

PLAN DE PRUEBAS – SISTEMA DE GESTIÓN DE CLIENTES

1. OBJETIVO

El objetivo de este plan de pruebas es asegurar que el sistema de gestión de clientes funcione correctamente, cumpla con los requisitos funcionales y no funcionales, y brinde una experiencia segura y eficiente al usuario final. Se busca identificar posibles errores, validar la integración entre los diferentes componentes y garantizar la calidad del producto antes de su entrega.

2. ALCANCE

El plan de pruebas abarca las siguientes funcionalidades y componentes del sistema:

- Registro de nuevos clientes en la base de datos.
- Edición y actualización de datos de clientes existentes.
- Consulta y visualización de la lista de clientes y detalles individuales.
- Eliminación de registros de clientes.
- Validaciones de formularios y mensajes de error.
- Interfaz de usuario (HTML, CSS, JavaScript).
- Seguridad en el manejo de datos y restricciones de acceso.
- Integración entre la interfaz, la lógica en PHP y la base de datos MySQL.
- Rendimiento y tiempos de respuesta del sistema.

3. ESTRATEGIA DE PRUEBAS

a. Pruebas Funcionales

- Verificar que el formulario de registro permita ingresar todos los campos requeridos y almacene correctamente la información.
- Probar la edición de datos de un cliente y confirmar que los cambios se actualicen en la base de datos.
- Consultar la lista de clientes y asegurarse de que se muestren todos los registros existentes.
- Visualizar los detalles de un cliente específico y comprobar que la información sea precisa.
- Eliminar un cliente y verificar que el registro desaparezca tanto de la interfaz como de la base de datos.
- Validar que los mensajes de éxito y error sean claros y específicos según la acción realizada.

b. Pruebas de Interfaz de Usuario

- Revisar que los formularios sean accesibles, tengan etiquetas claras y validaciones en el frontend.
- Comprobar la correcta disposición de los elementos en la pantalla (responsive design).
- Verificar la navegación entre las diferentes vistas del sistema (listado, registro, edición, detalles).
- Probar la compatibilidad en diferentes navegadores (Chrome, Firefox, Edge).

c. Pruebas de Seguridad

- Intentar acceder a páginas restringidas sin autenticación y verificar que el sistema lo impida.
- Probar la validación de entradas para evitar ataques de inyección SQL y scripts maliciosos.
- Comprobar que los datos sensibles no sean visibles ni accesibles desde el frontend.
- Revisar que las contraseñas (si aplica) se manejen de forma segura.

d. Pruebas de Rendimiento

- Registrar y consultar un gran número de clientes para evaluar la velocidad de respuesta.
- Medir el tiempo de carga de las páginas principales bajo diferentes condiciones de red.
- Observar el comportamiento del sistema ante múltiples solicitudes simultáneas.

e. Pruebas de Integración

- Verificar que los datos ingresados desde la interfaz se almacenen correctamente en la base de datos.
- Confirmar que los cambios realizados en la base de datos se reflejen en la interfaz de usuario.
- Probar la comunicación entre los archivos PHP, la base de datos MySQL y los scripts de JavaScript.

4. CRITERIOS DE ACEPTACIÓN

- Todas las funcionalidades deben ejecutarse sin errores ni interrupciones.
- Los datos deben guardarse, actualizarse, consultarse y eliminarse correctamente.
- La interfaz debe ser clara, intuitiva y fácil de usar.
- El sistema debe ser seguro ante intentos de manipulación o acceso no autorizado.
- El rendimiento debe ser adecuado para el uso esperado.

5. HERRAMIENTAS SUGERIDAS

- Navegador web (Chrome, Firefox, Edge) para realizar pruebas manuales de la interfaz.
- Gestor de base de datos como phpMyAdmin para verificar los registros y la integridad de los datos en MySQL.
- Herramientas para medir tiempos de carga (opcional): PageSpeed Insights, DevTools del navegador.