

ENTREGA DE AVANCE 6 DEL PROYECTO GRUPAL
INFORME DE ENTREGA: PROYECTO FINAL DE INGENIERÍA DE SOFTWARE

INTEGRANTES:

- Falconi Davila Ashley
- Martillo Alvares John
- Ortiz Moran Anthony
- Ynga Alay Alexander
- Villon Chele Emily

GRUPO: B

FECHA DE ENTREGA: 12/1/2026

1. INFORMACIÓN GENERAL Y ACCESO AL PROYECTO

El presente documento detalla la entrega del **Sistema de Gestión de Clientes**, una aplicación web diseñada para el registro y administración segura de información empresarial. El código fuente, la documentación técnica y los archivos de base de datos están disponibles en el siguiente repositorio público:

Enlace de GitHub: [Repositorio del Proyecto - Sistema de Gestión de Clientes](#)

2. POLÍTICA DE MANTENIMIENTO DEL PROYECTO

Para garantizar que el sistema funcione correctamente a lo largo del tiempo y sea fácil de reparar, se han establecido las siguientes políticas basadas en la técnica de **Tolerancia a Fallos**:

2.1. Gestión de Errores y Excepciones

El sistema cuenta con una "red de seguridad" (bloques *try-catch*) que captura cualquier fallo inesperado. Si ocurre un error, el programa no se detiene; en su lugar, muestra un mensaje sencillo al usuario y permite que la aplicación siga funcionando.

2.2. Registro Técnico de Fallos (Logging)

Cada vez que el sistema detecta un problema interno, este se anota automáticamente en un archivo llamado errores.log. Esto permite a los programadores revisar el historial de fallos sin interrumpir el uso del sistema.

2.3. Validación y Limpieza de Datos

Como medida preventiva, toda la información que el usuario ingresa se revisa dos veces: primero en el navegador y luego en el servidor. Esto asegura que no entren datos falsos, incompletos o maliciosos que puedan dañar la base de datos.

2.4. Seguridad de la Información

Se aplican "consultas preparadas" en la base de datos. Esta política impide que personas externas intenten robar o modificar información a través de ataques de inyección SQL.

3. INSTRUCCIONES PARA LA APERTURA DEL PROYECTO (IDE: VISUAL STUDIO CODE)

Para visualizar y ejecutar el código correctamente, siga estos pasos detallados:

3.1. Requisitos Previos

1. **XAMPP:** Necesario para ejecutar el servidor Apache (PHP) y el servidor de base de datos (MySQL).
2. **Visual Studio Code (VS Code):** Editor de texto recomendado para revisar el código comentado.

3.2. Configuración del Entorno

1. **Ubicación:** Copie la carpeta completa del proyecto en la ruta:
C:\xampp\htdocs\SistemaGestionClientes.
2. **Abrir en VS Code:**
 - a. Inicie Visual Studio Code.
 - b. Vaya al menú **Archivo > Abrir carpeta**.
 - c. Seleccione la carpeta del proyecto mencionada arriba.
3. **Base de Datos:**
 - a. Inicie Apache y MySQL desde el Panel de Control de XAMPP.
 - b. Desde su navegador, entre a <http://localhost/phpmyadmin>.
 - c. Cree una nueva base de datos llamada bd_clientes.
 - d. Importe el archivo SQL que se encuentra en la carpeta /scripts/ del proyecto.

3.3. Ejecución del Sistema

Para ver el sistema en funcionamiento, abra su navegador y escriba la siguiente dirección:

<http://localhost/SistemaGestionClientes/CapaPresentacion/index.html>

4. BITÁCORA DE ACTIVIDADES DEL GRUPO

A continuación, se presenta el progreso detallado de los componentes implementados en el proyecto:

Módulo / Componente	Descripción de la Actividad	Estado	Avance
Base de Datos (MySQL)	Diseño de tablas, relaciones y carga de datos de prueba.	Completado	100%
Capa de Datos (PHP)	Creación de la conexión segura y métodos de consulta.	Completado	100%
Capa de Negocio	Implementación de las reglas para guardar y editar clientes.	Completado	100%
Capa de Presentación	Diseño de la interfaz de usuario con HTML y CSS.	Completado	100%
Validaciones (JS)	Programación de alertas y revisión de formularios en tiempo real.	Completado	100%
Gestión de Excepciones	Configuración de mensajes de error amigables y archivos log.	Completado	100%
Documentación	Elaboración de manuales y comentarios detallados en el código.	Completado	100%

5. CONCLUSIÓN DE LA ENTREGA

El proyecto se entrega con el código fuente totalmente **comentado en español**, explicando la función de cada método y clase dentro de la arquitectura de capas (MVC). Se han realizado pruebas funcionales para asegurar que el registro, la edición y la eliminación de clientes operen sin inconvenientes bajo las políticas de mantenimiento establecidas.