



## Página Web “Rey Toro Delivery”

Asignatura: Taller de Desarrollo de Aplicaciones

Sección: B50-C3-D

Nombre del Docente: Patricio Antonio Araya Castro

Integrantes:     Ferdinando Stagnaro

                  Tomas Villagrán

                  David Seguel

                  Francisco Ruiz

Fecha de Entrega: 02-05-2024

## Contenido

1. Épicas.....	3
2. Historias de usuario y criterios de aceptación .....	3
3. Orden de Servicio.....	7
4. Product Backlog.....	12
5. Preparación del entorno .....	13

## 1. Épicas

- Navegación y Descubrimiento de Contenido: Funcionalidades relacionadas con información del servicio, información de contacto, navegación del contenido (interna y externa).
- Simulación de compra: Funcionalidades relacionadas a agregar y/o quitar productos de una simulación, al cálculo de una simulación de productos y al despliegue de información detallada de los productos.
- Formulario de contacto: Funcionalidades relacionadas a captación de información del potencial pedido del usuario e información de este, almacenándolas de forma local, entregando una notificación al usuario y enviando un correo electrónico al administrador.
- Gestión de Contenido: Funcionalidades relacionadas a agregar, leer, modificar y eliminar un producto mediante una interfaz.

## 2. Historias de usuario y criterios de aceptación

a) Exploración Informativa en la Landing Page:

- Como cliente interesado, quiero poder explorar la landing page para obtener información sobre los productos y servicios ofrecidos.

**Criterios de aceptación:**

➤ Visualización de información destacada:

- Dado que soy un cliente interesado. Cuando accedo a la landing page. Entonces veo información destacada sobre los productos más populares, promociones actuales o cualquier otro contenido relevante para captar mi interés.
- Dado que estoy explorando la landing page. Cuando desplazo hacia abajo. Entonces encuentro secciones informativas sobre la empresa, su historia, valores y cualquier otra información relevante que pueda influir en mi decisión de compra.

b) Exploración de Catálogo de Productos:

- Como cliente, quiero poder explorar el catálogo en línea para ver los productos disponibles.

**Criterios de aceptación:**

➤ Navegación intuitiva del catálogo:

- Como un cliente interesado, al acceder al catálogo en línea desde la landing page, encuentro una interfaz clara y fácil de usar que me permite navegar por las diferentes categorías y subcategorías de productos disponibles.
- Como un cliente interesado, al explorar el catálogo y hacer clic en una categoría específica, entonces se muestra información detallada del producto, con imágenes y descripciones breves.

c) Comunicación con el Vendedor:

- Como cliente interesado en un producto, quiero poder comunicarme con el vendedor para obtener más información y hacer consultas.

**Criterios de aceptación:**

- Formulario de contacto accesible:
  - Dado que soy un cliente interesado en un producto específico. Cuando accedo a la página de detalles del producto. Entonces encuentro un formulario de contacto claramente visible que puedo completar para enviar consultas al vendedor.
  - Dado que completo el formulario de contacto con mis consultas. Cuando lo envío. Entonces recibo una confirmación de que mi mensaje ha sido enviado, ya sea correctamente o no al vendedor.

2.) Simulación de compra:

a) Agregar productos a la simulación

- Como usuario quiero generar una simulación de compra para estimar el costo total de mi pedido.

**Criterios de aceptación:**

- Debe ser una página web (interfaz de usuario) clara que me facilite agregar y quitar productos a la simulación.
- Los productos deben ser seleccionados desde el mismo sitio en donde se visualizan.
- Se debe mostrar el costo total de los productos seleccionados de manera instantánea, en el detalle de la simulación.

b) Despliegue de información detallada del producto

- Como usuario quiero poder ver información detallada de un producto antes o durante la simulación de compra.

**Criterios de aceptación:**

- Debe haber un botón o enlace claramente visible que me permita acceder a la información detallada de un producto.
- La información detallada debe incluir características generales, precio, disponibilidad y cualquier otra información relevante.
- La información detallada debe ser desplegada de manera clara y legible en la página web.

3.) Formulario de contacto:

- Como comprador, quiero poder utilizar un formulario de contacto en el sitio web para solicitar más información sobre el servicio y sus condiciones, para así poder tomar una decisión de compra.

**Criterios de aceptación:**

- Debo encontrar fácilmente el formulario de contacto en la página.
- El formulario debe solicitar información del comprador / cotizante, tal como su: nombre, dirección de correo electrónico y mensaje detallando mi consulta sobre el servicio.
- Después de enviar el formulario, el usuario recibirá una alerta de que ha sido enviado un formulario.
- Como administrador del sitio web, necesito recibir notificaciones por correo electrónico cada vez que un usuario envíe un formulario de contacto, para poder responder a sus consultas o pedidos de manera oportuna.

**Criterios de aceptación:**

- Debo recibir un correo electrónico con los detalles del formulario enviado por el usuario, incluyendo su nombre, correo electrónico, mensaje y cualquier otra información relevante.
- Las notificaciones deben enviarse automáticamente al administrador tan pronto como se envíe el formulario.
- El contenido del correo electrónico debe ser claro y conciso.

4.) Gestión de Contenido:

- Como administrador, quiero poder agregar un producto al catálogo para poder aumentar la variedad de productos que ofrezco.

**Criterios de aceptación:**

- Debo poder acceder al sistema de administrador mediante un usuario y contraseña único.
- El sistema debe utilizar métodos de encriptación para proteger la información de inicio de sesión del administrador.
- Se deben implementar medidas de protección contra ataques de fuerza bruta y otras formas de intrusión.
- El acceso al sistema de administrador debe ser restringido y solo estar disponible para usuarios autorizados.
- Se deben mantener registros de actividad de inicio de sesión para monitorear el acceso al sistema y detectar posibles intentos de acceso no autorizados.
- Debe existir una interfaz clara que me solicite una imagen del producto, precio y comentarios del producto.

- Existirá una previsualización del producto a agregar en el catálogo.
- Si el producto ha sido agregado exitosamente al catálogo, debe existir una notificación al administrador.
  - Como administrador, quiero poder visualizar los productos del catálogo en una interfaz simple para tener información detallada de los elementos en el catálogo.

**Criterios de aceptación:**

- Debo poder acceder al sistema de administrador mediante un usuario y contraseña único.
- Debe existir una interfaz simplificada que muestre los productos con sus precios y sus respectivos detalles.
  - Como administrador, quiero poder modificar un producto del catálogo para poder realizar cambios según estime conveniente.

**Criterios de aceptación:**

- Debo poder acceder al sistema de administrador mediante un usuario y contraseña único.
- Debe existir una interfaz clara, que permita seleccionar un producto y el atributo a modificar.
- Existirá una previsualización del producto a modificar en el catálogo.
- Si el producto ha sido modificado exitosamente, debe existir una notificación al administrador.
- Al completar la modificación, se deben respaldar con fecha y hora los cambios realizados.
  - Como administrador, quiero poder eliminar un producto del catálogo para poder ajustarlo según la disponibilidad de Stock.

**Criterios de aceptación:**

- Debo poder acceder al sistema de administrador mediante un usuario y contraseña único.
- Debe existir una interfaz clara que me solicite el producto a eliminar.
- Existirá una confirmación para eliminar el producto.
- Si el producto ha sido eliminado exitosamente del catálogo, debe existir una notificación al administrador.

## 3. Orden de Servicio

### Datos del Cliente

<b>Nombre</b>	Rey Toro Delivery S.A.C - Rey Toro Delivery
<b>Rut</b>	76244001-6
<b>Teléfono</b>	+56 9 9241 7052
<b>Email</b>	restobar@reytoro.cl

### Datos de los Consultores técnicos (Coloque aquí la identificación de los integrantes del grupo)

<b>Nombre</b>	Francisco A. Ruiz Diaz
<b>Rut</b>	19640542-9
<b>Nombre</b>	Tomás Elías Villagrán Sánchez
<b>Rut</b>	18.661.972-2
<b>Nombre</b>	Ferdinando Stagnaro Espinoza
<b>Rut</b>	18.638.400-8
<b>Nombre</b>	David Seguel Blanco
<b>Rut</b>	20746235-7

### Observaciones generales

*Explique brevemente los objetivos y alcance del proyecto*

#### Los objetivos:

Objetivo principal: Desarrollar un conjunto de páginas web relacionadas que permitan al usuario y potencial cliente de Rey Toro Delivery, el informarse, navegar, cotizar mediante un simulador de pedidos y poder contactar al administrador mediante un formulario. También se generará una vista o interfaz de cara al administrador para que pueda generar un CRUD en el catálogo de productos.

Objetivos específicos:

1. Crear una Landing Page que muestre las condiciones generales del servicio.
2. Crear un catálogo web con un simulador de pedidos que permita al usuario informarse sobre productos, precios y simular una potencial compra.
3. Crear una página web con un formulario que permita al usuario ingresar sus datos personales y de contacto y que este al ser enviado genere un correo electrónico automáticamente en la bandeja de correos del administrador.
4. Generar una página web por el lado del administrador, que permita hacer un CRUD completo en la página del catálogo web. Junto con ello, diseñar una base de datos SQL para poder trabajar con el CRUD.

**El Alcance del proyecto**

El alcance del proyecto para desarrollar lo anterior debería contemplar los siguientes elementos:

1. Diseño de la arquitectura web: se definirán los diferentes componentes y tecnologías necesarias para desarrollar las páginas web, así como sus respectivas épicas, historias de usuario y criterios de aceptación relacionados a los requisitos del proyecto, además de las vistas, templates y lógica de negocio respectiva.
2. Base de datos: para la gestión de contenido por parte del administrador para que este pueda realizar un CRUD completo en la página del catálogo, se necesitará una base de datos para almacenar y gestionar los datos de los productos. El DBMS elegido será un sistema gestor de base de datos relacional, en este caso: MySQL.
3. Desarrollo de funcionalidades, tales como: navegación y descubrimiento de contenido, simulación de pedidos, además de funcionalidades relacionadas con la captación de información del usuario y potencial comprador mediante formulario con envío de correo electrónico automático.
4. Diseño e implementación de una interfaz de usuario atractiva y fácil de usar tanto por el lado del usuario como por el lado del administrador.
5. Integración con sistemas de terceros: se podrán integrar sistemas externos, en este caso mediante un sistema de envío de correo automático conectado con el formulario de contacto.
6. Pruebas y validación: se realizarán pruebas exhaustivas de la aplicación para garantizar y verificar su correcto funcionamiento.
7. Implementación y despliegue: se desplegará en un servidor que provea de hosting gratuito en donde alojar los sitios web a la medida del tamaño del proyecto y su complejidad.
8. Documentación y capacitación: se proporcionará documentación adecuada que incluya una guía de usuario detallada para facilitar el uso de la navegación en el sitio web por parte de los usuarios, así como una capacitación básica en su uso.



**Descripción del Servicio:**

Para efectuar el desarrollo de la aplicación se requiere del siguiente entorno de trabajo:

## ***1.- Describa la infraestructura necesaria para la implementación de la aplicación:***

La infraestructura necesaria para la implementación de las páginas web no contempla un gran tráfico web, ni posee otras aplicaciones que eleve el número de usuarios por lo cual se cree necesario los siguientes elementos:

1. Servidor web o servicio de Hosting: La aplicación necesitará un servidor web para alojar y servir el sitio web a los usuarios. Se pretende utilizar Apache para levantar la aplicación. Otra alternativa viable es un servicio de alojamiento en la nube como Amazon Web Services (AWS).
2. Base de datos: La aplicación necesitará una base de datos para almacenar y gestionar los datos de los productos y simulaciones de pedido hechas por los usuarios. Se considera el uso de una base de datos SQL, en este caso: MySQL por la baja complejidad del proyecto y el escaso requerimiento de datos que provee el aplicativo, así como por su gran compatibilidad con Apache o AWS.
3. Entorno de desarrollo: Los desarrolladores necesitarán un entorno de desarrollo en sus equipos locales para construir y probar la aplicación. Este entorno incluirá un editor de código como Visual Studio Code y las herramientas y librerías necesarias para desarrollar la aplicación con Express (Node.js), dada su alta compatibilidad y compenetración con el lenguaje de programación de alto nivel: JavaScript.
4. Herramientas de gestión de proyectos: Es necesario contar con herramientas para gestionar el proyecto, un sistema de gestión de tareas y un software de colaboración para equipos de desarrollo. Esta será Jira complementado para el desarrollo del diseño por el software Visual Paradigm.
5. Sistema de Control de Versiones: un sistema de control de versiones como Git y GitHub para trabajar en equipos con un repositorio común y poder dar seguimiento a errores y gestionar diversas versiones del proyecto que vayan surgiendo en el proceso de codificación. También nos proveerá de un sistema de copias de seguridad.

## ***2.- Describa la configuración necesaria del hardware para que funcione la aplicación:***

La configuración necesaria del hardware para la aplicación web utilizando el enfoque de Modelo-Vista-Controlador (MVC) mediante Express (Node.js) puede variar dependiendo del tamaño y la complejidad de la aplicación, así como del número de usuarios esperados. Sin embargo, a continuación, se describen algunas de las especificaciones recomendadas para un servidor web que pueda soportar la aplicación.

1. **Procesador:** Dado que la aplicación es de baja complejidad y no requiere un gran poder de procesamiento, un procesador de gama media o incluso un procesador de la generación anterior podría ser suficiente. Un procesador de al menos 4 núcleos y una frecuencia de al menos 2.5 GHz sería adecuado.
2. **RAM:** nuestro proyecto contará con 16GB, Dual Channel con 2 RAM de 8GB cada una.
3. **Almacenamiento:** Para el almacenamiento, contaremos con un SSD de 256 GB de almacenamiento para el sistema operativo, las aplicaciones y los datos de desarrollo. Además, se puede considerar la posibilidad de almacenamiento adicional en la nube como AWS para backups y datos adicionales generados por la aplicación.
4. **Conectividad:** se contará con una conexión de red rápida y confiable con fibra óptica para servir el sitio web a los usuarios de manera eficiente y eficaz.
5. **Sistema operativo:** optaremos por el sistema operativo SO GNU Linux, mediante distribución oficial Red Hat por su mejor nivel de ciberseguridad, fiabilidad y estabilidad en el uso y respaldo de la aplicación.

### ***3.- Describa el software necesario para el desarrollo, y posterior implementación de la aplicación:***

La configuración necesaria del hardware para la aplicación web utilizando el enfoque de Modelo-Vista-Controlador (MVC) mediante Express (Node.js) puede variar dependiendo del tamaño y la complejidad de la aplicación, así como del número de usuarios esperados. Sin embargo, a continuación se describen las tecnologías y software que utilizaremos para desarrollar y levantar la aplicación.

1. Para el desarrollo del Front End de la aplicación web se utilizará el stack de tecnologías de desarrollo: HTML5, CSS3 y JavaScript junto al Framework respectivo de este lenguaje de programación que es Node.js.
2. Express (Node.js): Por el lado del Back End se utilizará este framework web de alto nivel para el desarrollo rápido de aplicaciones web, basado en el patrón de diseño Modelo-Vista-Controlador (MVC). Express (Node.js) proporciona una estructura para la aplicación web, así como herramientas para la administración de bases de datos, autenticación, formularios, entre otras funcionalidades.
3. JavaScript: Express (Node.js) está escrito en JavaScript, por lo que se requerirá la instalación de JavaScript en el sistema para poder utilizar Express (Node.js). JavaScript es un lenguaje de programación de alto nivel, fácil de leer y escribir, y es el más popular en el desarrollo web.
4. Base de datos: Se considera el uso de una base de datos SQL, en este caso: MySQL por la baja complejidad del proyecto y el escaso requerimiento de datos que provee el aplicativo, así como por su gran compatibilidad con Apache o AWS.
5. Entorno de Desarrollo Integrado: Se utilizará Visual Studio Code como IDE.
6. Controlador de Versiones: Utilizaremos Git para controlar las diversas versiones del programa y sus módulos derivados. Además de mantener en línea el aplicativo y su desarrollo usaremos GitHub.
7. Servidor web: Levantaremos la aplicación con Apache, aunque sigue en evaluación la utilización de un servidor con alojamiento en la nube como Amazon Web Services (AWS).
8. Sistema Operativo: Se usará el SO GNU Linux con la distribución oficial Red Hat.

## 4. Product Backlog

El product backlog del proyecto puede observarse con detalle en el siguiente link:

Epic		+ Crear incidencia	
Incidencias sin epic			
<ul style="list-style-type: none"> <li>Navegacion y Descubrimiento de Contenido</li> <li>Simulacion de Compra</li> <li>Formulario de Contacto</li> <li>Gestion de Contenido</li> </ul>		0 incidencias   Estimación: 0 + Crear sprint	
<input type="checkbox"/> Backlog (11 incidencias)			
SCRUM-7 A01_Exploracion-LandingPage	NAVEGACION Y DESCU...	TAREAS POR HACER	
SCRUM-8 A02_Exploracion-Catalogo	NAVEGACION Y DESCU...	TAREAS POR HACER	
SCRUM-9 A03_Comunicacion-con-vendedor	NAVEGACION Y DESCU...	TAREAS POR HACER	
SCRUM-10 A04_Agregar-Productos-Simulacion	SIMULACION DE COMP...	TAREAS POR HACER	
SCRUM-11 A05_Despliegue-detalles-producto	SIMULACION DE COMP...	TAREAS POR HACER	
SCRUM-12 A06_Envio-Formulario	FORMULARIO DE CONT...	TAREAS POR HACER	
SCRUM-13 A07_Recibir-Notificaciones	FORMULARIO DE CONT...	TAREAS POR HACER	
SCRUM-15 A08_Interfaz-Productos	GESTION DE CONTENIDO	TAREAS POR HACER	
SCRUM-14 A09_Agregar-Producto	GESTION DE CONTENIDO	TAREAS POR HACER	
SCRUM-16 A10_Modificar-Producto	GESTION DE CONTENIDO	TAREAS POR HACER	
SCRUM-17 A11_Eliminar-Producto	GESTION DE CONTENIDO	TAREAS POR HACER	

- <https://inacapmail-fco-ruiz.atlassian.net/jira/software/projects/SCRUM/boards/1/backlog?epics=visible&atlOrigin=eyJpIjoiNzBkNWYxOTZmNTQ2NDNmMThiNjc2ZjFINGRjYjA1MGYiLCJwIjoiaaiJ9>

## 5. Preparación del entorno

### Preparación del entorno:

#### 1. Instalación de Apache y MySQL:

- Asegúrate de tener Apache y MySQL instalados en tu sistema. Lo instalaremos utilizando el gestor de paquetes "yum", de Linux. En este caso, en Red Hat:

```
sudo yum install httpd mysql-server
```

#### 2. Instalación de Node.js:

2.1 Agregar el repositorio Extra Packages for Enterprise Linux (EPEL):

```
sudo yum install epel-release
```

2.2 Después de agregar el repositorio, instala los paquetes necesarios para la instalación de Node.js:

```
sudo yum install curl gcc-c++ make
```

2.3 Descarga e instala Node.js utilizando el siguiente comando:

```
sudo yum install nodejs
```

#### 3. Instalación de Visual Studio Code:

- Para instalar hay que configurar el repositorio de Microsoft

3.1 Descarga e importa la clave GPG mediante el siguiente comando curl:

```
rpm --import https://packages.microsoft.com/keys/microsoft.asc
```

3.2 Agregar el repositorio vscode a /etc/yum.repos.d/vscode.repo:

```
[code]
```

```
name=Visual Studio Code
```

```
baseurl=https://packages.microsoft.com/yumrepos/vscode
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey= https://packages.microsoft.com/keys/microsoft.asc
```

3.3 Instalar la última versión de code:

```
sudo yum install -y code
```

3.4 Agregar *Live Server* a las extensiones de VSC para trabajar de forma local.

## 5. Configuración del Entorno Virtual:

- Para Node.js, generalmente no se utiliza un entorno virtual como en Python. Las dependencias se manejan a nivel de proyecto utilizando el archivo "package.json".

## 6. Creación del Proyecto Express:

- Instala express-generator (si aun no lo tienes instalado) con el siguiente código

```
npm install -g express-generator
```

6.1 Crea el proyecto Express con el siguiente comando

```
npx express-generator myproject
```

## 7. Instalación de Dependencias:

- Navega al directorio del proyecto con el siguiente comando:

```
cd myproject
```

7.1 Una vez estes en el directorio utiliza el siguiente código:

```
npm install
```

## 8. Configuración de MySQL:

- Configura la conexión a la base de datos MySQL en el archivo `app.js` o en un archivo de configuración separado utilizando un paquete como `mysql` o `sequelize`.

## 9. Desarrollo de la Aplicación:

- Desarrolla tus rutas, controladores y vistas utilizando Express y las tecnologías front-end que mencionaste (HTML5, CSS3, JavaScript).

## 10. Despliegue en Apache:

- Configura Apache para servir tu aplicación Express. Puedes hacer esto utilizando módulos como `mod\_proxy` o `mod\_proxy\_fcgi`.

## 11. Despliegue en la Nube:

- Si decides utilizar un servicio en la nube como AWS, sigue las instrucciones proporcionadas por el proveedor para desplegar tu aplicación. A menudo, esto implicará configurar un servidor EC2, instalar las dependencias necesarias y configurar Apache o Nginx para servir tu aplicación.