

MANUALES TÉCNICOS DE LA APLICACIÓN



Estructura de contenidos

Pág.

Intro

ducción	3
Mapa de contenido	4
1. Manual del usuario	5
1.1 Componentes	5
1.1.1 Alcance	6
1.1.2 Requisitos de conocimiento para manejar el sistema	6
1.1.3 Requerimientos de hardware y software del equipo cliente	6
1.1.4 Política de seguridad	7
1.1.5 Ingreso al sistema.....	7
1.1.6 Mapa de navegación	7
1.1.7 Descripción de las funciones del sistema	7
1.1.8 Información sobre la mesa de ayuda.....	7
1.1.9 Descripción de los mensajes de error.....	8
2. Manual técnico	8
2.1 Componentes	9
2.1.1 Alcance.....	9
2.1.2 Descripción de los procesos	9
2.1.3 Diagramas UML.....	9
2.1.4 Administración de usuarios.....	10
2.1.5 Modelo relacional de la base de datos.....	10
2.1.6 Descripción de la plataforma.....	10
2.1.7 Documentación del código fuente.....	10
2.1.8 Descripción de los acuerdos de niveles de servicios o ANS	10
3. Recomendaciones para realizar informes técnicos.....	11
4. Ejercicio de creación de un manual de usuario	12
5. Ejercicio de creación de un manual técnico.....	26
Glosario	31
Bibliografía.....	32
Control del documento.....	33

1.1.1 Alcance

Este proyecto tiene como objetivo desarrollar un sistema de gestión integral que abarcará la automatización de la solicitud de créditos por parte de los clientes para la compra de muebles, electrodomésticos y otros productos. El sistema permitirá establecer planes de pago a cuotas de manera eficiente, llevará un registro de las transacciones financieras y proporcionará a la empresa las herramientas necesarias para gestionar y controlar estos créditos de manera efectiva.

El alcance del proyecto también incluye la capacitación del personal en la utilización del sistema y la implementación de medidas de seguridad para proteger la información financiera de los clientes.

1.1.2 Requisitos de conocimiento para manejar el sistema

1. **Conocimiento Técnico Básico:** Deben comprender conceptos básicos de tecnología, como navegación en la web, uso de computadoras y dispositivos móviles, y la capacidad de realizar tareas en línea.
2. **Operación del Sistema:** Deben conocer las funciones y características específicas del sistema de facturación electrónica, incluyendo cómo ingresar y administrar datos, generar facturas, y realizar un seguimiento de las transacciones.
3. **Seguridad de Datos:** Deben estar al tanto de las mejores prácticas de seguridad de datos, incluyendo la importancia de contraseñas seguras y la protección de la información confidencial.
4. **Resolución de Problemas:** Deben saber cómo solucionar problemas comunes que puedan surgir durante el uso del sistema, como errores de entrada de datos o problemas de acceso.

Mantenimiento del Sistema: Deben entender cómo realizar tareas de mantenimiento básicas, como actualizaciones de software o resolución de problemas técnicos menores.

1.1.3 Requerimientos de hardware y software del equipo

Hardware:

1. Computador con mínimo de 4GB en RAM
2. Espacio libre en disco de 1 GB
3. Mínimo un procesador Celeron n4020

Requisitos de software:

1. Sistema operativo Windows 10 o posterior, android o iOS

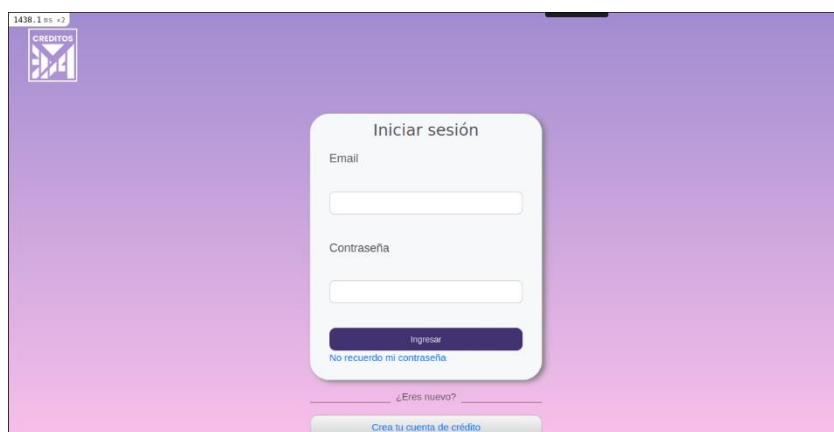
1.1.4 Política de seguridad

Este aplicativo solo será manejado por el administrador del almacén.


Los controles de seguridad de las aplicaciones son técnicas que mejoran la seguridad de la aplicación a nivel de codificación para que sea menos vulnerable a las amenazas.

1.1.5 Ingreso al sistema:

Para ingresar al sistema como clientes o administrador debemos iniciar sesión y si no cuentas con una cuenta, debemos registrarnos como administrador, una vez teniendo una cuenta creada ya puedes iniciar sesión.



1438.1 05 v2



Iniciar sesión

Email

Contraseña

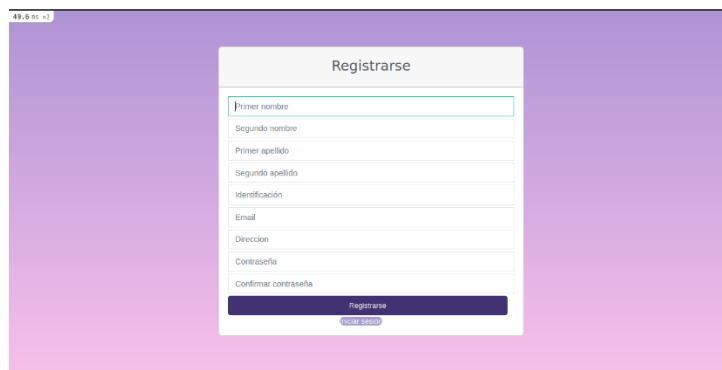
Ingresar

[No recuerdo mi contraseña](#)

¿Eres nuevo?

[Crea tu cuenta de crédito](#)

Aquí podrás diligenciar tus datos para poder registrarte, el correo para ser administrador ejemplo (pepito@admin.com) o te



49.6 05 v2

Registrarse

Primer nombre

Segundo nombre

Primer apellido

Segundo apellido

Identificación

Email

Dirección

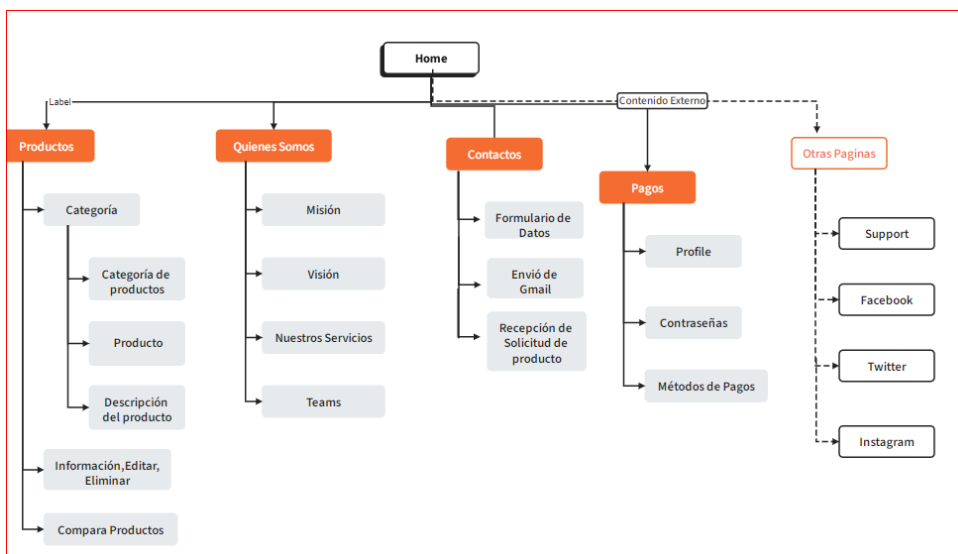
Contraseña

Confirmar contraseña

Registrarse

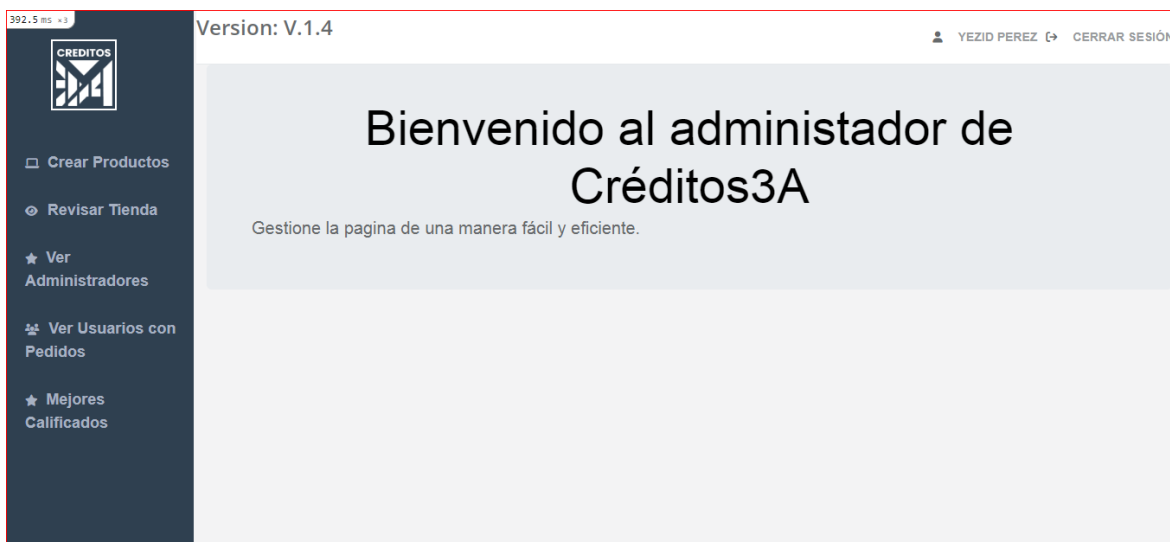
[Volver sesión](#)

1.1.6 Mapa de navegación

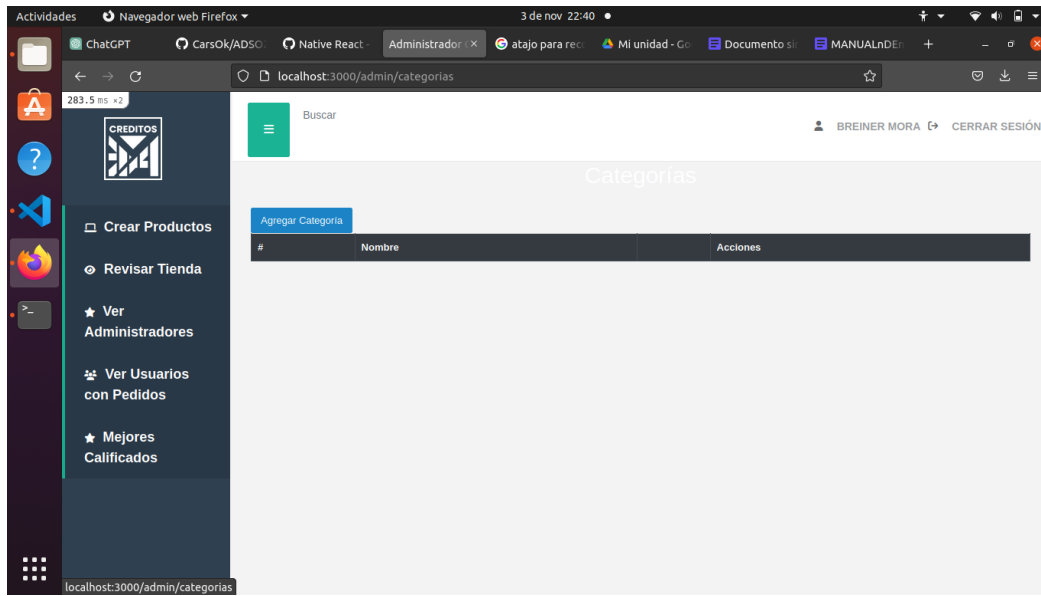


1.1.7 Descripción de las funciones del sistema

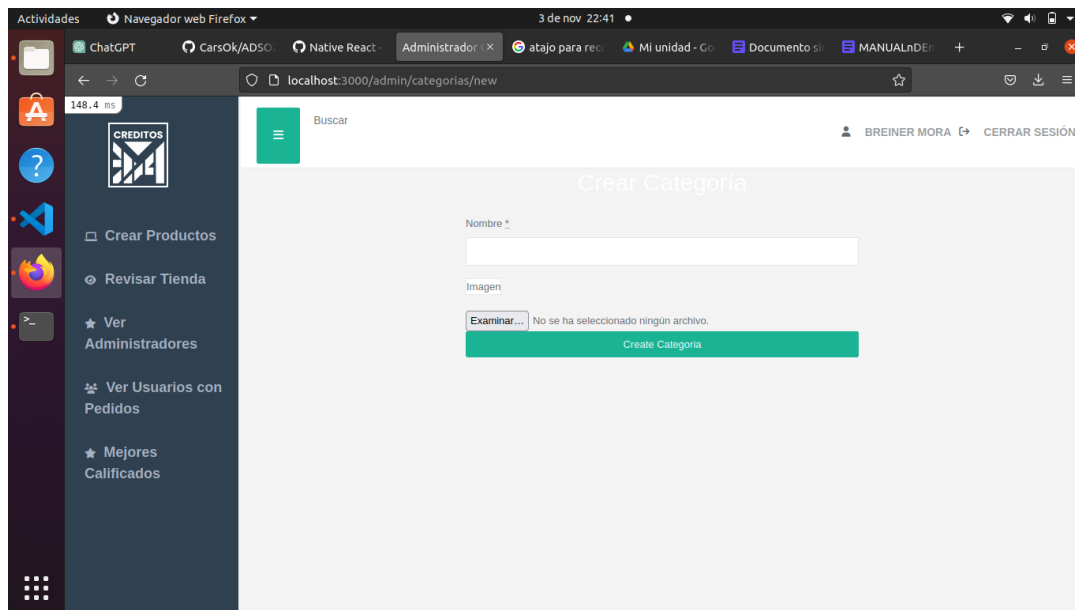
Esta es la sección en que los administradores van a operar la página web, podrán crear productos, ver los administradores que la operan, ver los pedidos y las calificaciones.



para crear los productos primero se debería crear las categorías que estarían asociadas a los diferentes productos como se ve en pantalla botón azul (agregar categoría).



Una vez demos click en el botón “agregar categoría” se abrirá esta sesión para agregar la categoría deseada, también podemos añadir una foto que identifique la categoría.



Cuando se crea la categoría saldrá la información y herramientas que son editar o eliminar tal y como se muestra en la pantalla, podemos editar la categoría o eliminarla.

Información de la Categoría

#	Nombre	Acciones
1	Closet	Editar Borrar

Productos de la Categoría

#	Nombre	Descripción
---	--------	-------------

[Crear productos](#)
[Ver todas las Categorías](#)

editar categoría:

Editar categoría

Crear Categoría

Nombre *

Imagen

[Seleccionar archivo](#)
Sin archivos seleccionados

[Update Categoría](#)

[Ver registro](#) | [Regresar](#)

para añadir productos le damos al botón crear productos.

Información de la Categoría

#	Nombre	Acciones
1	Closet	Editar Borrar

Productos de la Categoría

#	Nombre	Descripción
---	--------	-------------

[Crear productos](#)
[Ver todas las Categorías](#)

saldrá el formulario para ingresar la información del producto a crear.

Crear producto

Nombre *

TV

Descripción *

de ultima tecnologia

Precio *

50000000

Categoría *

Closet ✓

Avatar *


Seleccionar archivo mqdefault.jpg

Guardar

saldrá un apartado donde se mostrará la lista de todos los productos creados de una categoría específica.

Lista de Productos

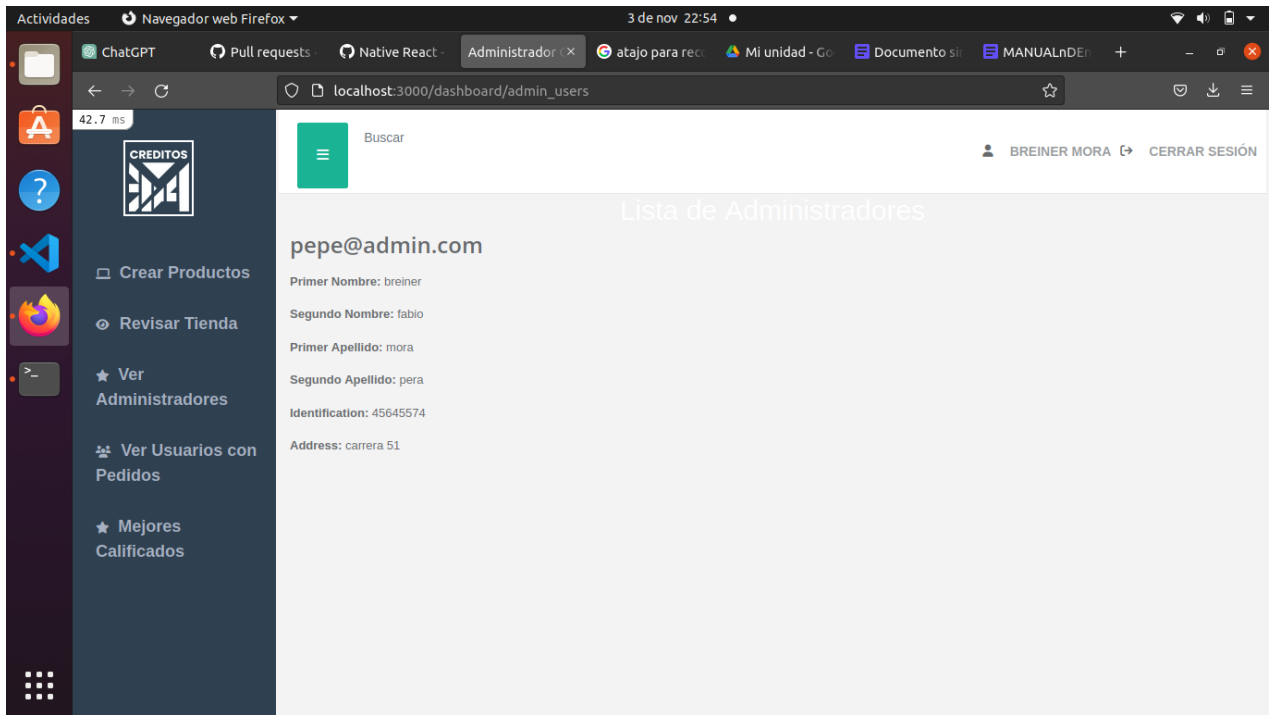
Crear Producto



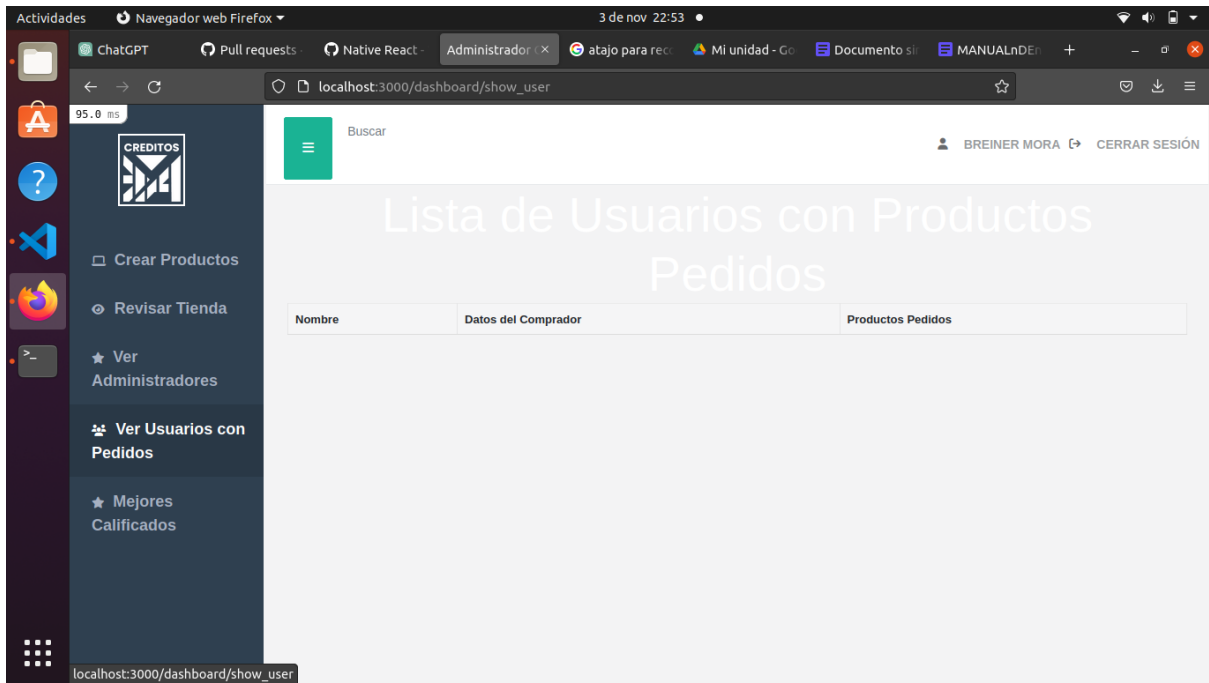
Cama doble
Alcoba Doble Denzel Gris, Nieve
Precio: \$2,500,000.00

Ver detalles

En este apartado se observarán todos los usuarios administradores registrados





En esta ventana se renderizan todos los usuarios que han hecho algún pedido, se mostrará el nombre del usuario y que producto pidió.



ya que los productos se van a clasificar, en este apartado se observarán que productos tienen las mejores calificaciones.

MEJORES PRODUCTOS CALIFICADOS

 <p>Cama doble</p> <p>Promedio de calificación en: ★ 4.0</p> <p>Descripción: Alcoba Doble Denzel Gris, Nieve</p> <p>Precio: \$2,500,000.00</p>	 <p>pc</p> <p>Promedio de calificación en: ★ 5.0</p> <p>Descripción: ryzen 5</p> <p>Precio: \$300,000.00</p>	 <p>macbook</p> <p>Promedio de calificación en: ★ 2.6666666666666665</p> <p>Descripción: de ultima tecnologia</p> <p>Precio: \$10,000,000.00</p>
--	--	--

Panel de administrador:

Información	Editar	Eliminar
Información	Editar	Eliminar
Información	Editar	Eliminar

En los botones de “editar y eliminar” solo el administrador tiene los permisos necesarios para editar o eliminar algún producto creado.

Editar	Eliminar
--------	----------

“Eliminar”:

El botón eliminar permite poder borrar permanente todo producto no deseado que el usuario adquirió.

Y el editar permite hacerle cambios en caso de error de digitación de datos.

Ver Productos

Este botón de “productos” tiene la función de al darle clic en productos muestra una vista del producto que adquirió.

A continuación, veremos las funciones del aplicativo:

El usuario una vez registrado e iniciado sesión puede navegar libremente por la página web.



En la opción de “nuevo” podremos registrar un usuario nuevo, pedirá datos personales e importantes y podremos especificar qué tipo de rol o manejo pueda tener en el aplicativo

Información sobre la mesa de ayuda:

Si se presentan problemas o alguna clase de error con el aplicativo, se puede contactar con nosotros a través de nuestros correos o número de celular:

-Correo: yezidperez@gmail.com / jesusvillanueva@gmail.com

-Celular: 3127338944 / 3218265527

2. Manual técnico

2.1.7 Alcance

Este proyecto tiene como objetivo desarrollar un sistema de gestión integral que abarcará la automatización de la solicitud de créditos por parte de los clientes para la compra de muebles, electrodomésticos y otros productos. El sistema permitirá establecer planes de pago a cuotas de manera eficiente, llevará un registro de las transacciones financieras y proporcionará a la empresa las herramientas necesarias para gestionar y controlar estos créditos de manera efectiva. El alcance del proyecto también incluye la capacitación del personal en la utilización del sistema y la implementación de medidas de seguridad para proteger la información financiera de los clientes.

Nombre del proyecto:

Sistema informático digital basado en Ruby y MYSQL para Administrar la solicitud y aprobación de préstamos en la empresa créditos 3ª.

OBJETIVO GENERAL:

Implementar el Sistema informático digital basado en Ruby y MYSQL para administrar la solicitud y aprobación de préstamos en la empresa créditos 3ª

OBJETIVOS ESPECÍFICOS:

1. Desarrollar el Sistema de Solicitud en Ruby on Rails y MySQL:

- Acción: Diseñar, desarrollar e implementar la parte del sistema de solicitud de préstamos utilizando el marco de Ruby on Rails y una base de datos MySQL.

- Plazo: Completar el desarrollo en los primeros 4 meses del proyecto.

2. Diseñar y Crear la Base de Datos de Solicitudes de Préstamos en MySQL:

- Acción: Diseñar y crear una base de datos en MySQL que almacene y administre las solicitudes de préstamos, siguiendo las mejores prácticas de modelado de datos.
- Plazo: Configurar la base de datos dentro de los primeros 2 meses.

3. Integrar el Sistema de Aprobación de Préstamos utilizando Ruby y MySQL:

- Acción: Desarrollar un sistema de aprobación de préstamos basado en Ruby que utilice datos de solicitudes y criterios de aprobación definidos, y almacene los resultados en la base de datos MySQL.
- Plazo: Integrar el sistema de aprobación en los primeros 6 meses.

4. Realizar Pruebas y Depuración del Sistema utilizando Metodología Ágil:

- Acción: Realizar pruebas exhaustivas del sistema utilizando una metodología ágil, identificar y corregir errores de manera iterativa.
- Plazo: Completar las pruebas y depuración en un período de 3 meses.

5. Capacitar al Personal en el Uso del Sistema:

- Acción: Proporcionar capacitación al personal de "Créditos 3ª" en la operación y administración del sistema basado en Ruby on Rails y MySQL.
- Plazo: Realizar la capacitación en un plazo de 2 meses.

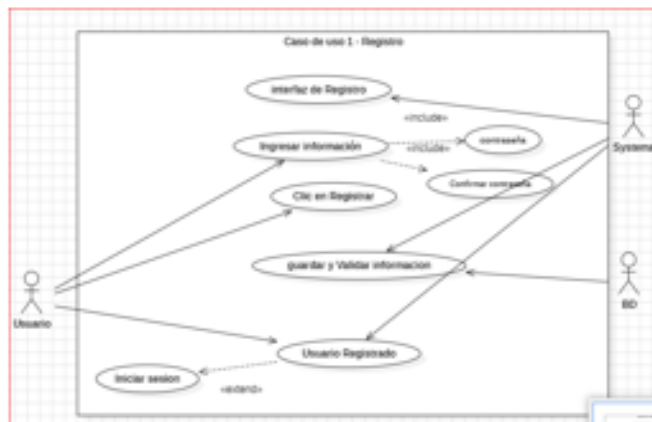
6. Lanzar el Sistema en Producción:

- Acción: Implementar el sistema en producción, permitiendo que los clientes realicen solicitudes de préstamo a través de la plataforma construida en Ruby y MySQL.
- Plazo: Lanzar el sistema en producción dentro de los primeros 7 meses.

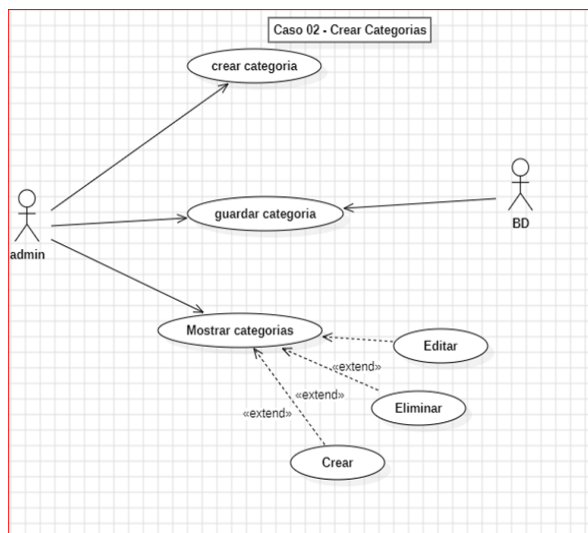
2.1.3 Diagramas UML

Diagramas de Caso de Uso:

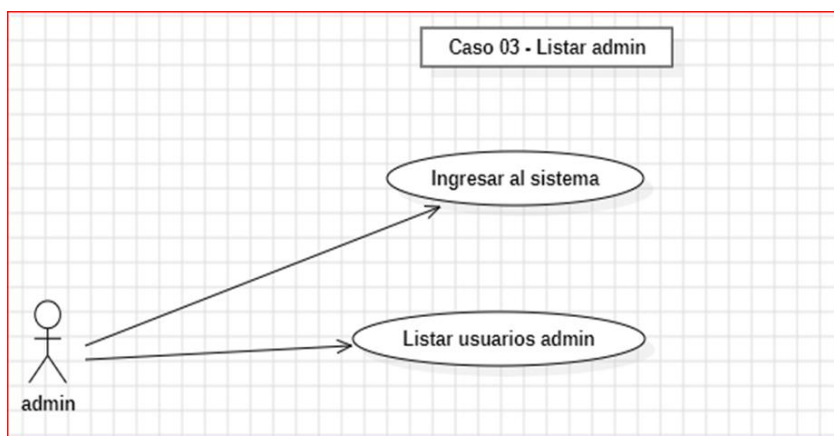
CASO DE USO:	001
TIPO DE CASO DE USO:	Primario
ACTORES:	Usuario
Descripción:	Registro de usuario
Precondiciones:	El usuario no está registrado en el sistema.
ACCIONES NORMALES DE LOS EVENTOS	
ACCIÓN DE ACTORES	RESPUESTA DEL SISTEMA
1 El usuario abre la interfaz de registro.	El sistema solicita iniciar sesión en el sistema.
2 El usuario ingresa su información personal, como su nombre, dirección de correo electrónico y contraseña	2
3 El usuario hace clic en el botón "Registrar".	El sistema guarda la información del usuario en la base de datos.
4 El usuario hace clic en el enlace de confirmación en el correo electrónico.	El sistema envía un correo electrónico de confirmación al usuario.
Poscondición	
<ul style="list-style-type: none"> El usuario está registrado en el sistema. El usuario puede iniciar sesión en el sistema. 	
DECISIONES ALTERNATIVAS	
Línea 1	<ul style="list-style-type: none"> Si el usuario ingresa información incorrecta, el sistema muestra un mensaje de error. Si el sistema no puede guardar la información del usuario en la base de datos, el sistema muestra un mensaje de error.



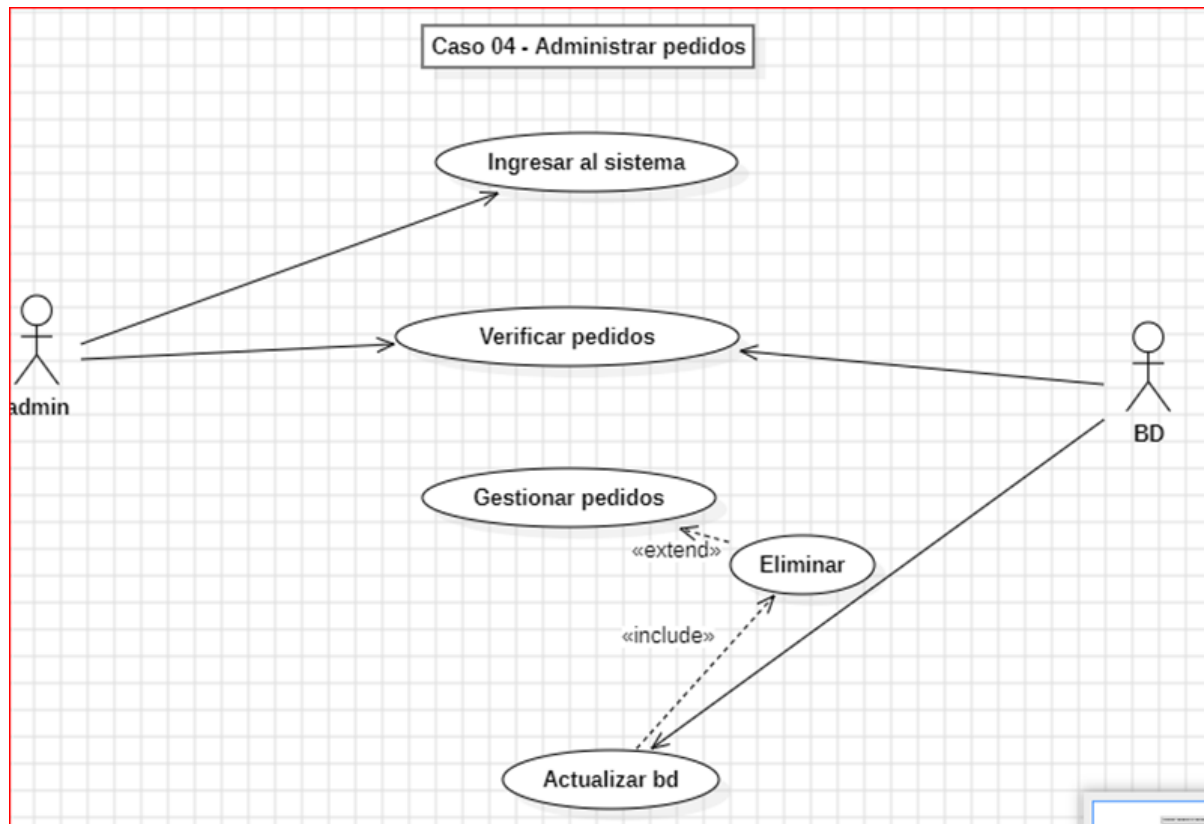
CASO DE USO:		002
TIPO DE CASO DE USO:		Primario
ACTORES:		Administrador
Descripción:		Crear categoría
Precondiciones:		El administrador tiene permiso para crear categorías.
ACCIONES NORMALES DE LOS EVENTOS		
ACCIÓN DE ACTORES		RESPUESTA DEL SISTEMA
1	El administrador accede a la interfaz de administración de categorías.	El sistema guarda la información del usuario en la base de datos.
2	El administrador ingresa el nombre de la nueva categoría.	2
3	El administrador selecciona el tipo de categoría.	
4	El administrador hace clic en el botón "Crear".	<p>Si la información es válida, el sistema crea una nueva categoría en la base de datos.</p> <p>El sistema muestra un mensaje de confirmación al administrador</p>
Poscondición		
<ul style="list-style-type: none"> Se crea una nueva categoría en la base de datos. 		
DECISIONES ALTERNATIVAS		
Línea 1	<ul style="list-style-type: none"> Si la información ingresada por el administrador no es válida, el sistema muestra un mensaje de error. El administrador puede corregir la información y volver a intentarlo. 	

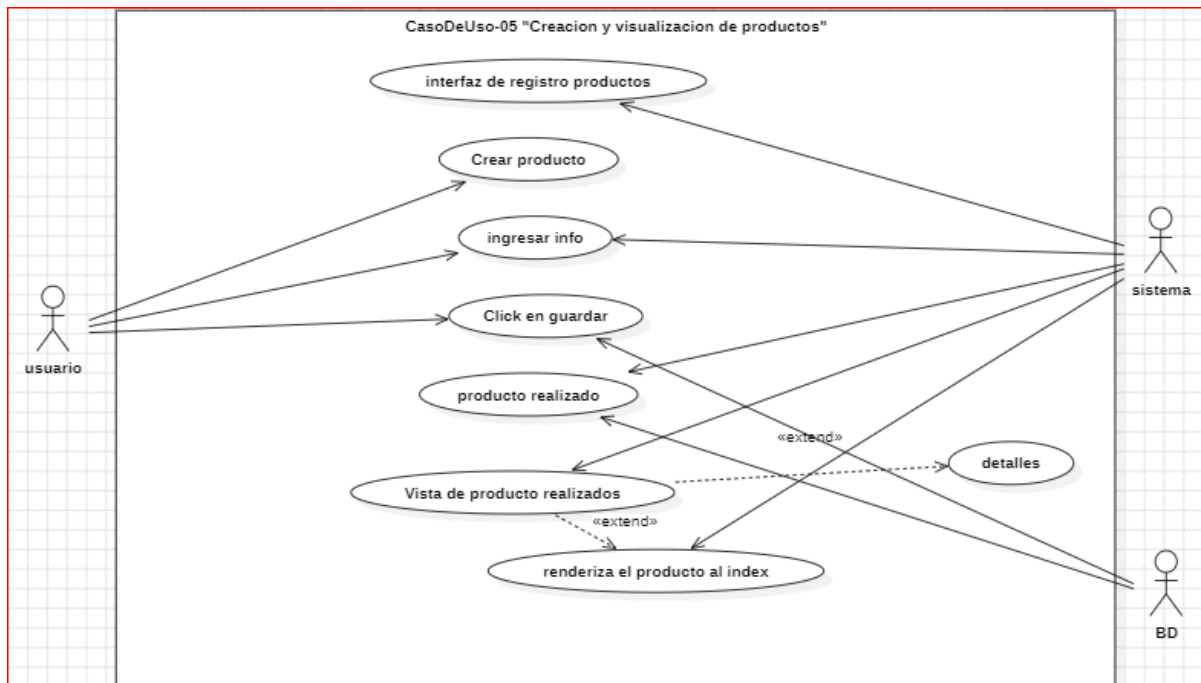


CASO DE USO:		003	
TIPO DE CASO DE USO:		Primario	
ACTORES:		Administrador	
Descripción:		Listar usuarios admin	
Precondiciones:		El administrador está autenticado en el sistema.	
ACCIONES NORMALES DE LOS EVENTOS			
ACCIÓN DE ACTORES		RESPUESTA DEL SISTEMA	
1	El administrador accede a la interfaz de administración de usuarios.	El sistema guarda la información del usuario en la base de datos.	
2	El administrador hace clic en el enlace "Listar usuarios admin".	2	El sistema muestra una lista de todos los usuarios admin en la base de datos.
3	El administrador puede ver la información de cada usuario admin, como el nombre, el apellido, el correo electrónico y el estado de la cuenta.		El sistema debe proporcionar una interfaz de administración de usuarios que permita al administrador ver una lista de todos los usuarios admin.
Poscondición			
<ul style="list-style-type: none">El administrador tiene una lista de todos los usuarios admin en la base de datos			
DECISIONES ALTERNATIVAS			
Línea 1	<ul style="list-style-type: none">La lista debe incluir la información de cada usuario admin, como el nombre, el apellido, el correo electrónico y el estado de la cuenta.		

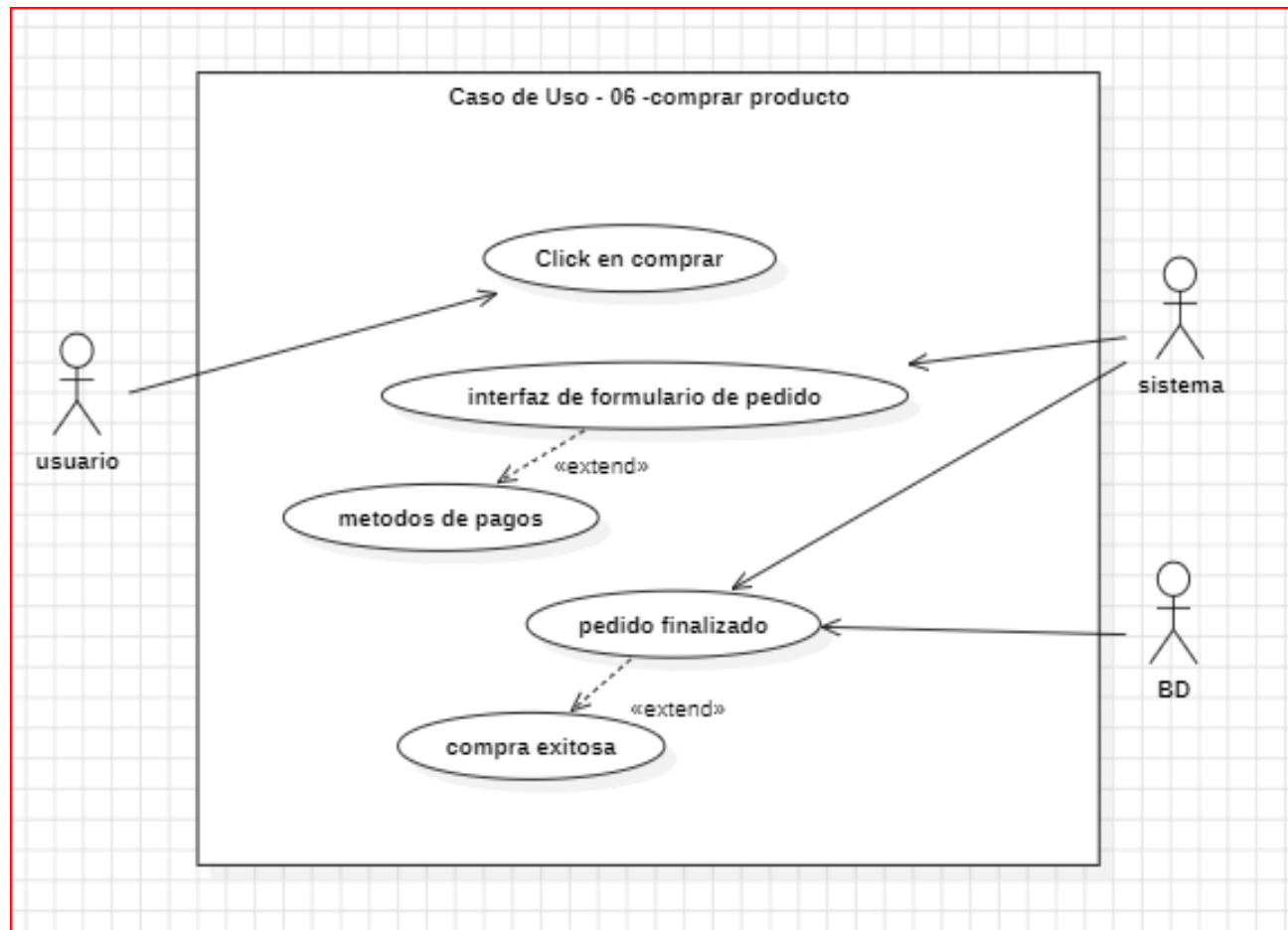


CASO DE USO:		004	
TIPO DE CASO DE USO:		Primario	
ACTORES:		Administrador	
Descripción:		Administrar pedidos	
Precondiciones:		El administrador del sistema quiere administrar los pedidos realizados por los usuarios.	
ACCIONES NORMALES DE LOS EVENTOS			
ACCIÓN DE ACTORES		RESPUESTA DEL SISTEMA	
1	El administrador accede a la página de administración de pedidos.	El sistema muestra una lista de todos los pedidos realizados.	
2	El administrador puede ver la información de cada pedido, como el número de pedido, la fecha de realización, el estado del pedido, el usuario que realizó el pedido y los productos incluidos en el pedido.	2	
3	El administrador puede realizar las siguientes acciones sobre un pedido:		
4	Ver detalles: El administrador puede ver los detalles completos de un pedido. Editar estado: El administrador puede cambiar el estado de un pedido.		Si la información es válida, el sistema crea una nueva categoría en la base de datos. El sistema muestra un mensaje de confirmación al administrador Eliminar: El administrador puede eliminar un pedido.
Poscondición			
. El administrador debe estar autenticado en el sistema.			
DECISIONES ALTERNATIVAS			
Línea 1	Si no hay ningún pedido realizado, el sistema muestra interfaz vacía.		





CASO DE USO:		005	
TIPO DE CASO DE USO:		Primario	
ACTORES:		Usuario	
Descripción:		Crear y visualizar productos	
Precondiciones:		El usuario debe estar autenticado en el sistema.	
ACCIONES NORMALES DE LOS EVENTOS			
ACCIÓN DE ACTORES		RESPUESTA DEL SISTEMA	
1	El usuario accede a la página de creación de productos.	El sistema guarda la información del usuario en la base de datos.	
2	El usuario ingresa la información del producto, como el nombre, la descripción, el precio, la imagen y la categoría.	2	
3	El usuario hace clic en el botón "Crear".		El sistema crea el producto y lo agrega a la base de datos.
4	El usuario puede ver la lista de todos los productos creados.		El sistema muestra la lista de productos creados.
Poscondición			
• Si el usuario no está autenticado, el sistema lo redirige a la página de inicio de sesión.			
DECISIONES ALTERNATIVAS			
Línea 1	• Si el usuario ingresa información incorrecta, el sistema muestra un mensaje de error. • Si el nombre del producto ya existe, el sistema muestra un mensaje de error.		



CASO DE USO:		006	
TIPO DE CASO DE USO:		Primario	
ACTORES:		Usuario	
Descripción:		Compra de un producto	
Precondiciones:		El usuario está registrado en el sistema y ha añadido un producto	
ACCIONES NORMALES DE LOS EVENTOS			
ACCIÓN DE ACTORES		RESPUESTA DEL SISTEMA	
1	El usuario hace clic en el botón "Comprar".	El sistema muestra el formulario de pedido.	
2	El usuario ingresa la información de envío y pago.	2	
3	El usuario hace clic en el botón "Pagar".		El sistema procesa el pago. El sistema envía un correo electrónico de confirmación al usuario.
Poscondición			
<ul style="list-style-type: none">El pedido se ha realizado correctamente.El producto se ha enviado al usuario.			
DECISIONES ALTERNATIVAS			
Línea 1	<ul style="list-style-type: none">Si el usuario no ha añadido el producto al carrito de la compra, el sistema mostrará un mensaje de error.Si el usuario ingresa información incorrecta en el formulario de pedido, el sistema mostrará un mensaje de error.Si el sistema no puede procesar el pago, el sistema mostrará un mensaje de error.		

Modelo relacional de la base de datos.

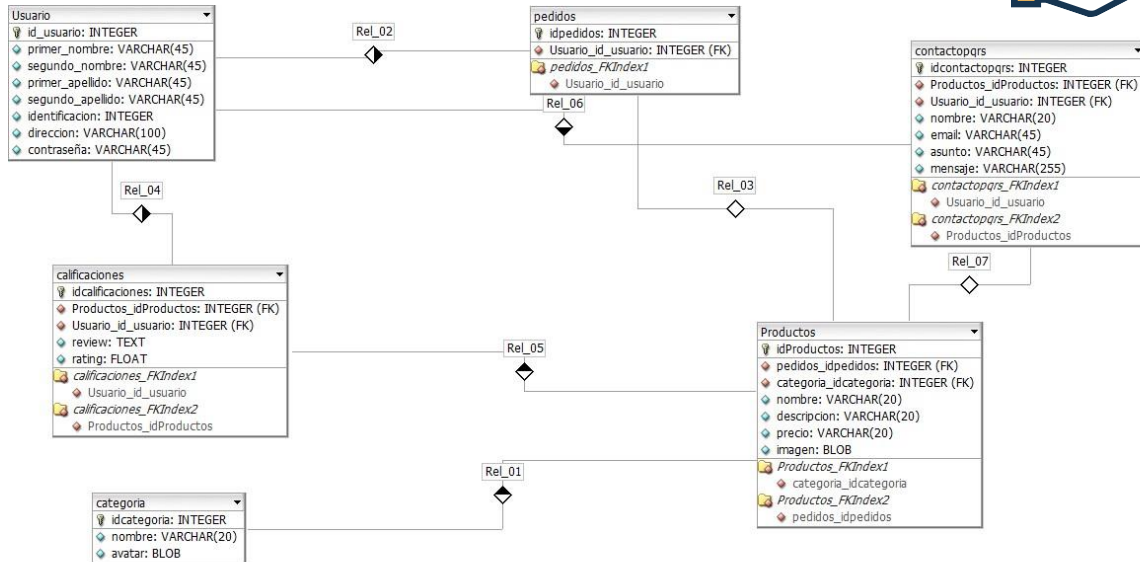
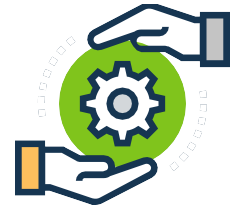
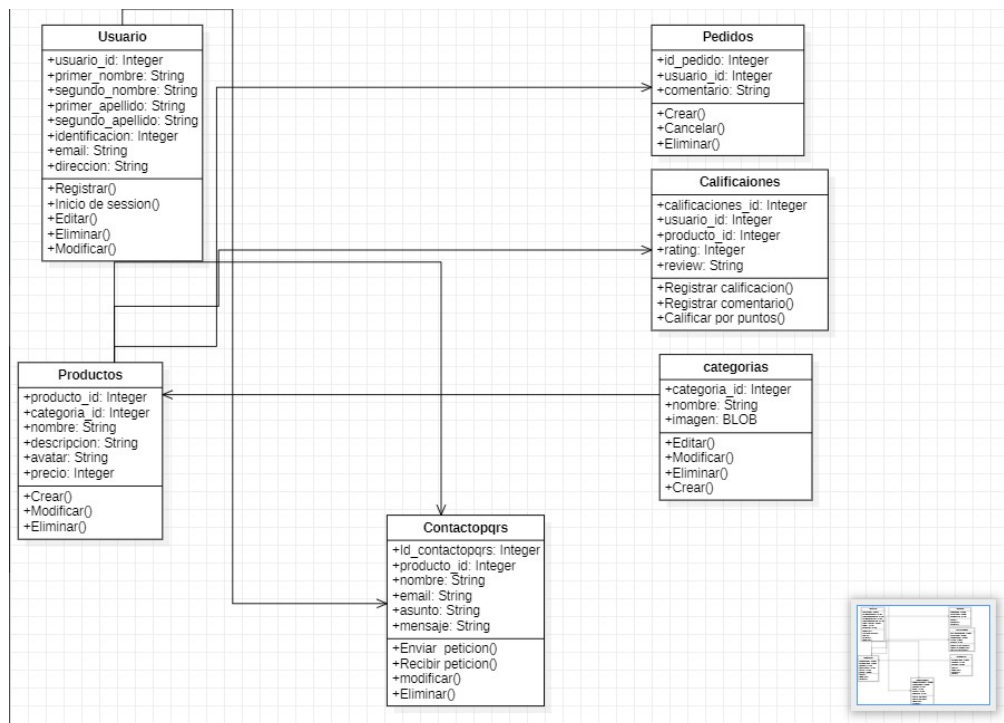
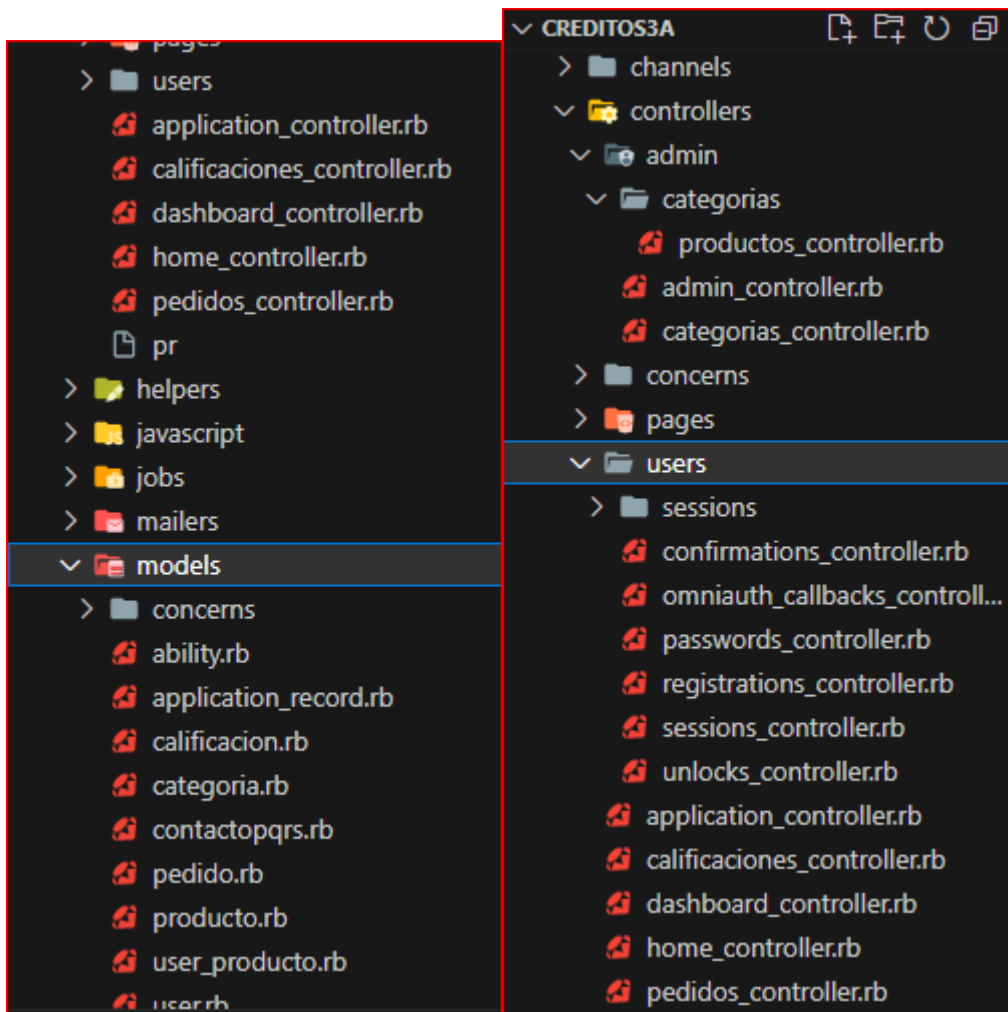


Diagrama de clases

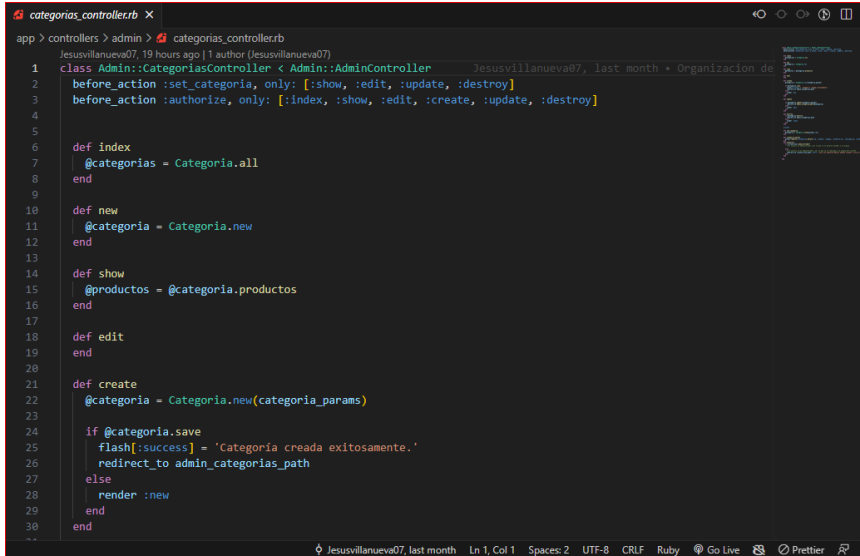


Documentación del código fuente

Este proyecto fue realizado con Patrones de arquitectura de software: MVC (Modelo, vista, controlador).



Controlador de Categoría:



```
1 class Admin::CategoriasController < Admin::AdminController
2   before_action :set_categoria, only: [:show, :edit, :update, :destroy]
3   before_action :authorize, only: [:index, :show, :edit, :create, :update, :destroy]
4
5
6   def index
7     @categorias = Categoria.all
8   end
9
10  def new
11    @categoria = Categoria.new
12  end
13
14  def show
15    @productos = @categoria.productos
16  end
17
18  def edit
19  end
20
21  def create
22    @categoria = Categoria.new(categoria_params)
23
24    if @categoria.save
25      flash[:success] = 'Categoria creada exitosamente.'
26      redirect_to admin_categorias_path
27    else
28      render :new
29    end
30  end
31 end
```

El código tiene las siguientes acciones:

- **index:** Esta acción muestra una lista de todas las categorías.
- **new:** Esta acción muestra un formulario para crear una nueva categoría.
- **create:** Esta acción crea una nueva categoría a partir de los datos proporcionados en el formulario.
- **show:** Esta acción muestra los detalles de una categoría específica.
- **edit.:** Esta acción muestra un formulario para editar una categoría existente.
- **update:** Esta acción actualiza una categoría existente a partir de los datos proporcionados en el formulario.
- **destroy:** Esta acción elimina una categoría existente.

Controlador Admin:

```
app > controllers > admin > admin_controller.rb
Jesusvillanueva07, 6 days ago | 1 author (Jesusvillanueva07)
1 class Admin::AdminController < ApplicationController
2   layout 'admin'
3   before_action :authenticate_user!
4
5   private
6
7   def require_admin
8     unless current_user
9       flash.now[:alert] = "Acceso Denegado. Debes estar conectado para acceder a esta página."
10      render 'denied_access', status: :forbidden
11    end
12  end
13 end
14
```

El código en la imagen es un controlador de Rails que se utiliza para controlar el acceso a una página de administración. El código tiene dos partes principales:

- La línea 1-3: class Admin::AdminController < ApplicationController

Esta línea define la clase Admin::AdminController como una subclase de la clase ApplicationController. Esto significa que Admin::AdminController hereda todos los métodos de ApplicationController.

- layout 'admin'

Esta línea establece el layout que se utilizará para la página de administración. En este caso, el layout se llama admin.

- before_action:authenticate_user!

Esta línea establece el método before_action para el controlador. El método before_action se ejecuta antes de que se ejecute cualquier otra acción en el controlador. En este caso, el método authenticate_user! se llama para verificar si el usuario actual está conectado.

En las líneas 6-12, define el método require_admin. Este método verifica si el usuario actual está conectado. Si el usuario no está conectado, el método establece un mensaje de alerta en el flash y redirige al usuario a una página de acceso denegado.

Modelo productos:

Como puede ver, nuestra validación nos permite saber que Producto no es válido sin un nombre, descripción, precio, categoría, avatar atributo.

Las validaciones a nivel de modelo son la mejor manera de garantizar que solo se guarden datos válidos en su base de datos.

Antes de guardar un objeto Active Record, Rails ejecuta sus validaciones. Si estas validaciones producen algún error, Rails no guarda el objeto.

```
class Producto < ApplicationRecord
  has_many :pedidos
  has_many :calificaciones
  #has_many :user_products
  #has_many :users, through: :user_products

  belongs_to :categoria
  mount_uploader :avatar, AvatarUploader
  validates :nombre, presence: true
  validates :descripcion, presence: true
  validates :precio, presence: true
  validates :categoria, presence: true
  validates :avatar, presence: true
  def calificaciones
    Calificacion.where(producto_id: id)
  end

  def rating_average
    if calificaciones.any?
      total_ratings = calificaciones.sum(:rating)
      total_ratings / calificaciones.count.to_f
    else
      0
    end
  end
end
```

Modelo Usuario:

```
class User < ApplicationRecord
  has_many :pedidos
  has_many :calificaciones
  #has_many :user_products
  #has_many :productos, through: :user_products

  before_create :set_default_administrador

  validates :first_name, presence: true
  validates :second_name, presence: true
  validates :last_name_1, presence: true
  validates :last_name_2, presence: true
  validates :identification, presence: true
  validates :address, presence: true
  has_one_attached :imagen
  attribute :administrador, :boolean, default: false

  # Include default devise modules. Others available are:
  # :confirmable, :lockable, :timeoutable, :trackable and :omniauthable
  devise :database_authenticatable, :registerable,
         :recoverable, :rememberable, :validatable

  private

  def set_default_administrador
    self.administrador = (self.email.split("@").last == "admin.com")
  end
end
```

La clase también tiene los siguientes métodos:

- **has_many:** Este método define una relación de muchos a muchos entre la clase User y la clase Pedido. Esto significa que un usuario puede tener muchos pedidos, y un pedido puede tener muchos usuarios.
- **has_many:** Este método define una relación de muchos a muchos entre la clase User y la clase Calificación. Esto significa que un usuario puede tener muchas calificaciones, y una calificación puede tener muchos usuarios.
- **before_create:** Este método se ejecuta antes de que se cree un nuevo usuario. En este caso, el método establece la propiedad administrador en false si el correo electrónico del usuario termina en admin.com.
- **validates:** Este método se utiliza para validar los datos del usuario. En este caso, el método verifica que las propiedades nombre, apellido, email, identificación y dirección no estén vacías.

Estructura de Directorios:

Directorio	Descripción
Controlador	Contiene la clase Controlador.
Modelo	Contiene las clases que están asociadas con la base de datos.
Vista	Contiene los directorios que implementan la vista de la aplicación.
Vista/css	Contiene la definición de los archivos CSS o estilos en cascada.
Vista/html.erb	Contiene los formularios de la aplicación web.
Vista/imágenes	Contiene las imágenes usadas en la aplicación.
Vista/jquery	Contiene la librería JQuery y JQuery UI, Bootstrap.
Index.html.erb	Es el arrancador de la aplicación.

MVC: Modelo, Vista, controlador.

Mesa de ayuda: personal dedicado a dar soporte funcional y técnico a los usuarios de una aplicación.

Plugin: componente de software que se puede “enchufar” o integrar a otro sistema obteniéndose un software ampliado o mejorado.

UML: acrónimo de Unifed Modeling Language. Lenguaje de modelado para procesos de diseño de sistemas.

Active-record: Active Record proporciona la capa objeto-relacional que sigue rigurosamente el estándar ORM: Tablas en Clases, Registros en Objetos, y Campos en Atributos.

Diagrama Entidad-Relación (DER): se usa como una herramienta de diseño para representar gráficamente a: los datos, las relaciones que existen entre ellos y algunas reglas de integridad.

Modelo entidad relación o MER: es un tipo de diagrama de flujo que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema.

PostgreSQL: es un sistema o motor de bases de datos compatible con los servicios de OVHcloud y la mayoría de las herramientas más populares del mercado. Es compatible con diversos modelos de datos para crear aplicaciones orientadas a objetos, potentes y escalables.

Ruby on Rails: es un framework o marco de creación de aplicaciones de código abierto, que funciona con Ruby. Se trata de un conjunto de bibliotecas que posibilitan el desarrollo de aplicaciones web usando Ruby, ya que está escrito en este mismo lenguaje.