



Nombre:

Jose fabian

Matricula:

2024-0181

Materia:

Programación III

Profesor:

Ing. Kelyn Tejada

Carrera:

Desarrollo de Software

INTRODUCCIÓN

En este documento se presenta el desarrollo de un **sistema modelo de librería online**, construido con PHP, MySQL y PDO, que forma parte del proyecto final de la asignatura Programación III.

El objetivo principal es aplicar la metodología **Agile–Scrum** en un proyecto de software real, definiendo estrategia de trabajo, planificación, historias de usuario, plan de pruebas y evidencias de automatización.

El sistema permite administrar un catálogo de libros y autores almacenados en una base de datos, mostrando pantallas como **inicio**, **listado de libros**, **autores** y **contacto**, además de un **login** para el administrador del sistema. A partir de este sistema se organizan las actividades del proyecto y se diseñan las pruebas manuales y automatizadas que validan su funcionamiento.

1.2 Nombre del proyecto de software

El sistema desarrollado se denomina:

Librería Online – Sistema de Gestión de Catálogo de Libros y Autores

Es un portal web que permite visualizar y gestionar la información de una librería académica de manera centralizada.

1.3 Tecnología para aplicar

Para implementar el sistema de librería online se utilizan las siguientes tecnologías:

- **Lenguaje de programación:** PHP 8 con programación estructurada.
- **Conexión a base de datos:** PDO (PHP Data Objects).
- **Base de datos:** MySQL, utilizando la base dblibreria.
- **Servidor web:** Apache, a través de XAMPP en entorno local.
- **Frontend:** HTML5, CSS3 y **Bootstrap** para el diseño responsivo.
- **Gestión del código:** Git y repositorio en GitHub.
- **Navegador de pruebas:** Google Chrome / Microsoft Edge.
- **Pruebas automatizadas:** Selenium WebDriver con ChromeDriver.

Estas herramientas permiten construir un sistema web sencillo pero completo, y además facilitan la automatización de pruebas sobre la interfaz.

1.4 Objetivo del proyecto

El objetivo general del proyecto es:

Desarrollar e implementar un sistema web de librería online que permita gestionar el catálogo de libros y autores, y mostrar la información al usuario de forma clara, rápida y organizada, aplicando la metodología Agile–Scrum y técnicas de pruebas manuales y automatizadas.

Objetivos específicos:

- Diseñar una interfaz web para listar libros y autores desde la base de datos dblibreria.
- Implementar un módulo de autenticación para el administrador del sistema.
- Permitir la consulta del catálogo mediante filtros y listados.
- Aplicar la metodología Scrum para planificar y organizar el desarrollo.
- Diseñar un plan de pruebas que incluya casos manuales y automatizados con Selenium.

1.5 Alcance del proyecto

Incluido en el alcance:

- Página de **inicio** con presentación de la librería y botones de acceso a libros y autores.
- Módulo de **login** para el administrador (pantalla de bienvenida con credenciales de acceso).
- Página de **listado de libros**, con información como código, título, tipo, autor, editorial, precio y total de ventas.
- Página de **autores**, donde se muestran los autores registrados con su código y una breve descripción.
- Página de **contacto**, con información de la librería y un formulario para enviar mensajes (guardado en una tabla de base de datos).
- Conexion con la base de datos mediante PDO y consultas a las tablas de libros, autores y contacto.

Fuera del alcance (no se implementa en esta versión):

- Registro y mantenimiento de usuarios finales (clientes).
- Procesos de compra en línea o pasarelas de pago.
- Módulo avanzado de administración con CRUD completo desde la web.
- Envío real de correos desde el formulario de contacto.

El sistema se centra en **mostrar la información de la librería** desde la base de datos y servir como base para la aplicación de Scrum y la automatización de pruebas.

1.6 Cronograma del proyecto

A continuación se presenta un cronograma de trabajo estimado para el desarrollo del sistema y la documentación.

Nº	Actividad	Fecha inicio	Fecha fin	Responsable
1	Análisis del sistema modelo de librería	01/12/2024	01/12/2024	Jose Fabian
2	Revisión de la base de datos dblibreria	01/12/2024	01/12/2024	Jose Fabian
3	Diseño y ajustes de las pantallas (inicio, libros, etc.)	02/12/2024	02/12/2024	Jose Fabian
4	Implementación de la conexión PDO y consultas	02/12/2024	02/12/2024	Jose Fabian
5	Configuración de Scrum (épicas, historias y tareas)	03/12/2024	03/12/2024	Jose Fabian
6	Diseño del plan de pruebas manuales	04/12/2024	05/12/2024	Jose Fabian
7	Configuración de Selenium y creación de scripts	06/12/2024	6/12/2024	Jose Fabian
8	Ejecución de pruebas y corrección de errores	7/12/2024	7/12/2024	Jose Fabian
9	Elaboración del documento final en PDF	8/12/2024	8/12/2024	Jose Fabian
10	Grabación del video de demostración	9/12/2024	9/12/2024	Jose Fabian

1.7 Definición del primer Release

El **primer Release** del sistema de Librería Online incluye las siguientes funcionalidades:

- Acceso a la aplicación a través del **login de administrador**.
- Página de **inicio** con presentación del sistema y botones para ver libros y autores.
- Listado de **libros** obtenido desde la base de datos dblibreria, mostrando código, título, tipo, autores, editorial, precio y total de ventas.
- Listado de **autores**, con nombre, código e información básica de cada autor.
- Página de **contacto** con datos de la librería y formulario que guarda los mensajes en la base de datos.
- Conexión a la base de datos usando **PHP + PDO**, centralizada en un archivo de configuración.

Requerimientos funcionales del Release 1

- **RF01:** El sistema debe permitir al administrador iniciar sesión introduciendo usuario y contraseña válidos.
- **RF02:** El sistema debe mostrar una página de inicio con la descripción de la librería y accesos a libros, autores y contacto.
- **RF03:** El sistema debe listar los libros almacenados en la tabla correspondiente de dblibreria.
- **RF04:** El sistema debe listar los autores registrados en la base de datos.
- **RF05:** El sistema debe permitir filtrar o buscar libros por diferentes criterios (ej. tipo, título o autor).
- **RF06:** El sistema debe mostrar una página de contacto con información de la librería.
- **RF07:** El formulario de contacto debe permitir enviar un mensaje y almacenarlo en la base de datos.

Requerimientos no funcionales

- **RNF01 – Usabilidad:** La interfaz debe ser limpia, legible y fácil de navegar, utilizando un diseño responsivo con Bootstrap.
- **RNF02 – Rendimiento:** El tiempo de carga de las páginas principales debe ser razonable en entorno local (menos de 3 segundos).
- **RNF03 – Seguridad básica:** Las credenciales del administrador no deben mostrarse en texto claro en la interfaz y la sesión debe controlarse desde PHP.
- **RNF04 – Mantenibilidad:** La conexión a la base de datos debe estar centralizada en un archivo de configuración (config.php) para facilitar cambios futuros.
- **RNF05 – Compatibilidad:** El sistema debe funcionar correctamente en los navegadores Chrome y Edge en sus versiones actuales.

2. Metodología Scrum

La gestión del proyecto se realiza utilizando la metodología **Agile–Scrum**, tomando como base el sistema modelo “**Librería Online**” desarrollado en PHP y MySQL.

2.1 Definir tareas a ejecutar

A partir de los requerimientos del sistema, el trabajo se divide en las siguientes tareas específicas (backlog técnico):

1. Crear repositorio del proyecto en GitHub.
2. Configurar el entorno local (XAMPP, base de datos dblibreria, carpeta del proyecto).
3. Revisar y documentar la estructura de la base de datos (tablas de libros, autores y contacto).
4. Ajustar la página de inicio (textos, botones “Ver libros”, “Ver autores” y “Contacto”).
5. Configurar la página de login del administrador y validar credenciales.
6. Implementar la página de **listado de libros** leyendo desde MySQL con PDO.
7. Implementar la página de **autores** leyendo desde MySQL con PDO.
8. Implementar la página de **contacto** y el guardado de mensajes en la base de datos.
9. Diseñar y documentar los **requerimientos funcionales y no funcionales** del sistema.
10. Registrar épicas, historias de usuario y tareas en Jira / Azure DevOps.
11. Diseñar el **plan de pruebas manuales** (casos de prueba por pantalla).
12. Configurar Selenium WebDriver (instalación de controlador y script base).
13. Crear scripts de pruebas automatizadas (login, listado de libros, autores y contacto).
14. Ejecutar pruebas, registrar resultados y corregir errores encontrados.
15. Preparar el documento final del proyecto (PDF) con evidencias.
16. Grabar el video de demostración del sistema y las pruebas.

2.2 Definir el equipo de trabajo

Aunque el proyecto es individual, se simulan los roles típicos de Scrum:

- **Product Owner (PO):**
 - Rol asumido por el estudiante.
 - Define los requerimientos del sistema, prioriza las historias de usuario y valida el incremento entregado.
- **Scrum Master:**
 - Rol asumido también por el estudiante.
 - Se encarga de que se sigan las prácticas de Scrum, organiza las ceremonias y gestiona los impedimentos.
- **Development Team (Equipo de Desarrollo):**
 - Compuesto por una sola persona (el estudiante).
 - Responsabilidades: desarrollo en PHP, conexión con MySQL, maquetación frontend y creación de scripts de automatización.
- **QA / Tester:**
 - El mismo estudiante asume este rol.
 - Diseña los casos de prueba, ejecuta pruebas manuales y automatizadas y documenta los resultados.

2.3 Herramientas que usarían

Para soportar el trabajo del equipo Scrum se definen las siguientes herramientas:

- **Jira:** para registrar épicas, historias de usuario, tareas y dar seguimiento al sprint mediante un tablero Scrum.
- **Git y GitHub:** control de versiones del código fuente y almacenamiento del proyecto.
- **Visual Studio Code:** editor principal para el desarrollo en PHP, HTML y CSS.
- **XAMPP (Apache + MySQL):** entorno local para ejecutar el sistema de librería y la base de datos dblibreria.
- **Selenium WebDriver + ChromeDriver:** automatización de pruebas funcionales de la interfaz web.
- **Navegador Google Chrome / Edge:** ejecución y validación visual del sistema.

Estas herramientas permiten organizar el trabajo, mantener el código centralizado y garantizar la trazabilidad entre requerimientos, tareas y pruebas.

2.4 Definir las épicas

Las historias de usuario se agrupan en las siguientes **épicas**:

- **Épica 1 – Gestión de acceso y seguridad básica**
 - Incluye todas las historias relacionadas con el login del administrador y el cierre de sesión.
- **Épica 2 – Consulta del catálogo de libros**
 - Agrupa las historias de visualización, filtrado y búsqueda de libros.
- **Épica 3 – Gestión de autores**
 - Contiene las historias relacionadas con el listado y visualización de autores.
- **Épica 4 – Página de contacto y comunicación**
 - Incluye las historias vinculadas al formulario de contacto y al almacenamiento de mensajes.
- **Épica 5 – Calidad y pruebas**
 - Reúne las historias que tienen que ver con el plan de pruebas, evidencia y automatización con Selenium.

2.5 Ceremonias de Scrum

Para este proyecto se plantea un **sprint único de dos semanas**. Las principales ceremonias son:

- **Sprint Planning:**
 - Fecha sugerida: Día 1 del sprint.
 - Actividad: Se define el objetivo del sprint ("Entregar un sistema de librería online funcional con pruebas básicas") y se seleccionan las historias de usuario que se trabajarán.
- **Daily Scrum (daily stand-up):**
 - Reunión diaria simulada de 10 a 15 minutos.
 - Preguntas guía:
 - ¿Qué hice ayer?
 - ¿Qué haré hoy?
 - ¿Qué impedimentos tengo?
- **Sprint Review:**
 - Fecha sugerida: último día del sprint.
 - Actividad: Se muestra el incremento del sistema funcionando (login, listado de libros, autores y contacto) y se revisa que se cumplan las historias comprometidas.

- **Sprint Retrospective:**

- Se realiza después de la sprint review.
- Se analizan los aciertos y fallos del proceso: administración del tiempo, uso de herramientas, calidad del código, etc., para mejorar en futuros proyectos.

2.6 Historias de usuario

A continuación se presentan **10 historias de usuario** para el sistema de Librería Online. Cada historia incluye criterios de aceptación y puntos de historia (escala Fibonacci: 1, 2, 3, 5, 8).

HU01 – Iniciar sesión como administrador

- **Como** administrador del sistema,
- **quiero** iniciar sesión introduciendo usuario y contraseña,
- **para** poder acceder al panel de administración de la librería.

Criterios de aceptación:

1. El formulario debe tener campos para usuario y contraseña.
2. Si las credenciales son correctas, el sistema redirige a la página de inicio de la librería.
3. Si las credenciales son incorrectas, se muestra un mensaje de error sin revelar detalles sensibles.

Puntos de historia: 3

HU02 – Ver página de inicio de la librería

- **Como** usuario del sistema,
- **quiero** ver una página de inicio con información básica de la librería,
- **para** entender de qué trata el portal y acceder rápidamente a libros, autores y contacto.

Criterios de aceptación:

1. La página de inicio debe mostrar un mensaje de bienvenida y descripción breve.
2. Deben existir botones o enlaces visibles para “Ver libros”, “Ver autores” y “Contacto”.
3. El diseño debe ser responsivo y legible en distintos tamaños de pantalla.

Puntos de historia: 2

HU03 – Listar todos los libros

- **Como** usuario,
- **quiero** ver un listado de todos los libros disponibles en la librería,
- **para** conocer el catálogo completo que existe en la base de datos.

Criterios de aceptación:

1. El listado debe mostrar, como mínimo: código, título, tipo, autores, editorial, precio y total de ventas.
2. Los datos deben obtenerse desde la base de datos dblibreria usando PDO.
3. Si no hay libros registrados, se debe mostrar un mensaje informativo.

Puntos de historia: 3**HU04 – Filtrar libros por tipo o búsqueda**

- **Como** usuario,
- **quiero** filtrar o buscar libros por campos como título, autor o tipo,
- **para** encontrar más rápido el libro que necesito.

Criterios de aceptación:

1. Debe existir un campo de búsqueda y/o combo para seleccionar el tipo de libro.
2. Al aplicar los filtros, el listado se actualiza mostrando solo los resultados que coinciden.
3. Si no hay coincidencias, el sistema muestra un mensaje indicando que no se encontraron resultados.

Puntos de historia: 5**HU05 – Visualizar autores registrados**

- **Como** usuario,
- **quiero** ver una página con los autores registrados,
- **para** conocer qué escritores forman parte del catálogo de la librería.

Criterios de aceptación:

1. Cada autor debe mostrar al menos nombre, código y una breve descripción.
2. Los datos deben ser obtenidos desde la base de datos.
3. El diseño debe presentar a los autores de forma clara, por ejemplo en tarjetas o bloques.

Puntos de historia: 3**HU06 – Ver detalle resumido de un autor**

- **Como** usuario,
- **quiero** poder identificar rápidamente la información principal de cada autor,
- **para** decidir si me interesa revisar sus libros en el catálogo.

Criterios de aceptación:

1. Cada tarjeta de autor debe mostrar su nombre destacado y el código asociado.
2. Debe incluir una breve descripción o resumen del autor.
3. La información debe ser legible y ordenada visualmente.

Puntos de historia: 2

HU07 – Ver página de contacto de la librería

- **Como** visitante,
- **quiero** ver los datos de contacto de la librería (dirección, teléfono, correo),
- **para** poder comunicarme en caso de dudas o solicitudes.

Criterios de aceptación:

1. La página debe mostrar claramente nombre de la librería, dirección, teléfono y correo electrónico.
2. La información debe estar separada en una sección específica dentro de la pantalla de contacto.
3. El diseño debe ser consistente con el resto del sitio.

Puntos de historia: 2

HU08 – Enviar mensaje desde el formulario de contacto

- **Como** usuario del sitio,
- **quiero** enviar un mensaje desde el formulario de contacto,
- **para** comunicarme con la librería de forma rápida.

Criterios de aceptación:

1. El formulario debe tener campos para nombre, correo y mensaje.
2. Todos los campos son obligatorios; si falta alguno, se muestra un mensaje de validación.
3. Al enviar un mensaje válido, el sistema debe guardarlo en la tabla correspondiente de la base de datos.
4. El usuario debe recibir una notificación visual de que su mensaje fue enviado correctamente.

Puntos de historia: 5

HU09 – Cerrar sesión como administrador

- **Como** administrador,
- **quiero** poder cerrar sesión de forma segura,
- **para** evitar que otra persona use el sistema con mis credenciales.

Criterios de aceptación:

1. Debe existir una opción para cerrar sesión (botón o enlace) cuando el administrador está autenticado.
2. Al cerrar sesión, la información de la sesión debe destruirse.
3. El sistema debe redirigir al usuario nuevamente a la pantalla de login.

Puntos de historia: 1

HU10 – Automatizar pruebas básicas con Selenium

- **Como** responsable de calidad del sistema,
- **quiero** automatizar pruebas de los flujos principales con Selenium,
- **para** validar rápidamente que la librería online funciona después de cada cambio.

Criterios de aceptación:

1. Debe existir al menos un script que automatice el flujo de login del administrador.
2. Debe existir al menos un script que automatice la navegación al listado de libros y verifique que la tabla se muestra.
3. Debe existir al menos un script que automatice la navegación a la página de contacto y la visualización del formulario.
4. Las pruebas deben ejecutarse exitosamente en el navegador configurado (Chrome / Edge).

Puntos de historia: 8

3. Plan de Pruebas

El propósito del plan de pruebas es asegurar que el sistema **Librería Online** cumpla con los requerimientos funcionales y no funcionales definidos en la planificación. Se aplicaron **pruebas manuales** y **pruebas automatizadas** utilizando Selenium WebDriver.

3.1 Lista de requerimientos funcionales y no funcionales a probar

Los siguientes requerimientos serán validados durante el proceso de prueba:

Requerimientos Funcionales (RF)

- **RF01:** Validar que el administrador pueda iniciar sesión con usuario y contraseña correctos.
- **RF02:** Validar que la página de inicio se muestre correctamente con sus botones y texto.
- **RF03:** Verificar que el listado de libros cargue todos los datos desde la base de datos dblibreria.
- **RF04:** Verificar que la página de autores cargue los registros correspondientes.
- **RF05:** Verificar que el sistema permita buscar o filtrar libros (si aplica).
- **RF06:** Validar que la página de contacto se visualiza correctamente.
- **RF07:** Validar que el formulario de contacto permita enviar un mensaje y lo guarde en la base de datos.

Requerimientos No Funcionales (RNF)

- **RNF01:** Usabilidad — La interfaz debe ser clara y fácil de navegar.
- **RNF02:** Tiempo de carga — Las páginas principales deben cargar en menos de 3 segundos en entorno local.
- **RNF03:** Seguridad básica — No se debe acceder a páginas internas sin iniciar sesión.
- **RNF04:** Compatibilidad — El sistema debe funcionar correctamente en Chrome o Edge.
- **RNF05:** Mantenibilidad — La conexión PDO debe estar centralizada en un archivo independiente.

3.2 Criterios de aceptación y rechazo de pruebas

Criterios de Aceptación

Una prueba se considera **aceptada** cuando:

- El resultado obtenido coincide exactamente con el resultado esperado.
- La pantalla o funcionalidad responde sin errores visibles.
- La base de datos refleja correctamente los cambios (en el caso del formulario de contacto).
- El flujo de navegación se comporta según lo definido en los requerimientos.

Criterios de Rechazo

Una prueba se considera **rechazada** cuando:

- El sistema arroja mensajes de error inesperados.
- Los datos no se cargan desde la base de datos.
- La información se muestra incompleta o deformada.
- El comportamiento real no coincide con el comportamiento esperado.
- Un script automatizado no puede ejecutar el flujo sin intervención manual.

3.3 Herramientas de pruebas (y por qué se usan)

Para realizar las pruebas se utilizaron las siguientes herramientas:

- **Selenium WebDriver:** Permite automatizar flujos como login, navegación a libros, autores y contacto.
- **ChromeDriver:** Ejecuta los scripts de automatización en el navegador.
- **Google Chrome / Edge:** Navegadores para pruebas manuales y automatizadas.
- **XAMPP (Apache + MySQL):** Para ejecutar el sistema y consultar la base de datos.
- **VS Code:** Para modificaciones y revisión del código durante pruebas.

Justificación del uso de Selenium:

Permite repetir pruebas críticas sin error humano y validar rápidamente que el sistema sigue funcionando después de cualquier cambio.

3.4 Cronograma de ejecución de pruebas

En el sprint se define el siguiente cronograma:

Nº	Actividad de prueba	Tipo	Fecha	Responsable
1	Pruebas manuales de login, inicio y navegación	Manual	8/12/2024	Jose Fabian
2	Pruebas manuales de libros, autores y contacto	Manual	8/12/2024	Jose Fabian
3	Instalación y configuración de Selenium WebDriver	Técnica	8/12/2024	Jose Fabian
4	Diseño de scripts automatizados	Automatizada	9/12/2024	Jose Fabian
5	Ejecución de scripts automatizados (login, libros, autores)	Automatizada	9/12/2024	Jose Fabian
6	Pruebas de regresión	Manual + Selenium	9/12/2024	Jose Fabian

3.5 Casos de prueba

Caso de prueba CP-01 — Login válido

- **Requerimiento:** RF01
- **Precondiciones:** El usuario administrador existe.
- **Pasos:**
 1. Abrir la página de login.
 2. Ingresar usuario y contraseña correctos.
 3. Hacer clic en “Iniciar sesión”.
- **Resultado esperado:** El sistema permite el acceso y muestra la página de inicio.

Caso de prueba CP-02 — Listado de libros cargado correctamente

- **Requerimiento:** RF03
- **Pasos:**
 1. Iniciar sesión.
 2. Navegar a la sección “Libros”.
- **Resultado esperado:** Se muestra una tabla con los datos de la base de datos.

Caso de prueba CP-03 — Enviar mensaje desde contacto

- **Requerimiento:** RF07
- **Pasos:**
 1. Abrir la página de contacto.
 2. Llenar nombre, correo y mensaje.
 3. Presionar “Enviar”.
- **Resultado esperado:**
 - El mensaje se guarda en la base de datos.
 - Se muestra un mensaje de confirmación.

3.6 Equipos de pruebas y responsabilidades

Aunque el proyecto es individual, se definen los roles de forma simulada:

- **Tester:** Estudiante
 - Ejecuta pruebas manuales y automatizadas.
 - Registra defectos encontrados.
 - Valida comportamientos del sistema.
- **Desarrollador:** Estudiante
 - Corrige errores y vuelve a ejecutar pruebas.
- **Product Owner:** Estudiante
 - Valida que las funcionalidades cumplan con los requerimientos.

3.8 Plan de automatización de pruebas

Para aplicar automatización se seleccionan los flujos más importantes:

Pruebas automatizadas incluidas:

1. **Login exitoso del administrador**
2. **Navegar a la página de libros y validar que la tabla existe**
3. **Navegar a autores y validar que se cargan los registros**
4. **Navegar a contacto y verificar que el formulario existe**

Estrategia de automatización

- Se utiliza Selenium WebDriver con ChromeDriver.
- Cada script automatiza un flujo completo.
- Se utilizan asserts básicos para verificar elementos presentes en pantalla.
- Las pruebas se ejecutan después de cada ajuste del sistema (regresión).

Resultado esperado de la automatización

- Reducción del tiempo de prueba.
- Validación rápida de las funciones principales.
- Confirmación de que el sistema es estable.

3.9 Ejecución y demostración de pruebas

Las pruebas se documentan mediante:

- Capturas del sistema durante las pruebas manuales.
- Capturas o video corto de Selenium ejecutando los scripts.
- Registro de los resultados en tablas de pruebas.

El objetivo es demostrar que el sistema cumple con los requerimientos definidos en el Release 1.

4. Evidencias del sistema y de las pruebas

En esta sección se presentan las evidencias del funcionamiento del sistema **Librería Online** y de las pruebas funcionales realizadas de forma manual sobre las principales pantallas: **inicio, login, libros, autores y contacto**.

4.1 Evidencias del sistema en funcionamiento

A continuación se muestran capturas de pantalla del sistema modelo desarrollado en PHP y MySQL, ejecutándose en el servidor local.

Figura 1. Página de inicio de la Librería Online

En esta captura se observa la **pantalla de inicio** del sistema, donde se presenta el nombre de la librería, un mensaje de bienvenida y los botones de navegación hacia las secciones principales (libros, autores, contacto, etc.).

Esta página sirve como entrada al sistema y permite al usuario entender de qué trata la aplicación.

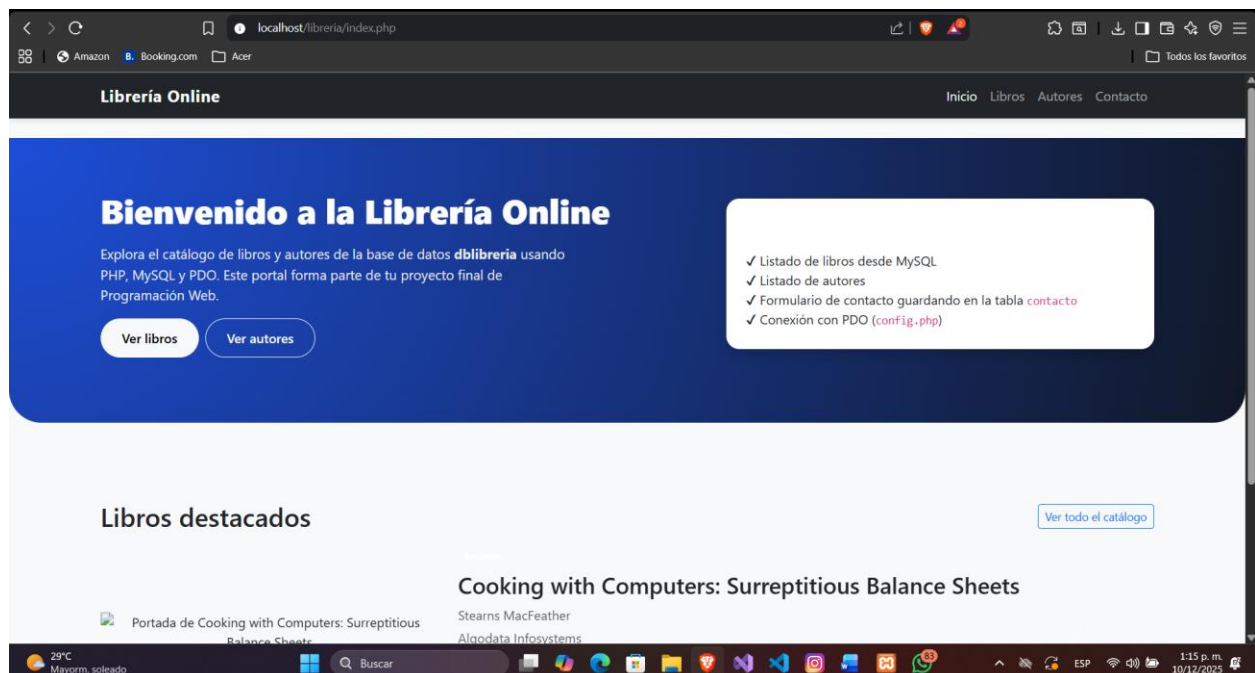


Figura 2. Pantalla de login del administrador

En esta captura se muestra la **pantalla de login**, donde el administrador debe introducir su usuario y contraseña para acceder al sistema.

Esta funcionalidad está relacionada con el requerimiento **RF01** (inicio de sesión del administrador) y garantiza un acceso controlado a ciertas opciones del sistema.

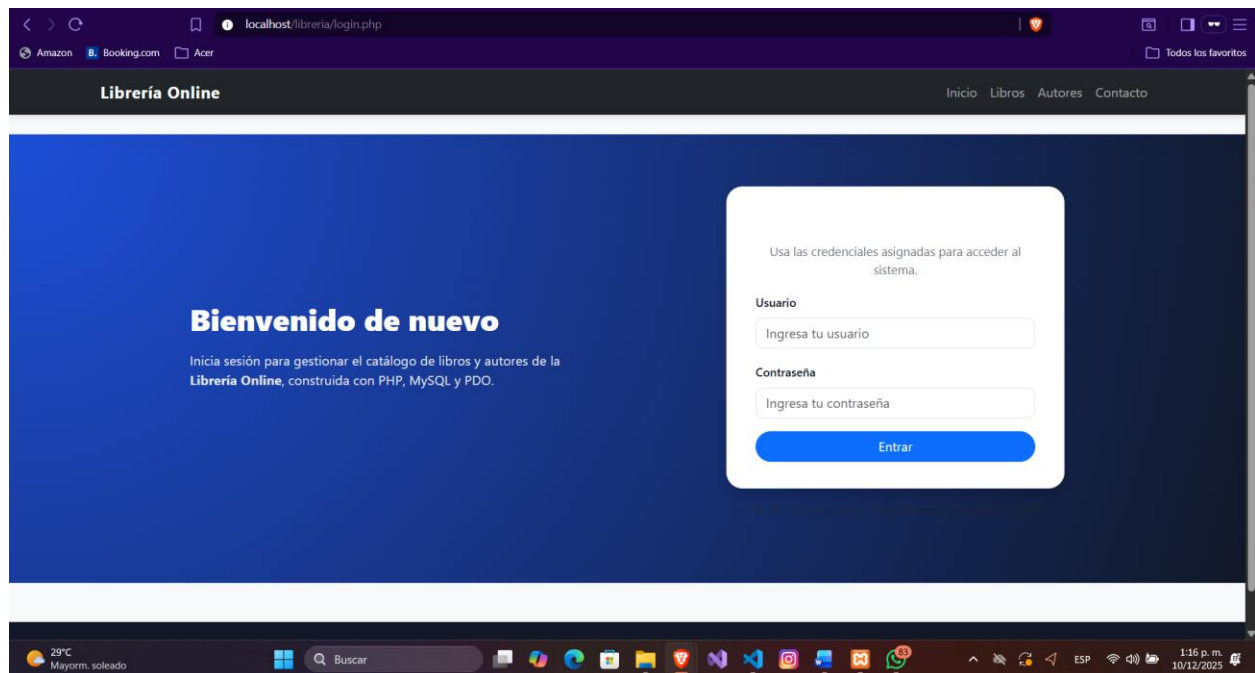
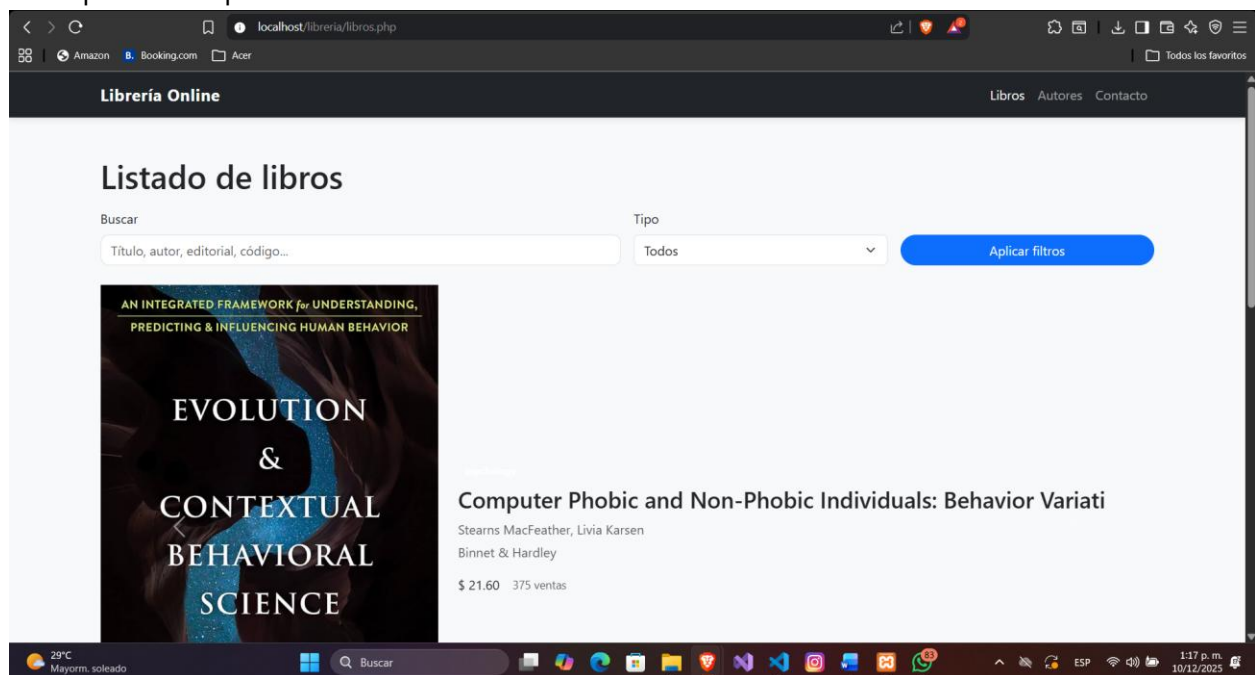


Figura 3. Listado de libros

En esta imagen se aprecia la **pantalla de libros**, donde se listan los registros almacenados en la base de datos dblibreria.

Se visualizan campos como código, título, tipo, autor, editorial, precio y total de ventas, lo cual corresponde al req



uerimiento **RF03** (listar libros).

localhost/libreria/autores.php

AmazonBooking.comAcer


Librería Online

LibrosAutoresContacto

Autores


Explora los autores disponibles en el catálogo de **dblibreria**.

22 autores registrados




Abraham Bennet
Código: 409-56-7008

Autor registrado en nuestra base de datos.
Contribuye al catálogo de la librería con
títulos de distintas temáticas.




Reginald Blotchet-Halls
Código: 648-92-1872

Autor registrado en nuestra base de datos.
Contribuye al catálogo de la librería con
títulos de distintas temáticas.



Cheryl Carson
Código: 238-95-776

Autor registrado en nuestra base de datos.
Contribuye al catálogo de la librería con



Michel DeFrance
Código: 722-51-5454

Autor registrado en nuestra base de datos.
Contribuye al catálogo de la librería con
títulos de distintas temáticas.

29°C
Mayorm. soleado

Buscar

1:19 p. m.
10/12/2025

Figura 5. Página de contacto

En esta imagen se observa la **página de contacto**, donde se muestran los datos de la librería (dirección, teléfono, correo, etc.) y el formulario para que el usuario envíe un mensaje.

Esta pantalla responde a los requerimientos **RF06** (mostrar página de contacto) y **RF07** (permitir el envío de un mensaje desde el formulario).

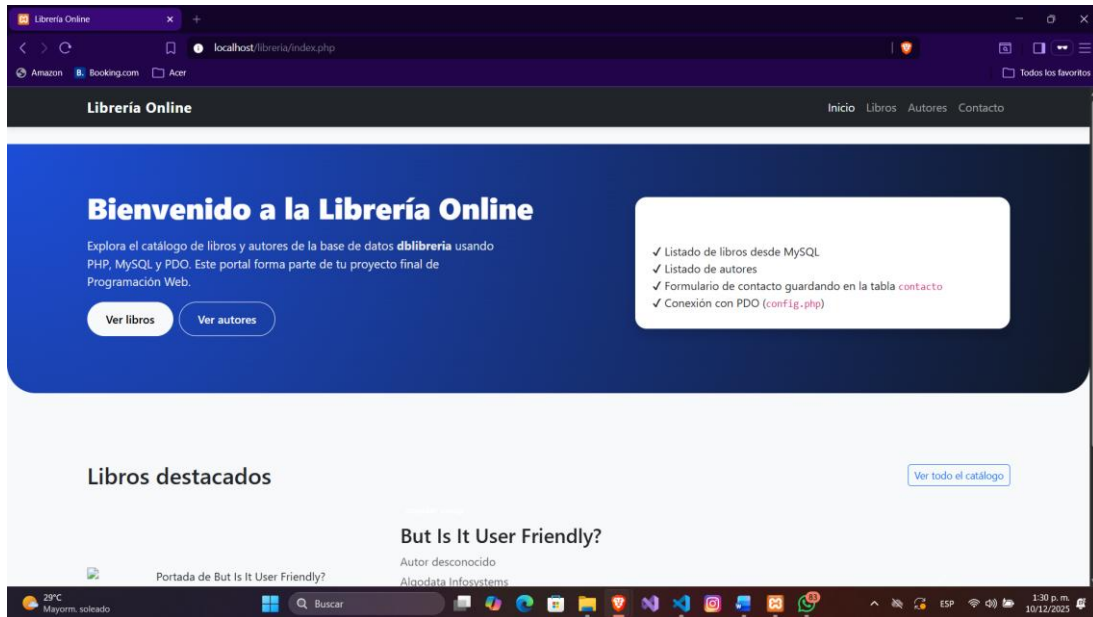
4.2 Evidencias de pruebas funcionales manuales

Además de mostrar el sistema en funcionamiento, se realizaron **pruebas funcionales manuales** sobre los flujos principales. A continuación se describen algunos casos de prueba y se relacionan con las capturas utilizadas como evidencia.

Prueba manual 1 – Login con credenciales válidas

- **Requerimiento probado:** RF01
- **Pasos ejecutados:**
 1. Abrir la página de login del sistema.
 2. Ingresar usuario y contraseña válidos del administrador.

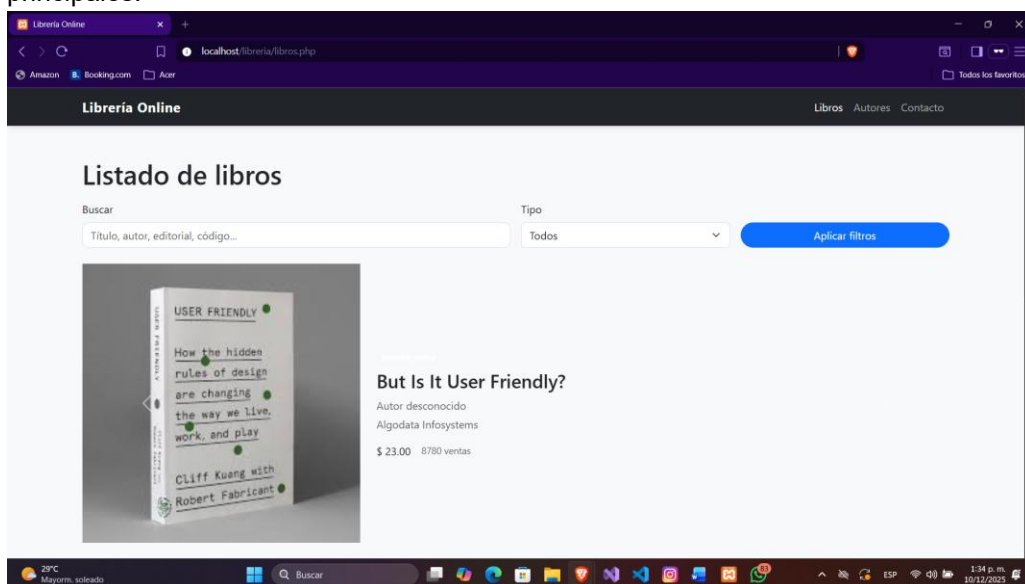
3. Presionar el botón “Iniciar sesión”.



- **Resultado esperado:** El sistema permite el acceso y muestra la página de inicio.
- **Resultado obtenido:** El sistema se comporta según lo esperado, redirigiendo correctamente a la página de inicio.

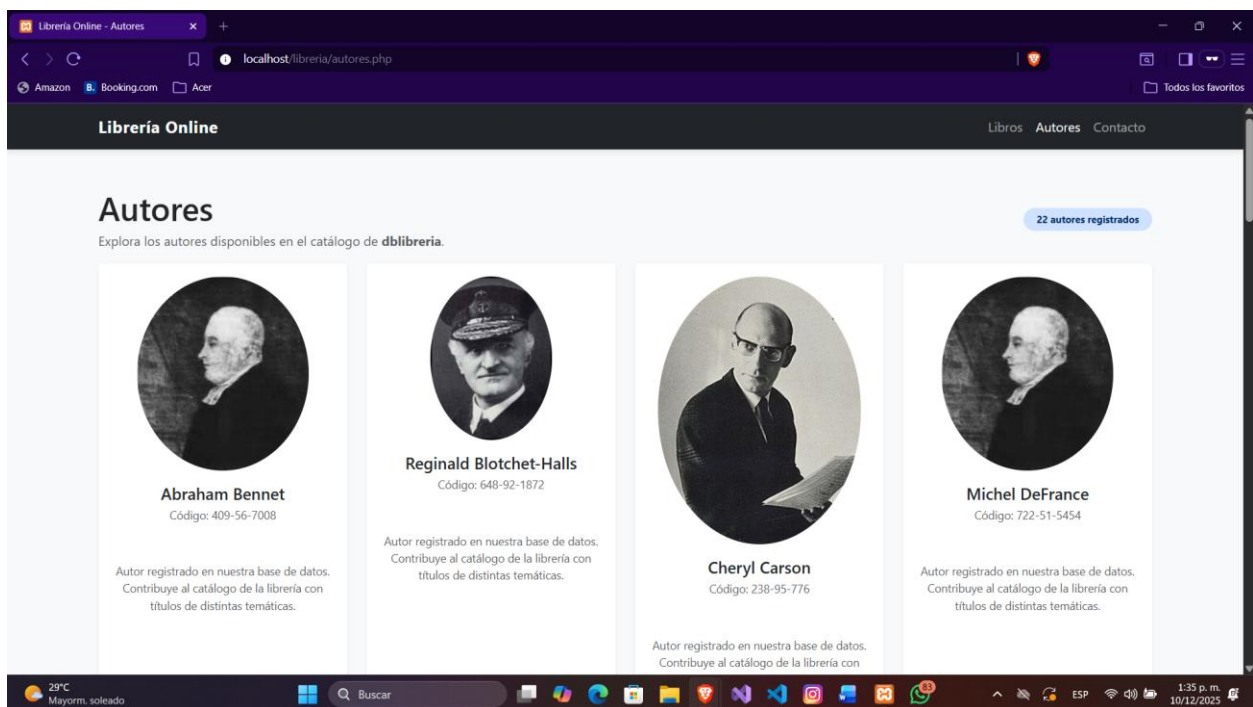
Prueba manual 2 – Visualización del listado de libros

- **Requerimiento probado:** RF03
- **Pasos ejecutados:**
 1. Desde la página de inicio, hacer clic en el botón o enlace “Libros”.
 2. Verificar que se muestre una tabla con los libros existentes.
- **Resultado esperado:** La tabla de libros se carga con los datos provenientes de la base de datos.
- **Resultado obtenido:** Los libros se muestran correctamente en la tabla con sus columnas principales.



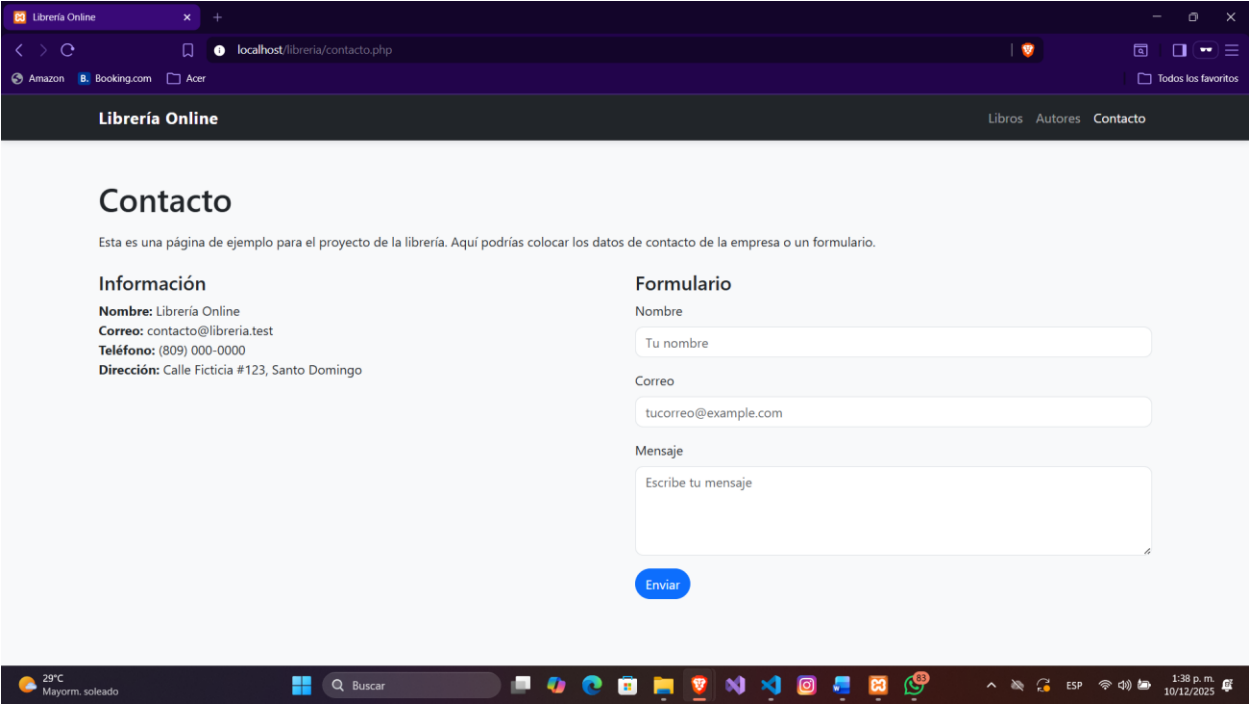
Prueba manual 3 – Visualización del listado de autores

- **Requerimiento probado:** RF04
- **Pasos ejecutados:**
 1. Desde el menú principal, acceder a la opción “Autores”.
 2. Verificar que se muestren los autores registrados.
- **Resultado esperado:** La pantalla debe mostrar claramente el nombre de los autores y su información relacionada.
- **Resultado obtenido:** La lista de autores se carga correctamente.



Prueba manual 4 – Acceso a la página de contacto

- **Requerimiento probado:** RF06
- **Pasos ejecutados:**
 1. Desde el menú o el pie de página, hacer clic en “Contacto”.
 2. Verificar que se muestren los datos de contacto de la librería y el formulario.
- **Resultado esperado:** La página debe mostrar la información de contacto y el formulario disponible.
- **Resultado obtenido:** La pantalla se muestra correctamente y el formulario es visible.



Prueba manual 5 – Envío de mensaje desde el formulario de contacto (si lo probaste)

Si llegaste a llenar y enviar el formulario:

- **Requerimiento probado:** RF07
- **Pasos ejecutados:**
 1. Abrir la página de contacto.
 2. Completar los campos de nombre, correo y mensaje.
 3. Presionar el botón “Enviar”.
- **Resultado esperado:** El sistema debe guardar el mensaje en la tabla correspondiente de la base de datos y mostrar una notificación.
- **Resultado obtenido:** *El sistema guardo satisfactoriamente el mensaje en la base de datos*

The screenshot shows a web browser window with the address bar displaying 'localhost/libreria/contacto.php'. The website has a dark header with the title 'Librería Online' and navigation links for 'Libros', 'Autores', and 'Contacto'. The main content area is titled 'Contacto' and includes a sub-header: 'Esta es una página de ejemplo para el proyecto de la librería. Aquí podrías colocar los datos de contacto de la empresa o un formulario.' Below this, there are two columns. The left column, titled 'Información', lists contact details for 'Librería Online': 'Nombre: Librería Online', 'Correo: contacto@libreria.test', 'Teléfono: (809) 000-0000', and 'Dirección: Calle Ficticia #123, Santo Domingo'. The right column, titled 'Formulario', contains three input fields: 'Nombre' (filled with 'José Alberto'), 'Correo' (filled with 'josealbertofabianvargas@gmail.com'), and 'Mensaje' (filled with 'Esto es una prueba manual'). A blue 'Enviar' button is located below the message field. The Windows taskbar at the bottom shows the date and time as '1:39 p. m. 10/12/2025'.

Librería Online

Libros Autores Contacto

Contacto

Esta es una página de ejemplo para el proyecto de la librería. Aquí podrías colocar los datos de contacto de la empresa o un formulario.

Información

Nombre: Librería Online
Correo: contacto@libreria.test
Teléfono: (809) 000-0000
Dirección: Calle Ficticia #123, Santo Domingo

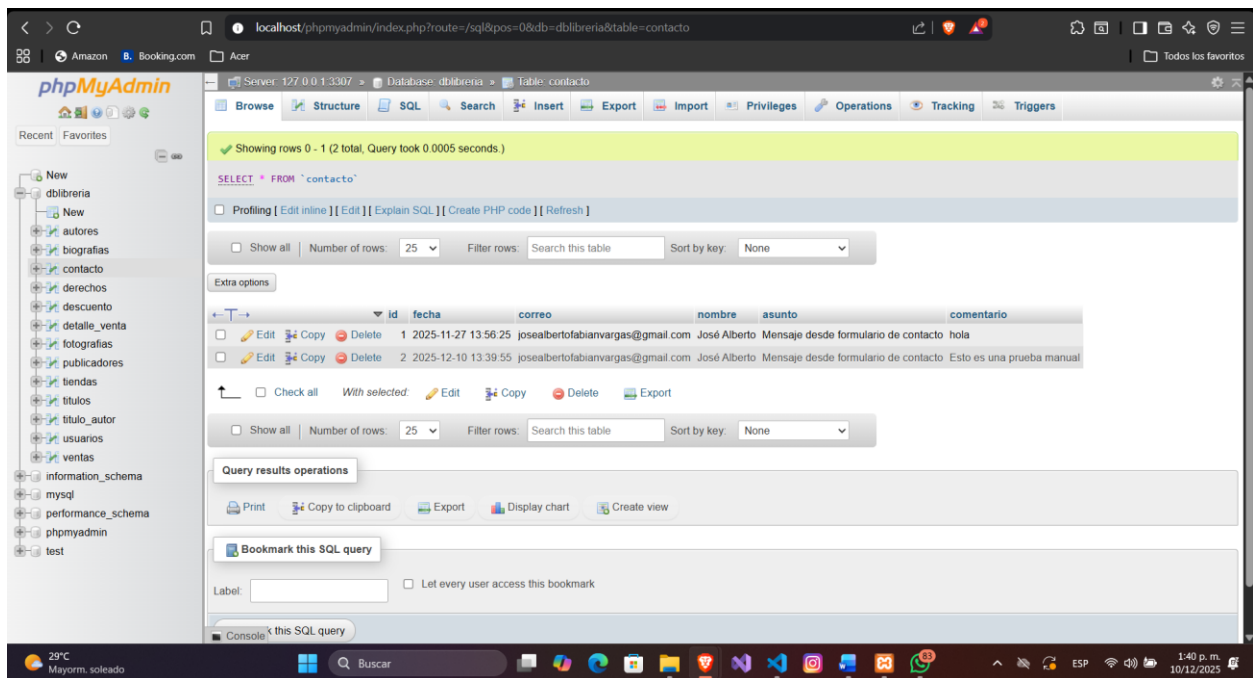
Formulario

Nombre
José Alberto

Correo
josealbertofabianvargas@gmail.com

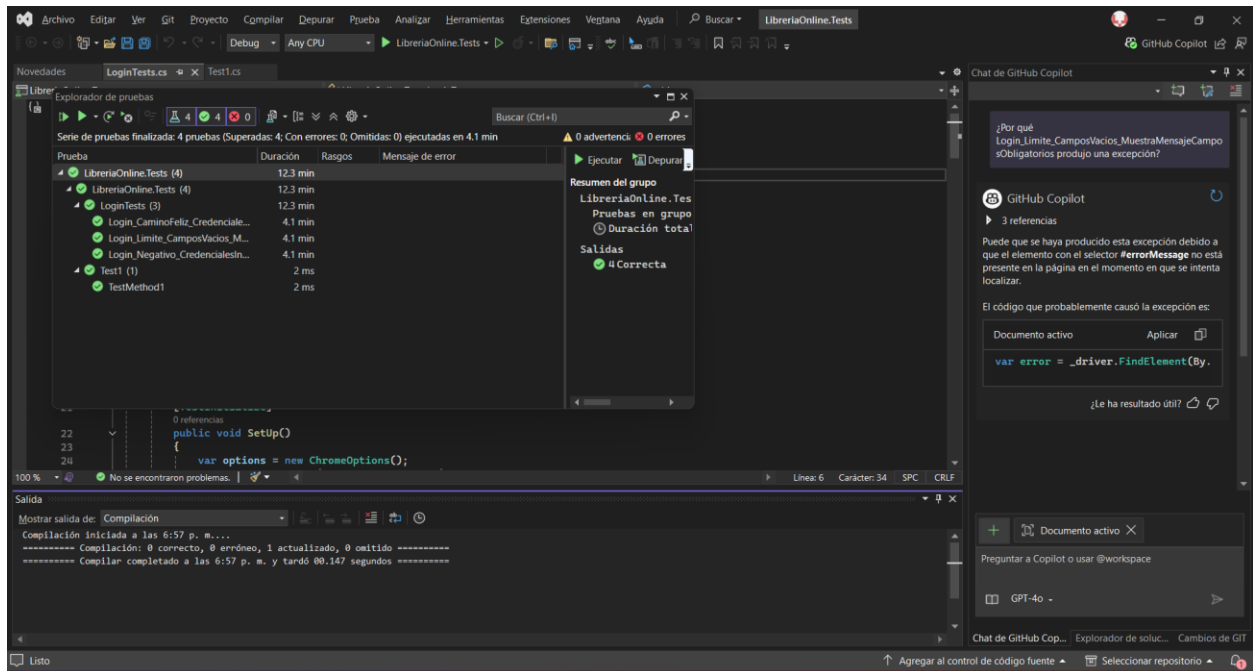
Mensaje
Esto es una prueba manual

Enviar



4.3 Evidencias de pruebas automatizadas con Selenium

Las pruebas automatizadas fueron implementadas utilizando **Selenium WebDriver**, **ChromeDriver** y el framework de pruebas **NUnit**. El objetivo de estas pruebas es validar el correcto funcionamiento del módulo de autenticación del sistema, verificando tanto escenarios positivos como negativos, así como límites funcionales relacionados con los campos del formulario de inicio de sesión.

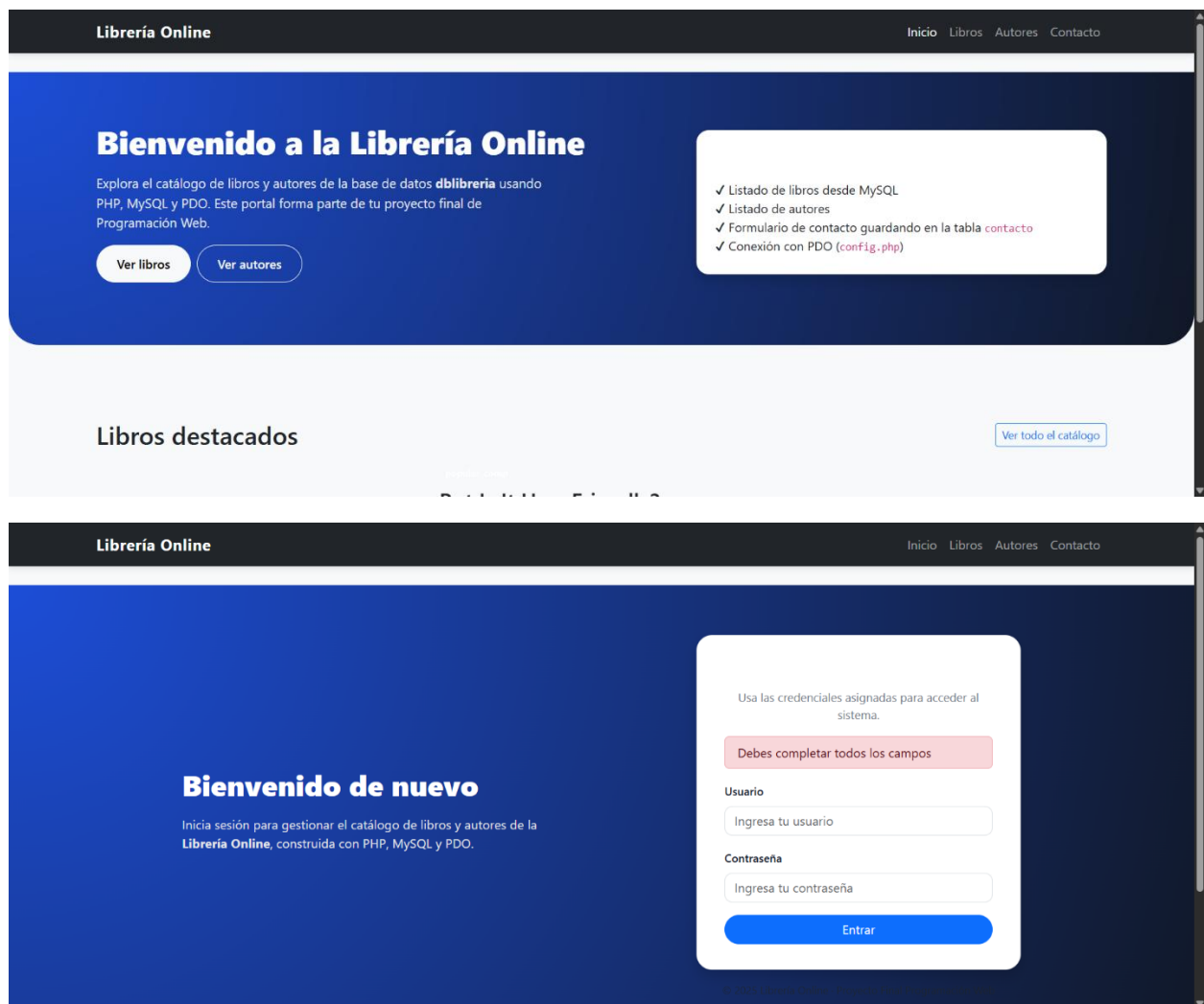


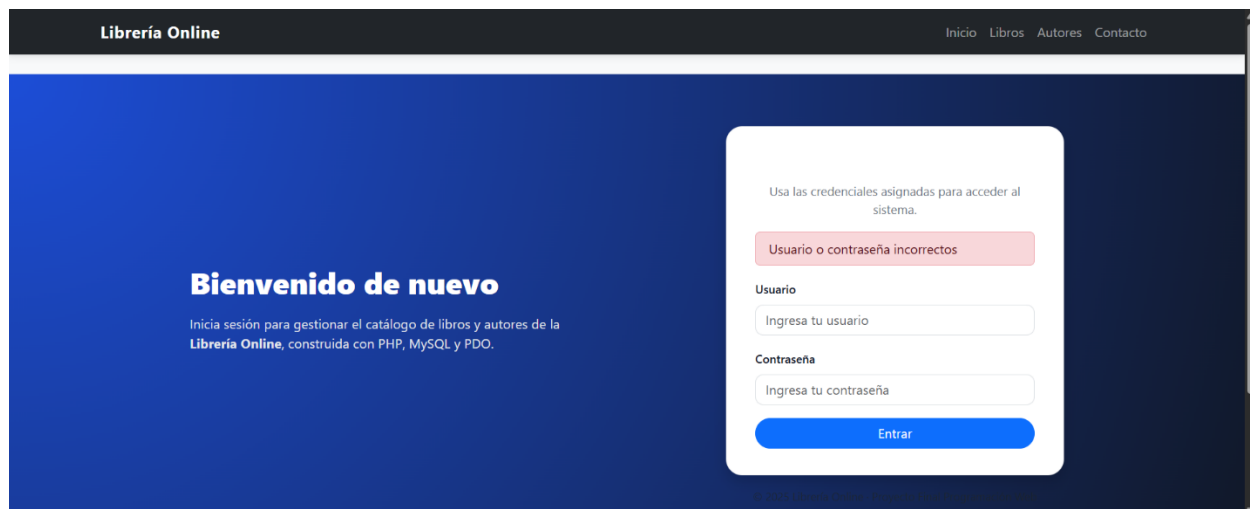
Resultados de las pruebas automatizadas

Todas las pruebas fueron ejecutadas correctamente dentro del Explorador de Pruebas de Visual Studio, validando:

- Inicio de sesión exitoso con credenciales válidas.
- Manejo de errores con credenciales incorrectas.
- Validación de campos vacíos.

Los resultados muestran que cada uno de los casos se ejecutó satisfactoriamente, sin errores en el flujo de autenticación.





5. Conclusiones

El desarrollo del sistema modelo **Librería Online** permitió aplicar de manera práctica los conceptos fundamentales de la asignatura Programación III, integrando PHP, MySQL y PDO para construir una aplicación funcional capaz de mostrar libros, autores y datos de contacto desde una base de datos real. A través de este proyecto fue posible comprender la importancia de planificar adecuadamente un software utilizando la metodología **Agile-Scrum**, definiendo tareas, épicas e historias de usuario para organizar el trabajo de forma estructurada.

Las pruebas funcionales manuales realizadas demostraron que el sistema cumple con los requerimientos establecidos para el primer release, ofreciendo una navegación clara, tiempos de respuesta adecuados y un comportamiento estable en cada una de sus pantallas principales. Además, se diseñó un plan de pruebas automatizadas con Selenium que servirá como base para validar de manera más eficiente los flujos críticos una vez se finalice la implementación completa del sistema.

En conclusión, este proyecto permitió integrar análisis, planificación, desarrollo y pruebas en un entorno controlado, reforzando las habilidades técnicas y metodológicas necesarias para el desarrollo de software profesional. El sistema, aunque sencillo, cumple su propósito como modelo de referencia y sienta las bases para futuras mejoras y automatización.