios.
- Probar los nuevos programas.
- Evaluar sistemas nuevos y existentes.
- Mejorar programas existentes.
- Realizar el mantenimiento correspondiente en los sistemas existentes, realizando el monitoreo y corrección de los defectos detectados, así como elaborar el código en lenguajes especializados (HTML, PHP, XML) para nuevos programas.
- Elaborar manuales operativos y especificaciones técnicas de los sistemas.
- Trabajar en conjunto con el resto del equipo, es decir, con Gerentes de Proyecto, Diseñadores Gráficos, otros Desarrolladores, Administradores de Bases de Datos y el personal de Ventas y Marketing.
- Elaborar informes sobre el progreso del proyecto.

## Tester:

- Detectar errores en el sistema
- Prueba de rendimiento en diferentes dispositivos.
- Identificar riesgos de errores de software.
- Ejecutar Test de usabilidad.
- Ejecutar Test de navegabilidad.
- Generar y ejecutar el proceso de datos de prueba
- Gestionar los ambientes de prueba.
- Enfoque de prevención de los defectos del software en lugar de enfoque de solo encontrar los defectos.
- Planifica y gestiona su propio grupo de trabajo
- Responde rápidamente a los cambios, incluyendo cambiar, agregar o mejorar los casos de prueba.
- 

## System Architect:

- Gestiona la parte técnica y arquitectónica de las Features.
- Participa en la definición de requerimientos funcionales, no funcionales y features.

- Facilita y ayuda a los equipos para que trabajen cómodos con la tecnología elegida
- Adelanta y deja todo listo para cuando comiencen a trabajar.
- Resuelve problemas que puedan surgir.
- Define y comunica una visión técnica y arquitectónica compartida
- Describe el contexto de la solución y la intención de la solución
- Analizan las ventajas y desventajas técnicas
- Determinan los componentes principales y los subsistemas
- Identifican las interfaces y las colaboraciones entre ellos.

### **Product Owner:**

Formación Empresarial, Gestión de Proyecto, ADA.

### **Developer:**

Gestión de Proyecto, ADA, Diseño Web, Programación Web, Base de Datos, S.O.

### **Tester:**

Programación, Diseño, S.O y Base de datos.

### **Product Manager:**

Formación Empresarial, Gestión de Proyecto, ADA.

### **System Architect:**

ADA, Gestión de Proyecto.

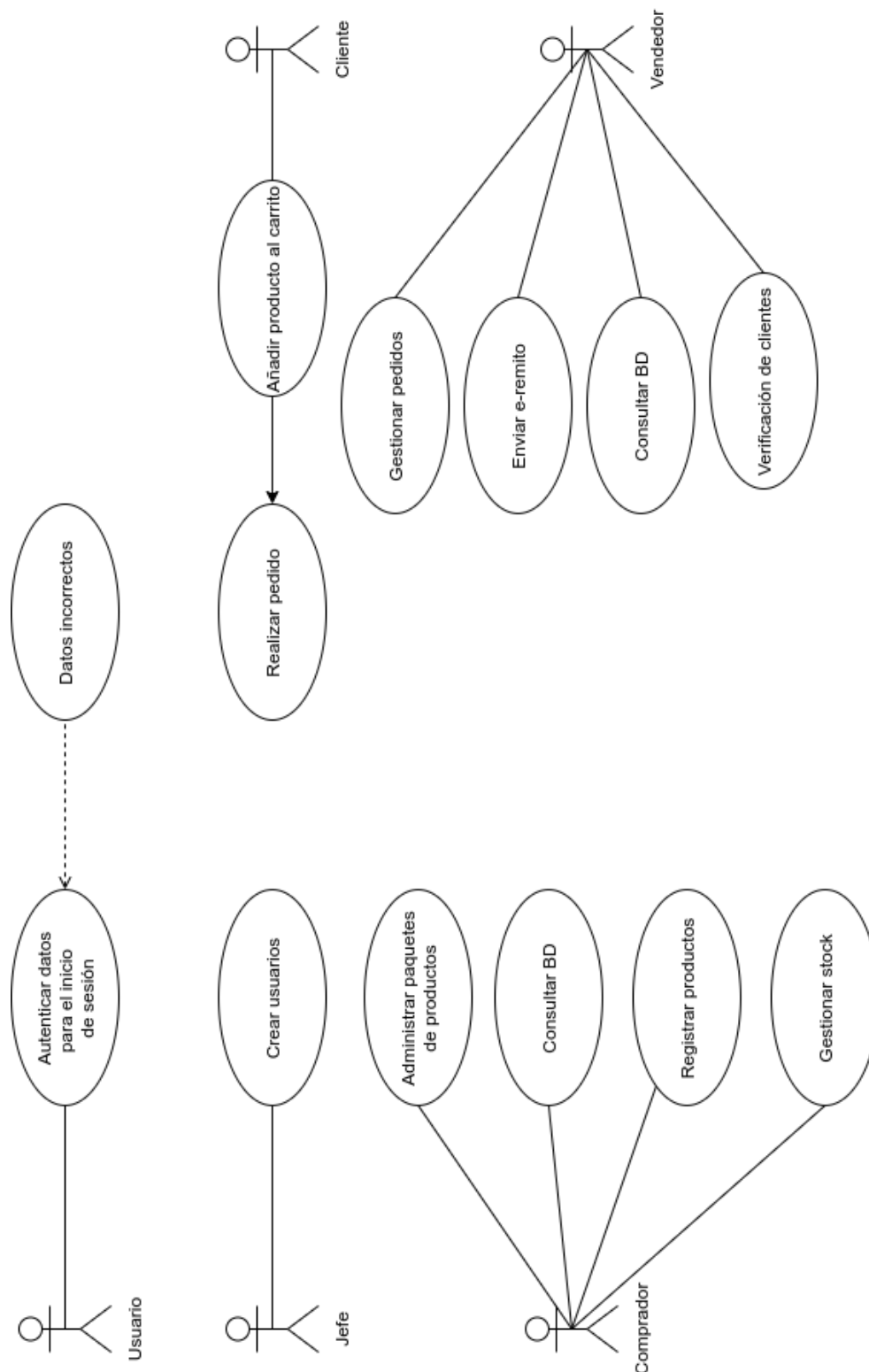
Kevin Alvarez: Tester y Developer.

Santiago Alvarez: Developer y Product Manager.

Mauricio Brun: Developer y System Architect.

Bruno Acosta: Product Owner.

## 5. Diagrama de casos de uso





Nombre:	El visitante se registra como usuario cliente.
Descripción:	Permite a un visitante el crearse un usuario tipo cliente
Actores:	Visitante
Precondiciones:	
No encontrarse con la sesion de una cuenta iniciada.	
Flujo normal:	
<ol style="list-style-type: none"> <li>1. El visitante ingresa a la pagina de registro para crear un usuario de tipo cliente.</li> <li>2. Luego de ingresar los datos, se envían a la base de datos.</li> <li>3. El usuario Jefe y/o Vendedor verificarán al cliente para que pueda tener su usuario.</li> </ol>	
Flujo alternativo:	
<ol style="list-style-type: none"> <li>1. Si el visitante no ingresa los datos necesarios en cualquiera de los campos, le saltará un mensaje de alerta avisando que el campo es obligatorio, en caso de no llenarlo, no podrá proceder con la creación de su usuario.</li> <li>2. Si el correo electrónico ingresado ya está en uso, le aparecerá un mensaje de alerta notificándole dicho hecho y deberá utilizar uno distinto.</li> </ol>	
Poscondiciones:	
El visitante habrá creado su usuario de tipo cliente e ingresará a él automaticamente.	

Nombre:	El visitante inicia sesión como cliente
Descripción:	Permite a un visitante ingresar como un usuario cliente.
Actores:	Visitante
Precondiciones:	Tener una cuenta registrada con anterioridad.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El visitante ingresa el correo y contraseña de su cuenta de usuario.</li> <li>2. Se valida la veracidad de los datos ingresados por el usuario en la base de datos.</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>1. Si el visitante no ingresa nada en el campo del correo electrónico, se le mostrará un mensaje emergente que dirá: "Ingrese el correo electrónico."</li> <li>1. Si el visitante no ingresa nada en el campo de la contraseña, se le mostrará un mensaje emergente que dirá: "Ingrese la contraseña."</li> <li>2. Si el correo ingresado no existe en la base de datos, se le mostrará un mensaje emergente que dirá: "No existe una cuenta con el correo electrónico ingresado."</li> <li>2. Si la contraseña no coincide con la que tiene asignada el usuario que posee dicho correo electrónico, se le mostrará un mensaje emergente que dirá: "Contraseña incorrecta."</li> </ol>
Poscondiciones:	El visitante habrá iniciado sesión y se le enviara a la página de inicio automáticamente.

Nombre:	Alta de usuarios por Jefe.
Descripción:	El permiso del Jefe de agregar usuarios de tipo Vendedor y Comprador (incluso cliente).
Actores:	Jefe
Precondiciones:	Estar ingresado como un usuario de tipo Jefe.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El Jefe ingresa el nombre del usuario, su contraseña (que deberá repetirla para verificarla), el correo electrónico, la fecha de nacimiento, dirección, el tipo de usuario y el año de ingreso (sólo para los usuarios distintos de Cliente).</li> <li>2. Se verificará la consistencia de los datos y se creará el usuario en cuestión.</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>1. Si alguno de los campos está vacío, se le mostrará un mensaje emergente que dirá: "Ingrese el dato".</li> <li>1. Si el campo de "confirmar contraseña" no coincide con el de "contraseña", se le mostrará un mensaje emergente que dirá: "Los campos no coinciden".</li> <li>2. Si el correo ingresado ya existe en la base de datos, se le mostrará un mensaje emergente que dirá: "El correo electrónico ya se encuentra en uso".</li> </ol>
Poscondiciones:	Se habrá creado el usuario.

Nombre:	El cliente agrega productos al carrito
Descripción:	Los clientes podrán seleccionar productos para agregarlos al carrito.
Actores:	Cliente
Precondiciones:	Tener una sesión iniciada como un usuario de tipo cliente.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El cliente selecciona el marco de un producto para visualizar su información y luego presiona el botón "agregar al carrito".</li> <li>2. El producto se encontrará en el carrito de forma temporal, el cliente podrá aumentar la cantidad de este que desee hasta que decida eliminarlo, comprarlo o cerrar la sesión de la cuenta.</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>1. Si el usuario desea agregar al carrito un producto para el cual no hay stock, no podrá hacerlo pues la función se encontrará inhabilitada (se encontrará en gris y no podrá apretar el botón), el cartel tendrá un cartel que avisará que no hay stock.</li> </ol>
Poscondiciones:	El usuario tendrá una lista temporal "carrito" con productos que él seleccionó para agregar.

Nombre:	Gestionar los pedidos de los clientes
Descripción:	El Jefe y Vendedor podrán gestionar los pedidos realizados por los clientes.
Actores:	Jefe y Vendedor
Precondiciones:	Tener una sesión de Jefe o Vendedor iniciada.
Flujo normal:	<ol style="list-style-type: none"> <li>1. El usuario Jefe o Vendedor entra la sección de pedidos realizados por los clientes.</li> <li>2. En la sección de pedidos serán capaces de ver los pedidos actualmente en proceso y los ya finalizados e interactuar con ellos.</li> <li>2. En los pedidos actuales podrá observar sus cuatro tipos de estados: pendiente, realizado, en transporte, entregado y cancelado.</li> <li>3. Para aquellos productos que se encuentren en estado "realizado" deberán presionar el botón "empaquetar", lo cual enviará la tarea de que se comience a empaquetar el producto y se envíe hacia el cliente.</li> <li>4. En cualquiera de los estados que se encuentre el pedido estará disponible la opción "cancelar", la cual lo deniega para poder hacer una devolución.</li> </ol>
Flujo alternativo:	<ol style="list-style-type: none"> <li>1. Si el pedido se encuentra en cualquiera de los otros estados distintos de "realizado", la opción "empaquetar" no estará disponible.</li> <li>2. Si el pedido se encuentra en estado cancelado se pasará de la lista "en proceso" a "finalizado".</li> </ol>
Poscondiciones:	Se habrá empaquetado y enviado el pedido al cliente correspondiente.

## 6. Diagrama de estados

Diagrama de estados del pedido.

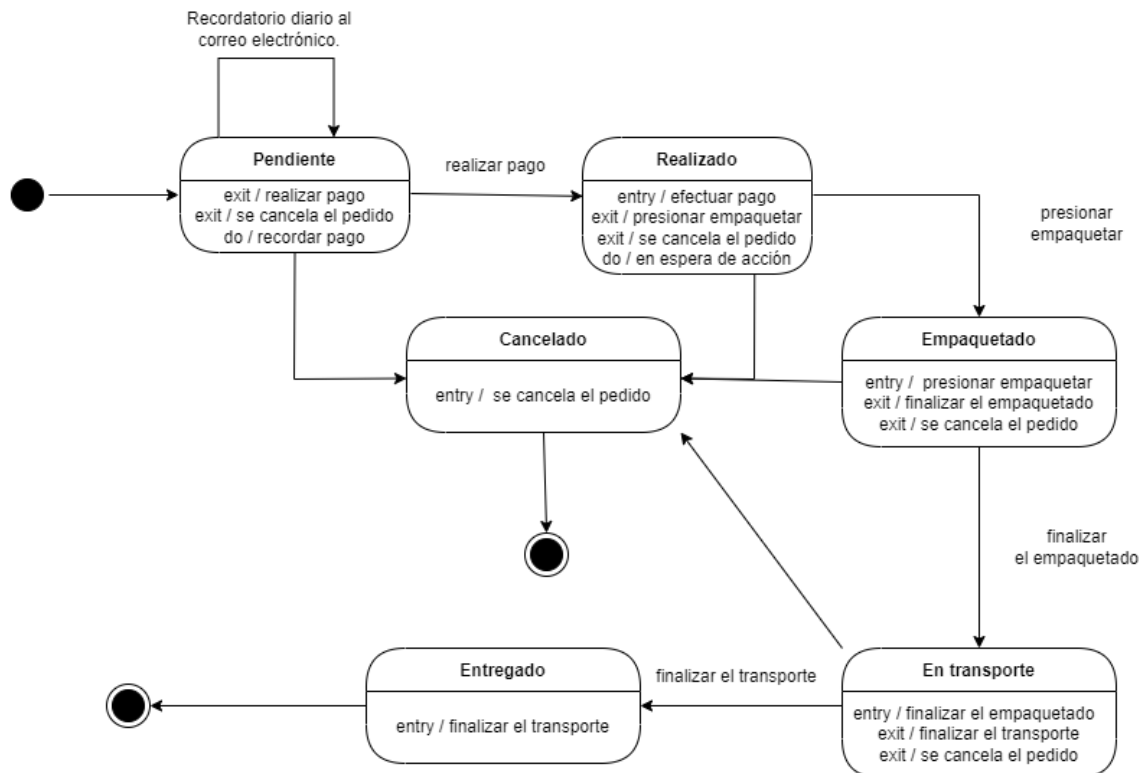


Diagrama de estados del registro.

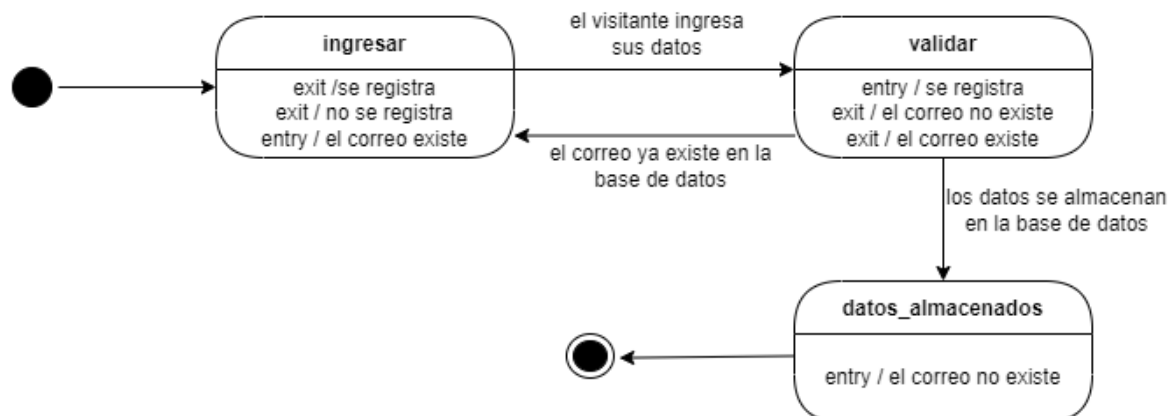


Diagrama de estados del remito.

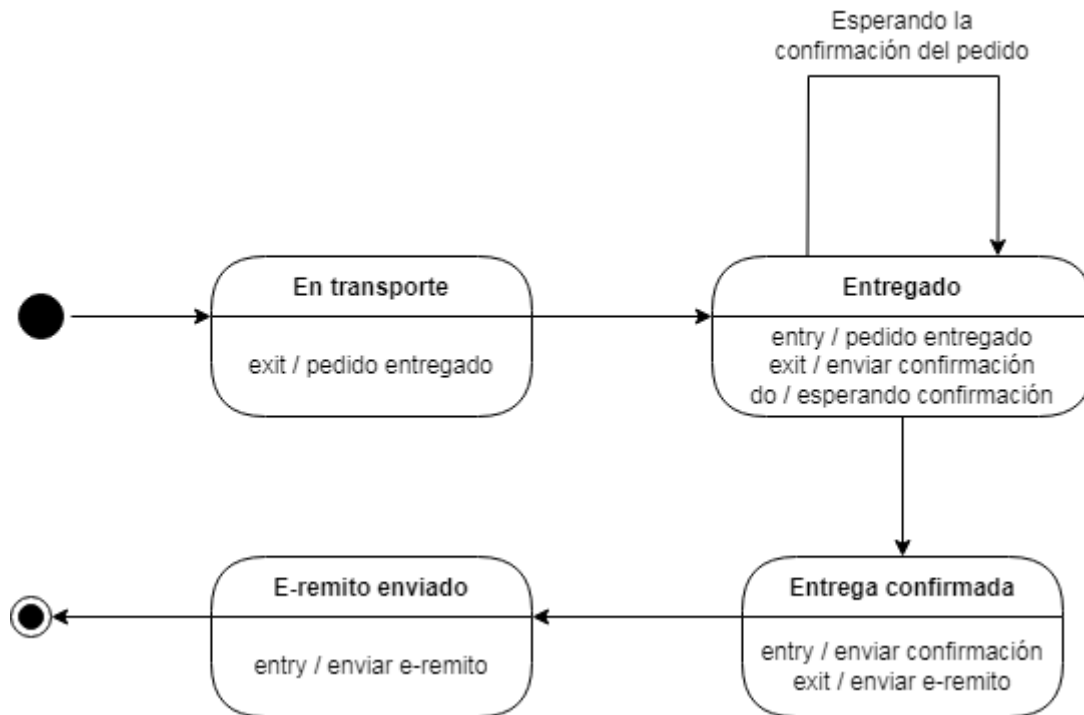
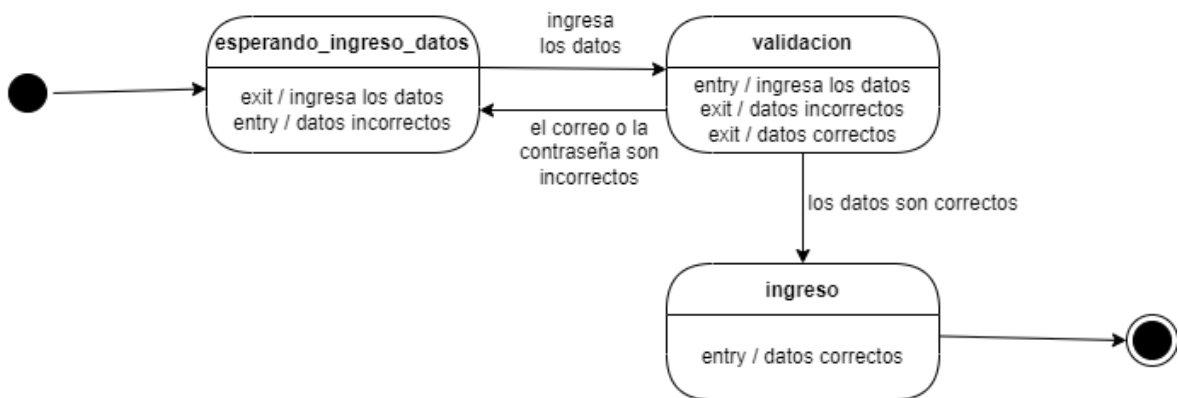


Diagrama de estados del inicio de sesión.



## 7. Diagrama de clases



## **8. Factibilidad**

### **8.1 Legal:**

Ley 9.739 - Propiedad literaria y artística

Ley N° 18331 - Ley de protección de datos personales.

Ley 16.074 - Seguro de accidentes de trabajo y enfermedades profesionales.

Estándar IEEE 830-1998.

### **8.2 Técnica:**

Nuestro proyecto es viable debido a que tenemos los recursos necesarios para poder llevarlo a cabo óptimamente. Contamos con profesionales capacitados en la gestión de proyectos web tanto como en el desarrollo del producto y la documentación correspondiente.

Equipos: Computadora de Oficina Intel p445207579

Software: Windows 10 Pro

### **8.3 Operativa**

Se le creará un manual de usuario con las especificaciones sobre el funcionamiento de todos los tipos de usuario, también se les dará capacitación al Jefe durante dos días, dos horas por día y la página web recibirá soporte y mantenimiento durante 30 días después de la compra.

## **8.4 Contrato de prestaciones de servicio.**

### **Acerca de las partes:**

La empresa Login System, representada por Mauricio Brun, mayor de edad, soltero, que vive en Felipe Sanguinetti 2739, identificado por la cédula número 5.434.099-9, cumple con las funciones de Project Manager de la empresa a quien en lo sucesivo se le denominará simplemente diseñador.

El Sr(a) Natalia Viera que en lo sucesivo se le denominará Seguridad Corporal.

## 9. Calidad de Software de Páginas Web.

### ¿Qué es el software?

Se define al software como un conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

Hay distintos tipos de software, entre ellos se encuentra el software de aplicación, de programación, de sistema, software malicioso y más tipos para los cuales no estaremos verdaderamente profundizando en esta presentación.

### Aplicaciones Web:

En la ingeniería de software se denomina aplicación web (al igual se le denomina como "Software Web") a aquella herramienta que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador. Es decir, es un programa que se codifica en un lenguaje interpretable por los navegadores web en la que se confía la ejecución al navegador.

En otras palabras, es un programa que se codifica en un lenguaje interpretable por los navegadores web en la que se confía la ejecución al navegador.

Cuando hablamos de software web nos referimos a aplicaciones que se instalan en servidores (propios o de alguna empresa que provea servicios de hosting), ordenadores dedicados a proveer servicios, realizar tareas, ante peticiones de otros ordenadores. Estas aplicaciones web o softwares web son desarrollados por programadores que utilizan entornos de programación Frontend y Backend.

En este caso, los datos y la lógica de negocio quedan centralizados en el propio servidor y los usuarios podrán acceder a ellos por medio de sus propios navegadores web.

Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios potenciales. Existen aplicaciones como los correos web, wikis, blogs, tiendas en línea y la propia Wikipedia que son ejemplos bastante conocidos de aplicaciones web.



Una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar formularios, participar en juegos diversos y acceder a gestores de base de datos de todo tipo.

## **Objetivo de Calidad de Software:**

Tiene como objetivo producir software de la mejor calidad posible, que cumpla, y si puede supere las expectativas de los usuarios. Esencialmente te permite saber cómo está construido su software por dentro, analizar el código, analizar sus componentes y cómo este interactúa con otros.

## **Calidad del producto:**

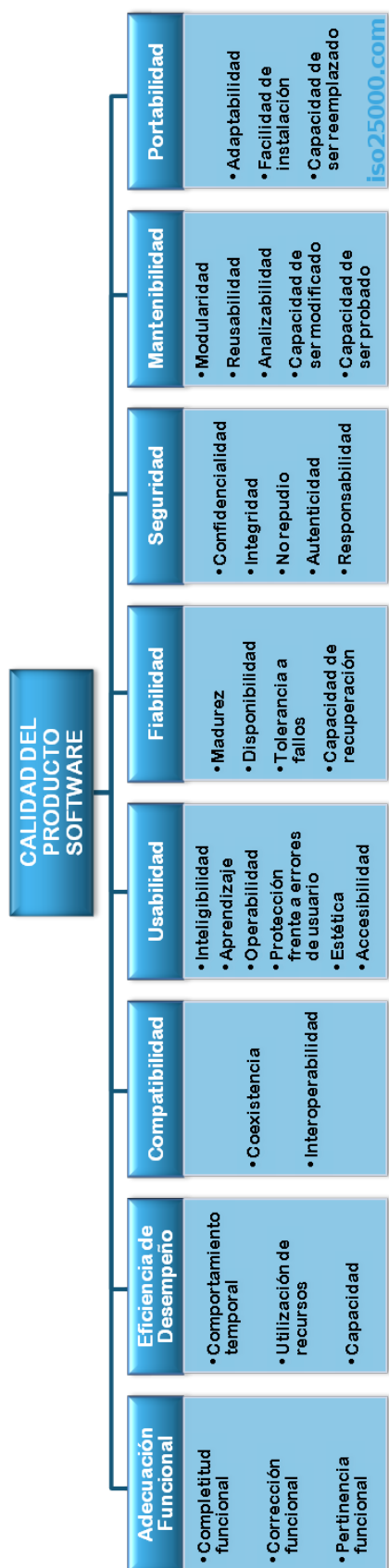
Para tener una calidad de software, se deberá chequear los requerimientos funcionales y no funcionales a la hora de empezar un proyecto y también ver los atributos de calidad.

## **Atributos de calidad:**

Todo lo que el cliente o persona que use el resultado de dicho proyecto le resulte cómodo, atractivo y pueda percibir lo que realmente busca en él. Estos se cumplen realizando y cumpliendo ciertos estándares de presentación y sus características.

## **Ejemplos de Atributos de calidad:**

Un claro ejemplo, es la **ISO 25010**, este es un conjunto de normas y estándares para que tu producto mejore en calidad según las necesidades del cliente, aquí podrás ver diferentes estándares y características y trabajar en ellas según lo que busque el cliente.





## Calidad del proceso:

Son los trabajos realizados para que el producto sea de calidad.

Aquí varios ejemplos y herramientas para la calidad del proceso.

**CMMI:** El CMMI o también llamado Capability Maturity Model Integration y es un modelo que ayuda a mejorar la calidad de proceso.

**MoProSoft:** Es un modelo de procesos que ayuda a desarrollar y hacer mantenimiento al software.

**ISO 15504:** Es una norma de desarrollo de software también conocido como Software Process Improvement Capability Determination, abreviado SPICE y es utilizado para dar normas y características para mejoras de calidad de proceso.

## Criterios:

### Seguridad

La Seguridad representa la capacidad de que el sistema no afecte su entorno y el de quien lo utiliza. Existen muchos sistemas de software en Internet que trabajan con usuarios mediante el acceso personalizado, por ejemplo las aplicaciones bancarias. Esto conlleva la introducción de una serie de medidas de seguridad implementadas dentro de las aplicaciones y de los servicios. En caso de no aplicar medidas de seguridad adecuadas, no sólo existe el riesgo de perjudicar la imagen de las organizaciones, si no también está latente el riesgo de afectar la economía de los usuarios del sistema de software. En la actualidad existen compañías que tienen grandes pérdidas económicas, debido principalmente a la falta o a la implementación inadecuada del aspecto de calidad seguridad.

### Disponibilidad

Define la probabilidad de que el sistema esté funcionando por un tiempo determinado. Sobre un sitio Web, los clientes no únicamente esperan disponibilidad las 24 horas del día, los siete días a la semana durante todo el año. Sólo una pérdida de tiempo de 10 minutos puede ocasionar grandes pérdidas, por



ejemplo en las aplicaciones de la bolsa de valores. En otros casos las pérdidas pueden ser de ventas de diferentes artículos, como libros (es el caso de Amazon), revistas, etc. La disponibilidad también implica que todos los usuarios puedan acceder a la información, no importando el navegador y la plataforma que utilicen. En el caso Shockwave Flash el cual es únicamente compatible con Microsoft Internet

Explorer y Netscape sobre Windows, los usuarios que trabajan sobre Unix no podrán acceder a la información que es manejada por este software. En este caso, también está latente la poca disponibilidad del sistema. Es claro que el hecho de hacer disponible un sitio desde todos los puntos de vista, implica adoptar lenguajes más robustos y estables, aunque la facilidad para desarrollar estas aplicaciones se vea reducida.

## **Escalabilidad**

Las necesidades para la escalabilidad han derivado una gama muy amplia con la que se cuenta en la actualidad de tecnología de innovación en cuanto a software y hardware. La industria ha desarrollado nuevos lenguajes de software, estrategias de

diseño, tecnologías de comunicación de datos y transferencia de protocolos. La necesidad surge cuando el número de clientes ó usuarios que utilizan los servicios de una aplicación en Web crece. Con el objetivo de evitar el riesgo de insuficiencia de recursos en el sistema, normalmente se programan crecimientos en la infraestructura.

Cuando un sistema de esta naturaleza esta preparado para crecer, lo hace sin complicaciones.

Este aspecto debe ser contemplado desde la planeación, análisis y el diseño, dentro del proceso de desarrollo de software.

## **Mantenibilidad**

Un aspecto novedoso que deriva del software basado en la Internet, son las nuevas versiones o actualizaciones. El software tradicional involucra mercadeo, precios, ventas y personal de instalación para los clientes. Sin embargo, este proceso es caro y el desarrollo de este software usualmente requiere modificaciones de mantenimiento, con sus estimados de tiempo. Es el caso, por ejemplo, de las múltiples versiones que manejan, las aplicaciones de Microsoft. El mantenimiento que se requiere para aplicaciones basadas en el web muchas veces debe ser

inmediato. El acceso de los clientes es algo que implica una actualización constante en cuanto a información y el servicio a los módulos. Los cambios que se realizan en mucho de los casos deben ser inmediatos y transparentes al usuario. En ocasiones extremas las aplicaciones deben ser actualizadas sin dejar de proporcionar el servicio. Por citar algunos ejemplos tenemos las aplicaciones que manejan ventas por Internet, las aplicaciones que manejan acciones de alguna casa de bolsa, ó las aplicaciones bancarias donde se realizan transacciones en línea.

## **Características de calidad**

ISO 9126-1:2001 plantea las siguientes características de calidad:

- 1) Funcionalidad.
- 2) Confiabilidad.
- 3) Facilidad de Uso.
- 4) Eficiencia.
- 5) Facilidad de Mantenimiento.
- 6) Portabilidad.

La Funcionalidad, es el conjunto de atributos que se refieren a la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones cumplen unos requerimientos o satisfacen unas necesidades implícitas. Las subcaracterísticas de la Funcionalidad son: Aptitud, Precisión, Interoperatividad, Conformidad, Seguridad y Trazabilidad.

La Confiabilidad, es el conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de rendimiento bajo unas condiciones especificadas durante un período definido. Las subcaracterísticas de la Confiabilidad son: Madurez, Tolerancia a fallas, Facilidad de Recuperación, Disponibilidad y Degradabilidad.

La Facilidad de Uso, es el conjunto de atributos que se refieren al esfuerzo necesario para usarlo, y sobre la valoración individual de tal uso, por un conjunto de usuarios definidos e implícitos. Las subcaracterísticas de la Facilidad de Uso son:

Comprensibilidad, Facilidad de aprendizaje, Operatividad, Explicitud, Adaptabilidad al usuario, Atractivo, Claridad, Facilidad de ayudas y Amigable al usuario.

La Eficiencia, es el conjunto de atributos que se refieren a las relaciones entre el nivel de rendimiento del software y la cantidad de recursos utilizados bajo unas condiciones predefinidas. Las subcaracterísticas de la Eficiencia son: Respecto al tiempo y Respecto a los recursos.

La Facilidad de Mantenimiento, es el conjunto de atributos que se refieren al esfuerzo necesario para hacer modificaciones especificadas. Las subcaracterísticas de la Facilidad de Mantenimiento son: Facilidad de análisis, Facilidad de cambio, Estabilidad y Facilidad de prueba.

La Portabilidad, es el conjunto de atributos que se refieren a la habilidad del software para ser transferido desde un entorno a otro. Las subcaracterísticas de la Portabilidad son: Adaptabilidad, Facilidad de instalación, Conformidad y Facilidad de Reemplazo.

La valoración de estas características es útil para que el usuario pueda definir los requerimientos del producto utilizando solamente las características que emplee en la práctica. Para algunos tipos de productos de software, hay determinadas características que no son significativas y las restantes no garantizan que con ellas comprendan todos los requerimientos de los productos de software, por lo que en cada caso habrá que completarlas con otras definiciones más específicas para esos productos o situaciones. No obstante el modelo tiene el nivel de abstracción suficiente como para que sea adaptable en la mayoría de las situaciones, siendo además, independiente de la tecnología.

Otra herramienta para la evaluación es ISO 25000:2005 (SQuaRE -Software Quality Requirements and Evaluation) que es una nueva serie de normas que se basa en ISO 9126 y en ISO 14598 (Evaluación del software). Uno de los principales objetivos de la serie SQuaRE es la coordinación y armonización del contenido de ISO 9126 y de ISO 15939:2002 (Measurement Information Model). ISO 15939 tiene un modelo de información que ayuda a determinar que se debe especificar durante la planificación, performance y evaluación de la medición. Para su aplicación, cuentan con los siguientes pasos: (1) Recopilar los datos, (2) Preparación de los datos y (3) Análisis de los datos.

La integración de ISO 9126 e ISO 15939 permiten plantear un proceso de 4 pasos:

1. Identificación de los requerimientos relacionados a la calidad del producto, es decir, seleccionar la parte del modelo de calidad que resulta relevante para la evaluación de calidad.
2. Identificación del contexto de interpretación. Es decir, selección de los valores de referencia y determinación de los objetivos especificados en un contexto determinado.
3. Uso de las medidas derivadas de la etapa de preparación de los datos.
4. Comparación y análisis de los resultados obtenidos respecto de un conjunto de valores de referencia.

SQuaRE incluye un estándar de requerimientos de calidad. Está compuesto por documentos agrupados en 5 tópicos:

1. Administración de la Calidad.
2. Modelo de Calidad.
3. Medidas de Calidad.
4. Requerimientos de Calidad.
5. Evaluación de la Calidad.

**Administración de la Calidad:** Provee una guía para planificar y administrar las evaluaciones del software.

**Modelo de Calidad:** Describe el modelo de calidad interno / externo y la calidad en uso. Presenta características y subcaracterísticas.

**Medidas de Calidad:** Medición de primitivas, Medidas para la calidad interna, Medidas para la calidad externa y Medidas para la calidad en uso.

**Requerimientos de Calidad:** Permite habilitar la calidad del software a ser especificado en términos de requerimientos de calidad durante todo el ciclo de vida de un proyecto de software o adquisición, mantenimiento y operación.

**Evaluación de la Calidad:** Evaluación de la Calidad, Proceso para desarrolladores, Proceso para compradores, Proceso para evaluadores y Documentación del módulo de evaluación.

Los pasos para la realización de una Evaluación de la Calidad del Software son:

1. Identificación del Producto de software que se pretende evaluar
2. Determinar los Requisitos aplicables
3. Relevar la información necesaria para el cálculo de las métricas de los requisitos aplicables
4. Evaluar la Calidad del Software usando las métricas respecto de los requisitos establecidos en el paso 2 y determinar su cumplimiento
5. Establecer las acciones correctivas respecto de los requisitos evaluados
6. Elaborar un Informe Final.

El Informe de toda esta evaluación, es el documento que refleja los objetivos, alcances, observaciones, recomendaciones y conclusiones del proceso de evaluación, relacionados con las áreas de informática. El Informe debe incluir suficiente información para que sea comprendido por los destinatarios esperados y facilitar las acciones correctivas.

Por último, se puede decir que la Evaluación de la Calidad del Software puede traer aparejado la certificación del software, lo cual permite mejorar el nivel de competitividad de la empresa con sus respectivas consecuencias

## **Administración de Calidad Total**

El término Administración de Calidad Total fue creado en 1985 por el Comando de Sistemas Aéreos Navales de la USAF (Fuerza Aérea de Estados Unidos), para describir el estilo que utiliza la administración japonesa para mejorar la calidad de sus productos de software. Actualmente el término ha tomado numerosos sentidos, de acuerdo a quién lo interpreta. En términos generales éste representa un estilo de dirección exitoso en cuanto a la implementación de la calidad en los productos de software y en la satisfacción del cliente. La base para este tópico es la creación de una cultura de calidad que integra a todos los miembros de la organización.

Desde 1980 un gran número de compañías de los Estados Unidos adoptaron el ACT.

La adopción de ISO 9000 por la Comunidad Europea como estándar administrativo y la aceptación de estos estándares por la iniciativa privada, han fomentado que cada vez más organizaciones implementen marcos de trabajo con base en el ACT.

Compañías en el marco industrial de la computación y la electrónica como Hewlett Packard, IBM, Motorola y Siemens (entre otras), han implementado satisfactoriamente ACT. Un claro ejemplo es IBM que con su división de AS/400, ven recompensado su esfuerzo en sus productos de software con reconocimientos como el Malcom Baldrige National Quality Award.

Los elementos principales del sistema ACT (mostrados en la figura 2.1) se concentran en los siguientes aspectos:

**Enfoque del cliente:** Si tenemos como objetivo principal la satisfacción total del cliente, nos enfrentamos a un problema que involucra varios aspectos técnicos y sociales. En este punto de acuerdo a la ACT se debe incluir estudios acerca de lo que quiere y necesita el cliente; de acuerdo a la especificación de sus requerimientos, medir y evaluar la satisfacción de ellos.

**Proceso:** El objetivo principal es tomar como base las experiencias para enriquecer los procesos de mejora. Con este enfoque es posible obtener una línea de producción cada vez más estable en el proceso de negocio y en el proceso de desarrollo. Los procesos de mejora son muy importantes ya que a través de ellos se podrán obtener productos de software de mayor calidad.

**Lado humano de la calidad:** Tiene como objetivo crear una compañía con una cultura de calidad. Aquí intervienen factores psicológicos y sociales que motiven y convenzan al personal de las ventajas que trae laborar con la calidad.

**Medidas y análisis:** Lo que busca evaluar y analizar, es cuantificar los resultados que se van generando en cada fase del proceso de desarrollo. Con ello se permite establecer predicciones más claras y objetivas. Otro aspecto que se beneficia, es el control de la calidad del proceso y del producto de software

## **10. OWASP.**

### **¿Qué es OWASP?**

OWASP (Open Web Application Security Project), es un proyecto sin ánimo de lucro a nivel mundial que busca mejorar la seguridad del software en general. Para esto, la organización se ha provisto de una serie de herramientas y documentos que explican cuáles son las brechas de seguridad más comunes en cualquier sistema de información. Sobra decir, que todos los materiales de OWASP están disponibles de manera libre (gratuita) para su libre consulta y uso.

### **¿Por qué es importante conocerlo?**

OWASP es una organización de talla mundial, no encontrarás un compendio de vulnerabilidades más grande que las que menciona OWASP tan bien documentadas. Muchas empresas dedicadas al desarrollo de software están concientes de esto, por lo cuál, no es nada raro que las buenas ofertas para nosotros los desarrolladores están cargadas de un componente que incluya conocimientos en OWASP.

### **¿Cuáles son exactamente sus contenidos?**

Actualmente OWASP tiene en realidad varios proyectos en los que resaltan las categorías Tool Projects, Code Projects y Documentation Projects. El proyecto de documentación más conocido es el TOP TEN, en el cuál se listan las 10 vulnerabilidades (security risks) más habituales y cómo prevenirlas. En este top, reconocerás términos como SQL INJECTION, Cross-Site Scripting (XSS) y Broken Authentication. Sin más, te dejo con esta lista de diez riesgos de seguridad o vulnerabilidades.

---

## **OWASP TOP 10**

El más famoso de los proyectos de esta metodología es conocido con el nombre OWASP TOP 10, que no es más que un listado de los problemas de seguridad más comunes en las aplicaciones web y ordenados de más a menos críticos.

Aquí una imagen completa de dicho top en el cual se enumeran aquellos problemas anteriormente mencionados, junto con su nombre

## **Inyección SQL**

Esto ocurre cuando la aplicación envía datos no confiables a un intérprete. Estas son fáciles de descubrir cuando se examina el código, pero más difíciles mediante test de intrusiones.

Las aplicaciones más susceptibles de sufrir inyecciones son las diseñadas de forma pobre y que carecen de validación de entradas. Pero, no es solo código lo que puede inyectarse en una aplicación web. También pueden detectarse llamadas al sistema que usen órdenes de comandos para ejecutar procesos e inyectar peticiones de ejecución de otros procesos.

## **Secuencia de comandos en sitios cruzados (XSS)**

Esto se produce cuando la aplicación incluye datos suministrados por el usuario en una página enviada al navegador sin validar apropiadamente el contenido.

Hay tres tipos:

Almacenados: el código se almacena en una base de datos y se produce cada vez que se recupera la información.

Reflejados: el código se refleja desde el servidor como mensaje de error o resultado de una petición, pero no se almacena.

Basados en DOM: el ataque se ejecuta como resultado de la modificación del DOM del OPentorno del navegador de la víctima. El código se ejecuta de forma inesperada.

## **Pérdida de autenticación y gestión de sesiones**

La autenticación y establecimiento de sesión permite que un usuario se identifique como usuario de la aplicación y sea reconocido como tal por los mecanismos de control de acceso. A menudo estos esquemas de autenticación y gestión de las sesiones contienen vulnerabilidades en las secciones de:



- Cierre de sesión
- Gestión de contraseñas
- Tiempo de desconexión
- Función de recordar contraseña
- Pregunta secreta
- Actualización de la cuenta

Puede ser difícil encontrar estas vulnerabilidades. Por ejemplo, si un sistema permite cambiar la clave con una función de recuperar contraseña, un usuario no legítimo podría usar esa función para hacerse pasar por usuario legítimo de la aplicación.

Es necesario proveer un par de atributos (usuario y contraseña) que identifiquen únicamente al usuario. Mediante entradas no validadas es posible conseguir el acceso sin conocer ninguno de esos dos atributos para la autenticación.

Si las credenciales de autenticación, identificación y sesión no se protegen mediante técnicas de cifrado, un usuario malintencionado podría suplantar la identidad del usuario legítimo.

## **Referencia directa insegura a objetos**

Las aplicaciones comprueban siempre que el usuario disponga de permisos sobre el objetivo. Un análisis de código mostraría si la autorización se verifica correctamente.

Para prevenir las referencias inseguras a objetos directos debe seleccionarse una forma de proteger el acceso:

- Usar referencias indirectas por usuario o sesión: esto impediría el acceso a recursos no autorizados directamente por los atacantes.
- Comprobar el acceso: debe asegurarse que el usuario está autorizado para acceder al objeto solicitado.

## **Falsificación de peticiones en sitios cruzados**

Cuando los navegadores envían credenciales de autenticación automáticamente, como en el caso de las cookies de sesión, los atacantes pueden crear páginas web que generan peticiones falsas indistinguibles de las auténticas.

La forma más sencilla de revisar la vulnerabilidad en una aplicación es verificar si cada enlace y formulario contiene un testigo (token) no predecible para cada usuario. Si no lo tiene, los atacantes pueden falsificar peticiones.

El análisis debe realizarse en enlaces y formularios que invoquen funciones que permitan cambiar estados. Estos son los principales objetivos de los ataques.

## **Configuración de seguridad defectuosa**

Una aplicación puede ser segura pero pueden encontrarse problemas de seguridad en ella debido a una configuración deficiente del servidor. Debemos cuestionar los siguientes aspectos de seguridad en el servidor:

- Errores de seguridad no parcheados en el software del servidor.
- Errores de seguridad en el software del servidor o malas configuraciones que permiten ataques de listado de directorio.
- Existencia de archivos por defecto, de respaldo o de ejemplo.
- Permisos inadecuados en archivos y directorios.
- Servicios innecesarios habilitados, como manejo de contenido y administración remota.
- Cuentas por defecto con contraseñas por defecto.
- Funciones administrativas o de depuración habilitadas o accesibles.
- Mensajes de error demasiado informativos.
- Certificados SSL y opciones de encriptación mal configurados.
- Uso de certificados auto firmados para la autenticación.
- Uso de certificados por defecto.
- Autenticación inadecuada con sistemas externos.

## **Almacenamiento criptográfico inseguro**

En muchas aplicaciones es necesario almacenar información con seguridad. Para ello, se recurre a bases de datos u otros soportes usando en muchos casos técnicas de cifrado. Pero se cometen errores al aplicar esas técnicas de cifrado.

Las áreas donde los errores son más comunes son:

- Al cifrar la información esencial
- Almacenamiento inseguro de claves, certificados y contraseñas
- Almacenamiento incorrecto de secretos en memoria
- Fuentes pobres de aleatorización
- Elección pobre de algoritmo
- Inventar el nuevo algoritmo de encriptación
- Fallo al incluir soporte para cambios en las claves de encriptación y otros procedimientos de mantenimiento requeridos.

- La encriptación se utiliza para proteger los activos más sensibles del sitio, que pueden ser comprometidos por una debilidad.

Para protegernos de ese problema, debemos:

Identificar las amenazas existentes y asegurar que los datos están cifrados de forma que se defiendan de las amenazas.

Asegurarnos de que las copias de seguridad almacenadas externamente están cifradas, con las claves gestionadas y almacenadas de forma separada.

Asegurarnos del uso adecuado de algoritmos y claves fuertes.

Garantizar que las contraseñas se almacenan en forma de hash con un algoritmo estándar robusto.

Garantizar que todas las claves y contraseñas son protegidas contra accesos no autorizados.

## **Fallo de restricción de acceso a URL**

Control de acceso es el mecanismo que, una vez autenticado el usuario, decide si puede acceder o no a un conjunto de información o si puede realizar determinadas tareas.

Los desarrolladores tienden a menospreciar la dificultad en la creación de los controles de acceso. Incluso, en muchas ocasiones, se crean como un aspecto evolutivo de la propia aplicación, sin un plan de implantación fuerte.

Una vez obtenido un error en la validación de permisos, se pueden realizar operaciones para las que no se ha dado permiso: borrar o añadir información e incluso tomar el control administrativo de la aplicación.

Para evitar esto se recomienda:

- Basar la autenticación y la autorización en roles para minimizar el esfuerzo necesario para mantener esas políticas.
- Las políticas deben ser configurables.
- Debe negarse por defecto todo acceso, y requerir el establecimiento explícito de permisos a usuarios y roles por cada página.
- Si la página forma parte de un proceso de varios pasos, debe verificarse que las condiciones de la misma se encuentren en el estado apropiado para permitir el acceso.

## **Protección insuficiente en la capa de transporte**

Un problema muy común en los sitios web es que usan HTTPS para la autenticación y luego continúan el resto de navegación mediante HTTP.

En otros casos, aunque la web realiza toda la comunicación vía HTTPS, responde igualmente si las peticiones se le realizan mediante HTTP. Esto permite ataques Man in The Middle.

La mejor forma de averiguar si una aplicación se encuentra protegida es comprobar si:

- Se utiliza SSL para proteger la aplicación.
- Se sigue usando SSL para todos los recursos de páginas y servicios privados.
- Solo se aportan algoritmos considerados fuertes.
- Todas las cookies de sesión tienen el atributo secure activado.
- El certificado del servidor es legítimo y está correctamente configurado para ese servidor.

Para ello debe:

- Emitirse por una entidad autorizada
- No haber expirado
- No haber sido revocado
- Se ajusta a todos los dominios utilizados por la aplicación.
- El servidor no admite conexiones sin SSL para los recursos privados.
- El servidor tiene implantada la política HSTS que obliga a los navegadores a comunicarse con el servidor exclusivamente mediante HTTPS.

## **Redirecciones y reenvíos no validados**

Las aplicaciones redirigen a los usuarios a otras páginas. Detectar redirecciones sin validar es fácil. Es necesario buscar redirecciones donde el usuario puede establecer la dirección URL completa.

Existen diversas maneras para efectuar un uso seguro de redirecciones y reenvíos:

- Evitando el uso de redirecciones y reenvíos.
- Si usamos, no involucrar parámetros manipulables por el usuario.
- Si los parámetros de destino no pueden evitarse, comprueba el valor facilitado y que el usuario esté autorizado.
- Se recomienda que el valor de cualquier parámetro de destino sea un valor de mapeo, en lugar de la dirección real y traducir en el servidor el destino real.

## 11. Evaluar costo página web:

Las páginas/aplicaciones de tiendas online web suelen tener un precio aproximado a mil dólares. En cuanto a la suma de las horas de trabajo que requirieron los miembros del grupo de proyecto, se estiman alrededor de 200 horas de trabajo en total, lo cual puede aumentar el precio del producto final en al menos 900 dólares.

El hosting y dominio suele costar alrededor de 80 dólares y quiénes se deben encargar de esto es la empresa Seguridad Corporal, por ende no se añadirá al precio final de la aplicación.

El soporte y mantenimiento de la página web costará 5 dólares la hora. Estas opciones estarán disponibles por determinada cantidad de tiempo, dependiendo del paquete que elijan.

Paquete básico:

- Aplicación web de Tienda Online funcional.
- Ayuda personalizada para la redacción de textos.
- SEO on page básico.
- Video tutoriales para aprender a gestionar tu web.

Precio: 2400 dólares.

Paquete intermedio:

- 3 meses de soporte
- SEO on page avanzado
- Asesoramiento SEM, explicación sobre los precios, beneficios, desventajas en su adquisición.

\*Incluye todo lo del paquete anterior\*

Precio: 3000 dólares.

Paquete avanzado:

- Diseño, redacción e implementación de secuencias automáticas de correos para e-mail marketing

- 1 año de soporte y mantenimiento post-entrega (disponibles, más no añadidos al precio. Cada hora se cobra por fuera del precio de la venta).
- 4 sesiones anuales de soporte, seguimiento de analíticas y coordinación de acciones.

\*Incluye todo lo del paquete anterior\*

Precio: 4000 dólares.

## Bibliografía.

<https://its.utu.edu.uy/sites/its.utu.edu.uy/files/2022-05/Proyecto%203%C2%BA%20EMT%20%C3%89nfasis%20WEB.pdf>  
<https://its.utu.edu.uy/sites/its.utu.edu.uy/files/2022-05/Proyecto%203%C2%BA%20EMT%20%C3%89nfasis%20WEB.pdf>  
<https://hireline.io/mx/enciclopedia-de-perfiles-ti/perfil-de-tester-software-tester>  
<https://www.euroinnova.uy/blog/actividades-de-un-tester>  
<https://trans-ti.com/2020/11/09/que-es-un-tester-de-software-y-cuales-son-sus-tareas-principales/>  
<https://www.productmanagers.es/que-es-un-product-manager/>  
<https://www.crehana.com/blog/marketing-digital/que-hace-product-manager/>  
<https://blogs.imf-formacion.com/blog/marketing/funciones-tareas-product-manager/>  
<https://rockcontent.com/es/blog/product-owner/>  
<https://www.eude.es/blog/responsabilidades-product-owner/>  
<https://beagilemyfriend.com/product-owner/>

<https://www.universia.net/ar/actualidad/orientacion-academica/ocupaciones-tecnologicas-que-hace-developer-software-1148915.html>

[https://docs.informatica.com/es\\_es/data-quality-and-governance/data-quality/10-2/\\_profiling-getting-started-guide\\_data-quality\\_10-2\\_ditamap/introduccion/interfaz\\_de\\_usuario\\_de\\_informatica\\_developer/tareas\\_de\\_informatica\\_developer.html](https://docs.informatica.com/es_es/data-quality-and-governance/data-quality/10-2/_profiling-getting-started-guide_data-quality_10-2_ditamap/introduccion/interfaz_de_usuario_de_informatica_developer/tareas_de_informatica_developer.html)

<https://www.freelancermap.com/blog/es/que-hace-arquitecto-software/>

<https://www.michaelpage.es/advice/profesi%C3%B3n/tecnolog%C3%ADa/perfil-de-arquitecto-de-software#:~:text=El%20puesto%20de%20Arquitecto%20de,desarrollo%20e%20implementaci%C3%B3n%20del%20software.>

<https://blog.pleets.org/article/conoce-owasp>

<https://liliseguridadinformatica.webnode.es/guia-de-buenas-practicas/disenio/a10/>

[https://owasp.org/www-pdf-archive/OWASP\\_Top\\_10\\_2007\\_Spanish.pdf](https://owasp.org/www-pdf-archive/OWASP_Top_10_2007_Spanish.pdf)

[https://es.wikipedia.org/wiki/OWASP\\_Top\\_10](https://es.wikipedia.org/wiki/OWASP_Top_10)

<https://www.arsys.es/blog/owasp>

## HOJA TESTIGO

**MATERIA: Análisis y Diseño de Aplicaciones WEB**

**Nombre del Profesor/a: Ana Gonzalez**

**Nota:**

[illegible]