



Departamento de Ciencias de la Computación (DCCO)

Ingeniería en Software

Análisis y Diseño de Software NRC 8311

**“SISTEMA DE VENTAS EN LÍNEA ECUABONSAI”.**

**Integrantes:**

Jairo Alvarado

David Calvopiña

Bryan Yanzapanta

Jonathan Zapata

**Tutor:**

Ing. Jenny Ruiz

Sangolquí, 2023

**Historial de Revisiones**

<b>Fecha</b>	<b>Versión</b>	<b>Descripción</b>	<b>Autor</b>
04/02/2023	1.0	Revisión documental Capítulo I, II	Jairo Alvarado David Calvopiña Bryan Yanzapanta Jonathan Zapata
07/02/2023	2.0	Revisión documental Capítulo I, II, III, IV	Jairo Alvarado David Calvopiña Bryan Yanzapanta Jonathan Zapata

## Índice

RESUMEN EJECUTIVO .....	5
INTRODUCCIÓN .....	6
CAPÍTULO I .....	7
1.1 Título del Proyecto .....	7
1.2. Sistema de Objetivos .....	7
1.2.1 Objetivo General .....	7
1.2.2 Objetivos Específicos .....	7
1.3 Alcance .....	7
1.4 Definición y Justificación del Problema .....	8
1.5 Presupuesto .....	9
1.5.1 Recursos Hardware y Software .....	9
1.5.2 Recursos Humanos .....	10
1.5.3 Factibilidad Económica .....	11
CAPÍTULO II .....	11
2.1 Modelamiento del Negocio y sus Entregables .....	11
2.1.1 Documentos de Caso de Uso Historias de Usuario .....	11
2.2 Definición de Requerimientos .....	14
2.2.1 Especificación de Requerimientos de Software .....	14
2.2.2 Especificación de Casos de Uso .....	24
2.3 Análisis y Diseño .....	25
2.3.1 Modelo Conceptual .....	25
2.3.2 Modelo Lógico .....	25

2.3.3 Modelo Físico .....	26
2.3.4 Script de la Base de Datos.....	27
2.4 Implementación .....	31
2.4.1 Modelo de Arquitectura .....	31
2.4.1 Modelo de Arquitectura Cliente-Servidor .....	32
CAPÍTULO III .....	32
3.1 Pruebas.....	32
3.1.1 Prueba de Caja Negra.....	32
3.1.2 Prueba de Caja Blanca .....	32
3.1.3 Documentación de Informe de Errores.....	32
3.1.4 Documentación Técnicas Caja Blanca y Caja Negra.....	33
CAPÍTULO IV .....	39
4.1 Conclusiones.....	39
4.2 Recomendaciones. ....	39

## **RESUMEN EJECUTIVO**

En el presente trabajo abordaremos el análisis y diseño de software en la creación de un sistema de ventas en línea para el emprendimiento Ecuabonsai, el mismo que está orientado en brindar como solución la automatización del proceso de ventas, debido a que el emprendimiento no cuenta con un local físico y manejan sus ventas mediante las redes sociales.

## **INTRODUCCIÓN**

El emprendimiento necesita aumentar la venta de sus productos, por lo que brindar un mejor servicio es una prioridad y esto se puede lograr a través de una página web que cuente con un sistema de ventas en línea, así como también que proporcione información relevante acerca del emprendimiento.

Es necesario decir que nos guiaremos en el proceso de análisis de requisitos previamente realizado en el semestre anterior, ya que este proceso de elicitación nos ayudó a identificar las necesidades a través de cada uno de los requisitos funcionales.

## CAPÍTULO I

### 1.1 Título del Proyecto

Sistema de ventas en línea Ecuabonsai

### 1.2. Sistema de Objetivos

#### 1.2.1 Objetivo General

Realizar un análisis y diseño de software, mediante la creación de una aplicación web para automatizar el proceso de ventas del emprendimiento Ecuabonsai.

#### 1.2.2 Objetivos Específicos

- Identificar el patrón de diseño con el que se va a realizar de acuerdo a los requisitos funcionales propuestos en el documento de especificación de requisitos.
- Aplicar las técnicas de las 5W y 2H, para obtener la matriz de historias de usuario (HU).
- Realizar casos de pruebas a las diferentes funcionalidades del sistema y reportar los errores encontrados.

### 1.3 Alcance

Las fases del proceso de Análisis y Diseño de Software que posibilitarán el desarrollo del proyecto son:

- **1.3.1 Etapa de análisis:** Tener un conocimiento claro del problema que se desea resolver para poder crear el sistema de venta en línea Ecuabonsai de tal manera que se identifiquen los componentes principales con los que contara el mismo.
- **1.3.2 Etapa de Diseño:** Utilizar la información recolectada y enfocarnos en la selección de un modelo que nos permita llevar a cabo nuestros requerimientos y al cumplimiento de las necesidades del cliente

- **1.3.3 Etapa de Desarrollo** Utilizar los modelos creados durante la etapa de diseño para el desarrollo del sistema de venta en línea donde su principal característica sea la creación de un catálogo que permita a los usuarios
  - Registrar una cuenta: El usuario podrá registrar todos sus datos
  - Visualizar un catálogo: El usuario podrá navegar por el catálogo y observar todas las características del producto y conforme se decida; seleccionar un producto para su posterior cuenta
  - Realizar compra: El usuario podrá añadir el producto o los productos a un carrito de compras.
- **1.3.4 Etapa de Pruebas:** Comprobar que efectivamente el sistema y los componentes que lo conforman cumplan los requerimientos establecidos en el documento SRS

#### **1.4 Definición y Justificación del Problema**

El emprendimiento Ecuabonsai necesita ampliar su reconocimiento como marca y agilizar el proceso de ventas mediante el diseño de una página web, la aplicación tendrá repercusiones en las ventas, publicidad y automatizar el registro de compra asimismo el sistema de ventas será más dinámico para el cliente, contando con un usuario, contraseña y accediendo al catálogo que se ira actualizado conforme a la llegada de nuevos productos o temporadas del año.

El impacto que va a tener el sistema de ventas en línea Ecuabonsai será dar a conocer al público en general donde obtener arboles bonsái al mejor precio y con la calidad del producto 100% garantizada, esta propuesta busca instruir a otros investigadores sobre el planteamiento de una página web la cual brindará una mejora a la imagen de la marca cliente.



## 1.5 Presupuesto

### 1.5.1 Recursos Hardware y Software

Recurso	Requisitos del sistema	Valor (USD)
Sistema operativo	Windows 10	20
Programa de ofimática	Microsoft Office 365	16
Aplicación para realizar gráficos	Lucid	8
IDE para desarrollo	Visual Studio Code	0
Reuniones	Zoom	0
		44

**Tabla 1.** Recursos Software

Recurso	Requisitos de los estudiantes	Valor (USD)
Computar personal primer estudiante	DELL	500
Computar personal segundo estudiante	DELL	900
Computar personal tercer estudiante	HP	900
Computar personal cuarto estudiante	HP	500

	2800
--	------

**Tabla 2. Recursos Hardware**

### **1.5.2 Recursos Humanos**

Nombre	Cargo	Funciones	Periodo de tiempo
Jairo Alvarado	Estudiante	Encargado de redactar, investigar, analizar e implementar el aplicativo software que cumple con los objetivos planteados.	24 de octubre del 2022 hasta la finalización, calificación y presentación del documento el 13 de febrero de 2023
David Calvopiña	Estudiante	Encargado de revisar y desarrollar las correcciones del documento de investigación.	24 de octubre del 2022 hasta la finalización, calificación y presentación del documento el 13 de febrero de 2023
Bryan Yanzapanta	Estudiante	Encargado de realizar las correcciones dentro del aplicativo software.	24 de octubre del 2022 hasta la finalización, calificación y presentación del documento el 13 de febrero de 2023
Jonathan Zapata	Estudiante	Líder del grupo, encargado de verificar que	24 de octubre del 2022 hasta la finalización,

		los demás miembros cumplan sus funciones, además de realizar las reuniones con el usuario y apoyar a las correcciones dentro del aplicativo software.	calificación y presentación del documento el 13 de febrero de 2023
--	--	---	--

**Tabla 3. Recursos Humanos**

### **1.5.3 Factibilidad Económica**

	Primer Mes	Segundo Mes	Tercer Mes	Cuarto Mes	Total
Computadores	2800	2800	2800	2800	1400
Recursos software	4970	4970	4970	4970	24850
Total					26250

**Tabla 4. Factibilidad económica**

## **CAPÍTULO II**

### **2.1 Modelamiento del Negocio y sus Entregables**



#### **2.1.1 Documentos de Caso de Uso Historias de Usuario**

HISTORIA DE USUARIO (HU)					
ITEM	USUARIO	STATUS			
REQ001	Admin	En proceso			
TIEMPO	PRIORIDAD	PROG. RESP			
8	Alta	Bryan Yanzapanta			
QUE	PARA QUE		COMO		
Gestionar el producto con sus datos; nombre, descripción, tamaño, edad, especie y precio	Vender el producto mediante un catálogo		Gestionar el producto mediante un CRUD que contenga todos los datos del producto,		
NOMBRE HISTORIA	Gestionar Producto				
PRUEBA	COMENTARIOS				
Pruebas Unitarias que validen los datos del producto	0				



**Figura 1:** Historia de usuario REQ001

HISTORIA DE USUARIO (HU)					
ITEM	USUARIO	STATUS			
REQ002	Usuario	En proceso			
TIEMPO	PRIORIDAD	PROG. RESP			
1	Alta	Jonathan Zapata			
QUE	PARA QUE		COMO		
Iniciar Sesión mediante un usuario y contraseña	Ingresar al Sistema		Solicitar usuario y contraseña		
NOMBRE HISTORIA	Iniciar Sesión				
PRUEBA	COMENTARIOS				
Pruebas Unitario que validen los datos del usuario	0				



**Figura 2:** Historia de usuario REQ002

HISTORIA DE USUARIO (HU)			
ITEM REQ003	USUARIO Cliente	STATUS En proceso	 
TIEMPO 2	PRIORIDAD Alta	PROG. RESP David Calvopina	
QUE Registrar Cuenta para acceder a todas las funciones del sistema,.	PARA QUE Realizar Pedidos y acceder a las demás funciones que existen dentro del mismo.	COMO Solicitar nombre y tipo de usuario y contraseña, en este caso debe colocar en tipo de usuario cliente ya que para los otros dos tipos es diferente el proceso	
NOMBRE HISTORIA Registrar Cuenta			
PRUEBA	Pruebas Unitarias que validen los datos del cliente		COMENTARIOS Las cuentas se abren principalmente para clientes, ya que los otros actores ya tienen una cuenta asignada.

**Figura 3:** Historia de usuario REQ003

HISTORIA DE USUARIO (HU)			
ITEM REQ004	USUARIO Admin	STATUS En proceso	 
TIEMPO 6	PRIORIDAD Alta	PROG. RESP Jairo Alvarado	
QUE Seleccionar los productos para generar las promociones.	PARA QUE Aumentar momentaneamente la demanda de aquellos productos seleccionados	COMO Seleccionar un producto 'x' y rebajar su precio ó vender 2 productos al precio de uno, todo esto mediante el CRUD	
NOMBRE HISTORIA Generar Promociones			
PRUEBA	Pruebas Unitarias que validen los datos del producto		COMENTARIOS 0

**Figura 4:** Historia de usuario REQ004

HISTORIA DE USUARIO (HU)			
ITEM REQ005	USUARIO Usuario	STATUS En proceso	 
TIEMPO 5	PRIORIDAD Alta	PROG. RESP Byan Yanzapanta	
QUE Seleccionar los productos para generar pago	PARA QUE Realizar la compra y obtener los productos deseados por el usuario	COMO Seleccionar un producto 'x' y la cantidad	
NOMBRE HISTORIA Generar Promociones			
PRUEBA	Pruebas Unitarias que validen la seleccion del producto		COMENTARIOS 0

**Figura 5:** Historia de usuario REQ005

## 2.2 Definición de Requerimientos

### 2.2.1 Especificación de Requerimientos de Software

#### Requisitos Funcionales

#### Requisito funcional 1

<b>Nombre</b>	Gestionar Producto	
<b>Id. Requerimiento</b>	REQ001	
<b>Objetivos</b> <b>Asociados -</b> <b>Propósito</b>	Administrar toda la información del producto bonsái.	
<b>Dependencias</b>	REQ003	
<b>Descripción</b>	El sistema deberá permitir la gestión del producto bonsái; registrando, actualizando, eliminando, y mostrando mediante reportes los productos bonsái.	
<b>Actor</b>	Administrador	
<b>Pre-Condiciones</b>	El administrador debe haber iniciado sesión en su cuenta.	
<b>Post-Condiciones</b>	N/A	
	Pasos	Acción

<b>Secuencia Normal</b>	1	El actor solicita gestionar todos los productos bonsáis disponibles para la venta.
	2	El sistema procede a mostrar todas las operaciones; como son ingresar, actualizar, eliminar, y mostrar reportes.
<b>Excepciones</b>	Pasos	Acción
	1	Si el actor solicita gestionar los productos bonsái y el sistema no funciona se despliega un mensaje de error, que describe que el sistema no se encuentra disponible por el momento.
<b>Rendimiento</b>	No existe un tiempo estimado.	
<b>Frecuencia</b>	Cada vez que se desee gestionar los productos bonsái.	
<b>Prioridad</b>	Alta	
<b>Comentarios</b>	Se mostrarán diferentes casos de uso en un segundo nivel.	

## Requisito funcional 2

<b>Nombre</b>	Iniciar Sesión
---------------	----------------

<b>Id. Requerimiento</b>	REQ002	
<b>Objetivos Asociados</b> <b>- Propósito</b>	Administrar cada cuenta para cada tipo de usuario en el sistema.	
<b>Dependencias</b>	REQ003	
<b>Descripción</b>	El sistema deberá administrar el inicio de sesión para diferentes actores. El proceso se realiza con el ingreso de datos previamente creados y cada actor será redirigido hacia su interfaz correspondiente.	
<b>Actor</b>	Administrador, Cliente, Vendedor	
<b>Pre-Condiciones</b>	Tener una cuenta ya registrada	
<b>Post-Condiciones</b>	N/A	
<b>Secuencia Normal</b>	Pasos	Acción
	1	El actor solicita al sistema iniciar sesión
	2	El sistema muestra dos campos a llenar: usuario y contraseña.
	3	El actor ingresa los datos correspondientes en los diferentes campos establecidos.



	4	El sistema valida los campos de usuario y contraseña y muestra la opción de iniciar de sesión
	5	El actor confirma su inicio de sesión
	6	El sistema valida con la base de datos los campos de usuario y contraseña, y redirecciona al actor hacia su interfaz correspondiente.
<b>Excepciones</b>	Pasos	Acción
	2	Si el actor no tiene una cuenta para iniciar, deberá crear una cuenta.
	5	Si el actor ingresa incorrectamente más de 3 veces, el sistema bloquea temporalmente el ingreso de usuario y contraseña.
	6	Si el sistema no inicia sesión, se debe volver a intentar el ingreso verificando que los datos sean los correctos, con un máximo de 3 intentos.
<b>Rendimiento</b>	El tiempo mínimo estimado para realizar el ingreso de sesión es de 2 minutos	
<b>Frecuencia</b>	En cada proceso de venta	

<b>Prioridad</b>	Alta
<b>Comentarios</b>	Es importante recalcar, que las cuentas tienen diferentes tipos, dependiendo del actor al que accede, por lo tanto, la información o páginas mostradas es distinta.

### Requisito funcional 3

<b>Nombre</b>	Registrar Cuenta
<b>Id. Requerimiento</b>	REQ003
<b>Objetivos Asociados - Propósito</b>	Registrar una cuenta individual para cada tipo de usuario en el sistema.
<b>Dependencias</b>	N/A
<b>Descripción</b>	Una vez que el actor se encuentre en el sistema necesitará tener una cuenta para poder acceder a las funciones que este ofrece, de lo contrario, el actor sólo podrá visualizar el catálogo de productos, pero no podrá realizar un pedido. En efecto, el sistema permitirá al actor crear una cuenta para poder realizar un pedido y acceder a las demás funciones que existen dentro del mismo.
<b>Actor</b>	Visitante

<b>Pre-Condiciones</b>	N/A	
<b>Post-Condiciones</b>	N/A	
<b>Secuencia Normal</b>	Pasos	Acción
	1	El visitante solicita al sistema registrar una cuenta.
	2	El sistema procede a mostrar un campo de registro con los datos de usuario (nombre de usuario, tipo de usuario) y contraseña.
	3	El visitante ingresa sus datos en los diferentes campos.
	4	El sistema valida que todos los campos estén ingresados de manera correcta y muestra la opción de registrar.
	5	El visitante procede a confirmar el registro de la cuenta, cambiando su estado a cliente.
	6	El sistema asigna automáticamente un ID cliente luego del registro y todos sus datos son guardados en la base de datos.
	Pasos	Acción

<b>Excepciones</b>	3	Si el visitante ingresa incorrectamente los datos, el sistema pedirá reingresar bien los datos hasta que sean válidos (con caracteres necesarios, datos reales, entre otras validaciones)
	5	Si el sistema no se encuentra disponible se muestra un mensaje de error en la confirmación de registro.
<b>Rendimiento</b>	El tiempo estimado para el registro de cuenta es de unos 5 minutos.	
<b>Frecuencia</b>	Cada vez que un visitante quiere hacer una compra.	
<b>Prioridad</b>	Media	
<b>Comentarios</b>	Las cuentas se abren principalmente para clientes, ya que los otros actores ya tienen una cuenta asignada, pero en caso de querer crear una cuenta a ellos, su proceso es externo al proceso de compra, por ello no está añadido.	

#### Requisito funcional 4

<b>Nombre</b>	Generar Promociones
<b>Id. Requerimiento</b>	REQ004

<b>Objetivos Asociados - Propósito</b>	Proporcionar ofertas a ciertos productos, especialmente en fechas especiales.	
<b>Dependencias</b>	REQ001	
<b>Descripción</b>	<p>El administrador genera diferentes promociones que se guardarán en la base de datos del sistema, estas promociones se activan en los productos del catálogo de bonsáis de acuerdo a fechas o momentos, que el administrador puede cambiar.</p> <p>Además de la generación, el administrador configura otros datos de las promociones de acuerdo a especificaciones internas de la empresa (finanzas, marketing).</p>	
<b>Actor</b>	Administrador	
<b>Pre-Condiciones</b>	Tener un catálogo donde realizar promociones	
<b>Post-Condiciones</b>	Mantener un orden establecido de las promociones generadas.	
<b>Secuencia Normal</b>	Pasos	Acción
	1	El administrador ingresa a la página de añadir promociones.
	2	El sistema muestra la opción de generar una promoción sobre un producto bonsái donde se

		mostrarán los datos a llenar como son fecha, descuento, tiempo, desarrolló, descripción.
	3	El administrador genera una nueva promoción, ingresando los de datos explicando la razón de la promoción y sus características
	4	El sistema valida la información recibida de los campos ingresados
	5	El sistema crea la promoción en el catálogo para informar a los clientes.
<b>Excepciones</b>	Pasos	Acción
	4	En caso de ser secreto, no presentará la promoción en el catálogo.
<b>Frecuencia</b>	1 o 2 veces al mes, de acuerdo a las necesidades del propietario	
<b>Prioridad</b>	Baja	
<b>Rendimiento</b>	No existe un tiempo estimado ya que depende del administrador y de las promociones que esté manejando actualmente	

<b>Urgencia</b>	Específica para una fecha importante
<b>Comentarios</b>	El caso de uso también podría eliminar y modificar, pero comúnmente las promociones son por periodos ya específicos y programados, por lo que no se tienen a borrar ni modificar

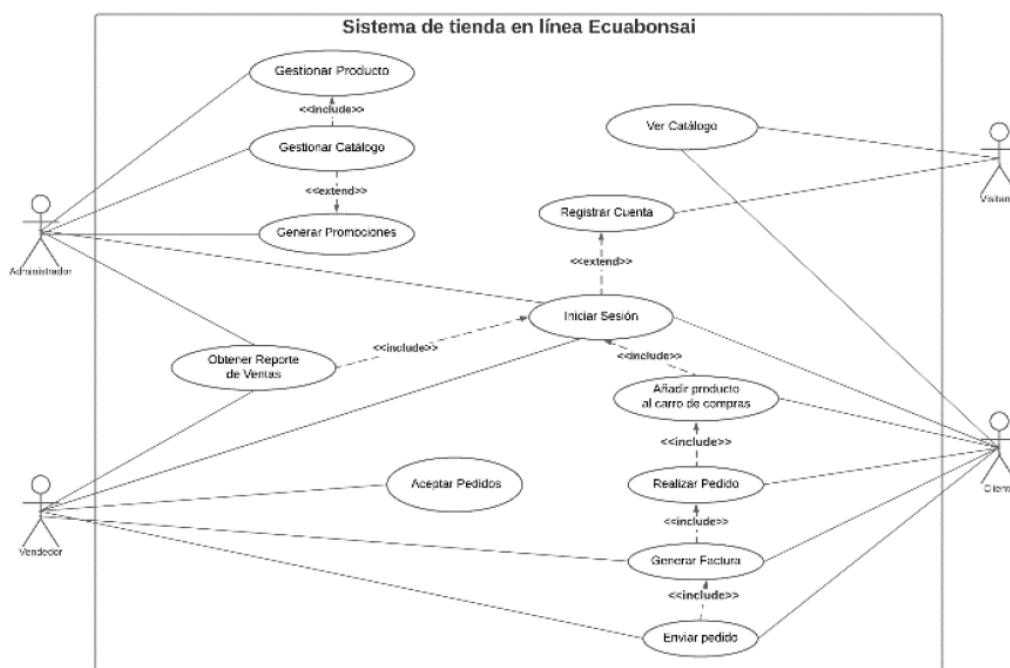
### Requisito funcional 5

<b>Caso de Uso</b>	Añadir Producto al Carro de Compras	
<b>Número</b>	REQ005	
<b>Objetivos Asociados - Propósito</b>	Administrar los productos bonsái en un carrito, donde se calcule el costo por artículo/s que hayan sido registrados.	
<b>Dependencias</b>	REQ001	
<b>Descripción</b>	El cliente elige los productos que desea adquirir, para ello añade productos a un carro de compras, eligiendo de acuerdo al catálogo presentado, este carro de compras calcula el costo total de todo lo que se le añada.	
<b>Actor</b>	Cliente	
<b>Pre-Condiciones</b>	Tener una lista de productos	
<b>Post-Condiciones</b>	Que el producto sea presentado como listo para la venta	
<b>Secuencia Normal</b>	Pasos	Acción
	1	El cliente elige productos del catálogo mostrado.
	2	El sistema muestra la opción de elegir la cantidad de productos que se desea añadir.

	3	Una vez que el cliente selecciona la cantidad, se añade al carrito de compras.
	4	El sistema añade el producto o productos y calcula la cantidad total.
<b>Excepciones</b>	Pasos	Acción
	3	Si el cliente desea cambiar la cantidad o quitar el producto o productos puede hacerlo seleccionando el carrito de compras, Es decir el proceso de modificación puede realizarse al entrar al carrito en otro momento.
<b>Rendimiento</b>	No existe un tiempo estimado ya que depende del actor y de los productos que desee añadir al carrito.	
<b>Frecuencia</b>	En cada proceso de venta	
<b>Prioridad</b>	Alta	
<b>Comentarios</b>	En este caso de uso superpone la existencia de un modelo programado de carro de compras, donde se toma en cuenta el añadir para el desarrollo principal, pero también se puede eliminar y modificar.	

### 2.2.2 Especificación de Casos de Uso

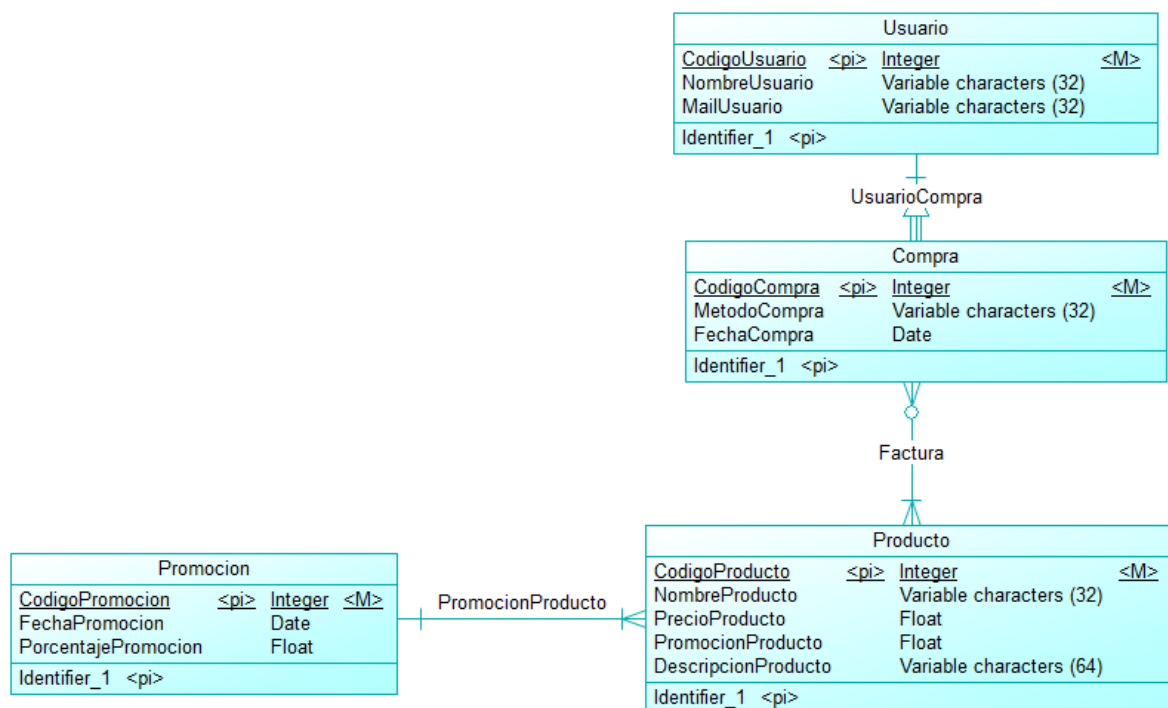




**Figura 6:** Diagrama de casos de uso

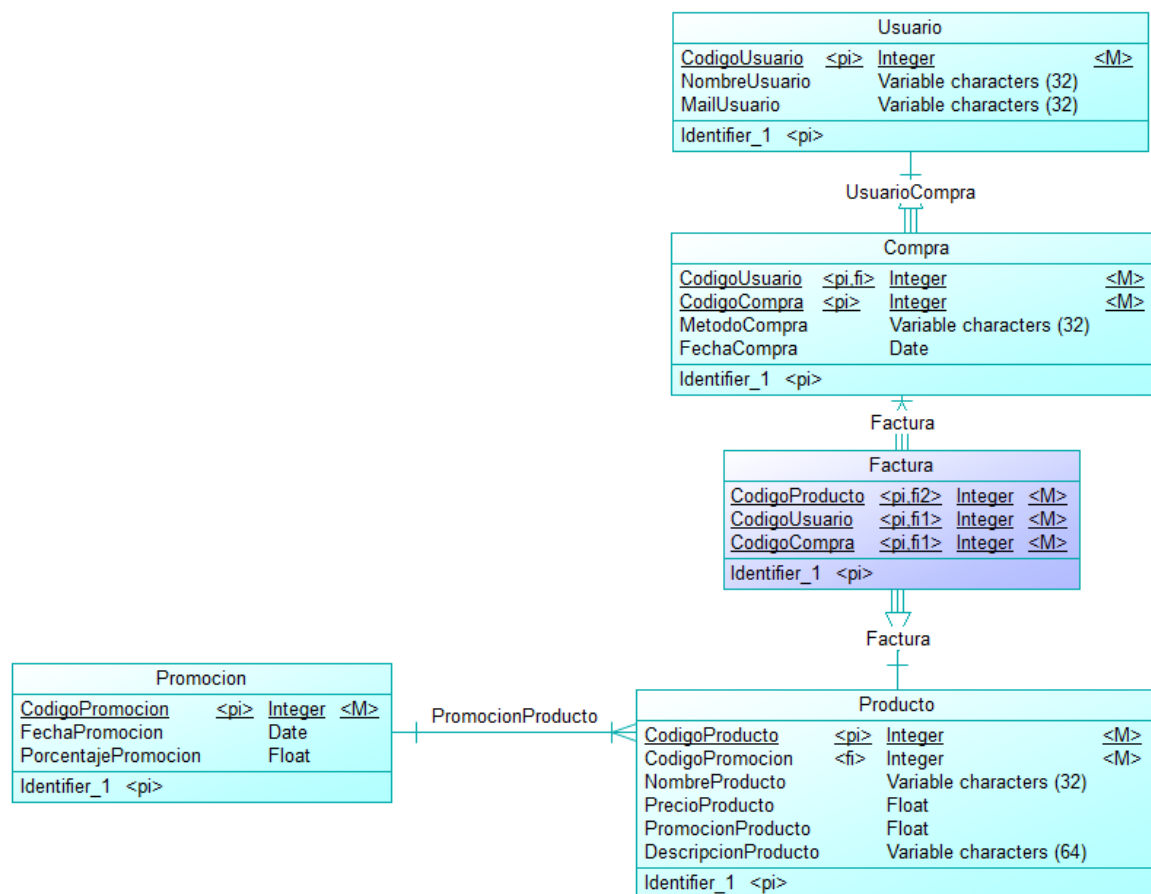
## 2.3 Análisis y Diseño

### 2.3.1 Modelo Conceptual



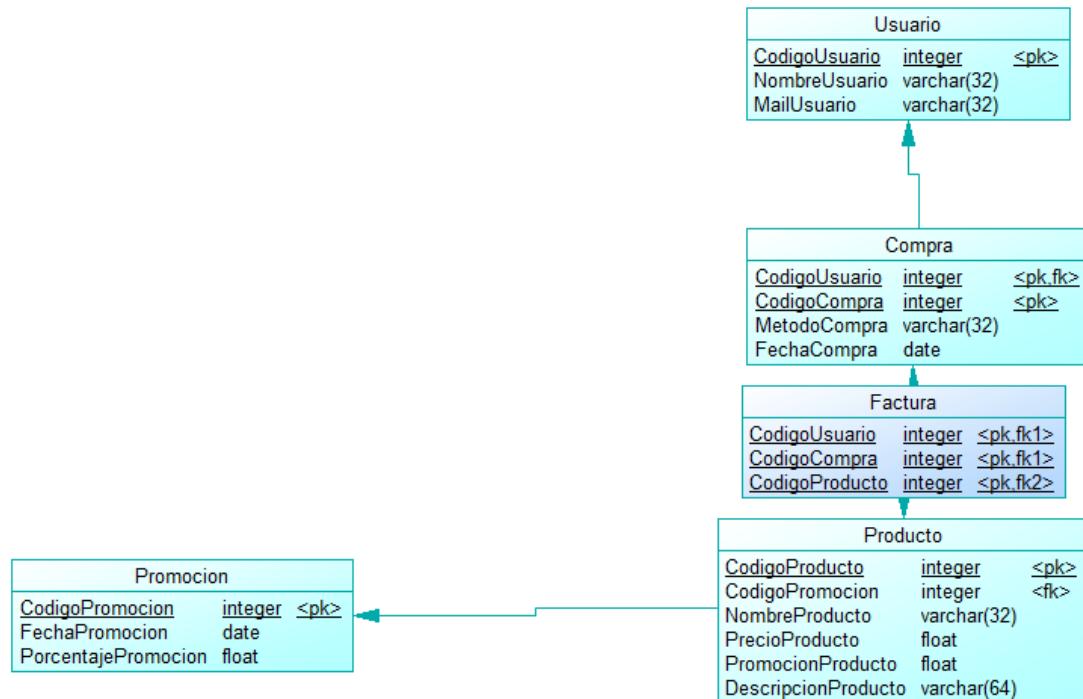
**Figura 7:** Modelo Conceptual

### 2.3.2 Modelo Lógico



**Figura 8:** Modelo Lógico

### 2.3.3 Modelo Físico



**Figura 9:**Modelo Físico

### 2.3.4 Script de la Base de Datos

```

/*=====*/
/* Table: COMPRA */
/*=====*/
create table COMPRA
(
    CODIGOUSUARIO      integer      not null,
    CODIGOCOMPRA       integer      not null,
    METODOCOMPRA       varchar(32)   null,
    FECHACOMPRA        date          null,
    constraint PK_COMPRA primary key (CODIGOUSUARIO, CODIGOCOMPRA)
);

/*=====*/
/* Index: COMPRA_PK */
/*=====*/
create unique index COMPRA_PK on COMPRA (
    CODIGOUSUARIO ASC,
    CODIGOCOMPRA  ASC
);

/*=====*/
/* Index: RELATIONSHIP_1_FK */
/*=====*/

```

```

create index RELATIONSHIP_1_FK on COMPRA (
CODIGOUSUARIO ASC
);

/*=====*/
/* Table: FACTURA */
/*=====*/
create table FACTURA
(
    CODIGOUSUARIO      integer          not null,
    CODIGOCOMPRA       integer          not null,
    CODIGOPRODUCTO     integer          not null,
    constraint PK_FACTURA primary key clustered (CODIGOUSUARIO, CODIGOCOMPRA,
CODIGOPRODUCTO)
);

/*=====*/
/* Index: FACTURA_PK */
/*=====*/
create unique clustered index FACTURA_PK on FACTURA (
CODIGOUSUARIO ASC,
CODIGOCOMPRA ASC,
CODIGOPRODUCTO ASC
);

/*=====*/
/* Index: FACTURA_FK */
/*=====*/
create index FACTURA_FK on FACTURA (
CODIGOUSUARIO ASC,
CODIGOCOMPRA ASC
);

/*=====*/
/* Index: FACTURA2_FK */
/*=====*/
create index FACTURA2_FK on FACTURA (
CODIGOPRODUCTO ASC
);

/*=====*/
/* Table: PRODUCTO */
/*=====*/
create table PRODUCTO
(
    CODIGOPRODUCTO     integer          not null,
    CODIGOPROMOCION    integer          not null,
    NOMBREPRODUCTO     varchar(32)      null,

```

```

    PRECIOPRODUCTO      float                null,
    PROMOCIONPRODUCTO   float                null,
    DESCRIPCIONPRODUCTO varchar(64)          null,
    constraint PK_PRODUCTO primary key (CODIGOPRODUCTO)
);

/*=====*/
/* Index: PRODUCTO_PK */
/*=====*/
create unique index PRODUCTO_PK on PRODUCTO (
CODIGOPRODUCTO ASC
);

/*=====*/
/* Index: PROMOCIONPRODUCTO_FK */
/*=====*/
create index PROMOCIONPRODUCTO_FK on PRODUCTO (
CODIGOPROMOCION ASC
);

/*=====*/
/* Table: PROMOCION */
/*=====*/
create table PROMOCION
(
    CODIGOPROMOCION      integer                not null,
    FECHAPROMOCION       date                  null,
    PORCENTAJEPROMOCION  float                 null,
    constraint PK_PROMOCION primary key (CODIGOPROMOCION)
);

/*=====*/
/* Index: PROMOCION_PK */
/*=====*/
create unique index PROMOCION_PK on PROMOCION (
CODIGOPROMOCION ASC
);

/*=====*/
/* Table: USUARIO */
/*=====*/
create table USUARIO
(
    CODIGOUSUARIO        integer                not null,
    NOMBREUSUARIO        varchar(32)            null,
    MAILUSUARIO          varchar(32)            null,
    constraint PK_USUARIO primary key (CODIGOUSUARIO)
);

```

```
/*=====*/
/* Index: USUARIO_PK */
/*=====*/
create unique index USUARIO_PK on USUARIO (
CODIGOUSUARIO ASC
);

alter table COMPRA
  add constraint FK_COMPRA_RELATIONS_USUARIO foreign key (CODIGOUSUARIO)
    references USUARIO (CODIGOUSUARIO)
    on update restrict
    on delete restrict;

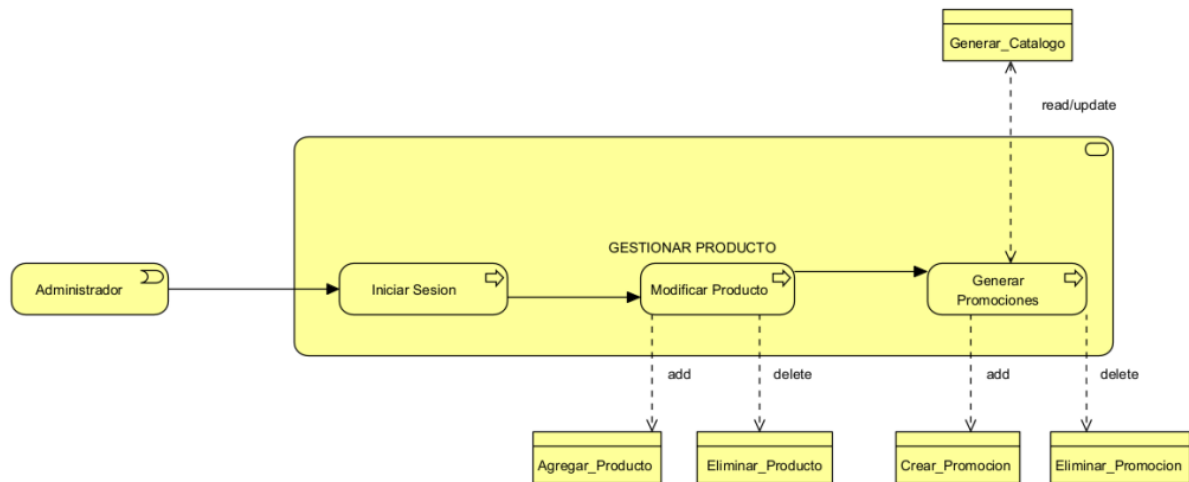
alter table FACTURA
  add constraint FK_FACTURA_FACTURA_COMPRA foreign key (CODIGOUSUARIO,
CODIGOCOMPRA)
    references COMPRA (CODIGOUSUARIO, CODIGOCOMPRA)
    on update restrict
    on delete restrict;

alter table FACTURA
  add constraint FK_FACTURA_FACTURA2_PRODUCTO foreign key (CODIGOPRODUCTO)
    references PRODUCTO (CODIGOPRODUCTO)
    on update restrict
    on delete restrict;

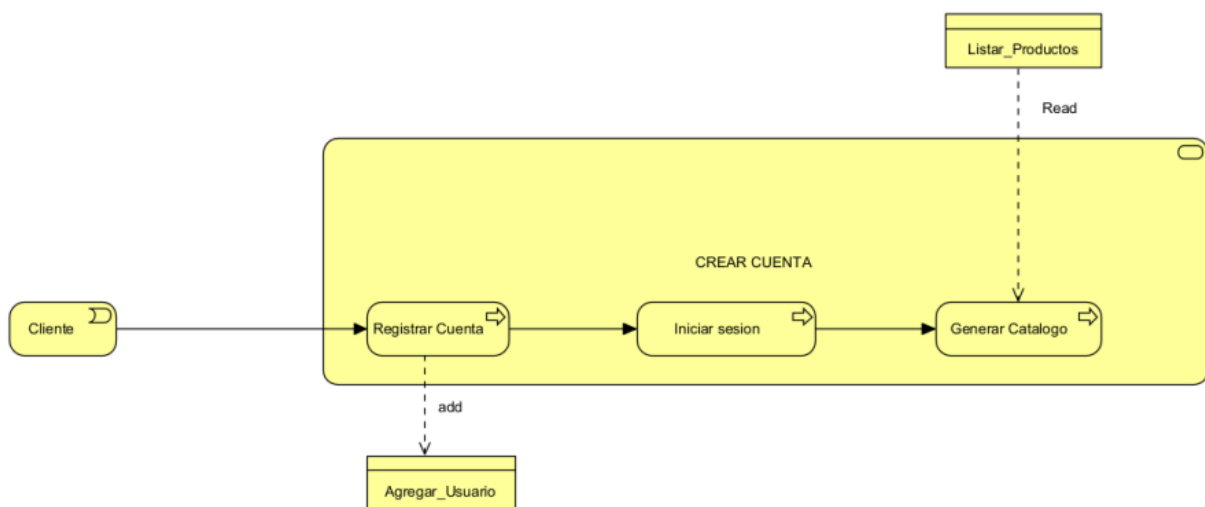
alter table PRODUCTO
  add constraint FK_PRODUCTO_PROMOCION_PROMOCIO foreign key (CODIGOPROMOCION)
    references PROMOCION (CODIGOPROMOCION)
    on update restrict
    on delete restrict;
```

## 2.4 Implementación

### 2.4.1 Modelo de Arquitectura General

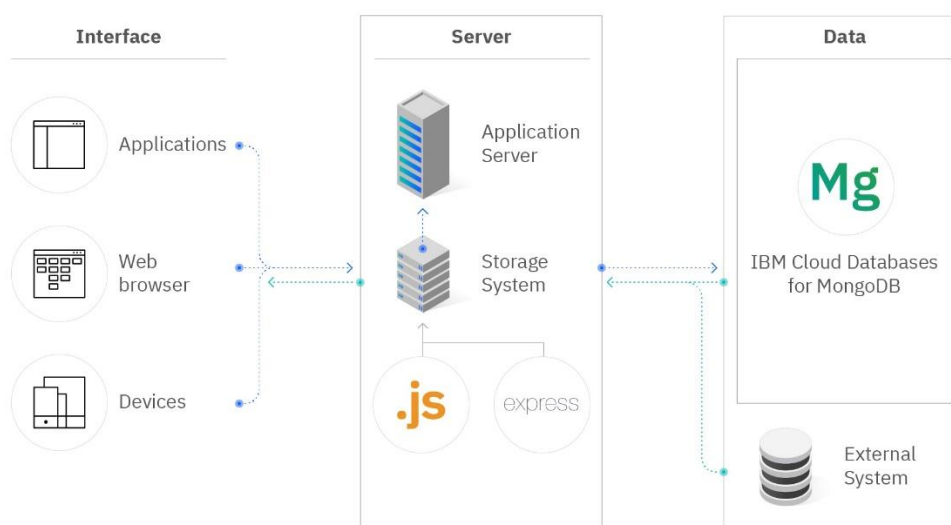


**Figura 10:** Arquitectura Administrador



**Figura 11:** Arquitectura Usuario

### 2.4.2 Modelo de Arquitectura Cliente-Servidor



**Figura 12:** Arquitectura Web. Diagrama Elaborado por IBM Cloud Databases for MongoDB

## CAPÍTULO III

### 3.1 Pruebas

#### 3.1.1 Prueba de Caja Negra

El resultado de las pruebas de Caja Negra realizadas al sistema se refleja en el documento adjunto “G5\_CajaNegra\_V1.0.docx” dentro del repositorio GitHub 8311\_G5\_ADSW.

#### 3.1.2 Prueba de Caja Blanca

El resultado de las pruebas de Caja Blanca realizadas al sistema se refleja en el documento adjunto “G5\_CajaBlanca\_V1.0.docx” dentro del repositorio GitHub 8311\_G5\_ADSW.

#### 3.1.3 Documentación de Informe de Errores

#### Función de la Plantilla



La función de esta plantilla es estandarizar y formalizar todos los aspectos que conformarán un reporte de errores de las pruebas que se aplicaron al aplicativo web: “Sistema de Ventas en Línea Ecuabonsai”.

El reporte de errores realizadas al sistema se refleja en el documento adjunto “G5\_REPORTE DE ERRORES\_V1.0.docx”.

### 3.1.3.1 Enlace: video demostrativo de evaluación por pares

<https://uespe->

[my.sharepoint.com/:v:/g/personal/jaruiz\\_espe\\_edu\\_ec/Ee67qPBuqvJCkeh6iXv5HfwBFp1HQPr\\_KYP5Z33d9XWH5g?e=S2sCD2](https://my.sharepoint.com/:v:/g/personal/jaruiz_espe_edu_ec/Ee67qPBuqvJCkeh6iXv5HfwBFp1HQPr_KYP5Z33d9XWH5g?e=S2sCD2)

### 3.1.4 Documentación Técnicas Caja Blanca y Caja Negra

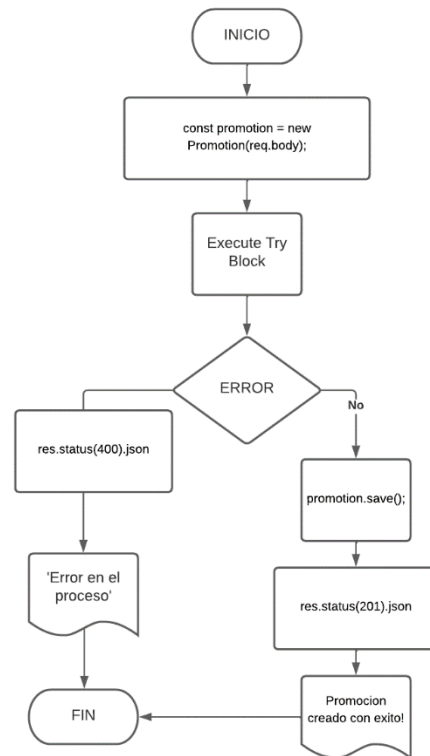
#### 3.1.4.1 TÉCNICA DE CAJA BLANCA: Cobertura de caminos

Requisito Generar Promociones

- Código fuente

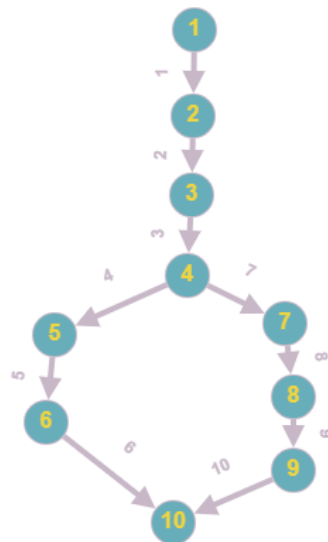
```
server > controllers > JS promotionController.js > ...
1  const Promotion = require('../models/Promotion');
2
3  // crear promocion
4
5  exports.create = async (req, res) => {
6      const promotion = new Promotion(req.body);
7
8      try {
9          await promotion.save();
10         res.status(201).json({
11             message: "Promocion creado con exito!"
12         });
13     } catch (error) {
14         res.status(400).json({
15             message: 'Error en el proceso'
16         });
17     }
18 }
```

- Diagrama de flujo



**Figura 13:** Diagrama de flujo Generar Promoción

- Grafo



**Figura 14:** Grafo función Generar Promoción

- Rutas

**R1:** 1, 2, 3, 4, 5, 6

**R2:** 1,2,3,7,8,9,10

- Complejidad Ciclomática

Se puede calcular de las siguientes formas:

- $V(G) = \text{número de nodos predichados(decisiones)} + 1$

$$V(G) = 1 + 1 = 2$$

- $V(G) = A - N + 2$

$$V(G) = 10 - 10 + 2 = 2$$

DONDE:

**P:** Número de nodos predichado

**A:** Número de aristas

**N:** Número de nodos

### 3.1.4.2 TÉCNICA DE CAJA NEGRA: Clases equivalentes

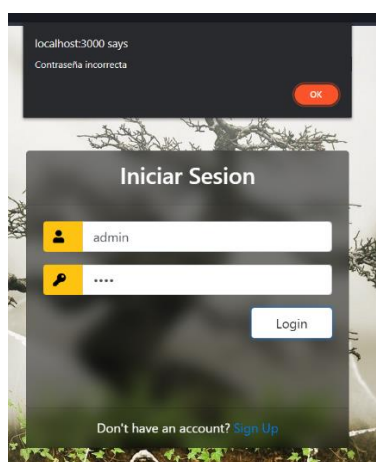
Generar Promociones

Condición de entrada	Clases validas	Clases Invalidas
Generar_Promocion	(1) Seleccionar Producto (2) Seleccionar Promoción (3) Clic en generar promoción	(4) No seleccionar producto (5) clic sobre botón distinto al de “Generar Promoción”

Nº Caso	Clase de equivalencia	Agregar-comida	Respuestas
1	1,2,3	Árbol Libre, 35%, Generar Promoción	Promoción creada con Éxito
2	4,5	-	Error en el proceso
3	1,2,5	Paz, 45%, - Clic en otro botón que no sea Generar Promoción	Error en el proceso
4	4,3	-, Generar Promoción	Error en el proceso

### Casos de prueba

#### Caso de prueba 1



**Figura 15:** Captura del aplicativo con el caso de prueba 1 Iniciar Sesión

## Caso de prueba 2

Crear Actualizar Consultar Eliminar User view

Seleccionar un producto\*

Pino Blanco  
Serissa  
Olivo

Subir Imagen

Choose Files No file chosen

Descripcion del producto

Seleccionar Promoción

10%

Enviar

### Editar Producto

ID

630aaf6e5f10e63de410e8dc

Producto

Pino Blanco

Descripcion

Garantizar un lugar exterior muy luminoso

Foto

Choose Files No file chosen

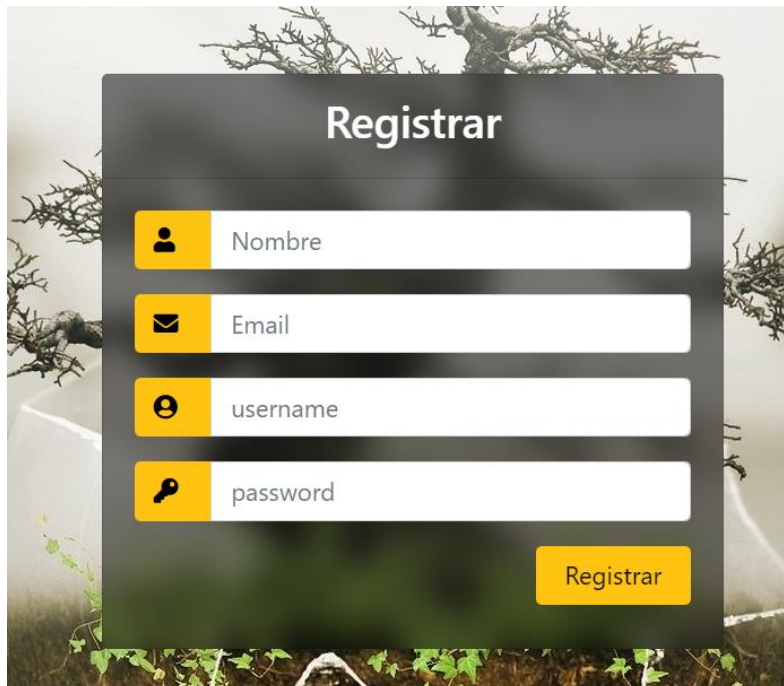
Promocion

10%

Actualizar Cancelar

**Figura 16:** Captura del aplicativo con el caso de prueba 2 Gestionar Producto

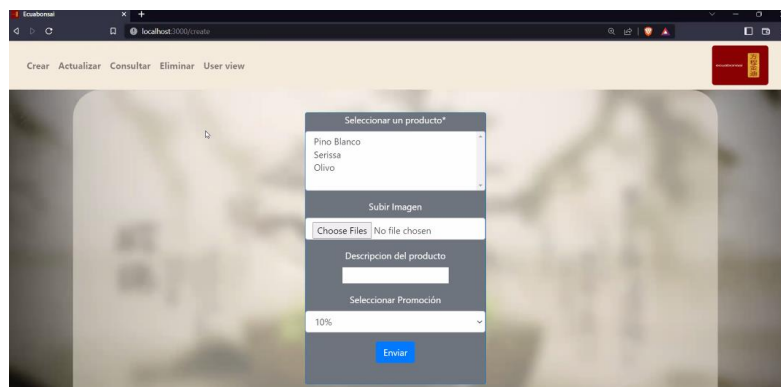
### Caso de prueba 3



The image shows a registration form titled "Registrar" overlaid on a background of a tree. The form has four input fields, each with a yellow icon on the left: a person icon for "Nombre", an envelope icon for "Email", a person icon for "username", and a key icon for "password". A yellow "Registrar" button is located at the bottom right of the form.

**Figura 17:** Captura del aplicativo con el caso de prueba 3 Registrar Cuenta

### Caso de prueba 4



The image is a screenshot of a web browser displaying a form titled "Seleccionar un producto\*". The form is part of a larger application with a navigation bar at the top containing links: "Crear", "Actualizar", "Consultar", "Eliminar", and "User view". The form itself has several sections: a dropdown menu for product selection with options "Pino Blanco", "Serissa", and "Olivo"; a "Subir Imagen" section with a "Choose Files" button and the text "No file chosen"; a "Descripcion del producto" text input field; a "Seleccionar Promoción" dropdown menu with "10%" selected; and a blue "Enviar" button at the bottom.

**Figura 18:** Captura del aplicativo con el caso de prueba 4 Generar Promociones

## **CAPÍTULO IV**

### **4.1 Conclusiones**

Tras desarrollar el aplicativo software Ecuabonsai se lograron apreciar las siguientes conclusiones:

- Mayor comodidad para los clientes, ya que pueden comprar productos desde la comodidad de sus hogares, sin tener que desplazarse a una tienda física.
- Seguimiento más eficiente de las ventas y los pedidos, lo que ayuda al emprendimiento Ecuabonsai tomar decisiones informadas.
- Mejora de la eficiencia y la efectividad del emprendimiento, y también ofrece a los clientes una experiencia de compra más cómoda y conveniente.

### **4.2 Recomendaciones.**

- Optimización para dispositivos móviles, debido a que cada vez más personas acceden a internet a través de dispositivos móviles, por lo que es importante optimizar el sistema de ventas en línea para su uso en estos dispositivos.
- Integrar el sistema de ventas en línea con otras plataformas, como redes sociales y servicios de envío, para mejorar la eficiencia y la conveniencia para los clientes.
- Adoptar un enfoque constante y comprometido con la mejora continua para lograr un desempeño óptimo del equipo.

