

MANUAL TÉCNICO

****

**Versión del sistema:** 1.0

**Fecha de elaboración:** 05/07/2017

**Área de elaboración:** Aula UAS

**Elaborado por:** Alejandro Onofre Cornejo

# **INTRODUCCIÓN**

En el siguiente manual se explican algunos de los aspectos más importantes en el diseño de del proyecto de un sitio web para la empresa “Eddy Burguer”, Dentro de este manual manual, se documentaron diagramas que le darán a conocer la forma lógica del funcionamiento del sistema que ayudará a su mejora y aprovechamiento del mismo.

Se recomienda consultar el manual de uso de las siguientes tecnologías web como: PHP, SGBD MySQL, CodeIgneter, para tener un mejor conocimiento de uso acerca de la aplicación web “Eddy Burguer”.

El presente proyecto está seguido de diferentes normas o estándares que generaron documentos durante el desarrollo del mismo.

# **CONTENIDO**

[INTRODUCCIÓN 2](#_Toc487477476)

[CONTENIDO 3](#_Toc487477477)

[INTRODUCCIÓN 4](#_Toc487477478)

[OBJETIVOS GENERALES Y ESPECÍFICOS DEL SISTEMA 5](#_Toc487477479)

[NORMAS, POLÍTICAS Y PROCEDIMIENTOS 7](#_Toc487477480)

[FUNDAMENTACIÓN DE LA TECNOLOGÍA UTILIZADA 10](#_Toc487477481)

[DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA 12](#_Toc487477482)

[ESPECIFICACIÓN DE REQUISITOS 13](#_Toc487477483)

[VISTA FUNCIONAL 18](#_Toc487477484)

[VISTA LÓGICA 36](#_Toc487477485)

[MODELO LÓGICO DE DATOS 49](#_Toc487477486)

[DISEÑO DE PANTALLAS 51](#_Toc487477487)

[DESCRIPCIÓN DE CAMPOS REQUERIDOS POR PANTALLA 52](#_Toc487477488)

[VISTA DE IMPLEMENTACIÓN 58](#_Toc487477489)

[VISTA DE DESPLIEGUE 58](#_Toc487477490)

[GLOSARIO DE TÉRMINOS 60](#_Toc487477491)

## **INTRODUCCIÓN**

El sitio web le brinda mayor publicidad a la organización, además, le permite a la empresa entrar al mundo de las nuevas tecnologías como lo son las ventas a través de internet (online), dicho sitio web cuenta con las secciones de inicio, conócenos, instalaciones, productos, contacto, ofertas, FAQ (preguntas frecuentes), galería y login.

El desarrollo de este sitio web fue para la empresa “Eddy Burguer”, la cual está localizada en la avenida Hidalgo esquina con Aquiles Serdán, en la ciudad de Acámbaro Guanajuato; la organización tiene el giro de la venta de alimentos y bebidas para personas de cualquier edad.

## **OBJETIVOS GENERALES Y ESPECÍFICOS DEL SISTEMA**

**Objetivo** **general**

Desarrollar una aplicación web en un periodo de tres meses, durante los meses mayo-agosto de 2017, utilizando el lenguaje de programación PHP y el SGBD MySql para la funcionalidad, y el uso de las tecnologías HTML, CSS y JavaScript, para crear el diseño de la aplicación y de esta forma, mejorar la administración de productos e incrementar las ventas y reconocimiento de la empresa Eddy Burguer hasta en un 10%, en el próximo año.

**Objetivos específicos**

* Desarrollar una modulo para la gestión de ventas, el cual incluya las funciones eliminar, insertar y listar ventas, integrando el lenguaje de programación PHP y SGBD MySql, en un periodo de 3 a 5 días, para que, de esta forma, la empresa sea capaz de mejorar la administración de ventas hasta en un 50% en el próximo año.
* Desarrollar una modulo para la gestión de pedidos, el cual incluya las funciones eliminar, insertar y listar pedidos, en un periodo de 3 a 7 días, integrando el lenguaje de programación PHP y SGBD MySql, para que, de esta forma, la empresa pueda implementar la estrategia de pedidos y de esta forma, incrementar sus ventas hasta en un 10% en el próximo año.
* Desarrollar una modulo para la gestión de empleados, el cual incluya las funciones eliminar, insertar, editar y listar empleados, en un periodo de 3 a 5 días, utilizando el lenguaje de programación PHP y SGBD MySql, para ayudar a la administración de las ventas hasta en un 10% y de forma indirecta, a la administración de empleados, en los próximos 12 o 14 meses.
* Desarrollar una modulo para la gestión de clientes, el cual incluya las funciones eliminar, insertar, editar y listar clientes, integrando el lenguaje de programación PHP y SGBD MySql, en un periodo de 3 a 4 días, para mejorar el control sobre los pedidos.
* Desarrollar una modulo para la gestión de proveedores, el cual incluya las funciones eliminar, insertar, editar y listar proveedores, en un periodo de 2 a 3 días, integrando el lenguaje de programación PHP y SGBD MySql, para mejorar hasta un 90% la administración de las marcas, en el próximo año.
* Desarrollar una modulo para la gestión de Marcas, el cual incluya las funciones eliminar, insertar, editar y listar marcas, integrando el lenguaje de programación PHP y SGBD MySql, en un periodo de 2 a 4 días, para mejorar la administración de marcas asta en un 90% en el próximo año.
* Desarrollar una modulo para la gestión de categorías, el cual incluya las funciones eliminar, insertar, editar y listar categorías, integrando el lenguaje de programación PHP y SGBD MySql, en un periodo de 2 a 3 días, y de esta forma, mejorar la gestión del stock asta en un 30% en los próximo 7 u 8 meses.
* Ayudar a mejorar el reconocimiento de la empresa asta en un 10%, a través del desarrollo de un apartado publicitario (sitio web), en un periodo de 5 a 7 días, implementando las tecnologías HTML y CSS.
* Desarrollar un método para la generación de reportes de venta, implementando el lenguaje de programación PHP y el SGBD MySql, y las tecnologías web HTML, CSS y JavaScript, para darle diseño al reporte, en un periodo de 6 a 7 dias, para mejorar la experiencia del usuario asta en un 15%.
* Desarrollar un método para la generación de reportes de pedido, implementando el lenguaje de programación PHP y el SGBD MySql, y las tecnologías web HTML, CSS y JavaScript, para darle diseño al reporte, en un periodo de 6 a 7 días, para mejorar la experiencia del usuario asta en un 15%.

## 

## **NORMAS, POLÍTICAS Y PROCEDIMIENTOS**

**CMMI**

CMMI Acelera la introducción en las organizaciones de producción de software, Mantienen la competitividad entre las organizaciones, Habilitan organizaciones públicas y privadas, para que hagan mejoras en sus prácticas, Fomentan la adopción de estándares de excelencia.

**ISO 9001**

Está pensada para que, las organizaciones que la apliquen, puedan garantizar su capacidad de ofrecer productos y servicios que cumplen con las exigencias de sus clientes, gracias a una certificación internacional que les brinde prestigio y garantías de calidad.

**IEEE 830**

El estándar IEEE 830-1998 para el SRS (en inglés) o ERS (Especificación de requerimientos de software) es un conjunto de recomendaciones para la especificación de los requerimiento o requisitos de software el cual tiene como producto final la documentación de los acuerdos entre el cliente y el grupo de desarrollo para así cumplir con la totalidad de exigencias estipuladas.

**IEEE 829**

Esta norma se aplica a todos los sistemas basados en software. Se aplica a los sistemas y software que están siendo desarrollados, adquiridos, operados, mantenidos y reutilizados. Al llevar a cabo el proceso de prueba, es importante examinar el software en sus interacciones con las otras partes del sistema. Esta norma identifica las consideraciones del sistema que los procesos de prueba y las tareas se ocupan en el sistema y la corrección de software y otros atributos (por ejemplo, integridad, exactitud, consistencia y capacidad de prueba), la determinación y la documentación de ensayo resultante aplicable.

**ISO 27000**

Esta norma proporciona una visión general de las normas que componen la serie 27000, indicando para cada una de ellas su alcance de actuación y el propósito de su publicación. Recoge todas las definiciones para la serie de normas 27000 y aporta las bases de por qué es importante la implantación de un SGSI, una introducción a los Sistemas de Gestión de Seguridad de la Información, una breve descripción de los pasos para el establecimiento, monitorización, mantenimiento y mejora de un SGSI.

**DEFINICIÓN DE LAS REGLAS DEL NEGOCIO IMPLEMENTADAS**

* Si el producto enviado no era lo que el cliente pidió, se le podrá realizar el cambio sin costos adicionales.
* Para el proceso de devolución, se realizará un cambio en el registro del pedido.
* Los costos del envío están sujetos a la cantidad monetaria comprada.

## **FUNDAMENTACIÓN DE LA TECNOLOGÍA UTILIZADA**

**PHP**

1. Para desarrollar el sitio web se eligió un lenguaje de programación que es PHP ya que es un lenguaje total mente libre.
2. Es abierto ya que cumple con los estándares de programación como la orientación a objetos.
3. PHP es uno de los lenguajes de programación que es más utilizado en sitio web.
4. No es muy necesario tener un entorno de desarrollo para programar en PHP
5. También se eligió por que no se necesita más que apache y MySQL para hacer un proyecto.

**Framework CodeIgniter**

1. Se eligió un Framework que es CodeIgniter ya que es un Framework de los más fáciles.
2. CodeIgniter cuenta con librerías que son extensas para ayudar a programar más fácil los apartados de sitio web.
3. Trabaja con el lenguaje de programación PHP esto hace que a los programadores mejoren la forma de trabajar y con mayor velocidad.
4. CodeIgniter maneja un patrón de diseño que es MVC esto hace que los programadores puedan tener los sitios web más prontos y sin cometer muchos errores ya que es como una estructura diseñada.
5. También te permite reutilizar código y es fácil de editar el código.

**MySQL**

1. Se eligió un sistema gestor de base de datos (MySQL) que sea libre, relacional y multiusuario.
2. Es uno de los más utilizados para los sitios web que se suben a internet.
3. El sistema gestor de base de datos es libre.
4. Tiene una gran cantidad de herramientas para administrar todas las bases de datos que se encuentren dentro de él.
5. También este sistema gestor de base de datos soporta diferentes sistemas operativos y es muy fácil su instalación y configuración.

**EsHost**

1. Es Host se eligió este hosting ya que da un plan de hosting gratuito
2. cuenta con 1000 MB de espacio en el disco
3. diseñado para soportar PHP y MySQL
4. En espacio dan 10 Gb para la transferencia de los archivos del sitio web
5. Es un host que no manejan publicidad esto ayuda al desarrollador a controlar mejor su sitio web.

**IDE NetBeans**

1. El entorno de desarrollo que se eligió fue NetBeans ya que se usó para desarrollar proyectos anteriores y dio buenos resultados
2. Además de que este entorno de desarrollo puede crear diferentes proyectos y con diferentes lenguajes de programación
3. El IDE proporciona una gran cantidad de herramientas para los desarrolladores de PHP.
4. Este entorno de desarrollo también se eligió porque es gratuito.
5. Puede abrir proyectos con el Framework CodeIgniter.

## **DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA**

|  |  |
| --- | --- |
| Actor del negocio | Descripción |
| Administrador | Este actor será el encargado administrar el sitio web.  Administrará el contenido de la aplicación web, para mantener actualizados tanto la información como los productos.  Algunas características que deberán cumplir los usuarios que administrarán la aplicación son: tener conocimientos sobre el manejo de equipo de cómputo, contar con una previa capacitación en el uso del nuevo sistema, ser ordenado, ser una persona de confianza para la organización, conocer bien el funcionamiento del negocio. |
| Cliente | Este actor visualiza información además puede consultar y comprar productos publicados en el sitio web.  Consultará el contenido de la aplicación y podrá realizar pedidos de los productos que se ofrecen.  Algunas características que deberán cumplir los clientes o usuarios que consulten la aplicación son: tener conocimientos sobre el manejo de equipo de cómputo. |

## **ESPECIFICACIÓN DE REQUISITOS**

1. **Descripción de los Requisitos Funcionales**.

Los requerimientos son las funcionalidades ayudan a resolver el problema que tiene el cliente, así como también lo que va a hacer el sistema. Para recolectar los requerimientos funcionales de un sistema se tiene que hacer una cita con el cliente y aplicar ya sea una encueste o una entrevista para así saber cuáles son los requerimientos funcionales y no funcionales

|  |  |  |
| --- | --- | --- |
| ID DEL REQUERIMIENTO | NOMBRE | DESCRIPCIÓN |
| RF\_01 | Consultar tablas. | Los usuarios podrán consultar las tablas con las que cuenta la base de datos. |
| RF\_02 | Consultas. | Los usuarios podrán observar o consultar los datos que contiene cada una de las tablas de la base de datos. |
| RF\_03 | Altas. | El administrador podrá realizar altas dentro de las tablas de la base de datos, ingresando correctamente los datos que cada una de ellas solicita, el id de la tabla se ingresará automáticamente debido a que este será auto incrementable. |
| RF\_04 | Actualizaciones. | El administrador podrá realizar actualizaciones en los datos de las tablas de la base de datos, llenando los requisitos que se piden en cada una de las tablas, dichas actualizaciones se realizarán por medio de la llave primaria (id). |
| RF\_05 | Bajas. | El administrador podrá realizar bajas en los datos que contienen las tablas de las bases de datos, dichas bajas serán realizadas por medio de la llave primaria (id). |
| RF\_06 | Consultar empleados. | El administrador podrá consultar los empleados que se tienen registrados en la base de datos. |
| RF\_07 | Agregar empleados. | El administrador podrá realizar registros de nuevos empleados, ingresando el nombre, password, email y el id del nuevo usuario se ingresará solo debido a que será auto incrementable. |
| RF\_08 | Actualizar empleados. | El administrador podrá realizar modificaciones a los empleados de la base de datos por medio del id. |
| RF\_09 | Baja de empleados. | El administrador podrá eliminar o dar de baja a los empleados de la base de datos por medio del id. |
| RF\_10 | LogIn. | Los usuarios podrán ingresar a la base de datos ingresando su email o nombre de usuario y su password. |
| RF\_11 | Consultar clientes. | El administrador podrá consultar los clientes que se tienen registrados en la base de datos. |
| RF\_12 | Agregar clientes. | El administrador podrá realizar registros de nuevos clientes, ingresando el nombre, password, email y el id del nuevo usuario se ingresará solo debido a que será auto incrementable. |
| RF\_13 | Actualizar clientes. | El administrador podrá realizar modificaciones a los clientes de la base de datos por medio del id. |
| RF\_14 | Baja de clientes. | El administrador podrá eliminar o dar de baja a los clientes de la base de datos por medio del id. |
| RF\_15 | Registrar ventas. | El empleado podrá registrar una venta, esto será posible ingresando el total de la venta, el monto con el que se pagó, el cambio que se va a regresar se realizará automáticamente con el total de la venta menos el total con el que el cliente pague, se va a ingresar la fecha de la venta, al igual se registrará al empleado encargado de realizar dicha venta. El id de la venta será ingresado automáticamente ya que este será auto incrementable. |

1. **Descripción de los Requisitos No Funcionales.**

|  |  |  |
| --- | --- | --- |
| ID DEL REQUERIMIENTO | NOMBRE | DESCRIPCIÓN |
| RNF\_01 | Desempeño. | El sitio web realizará las operaciones solicitadas en un lapso de tiempo no mayor a 10 segundos. |
| RNF\_02 | Disponibilidad. | El sitio web estará disponible 24/7 los 365 días del año. |
| RNF\_03 | Usabilidad. | El sitio web proporcionará mensajes de error que serán informativos y orientados a usuario final, además el sitio web contendrá interfaces gráficas de acuerdo a una plantilla además de visualizarse en diferentes dispositivos móviles. Será fácil navegar en las páginas del sitio web. |
| RNF\_04 | Idioma. | El sitio web presentará el contenido en el idioma Español. |
| RNF\_05 | Seguridad. | Los datos que contenga la base de datos se mantendrán íntegros. El sitio web no revelará datos personales de los clientes distintos de los nombres y números de referencia. |
| RNF\_06 | Navegador. | El sitio web podrá ser abierto en cualquier navegador web. |
| RNF\_07 | Mantenibilidad. | La mantenibilidad del sitio queda en manos del administrador del sistema, para lo cual se  capacitará a esta persona. Si el mantenimiento va más allá de las funcionalidades del  administrador la empresa deberá contactar al equipo de desarrollo del sitio. |
| RNF\_08 | Portabilidad. | La aplicación deberá funcionar en los sistemas operativos: Windows, Linux y Mac OS. Siempre  y cuando la máquina cuente con un explorador (Google Chrome, Mozilla Firefox, Safari e  Internet Explorer) en las versiones que se especificaron anteriormente para cada uno de ellos. El sitio web será sencillo de instalar. |
| RNF\_09 | Interface. | El sitio web será atractivo, es decir, contendrá imágenes llamativas que captarán la atención del cliente, así también, el sitio web contará con poco texto. |
| RNF\_10 | Colores. | El sitio web tendrá colores establecidos por el cliente, los cuales serán acorde a la organización a la que se realiza el proyecto. |

## **VISTA FUNCIONAL**

**Tabla 1. Descripción de bajo nivel del caso de uso: CU-01.**

|  |  |
| --- | --- |
| CU-01 | |
| Nombre del caso de uso | Alta de un producto. |
| Actores | * Administrador. * Sistema. |
| Resumen | 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de productos. 7. El sistema muestra la página de productos. 8. El administrador selecciona la opción de “Agregar producto”. 9. El sistema muestra el formulario para agregar un nuevo producto. 10. El administrador introduce los datos necesarios para agregar un nuevo producto. 11. El administrador da clic en la opción de “Guardar producto”. 12. El sistema valida los datos introducidos. 13. El sistema guarda la información introducida en la Base de Datos. 14. El sistema muestra la página actualizada de los productos. |
| Precondiciones | El administrador de la empresa deberá contar con la información necesaria para dar de alta un producto que se desee agregar en el sistema como el nombre del producto, precio, color, stock, marca y categoría a la cual pertenece el mismo. |
| Poscondiciones | El administrador tendrá que verificar que el o los productos dados de alta se hayan registrado correctamente en el sistema; esto al aparecer el producto en la lista de productos una vez registrado. |

**Tabla 2. Descripción de bajo nivel del caso de uso: CU-02.**

|  |  |
| --- | --- |
| CU-02 | |
| Nombre del caso de uso | Baja de un producto. |
| Actores | * Administrador. * Sistema. |
| Resumen | 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de productos. 7. El sistema muestra la página de los productos listados. 8. El administrador localiza el producto a eliminar. 9. El administrador da clic en la opción de “Eliminar producto”. 10. El sistema muestra un mensaje de confirmación. 11. El administrador da clic en confirmar para eliminar el producto. 12. El sistema elimina el producto en la Base de Datos. 13. El sistema redirige a la página de productos ya actualizados. |
| Precondiciones | El administrador tendrá que tener a la mano la información del o los productos que se requieran dar de baja para que estos sean eliminados del sistema como el nombre del producto. |
| Poscondiciones | El administrador tendrá que verificar que el o los productos dados de baja se hayan eliminado correctamente en el sistema; esto al mostrar un mensaje donde se informe que la información del producto ha sido eliminada. |

**Tabla 3. Descripción de bajo nivel del caso de uso: CU-03.**

|  |  |
| --- | --- |
| CU-03 | |
| Nombre del caso de uso | Editar un producto. |
| Actores | * Administrador. * Sistema. |
| Resumen | 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de productos. 7. El sistema muestra la página de los productos listados. 8. El administrador localiza el producto a editar. 9. El administrador da clic en la opción de “Editar producto”. 10. El sistema muestra el formulario para editar un producto con los datos correspondientes a modificar. 11. El administrador modifica los datos necesarios del producto. 12. El administrador da clic en la opción de “Guardar cambios”. 13. El sistema valida los datos modificados. 14. El sistema guarda los datos en la Base de Datos. 15. El sistema redirige a la página de productos ya actualizados. |
| Precondiciones | El administrador tendrá que tener a la mano la información del o los productos que se requieran editar para que estos puedan ser modificados en el sistema como el nombre del producto, precio, color, stock, marca y categoría a la cual pertenece el mismo. |
| Poscondiciones | El administrador tendrá que verificar que el o los productos editados se hayan modificado en el sistema; esto al aparecer la información del producto actualizada en la lista de productos una vez guardada. |

**Tabla 4. Descripción de bajo nivel del caso de uso: CU-04.**

|  |  |
| --- | --- |
| CU-04 | |
| Nombre del caso de uso | Consultar producto. |
| Actores | * Administrador. * Sistema. |
| Resumen | 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de productos. 7. El sistema muestra la página de los productos listados. 8. El administrador localiza el producto a consultar. 9. El administrador da clic en la opción de “Consultar producto”. 10. El sistema muestra una página donde se muestra la información del producto como el nombre del producto, precio, color, stock, marca y categoría a la cual pertenece el mismo. 11. El administrador consulta a información que necesita del producto. 12. El administrador da clic en la opción “Volver a la lista” para regresar a la lista de productos. 13. El sistema muestra la página donde se encuentran todos los productos listados. |
| Precondiciones | El administrador tendrá que tener a la mano la información del o los productos que se requieran consultar para que estos puedan ser visualizados en el sistema con su información correspondiente como el nombre del producto, precio, color, stock, marca y categoría a la cual pertenece el mismo. |
| Poscondiciones | El administrador tendrá que verificar que el o los productos consultados contengan la información correcta; esto al consultar él seleccionado y ver que el nombre del producto, precio, color, stock, marca y categoría a la cual pertenece el mismo sean los verídicos. |

**Tabla 5. Descripción de bajo nivel del caso de uso: CU-05.**

|  |  |
| --- | --- |
| CU-05 | |
| Nombre del caso de uso | Alta de un proveedor. |
| Actores | * Administrador. * Sistema. |
| Resumen | 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de proveedores. 7. El sistema muestra la página de proveedores. 8. El administrador selecciona la opción de “Agregar proveedor”. 9. El sistema muestra el formulario para agregar un nuevo proveedor. 10. El administrador introduce los datos necesarios para agregar un nuevo proveedor. 11. El administrador da clic en la opción de “Guardar proveedor”. 12. El sistema valida los datos introducidos. 13. El sistema guarda la información introducida en la Base de Datos. 14. El sistema muestra la página actualizada de los proveedores. |
| Precondiciones | El administrador de la empresa deberá contar con la información necesaria para dar de alta un proveedor que se desee agregar en el sistema como el nombre del proveedor, email, RFC, teléfono y la dirección del mismo. |
| Poscondiciones | El administrador tendrá que verificar que el o los proveedores dados de alta se hayan registrado correctamente en el sistema; esto al aparecer el proveedor en la lista de proveedores una vez registrado. |

**Tabla 6. Descripción de bajo nivel del caso de uso: CU-06.**

|  |  |
| --- | --- |
| CU-06 | |
| Nombre del caso de uso | Baja de un proveedor. |
| Actores | * Administrador. * Sistema. |
| Resumen | 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de proveedores. 7. El sistema muestra la página de los proveedores listados. 8. El administrador localiza el proveedor a eliminar. 9. El administrador da clic en la opción de “Eliminar proveedor”. 10. El sistema muestra un mensaje de confirmación. 11. El administrador da clic en confirmar para eliminar el proveedor. 12. El sistema elimina el proveedor en la Base de Datos. 13. El sistema redirige a la página de proveedores ya actualizados. |
| Precondiciones | El administrador tendrá que tener a la mano la información del o los proveedores que se requieran dar de baja para que estos sean eliminados del sistema como el nombre del proveedor. |
| Poscondiciones | El administrador tendrá que verificar que el o los productos dados de baja se hayan eliminado correctamente en el sistema; esto al mostrar un mensaje donde se informe que la información del proveedor ha sido eliminada. |
| Requisitos especiales |  |

**Tabla 7. Descripción de bajo nivel del caso de uso: CU-07.**

|  |  |
| --- | --- |
| CU-07 | |
| Nombre del caso de uso | Editar un proveedor. |
| Actores | * Administrador. * Sistema. |
| Resumen | 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de proveedores. 7. El sistema muestra la página de los proveedores listados. 8. El administrador localiza el proveedor a editar. 9. El administrador da clic en la opción de “Editar proveedor”. 10. El sistema muestra el formulario para editar un proveedor con los datos correspondientes a modificar. 11. El administrador modifica los datos necesarios del proveedor. 12. El administrador da clic en la opción de “Guardar cambios”. 13. El sistema valida los datos modificados. 14. El sistema guarda los datos en la Base de Datos. 15. El sistema redirige a la página de proveedores ya actualizados. |
| Precondiciones | El administrador tendrá que tener a la mano la información del o los proveedores que se requieran editar para que estos puedan ser modificados en el sistema como el nombre del proveedor, email, RFC, teléfono y la dirección del mismo. |
| Poscondiciones | El administrador tendrá que verificar que el o los proveedores editados se hayan modificado en el sistema; esto al aparecer la información del proveedor actualizada en la lista de proveedores una vez guardada. |

**Tabla 8. Descripción de bajo nivel del caso de uso: CU-08.**

|  |  |
| --- | --- |
| CU-08 | |
| Nombre del caso de uso | Consultar proveedor. |
| Actores | * Administrador. * Sistema. |
| Resumen | 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de proveedores. 7. El sistema muestra la página de los proveedores listados. 8. El administrador localiza el proveedor a consultar. 9. El administrador da clic en la opción de “Consultar proveedor”. 10. El sistema muestra una página donde se muestra la información del producto como el nombre del proveedor, email, RFC, teléfono y la dirección del mismo. 11. El administrador consulta a información que necesita del proveedor. 12. El administrador da clic en la opción “Volver a la lista” para regresar a la lista de proveedores. 13. El sistema muestra la página donde se encuentran todos los proveedores listados. |
| Precondiciones | El administrador tendrá que tener a la mano la información del o los proveedores que se requieran consultar para que estos puedan ser visualizados en el sistema con su información correspondiente como el nombre del proveedor, email, RFC, teléfono y la dirección del mismo. |
| Poscondiciones | El administrador tendrá que verificar que el o los proveedores consultados contengan la información correcta; esto al consultar él seleccionado y ver que el nombre del producto, precio, color, stock, marca y categoría a la cual pertenece el mismo sean los verídicos. |

**Tabla 9. Descripción de bajo nivel del caso de uso: CU-09.**

|  |  |
| --- | --- |
| CU-09 | |
| Nombre del caso de uso | Alta de un pedido. |
| Actores | * Administrador. * Sistema. |
| Resumen | 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de pedidos. 7. El sistema muestra la página de pedidos. 8. El administrador selecciona la opción de “Agregar pedido”. 9. El sistema muestra el formulario para agregar un nuevo pedido. 10. El administrador introduce los datos necesarios para agregar un nuevo pedido. 11. El administrador da clic en la opción de “Guardar pedido”. 12. El sistema valida los datos introducidos. 13. El sistema guarda la información introducida en la Base de Datos. 14. El sistema muestra la página actualizada de los pedidos. |
| Precondiciones | El administrador de la empresa deberá contar con la información necesaria para dar de alta un pedido que se desee agregar en el sistema como el id del pedido, el status, la fecha del pedido y la fecha de entre, la venta a la cual pertenece y el cliente a quien se realiza el mismo. |
| Poscondiciones | El administrador tendrá que verificar que el o los pedidos dados de alta se hayan registrado correctamente en el sistema; esto al aparecer el pedido en la lista de pedidos una vez registrado. |

**Tabla 10. Descripción de bajo nivel del caso de uso: CU-10.**

|  |  |
| --- | --- |
| CU-10 | |
| Nombre del caso de uso | Baja de un proveedor. |
| Actores | * Administrador. * Sistema. |
| Resumen | 1. El administrador entra al sistema. 2. El sistema muestra la página inicial. 3. El administrador introduce su usuario y su contraseña. 4. El sistema valida los datos registrados. 5. El sistema muestra el panel de administración. 6. El administrador ingresa a la sección de pedidos. 7. El sistema muestra la página de los pedidos listados. 8. El administrador localiza el pedido a eliminar. 9. El administrador da clic en la opción de “Eliminar pedido”. 10. El sistema muestra un mensaje de confirmación. 11. El administrador da clic en confirmar para eliminar el pedido. 12. El sistema elimina el pedido en la Base de Datos. 13. El sistema redirige a la página de pedidos ya actualizados. |
| Precondiciones | El administrador tendrá que tener a la mano la información del o los pedidos que se requieran dar de baja para que estos sean eliminados del sistema como el id del pedido, el status, la fecha del pedido y la fecha de entre, la venta a la cual pertenece y el cliente a quien se realiza el mismo. |
| Poscondiciones | El administrador tendrá que verificar que el o los pedidos dados de baja se hayan eliminado correctamente en el sistema; esto al mostrar un mensaje donde se informe que la información del pedido ha sido eliminada. |

## **VISTA LÓGICA**

**I. Patrones de diseño.**

Un patrón de diseño es un conjunto de reglas que describen como afrontar tareas y solucionar problemas que surgen durante el desarrollo del software. Existen varias clasificaciones de los patrones, una de ellas es según su finalidad, dentro de la cual se consideran 3 conjuntos fundamentales:

* + **Patrones de Creación.**

Estos patrones crean objetos, evitando dicha función y la instanciación directa por parte de los desarrolladores. Ello proporciona a nuestros programas una mayor flexibilidad para decidir que objetos usar. Dentro de este grupo los más conocidos son: Factory, Abstract Factory, Builder, Prototype y Singlenton.

* + **Patrones Estructurales.**

Guían el desarrollo de la aplicación bajo una determinada estructura. Entre ellos se encuentran: Adapter, Bridge, Composite, Decorator, Facade, Flayweight y Proxy.

* + **Patrones de Comportamiento.**

Enfocados principalmente en algoritmos y en la asignación de responsabilidades entre los objetos. Describen el comportamiento de objetos, clases y la comunicación entre ellos.

Estos se clasifican de la siguiente forma:

* + - **A nivel de clases:** Usan la herencia como vía para distribuir el comportamiento. El Interpreter y el Template Method constituyen ejemplos de este tipo de patrón.
    - **A nivel de objetos:** Usan la composicíon, más la herencia p’ara llevar a cabo las tareas. Muestra de ellos el Command, Iterator, Chain of Responsibility, Strategy y el State.

**Patrones de Diseño Detallados**

**MVC (Modelo Vista Controlador)**

1. Descripción del patrón.

Es un patrón de arquitectura de las aplicaciones de software, este patrón separa la lógica de negocio de la interfaz de usuario; así como incremente la reutilización y flexibilidad.

En pocas palabras, el patrón de diseño MVC organiza el código en base a su función. Por lo que este patrón separa el código en tres capas:

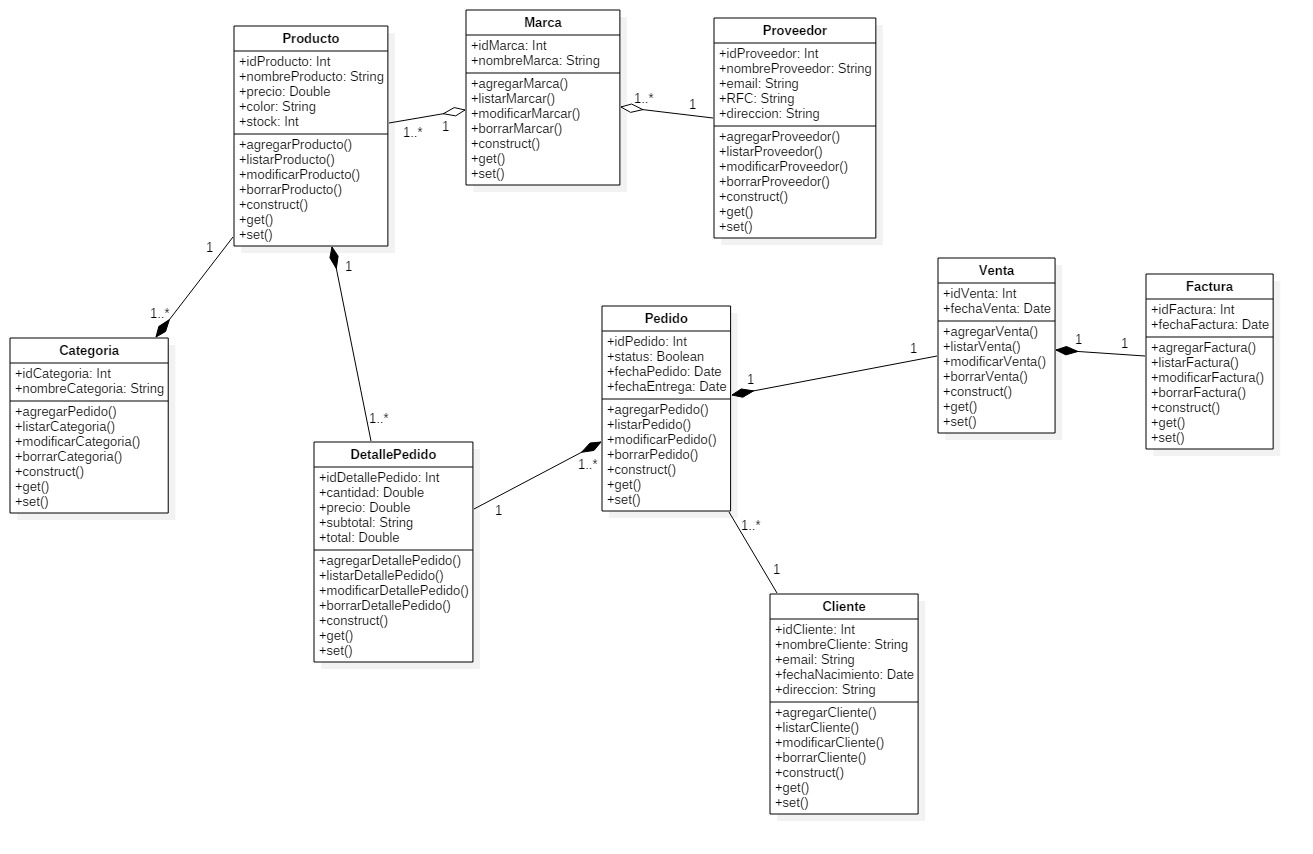
* + La capa del **modelo** define la lógica de negocio (la base de datos pertenece a esta capa).
  + La **vista** es lo que utilizan los usuarios para interactuar con la aplicación (los gestores de plantillas pertenecen a esta capa).
  + El **controlador** es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario.

1. ¿Dónde y por qué se emplea?

Este patrón de diseño se emplea en la estructura del proyecto, el cual permite separar la funcionalidad del proyecto de las interfaces del mismo.

Este patrón es empleado en el proyecto, debido a que es un patrón de gran utilidad en el diseño del proyecto integrador, así como en su estructura ya que se contará con un mejor orden de las aceptadas y archivos generados durante el desarrollo del proyecto; permitiendo así la separación de las acciones del usuario y la lógica de la aplicación.

1. Diagrama de clases donde se implementa el patrón.

****

**II. Mecanismos de diseño.**

Estos mecanismos son el resultado del refinamiento realizado sobre los mecanismos del análisis. Cabe destacar que este proceso se basa en las restricciones impuestas por el ambiente de implementación.

Algunos ejemplos de estos mecanismos son los siguientes:

* + **Mecanismo de seguridad:**

La seguridad es un aspecto importante en el desarrollo de un proyecto ya que se deben contar con las medidas necesarias para evitar posibles vulnerabilidades que puedan presentarse en este mismo.

Así mismo, esta herramienta de seguridad y/o control permite detectar, prevenir o recobrarse de alguna vulnerabilidad presentada en el proyecto.

* + **Mecanismo de Acceso a Datos:**

El acceso a la información es otro aspecto importante que debe tomarse en cuenta debido a que no todos los usuarios pueden tener acceso a esta misma, por lo que es necesario definir quiénes serán los actores en cada uno de los procesos que son desarrollados en el proyecto.

**Mecanismos de Diseño Detallados**

1. Descripción de los mecanismos identificados.
   * **Mecanismo de seguridad:**

Algunas de las medidas más importantes que se implementarán en el proyecto sobre los mecanismos de seguridad son las siguientes:

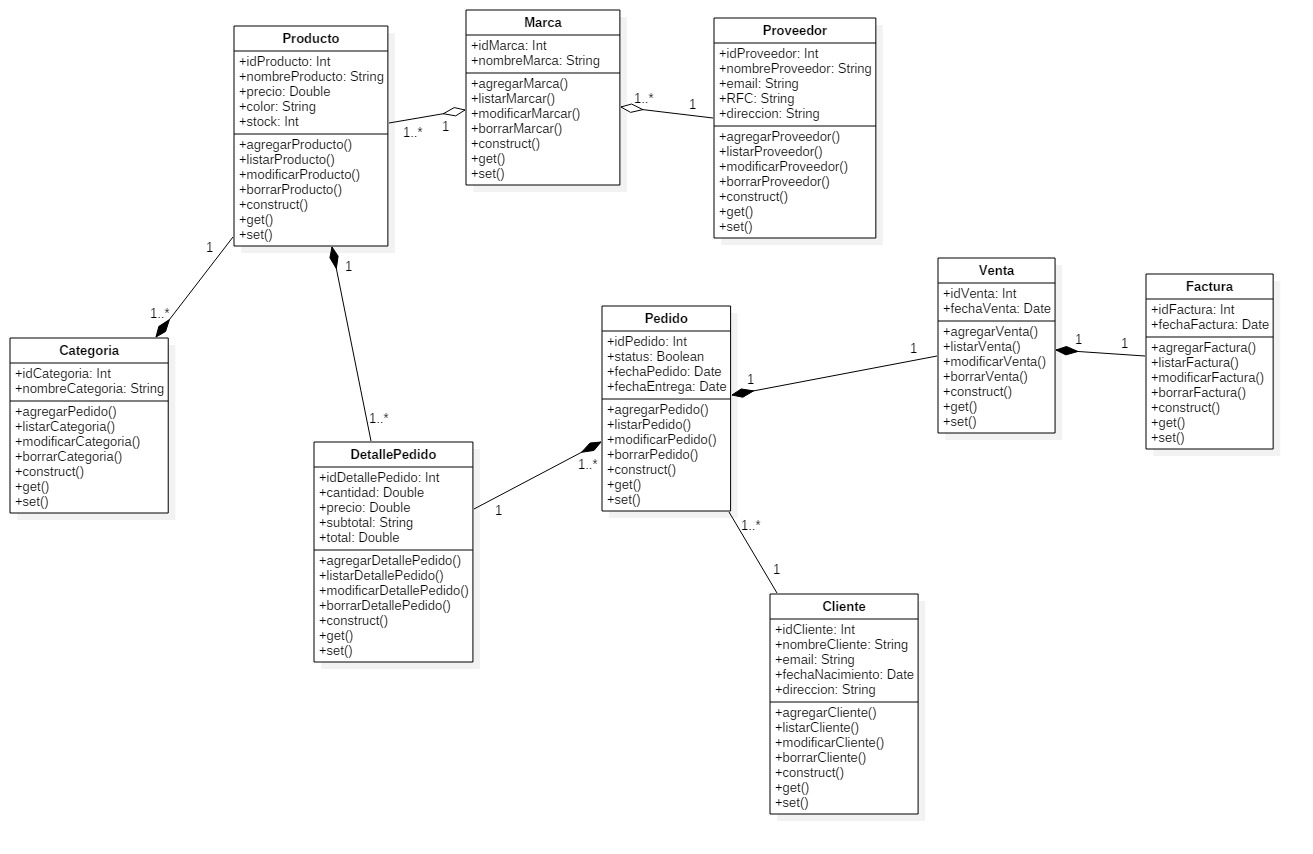
* + - **Restricciones en el sitio:** En el proyecto es importante aplicar restricciones a los usuarios, con la finalidad de dar los permisos necesarios para cada uno de éstos, lo que permitirá que la información sea proporcionada a las personas adecuadas y con sus permisos respectivos sobre la información que se manipula.
    - **Rastreo del paso de datos:** Esta medida es importante, ya que debe mantenerse el conocimiento del proceso del manejo de la información; con la finalidad de conocer el origen de los datos, para así mismo identificar si la información que se manipula es de confianza.
    - **Filtración de entradas:** Es muy importante el filtrado de la entrada de la información en un proyecto ya que si se realiza esta medida pueden eliminarse riesgos de que los datos dañados sean usados para provocar el mal funcionamiento de la aplicación.
  + **Mecanismo de Acceso a Datos:**

Es importante tomar en cuenta la siguiente medida identificada dentro del mecanismo de Acceso a Datos:

* **Diagrama de caso de uso:** Es importante definir cada uno de los procesos que serán ejecutados dentro de la aplicación, debido a que debe tomarse en cuenta los involucrados que participarán en cada proceso y el flujo de actividades que debe seguir para completar este mismo.

1. Documentación de los mecanismos más importantes.

* **Vista estática.**

****

* **Descripción de cada elemento mostrado en la vista estática.**

El diagrama de clases es la estructura estática que describe la estructura del sistema mostrando sus clases, atributos, operaciones y/o métodos y la relación que hay entre sus objetos; por lo que es importante describir cada una de sus clases:

* + **Categoría:**

Esta clase cuenta con dos atributos: idCategoria (int) y nombreCategoria (String), los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

* + - agregarCategoria(): Método que permitirá al usuario dar de alta una categoría en el sistema.
    - listarCategoria(): Método que permitirá al usuario mostrar todas las categorías con las cuales cuenta el sistema.
    - modificarCategoria(): Método que permitirá al usuario editar la información de una categoría que se encuentre registrada en la aplicación.
    - borrarCategoria(): Método que permitirá al usuario eliminar una categoría que se encuentre registrada en la aplicación.
    - construct(), getters() y setters(): Son los métodos de acceso de cada uno de los atributos con los que cuenta una categoría registrada en el sistema.
  + **Producto:**

Esta clase cuenta con dos atributos: idProducto, nombreProducto, precio, color y stock, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

* + - agregarProducto(): Método que permitirá al usuario dar de alta un producto en el sistema.
    - listarProducto(): Método que permitirá al usuario mostrar todos los productos con las cuales cuenta el sistema.
    - modificarProducto(): Método que permitirá al usuario editar la información de un producto que se encuentre registrado en la aplicación.
    - borrarProducto(): Método que permitirá al usuario eliminar un producto que se encuentre registrado en la aplicación.
    - construct(), getters() y setters(): Son los métodos de acceso de cada uno de los atributos con los que cuenta un producto registrado en el sistema.
  + **Marca:**

Esta clase cuenta con dos atributos: idMarca y nombreMarca, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

* + - agregarMarca(): Método que permitirá al usuario dar de alta una marca en el sistema.
    - listarMarca(): Método que permitirá al usuario mostrar todas las marcas con las cuales cuenta el sistema.
    - modificarMarca(): Método que permitirá al usuario editar la información de una marca que se encuentre registrada en la aplicación.
    - borrarMarca(): Método que permitirá al usuario eliminar una marca que se encuentre registrada en la aplicación.
    - construct(), getters() y setters(): Son los métodos de acceso de cada uno de los atributos con los que cuenta una marca registrada en el sistema.
  + **Proveedor:**

Esta clase cuenta con dos atributos: idProveedor, nombreProveedor, email, RFC, y dirección, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

* + - agregarProveedor(): Método que permitirá al usuario dar de alta un proveedor en el sistema.
    - listarProveedor(): Método que permitirá al usuario mostrar todos los proveedores con las cuales cuenta el sistema.
    - modificarProveedor(): Método que permitirá al usuario editar la información de un proveedor que se encuentre registrado en la aplicación.
    - borrarProveedor(): Método que permitirá al usuario eliminar un proveedor que se encuentre registrado en la aplicación.
    - construct(), getters() y setters(): Son los métodos de acceso de cada uno de los atributos con los que cuenta un proveedor registrada en el sistema.
  + **DetallePedido:**

Esta clase cuenta con dos atributos: idDetallePedido, cantidad, precio, subtotal y total, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

* + - agregarDetallePedido(): Método que permitirá al usuario dar de alta una detalle de un pedido en el sistema.
    - listarDetallePedido(): Método que permitirá al usuario mostrar todos los detalles de los pedidos con los cuales se cuentan.
    - modificaDetallePedido(): Método que permitirá al usuario editar la información de un detalle de pedido que se encuentre registrado en la aplicación.
    - borrarDetallePedido(): Método que permitirá al usuario eliminar un detalle de un pedido que se encuentre registrado en la aplicación.
    - construct(), getters() y setters(): Son los método de acceso de cada uno de los atributos con los que cuenta un producto registrada en el sistema.
  + **Pedido:**

Esta clase cuenta con dos atributos: idPedido, status, fechaPedido y fecha Entrega, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

* + - agregarPedido(): Método que permitirá al usuario dar de alta un pedido en el sistema.
    - listarPedido(): Método que permitirá al usuario mostrar todos los pedidos con las cuales cuenta el sistema.
    - modificarPedido(): Método que permitirá al usuario editar la información de un pedido que se encuentre registrado en la aplicación.
    - borrarPedido(): Método que permitirá al usuario eliminar un pedido que se encuentre registrado en la aplicación.
    - construct(), getters() y setters(): Son los métodos de acceso de cada uno de los atributos con los que cuenta un pedido registrado en el sistema.
  + **Cliente:**

Esta clase cuenta con dos atributos: idCliente, nombreCliente, email, fechaNacimiento y direccion, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

* + - agregarCliente(): Método que permitirá al usuario dar de alta un cliente en el sistema.
    - listarCliente(): Método que permitirá al usuario mostrar todos los clientes con las cuales cuenta el sistema.
    - modificarCliente(): Método que permitirá al usuario editar la información de un cliente que se encuentre registrado en la aplicación.
    - borrarCliente(): Método que permitirá al usuario eliminar un cliente que se encuentre registrado en la aplicación.
    - construct(), getters() y setters(): Son los métodos de acceso de cada uno de los atributos con los que cuenta un cliente registrada en el sistema.
  + **Venta:**

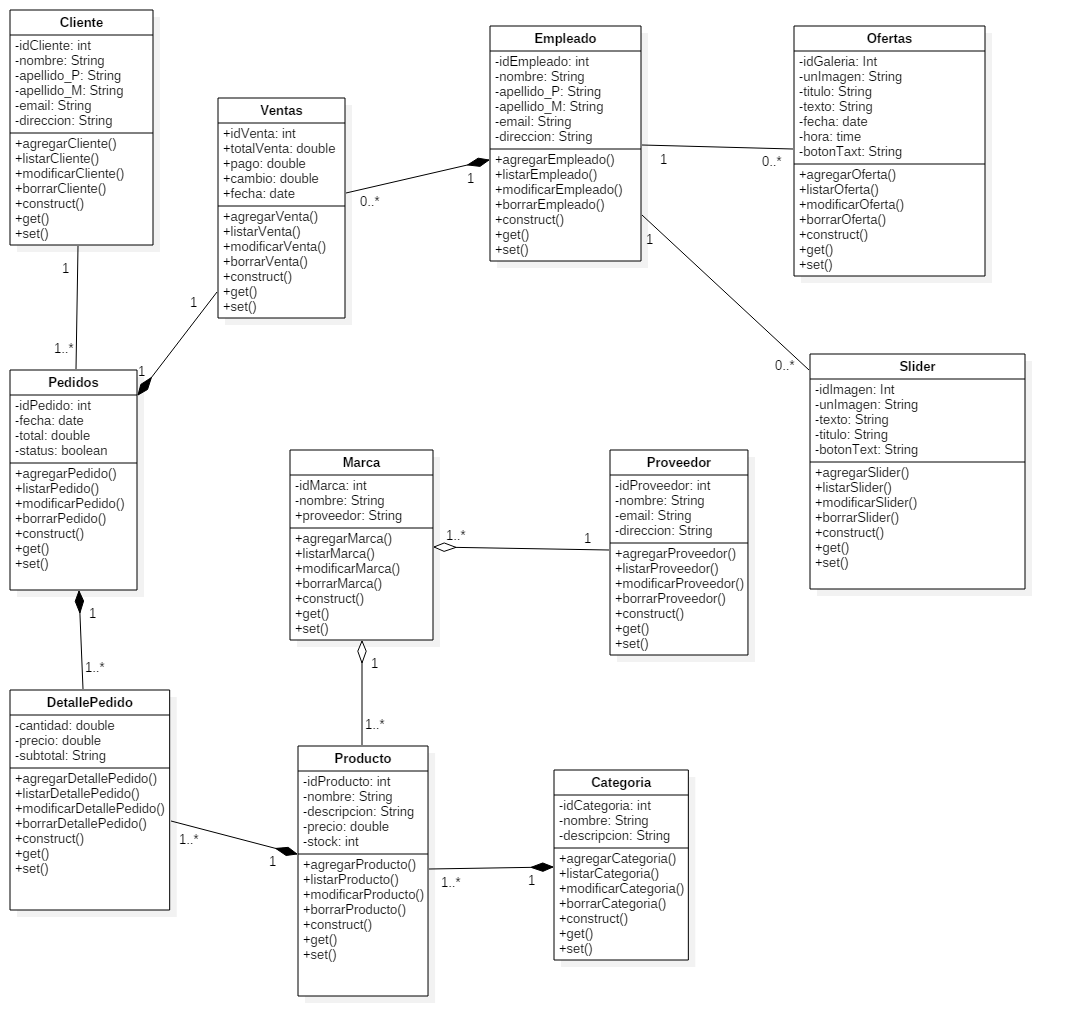
Esta clase cuenta con dos atributos: idVenta y fechaVenta, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

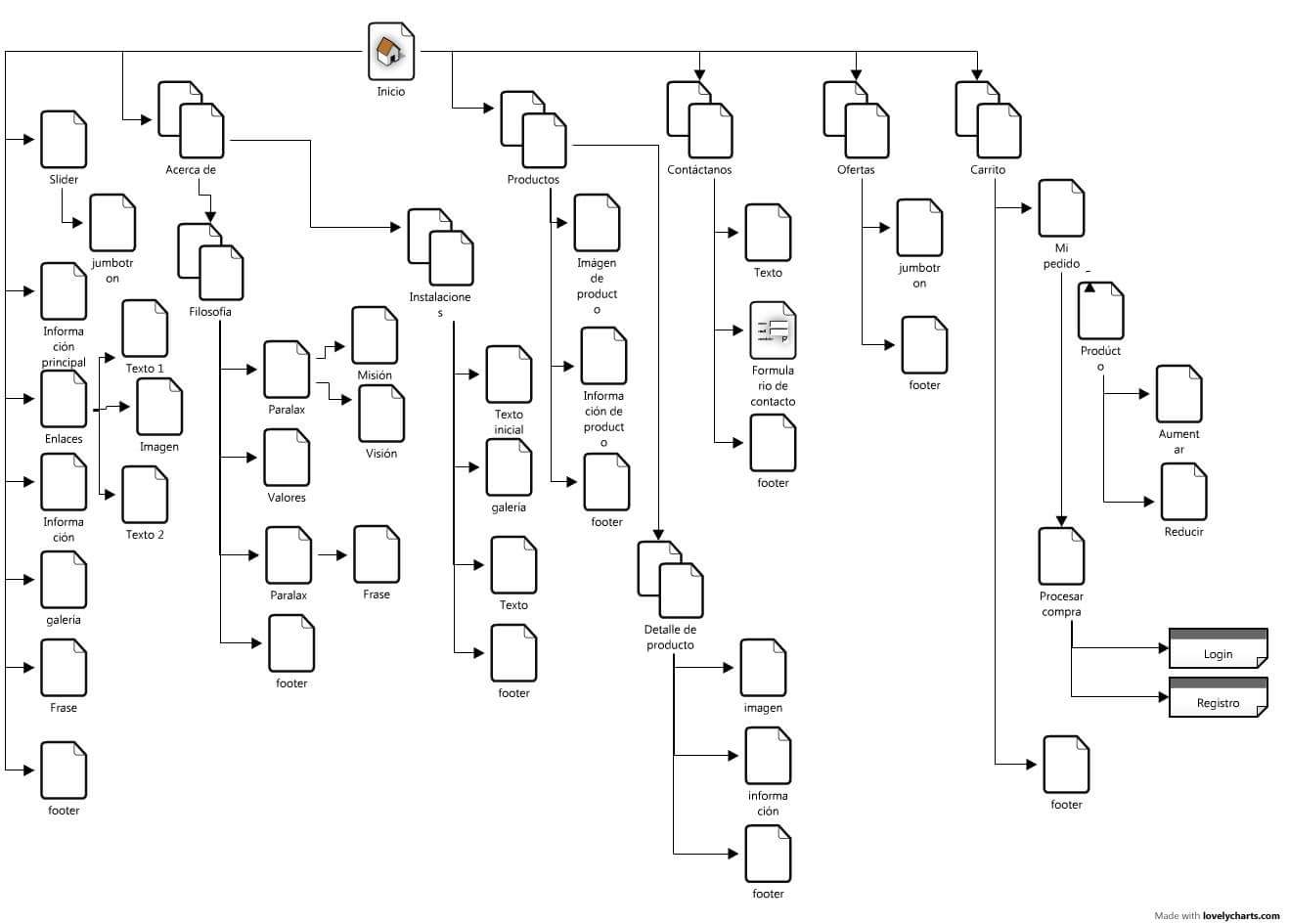
* + - agregarVenta(): Método que permitirá al usuario dar de alta una venta en el sistema.
    - listarVenta(): Método que permitirá al usuario mostrar todas las ventas con las cuales cuenta el sistema.
    - modificarVenta(): Método que permitirá al usuario editar la información de una venta que se encuentre registrada en la aplicación.
    - borrarVenta(): Método que permitirá al usuario eliminar una venta que se encuentre registrada en la aplicación.
    - construct(), getters() y setters(): Son los métodos de acceso de cada uno de los atributos con los que cuenta una venta registrada en el sistema.
  + **Factura:**

Esta clase cuenta con dos atributos: idFactura y FechaFactura, los cuales serán procesados en cada uno de sus métodos; los cuales, son los siguientes:

* + - agregarFactura(): Método que permitirá al usuario dar de alta una factura en el sistema.
    - listarFactura(): Método que permitirá al usuario mostrar todas las facturas con las cuales cuenta el sistema.
    - modificarFactura(): Método que permitirá al usuario editar la información de una factura que se encuentre registrada en la aplicación.
    - borrarFactura (): Método que permitirá al usuario eliminar una factura que se encuentre registrada en la aplicación.
    - construct(), getters() y setters(): Son los métodos de acceso de cada uno de los atributos con los que cuenta una factura registrada en el sistema.

## **MODELO LÓGICO DE DATOS**

****

**MODELO FÍSICO DE DATOS**

## **DISEÑO DE PANTALLAS**

El diseño de las pantallas fue relacionado a diferentes páginas de comida ya que se fue escogiendo el diseño, así como también los colores que se escogieron para el sitio web se sacó una conclusión de que los colores como el anaranjado, amarillo generan hambre al usuario al entrar al sitio web.

Este diseño que se implemento fue se selección porque es muy interactivo con todos los usuarios que entraran a la aplicación web.

El diseño que se eligió fue porque es muy intuitivo con el usuario además de tener una gran variedad imágenes donde se muestran los productos que se encuentran dentro del Eddy Burger

## **DESCRIPCIÓN DE CAMPOS REQUERIDOS POR PANTALLA**

**Inicio**

****

En esta pantalla tiene un apartado donde se muestran los productos que se venden en la empresa, así como también tiene un apartado donde se muestra conócenos, productos y contactos para que tenga mayor interacción.

**Acerca de**

****

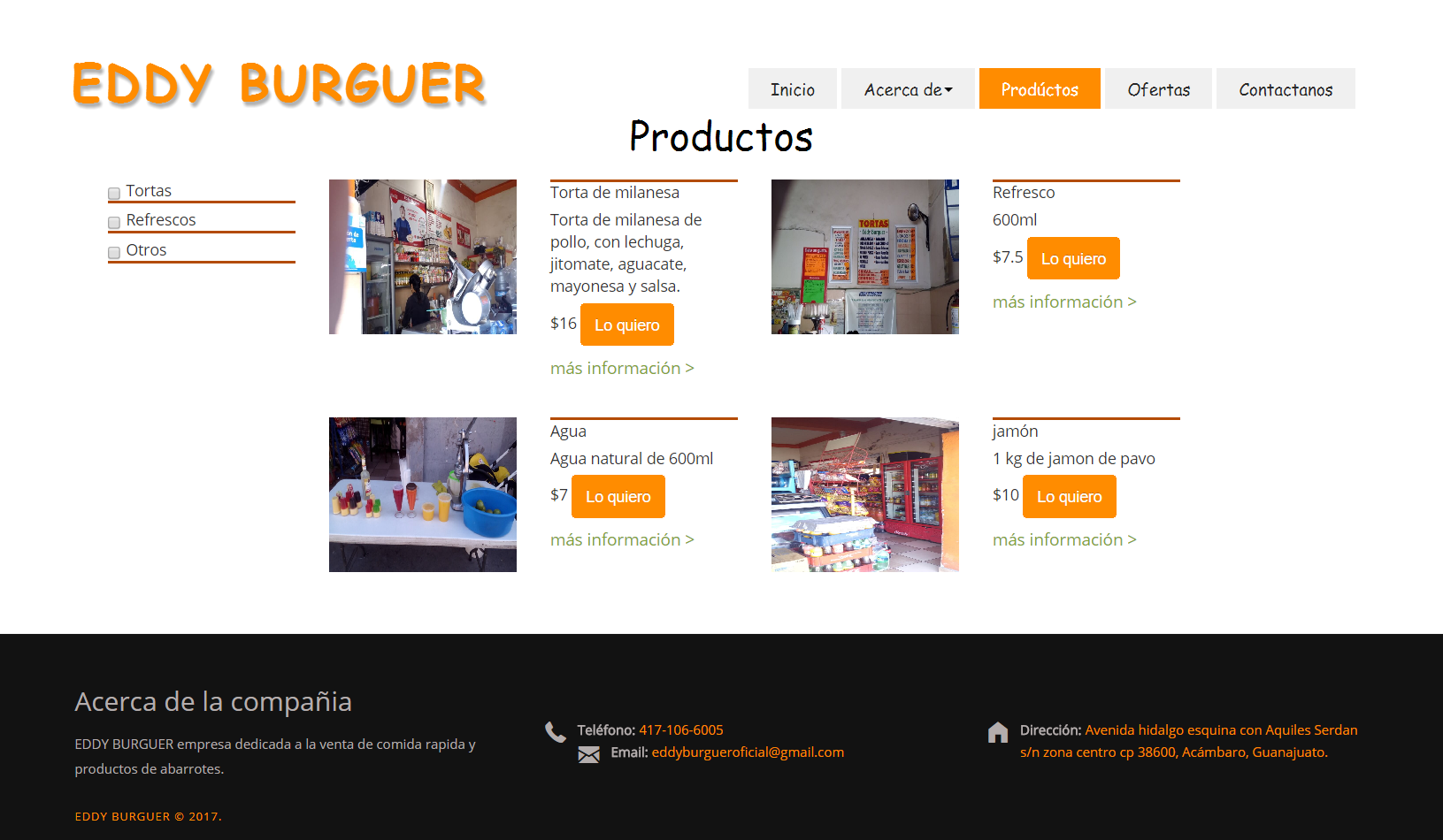
En este apartado se encuentra la filosofía no tiene campos como formularios que pueda llenar el usuario ya que solo es información de la empresa.

**Instalaciones**

****

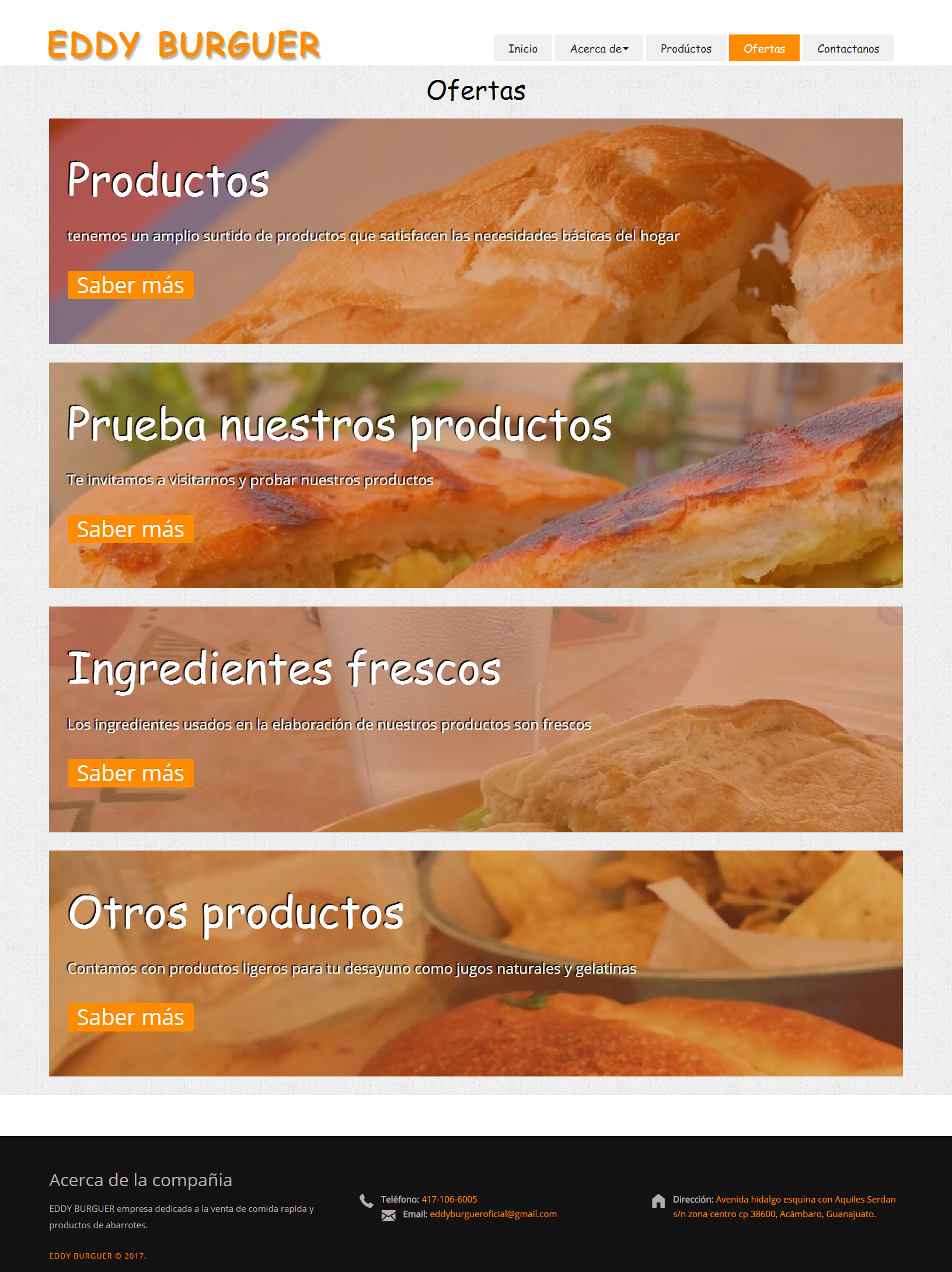
En esta pantalla se muestran todas las estaciones de la empresa solo se muestran las imágenes de las instalaciones no cuenta con formularios

**Productos**

****

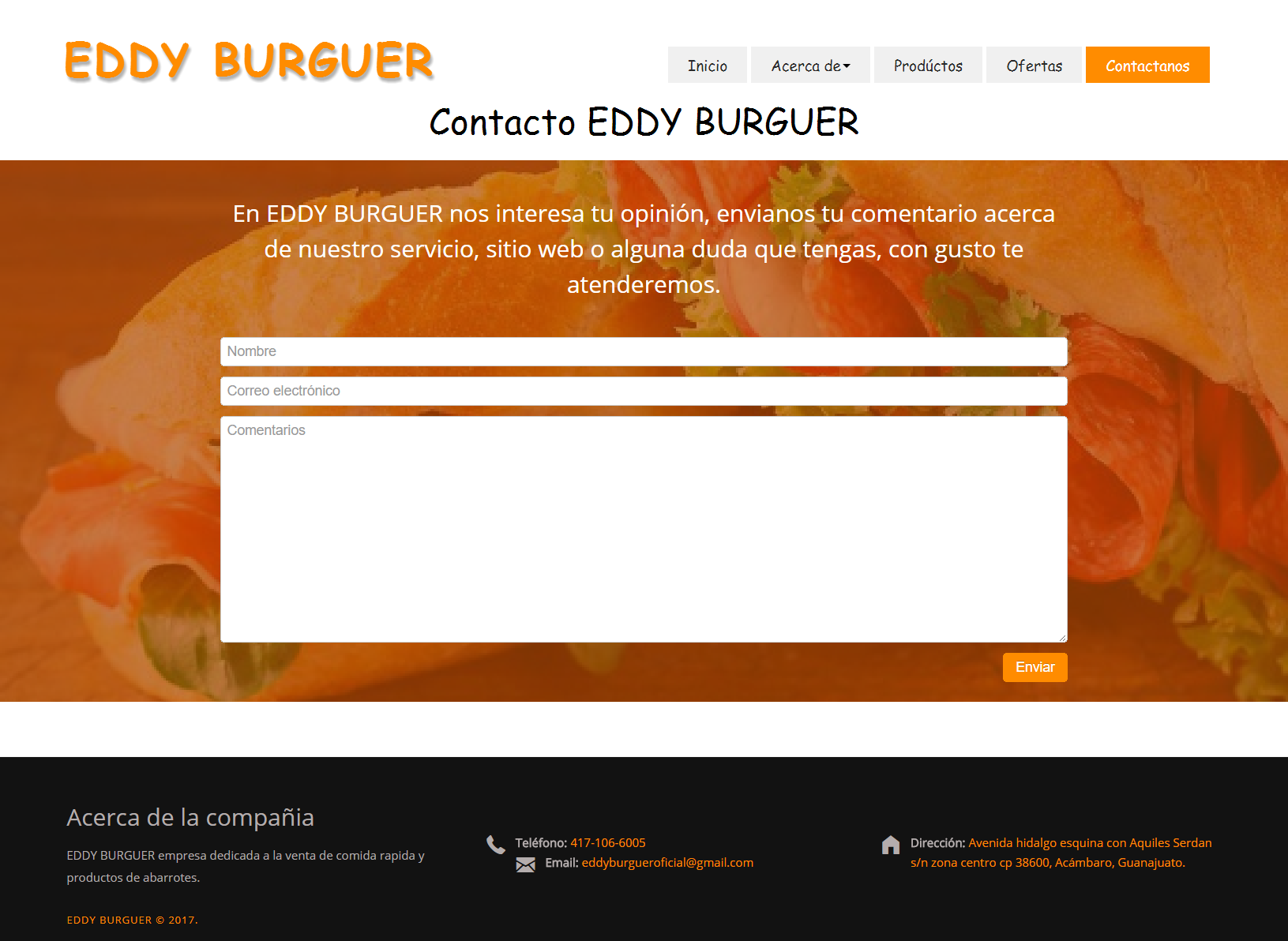
En esta pantalla se muestran los productos que hay en venta en la empresa así como también se muestra una descripción del producto.

**Ofertas**

****

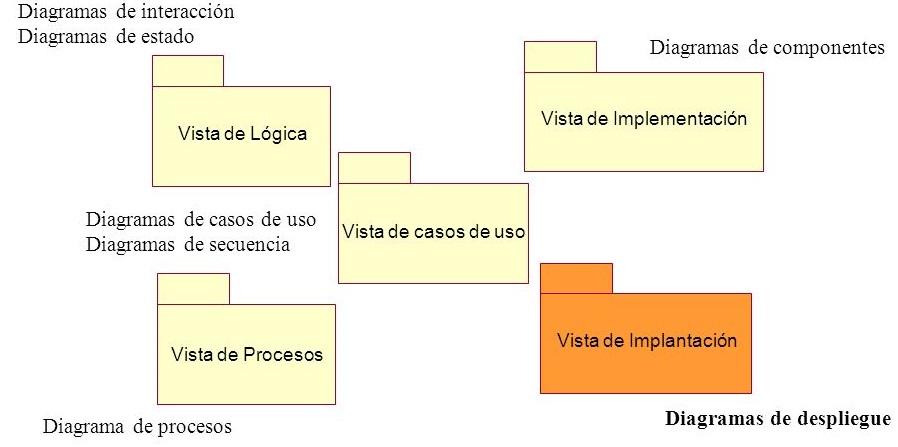
Aquí se muestra un apartado donde se dan a conocer las ofertas que hay en el día o en la semana.

**Contáctanos**

****

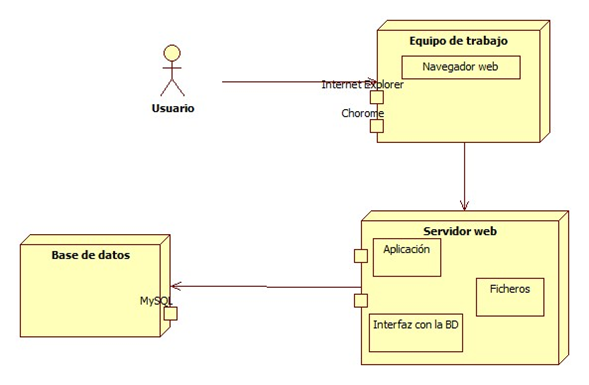
Este este último apartado donde se muestra un formulario para que el usuario pueda contactar a la empresa donde tiene que insertar su nombre su correo y su comentario.

## **VISTA DE IMPLEMENTACIÓN**

****

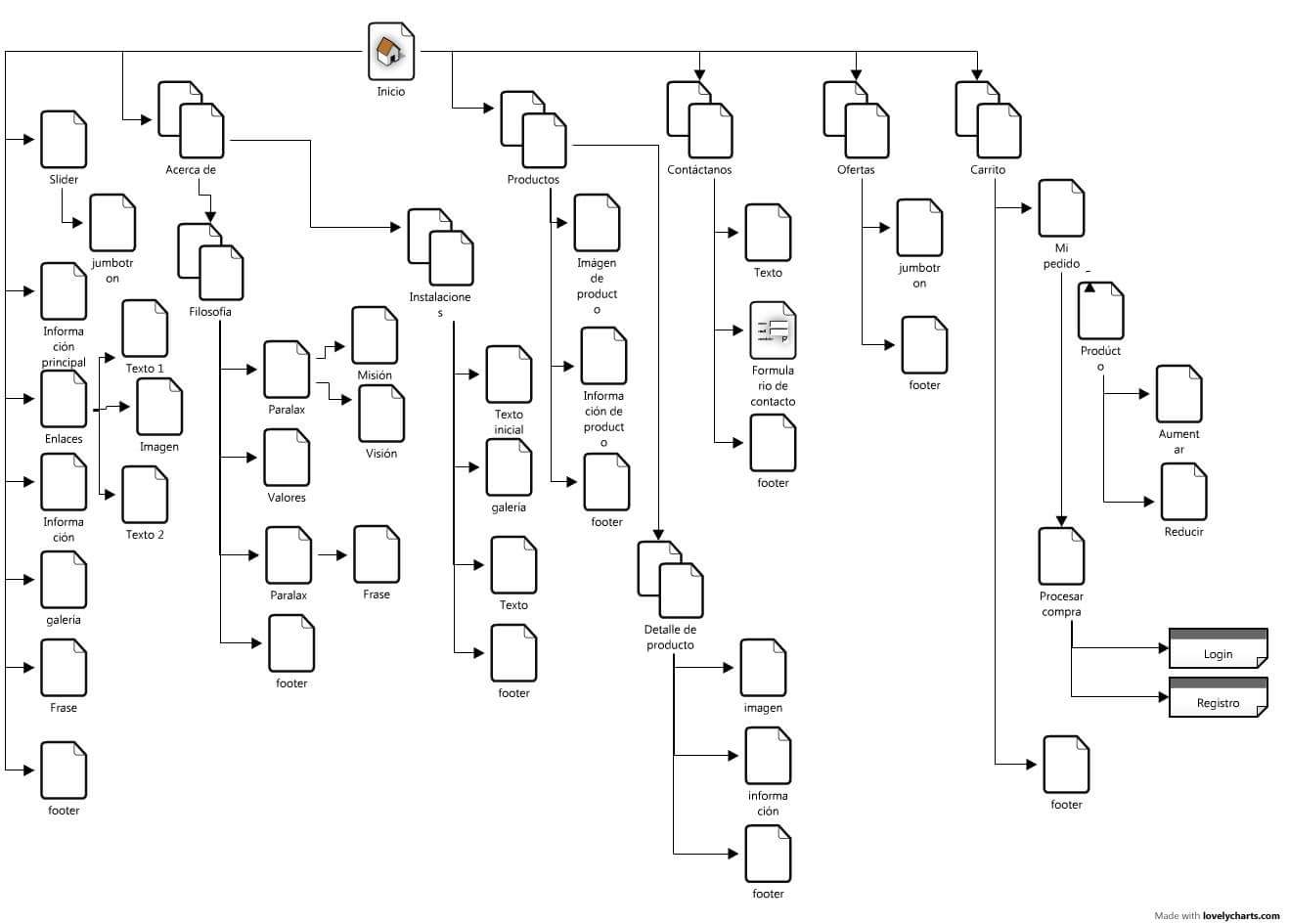
Se selecciona diferentes vistas pero la vista de implementación contiene el diagrama de componentes ya que tiene como se estructura de cómo está conformado el sitio web y que archivos debe tener en cada una de las carpetas del proyecto que se realizó.

## **VISTA DE DESPLIEGUE**

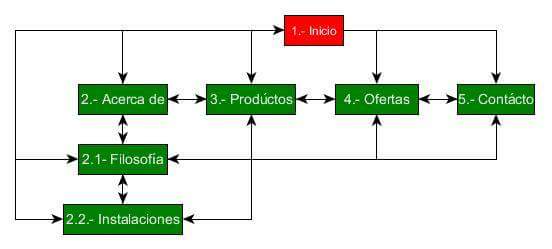
****

El usuario entrará a la aplicación web por medio de un navegador ya sea Chrome, Mozilla Fire Fox para conectarse al servidor web así se mostrará el sitio web en línea, la aplicación contará con su interfaz y con su base de datos que contendrá todos los productos que se encuentran dentro de la empresa.

El sitio web se pondrá dentro de un servidor donde se podrá modificar su base de datos por si hay algún error, así poder gestionar mejor los productos que se encuentran dentro de la base de datos.

****

**DIAGRAMA DE NAVEGACIÓN DEL SISTEMA**



## **GLOSARIO DE TÉRMINOS**

**Diagrama caso de uso:** Un diagrama de clase muestra un conjunto de clases, interfaces, y colaboraciones y sus relaciones entre ellos.

**Diagrama de componentes:** Un Diagrama de Componente es, como su nombre lo indica, un esquema o diagrama que muestra las interacciones y relaciones de los componentes de un modelo.

**Diagrama de navegación:** Los mapas de navegación proporcionan interfaz de navegación2 una representación esquemática interfaz de navegación1 de la estructura del hipertexto, indicando los principales conceptos incluidos en el espacio de la información y las interrelaciones que existen entre ellos.

**Interfaz:** Se denomina interfaz al conjunto de elementos de la pantalla que permiten al usuario realizar acciones sobre el Sitio Web que está visitando.

**MySQL:** ySQL es el servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB. MySQL AB es una empresa cuyo negocio consiste en proporcionar servicios en torno al servidor de bases de datos MySQL.

**Base de datos:** Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

**Diagrama de despliegue:** es un tipo de diagrama del [Lenguaje Unificado de Modelado](https://www.ecured.cu/Lenguaje_Unificado_de_Modelado) que muestran las relaciones físicas de los distintos nodos que componen un [sistema](https://www.ecured.cu/Sistema) y el reparto de los componentes sobre dichos nodos.

**Aplicación:** Programa preparado para una utilización específica, como el pago de nóminas

**Rol**: Función que alguien o algo desempeña