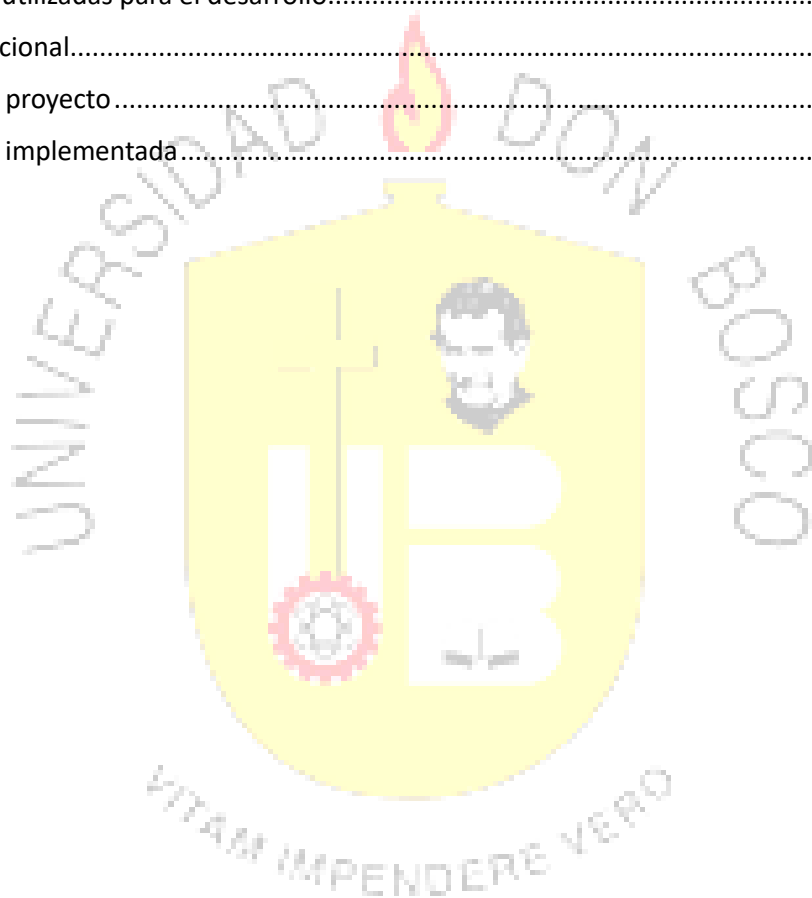


Manual Tecnico

Índice

| | |
|--|----|
| Manual Tecnico | 0 |
| Introducción | 2 |
| Objetivos | 3 |
| Objetivos Especifico | 3 |
| Objetivo Generales | 3 |
| Descripción general de los procesos a implementar en la aplicación web | 4 |
| Herramientas utilizadas para el desarrollo..... | 5 |
| Diagrama relacional..... | 6 |
| Estructura del proyecto..... | 7 |
| Programación implementada..... | 10 |



Introducción

En la Universidad Don Bosco (UDB), la experiencia gastronómica de los estudiantes, docentes y personal administrativo es fundamental para su bienestar y rendimiento. Para mejorar esta experiencia y facilitar el acceso a los servicios de alimentación dentro del campus, se propone el desarrollo de un Sistema de Control de Cafetines.

Este sistema tiene como objetivo principal organizar y optimizar la gestión de los locales de comida presentes en la Universidad, así como facilitar el proceso de pedido y reserva de productos para la comunidad universitaria. Además, se busca integrar funcionalidades especiales para la organización de eventos dentro del campus, asegurando una experiencia gastronómica fluida y satisfactoria para todos los usuarios.

El Sistema de Control de Cafetines no solo busca mejorar la eficiencia en la gestión de la alimentación dentro de la UDB, sino también contribuir al bienestar y comodidad de toda la comunidad universitaria, promoviendo así un ambiente propicio para el aprendizaje y el desarrollo personal y académico.

Objetivos

Objetivos Especifico

- Registro de locales y productos: Desarrollar un sistema para que los administradores puedan agregar, modificar y eliminar información sobre locales y productos disponibles en la Universidad.
- Funcionalidades de pedidos y reservas: Implementar una interfaz para que los usuarios registrados puedan hacer pedidos o reservas de productos, con opciones para seleccionar productos, horarios de retiro y métodos de pago.
- Gestión de eventos especiales: Integrar una función que permita a los usuarios organizar eventos especiales y realizar pedidos para dichos eventos, estableciendo montos de consumo y procesos de pago anticipado.

Objetivo Generales

- Seguridad de datos: Garantizar la seguridad y confidencialidad de la información mediante medidas como la encriptación de datos y la autenticación segura.
- Optimización de la experiencia del usuario: Mejorar la usabilidad y accesibilidad del sistema para usuarios y administradores, mediante pruebas de usabilidad y retroalimentación constante.
- Facilitación de la gestión administrativa: Desarrollar herramientas para que los administradores puedan gestionar locales, productos, pedidos y eventos, generando informes detallados para la toma de decisiones informadas.

Descripción general de los procesos a implementar en la aplicación web

1. Módulo de Autenticación y Gestión de Usuarios:

Este módulo permite a los usuarios registrarse e iniciar sesión en la plataforma de la cafetería. Los usuarios pueden tener uno de los tres roles: administrador, docente o alumno.

2. Módulo de Administración de Productos:

Este módulo está diseñado específicamente para el rol de administrador. Permite al administrador agregar, modificar o eliminar productos disponibles en la cafetería. Esto incluye la actualización de información del producto, como nombre, descripción, precio, y gestión de imágenes asociadas.

3. Módulo de Reservas y Compras:

Este módulo está disponible para los roles de docente y alumno. Permite a los usuarios buscar, reservar y comprar productos disponibles en la cafetería. Los usuarios pueden ver los productos disponibles, agregarlos al carrito de compras, realizar reservas para recoger en un momento específico o comprar productos directamente.

4. Módulo de Gestión de Reservas:

Permite a los usuarios visualizar y escoger sus reservas. Los usuarios pueden ver los productos disponibles para poder reservarlos.

Herramientas utilizadas para el desarrollo



Es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

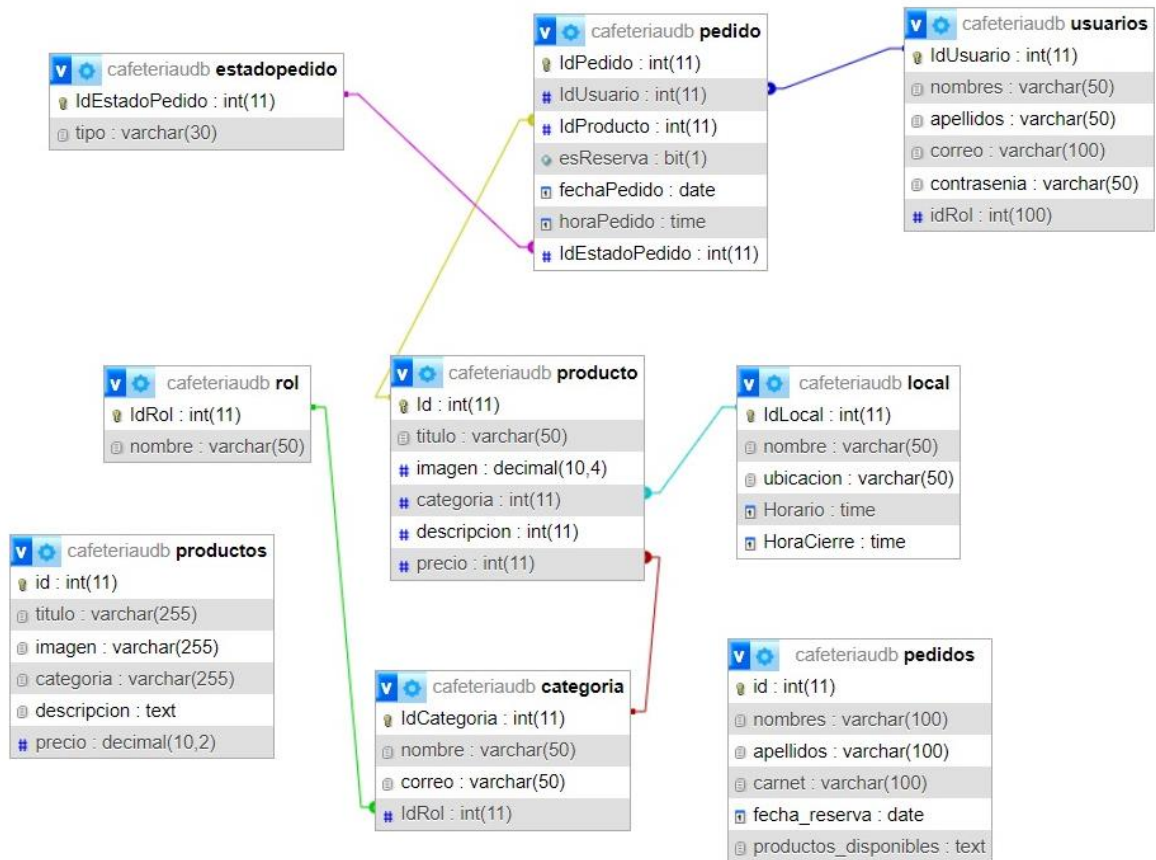


Ofrece como herramienta para alojar, revisar y gestionar proyectos de software utilizando el sistema de control de versiones Git.



phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL y MariaDB a través de páginas web, utilizando un navegador web.

Diagrama relacional



Estructura del proyecto

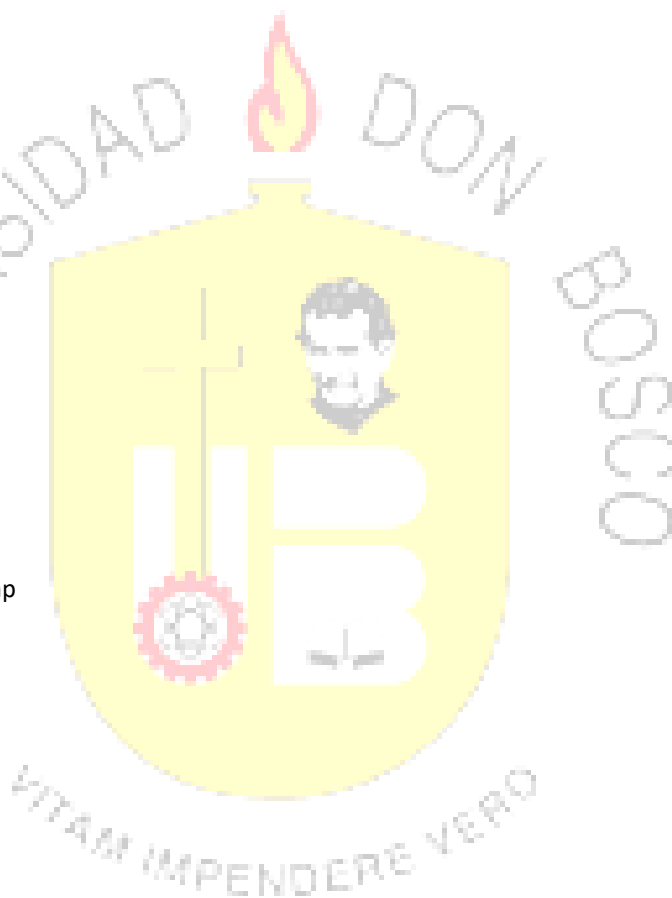
La estructura del proyecto en PHP sigue una organización lógica que facilita el desarrollo y mantenimiento del código. A continuación, describo la disposición de los archivos y carpetas del proyecto:

alumno/

- Cafetería_1.php
- Cafetería_2.php
- Cafetería_3.php
- Cafetería_4.php
- Conexión.php
- Reservacion.php
- Reservacion2.php
- Reservacion3.php
- Reservacion4.php
- Sobre_Nosotros.php

docente/

- Cafetería_1.php
- Cafetería_2.php
- Cafetería_3.php
- Cafetería_4.php
- Conexión.php
- Reservacion.php
- Reservacion2.php
- Reservacion3.php
- Reservacion4.php



- Sobre_Nosotros.php

empleado/

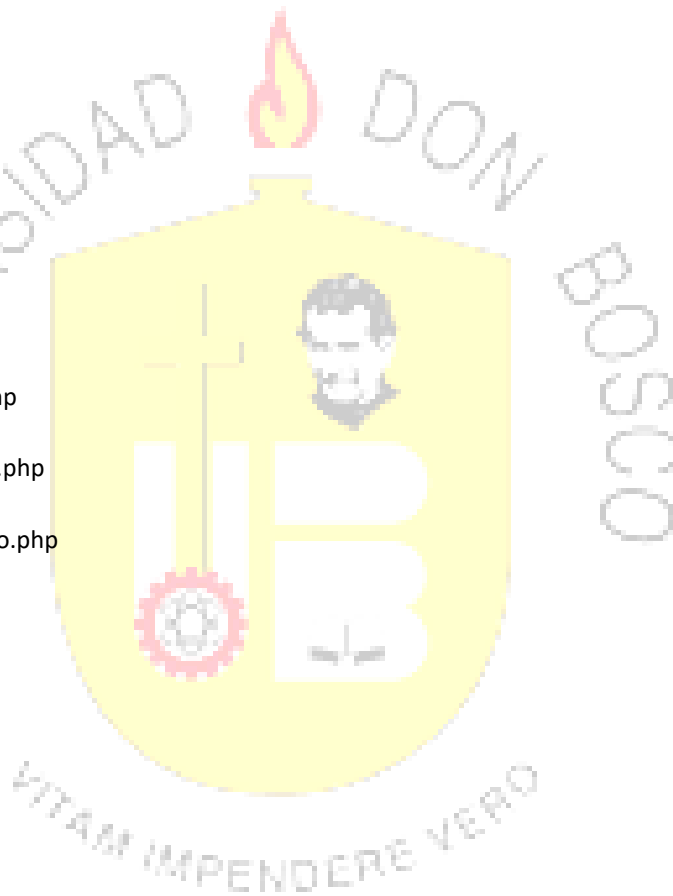
- Cafetería_1.php
- Cafetería_2.php
- Cafeteria_3.php
- Cafetería_4.php
- Conexión.php
- Reservacion.php
- Reservacion2.php
- Reservacion3.php
- Reservacion4.php
- Sobre_Nosotros.php
- eliminar_producto.php
- modificar_producto.php

estilos/

- SobreNosotros.css
- principal.css
- registro.css
- estilo.css
- estilos.css

imagen/

- cafetería 4
- cafetería 4
- “ ”



- “ ”

js/

- Scripts.js

/

- Avance de Proyecto
- Conexión.php
- Iniciarsesion.php
- LÉAME.md
- Registrarse.php
- index.php
- Navegación de archivos del repositorio LÉAME



Programación implementada

En nuestro proyecto, optamos por utilizar PHP puro y aprovechar al máximo las capacidades de la Programación Orientada a Objetos (POO) que ofrece este lenguaje. Aquí te explico cómo lo implementamos:

1. Clases y Objetos:

- Creamos clases para representar entidades y componentes del sistema. Por ejemplo, tenemos una clase **Usuario** que encapsula la información y las acciones relacionadas con los usuarios del sistema.
- Utilizamos la palabra clave **new** para instanciar objetos a partir de estas clases. Por ejemplo, podemos crear un objeto **usuario1** a partir de la clase **Usuario**.

2. Encapsulamiento:

- Aplicamos el principio de encapsulamiento para ocultar los detalles internos de nuestras clases y exponer solo la funcionalidad necesaria.
- Definimos propiedades y métodos como públicos, privados o protegidos según su nivel de acceso requerido.

3. Herencia:

- Aprovechamos la herencia para compartir atributos y métodos comunes entre clases relacionadas y evitar la duplicación de código.
- Por ejemplo, tenemos una clase base **Persona** de la cual heredan las clases **Alumno**, **Docente** y **Empleado**, cada una con sus propias características adicionales.

4. Polimorfismo:

- Implementamos el polimorfismo para permitir que objetos de diferentes clases respondan de manera diferente a los mismos métodos.
- Esto nos permite tratar objetos de diferentes tipos de manera uniforme, según su interfaz común.

5. Abstracción:

- Utilizamos la abstracción para definir interfaces y clases abstractas que establecen un conjunto de métodos que deben ser implementados por las clases concretas.

- Esto nos permite definir comportamientos comunes y establecer contratos claros para las clases que implementan estas interfaces.

Ventajas de Utilizar POO en PHP puro:

- **Organización y Mantenimiento:** La POO nos ayuda a organizar nuestro código en unidades lógicas y cohesivas, lo que facilita su comprensión y mantenimiento.
- **Reutilización del Código:** Podemos reutilizar clases y objetos en diferentes partes de nuestro proyecto, lo que reduce la duplicación de código y mejora la eficiencia del desarrollo.
- **Flexibilidad y Escalabilidad:** La POO nos brinda un enfoque flexible y escalable para el desarrollo de aplicaciones, lo que nos permite adaptar y expandir nuestro código según sea necesario.

La implementación de la Programación Orientada a Objetos en PHP puro nos ha permitido crear una estructura más organizada y modular para nuestro proyecto, lo que facilita su desarrollo y mantenimiento a largo plazo.

