

Introducción

El Proyecto 1 tiene como finalidad aplicar de manera práctica los conocimientos adquiridos en el curso, mediante la construcción de una plataforma que simule la interfaz y los flujos principales de una plataforma bancaria en línea.

El proyecto está orientado al diseño, usabilidad y accesibilidad de las interfaces, siguiendo buenas prácticas de HTML5 semántico, CSS3 y JavaScript. Además, incorpora el enfoque mobile-first, garantizando que la experiencia inicial esté optimizada para dispositivos móviles y luego escale a tabletas y computadoras de escritorio.

Objetivo general

Desarrollar un prototipo visual interactivo que simule los flujos esenciales de una aplicación bancaria en línea, aplicando los principios de accesibilidad, usabilidad, diseño responsive y buenas prácticas de desarrollo web.

Objetivos específicos

- Implementar formularios de registro, inicio de sesión y recuperación de contraseña con validaciones y mensajes de error claros.
- Construir módulos visuales para cuentas, tarjetas y transferencias internas, con estructuras de datos simuladas.
- Aplicar un diseño responsive mobile-first, utilizando Flexbox y CSS Grid para la adaptación a distintos dispositivos.
- Incorporar elementos de accesibilidad (WAI-ARIA, atributos alt, tabindex, contraste, navegación por teclado) siguiendo estándares internacionales (WCAG 2.1 AA).
- Asegurar una estructura HTML semántica coherente, empleando etiquetas adecuadas y jerarquías correctas de encabezados.
- Establecer un estándar de componentes reutilizables (botones, tarjetas, tablas, formularios, modales) diseñados por los estudiantes.
- Consolidar buenas prácticas de organización de código, documentación y despliegue en plataformas como GitHub Pages o Netlify.

Parte 1 – Módulos del sistema

Registro

El sistema deberá permitir el registro de usuarios mediante un **formulario de inscripción** que contenga todos los campos obligatorios para identificar al cliente. El formulario deberá validar cada usuario y mostrar mensajes de error claros en caso de inconsistencias.

Al completar correctamente el formulario, el sistema deberá mostrar una **ventana emergente** con los **Términos y Condiciones** cargados desde un archivo PDF. El usuario deberá **leerlos y aceptarlos** mediante una casilla de verificación antes de poder finalizar el proceso. En caso de no aceptar, no se completará el registro.

Una vez enviados los datos, se mostrará una **pantalla de confirmación** que indique que el registro se realizó con éxito, y posteriormente se redirigirá a la página de **inicio de sesión**.

Campos requeridos

1. **Tipo de identificación** (select obligatorio)
 - Opciones: Nacional, DIMEX, Pasaporte
2. **Número de identificación** (input obligatorio)
 - Nacional: #-####-####
 - DIMEX: 11–12 dígitos numéricos
 - Pasaporte: 6–12 caracteres alfanuméricos en mayúscula
3. **Username** (input obligatorio)
 - 4–20 caracteres
 - Minúsculas, números, símbolos permitidos: . _ -
4. **Nombre completo** (nombre + apellido1 + apellido 2)
5. **Fecha de nacimiento** (date obligatorio)
6. **Correo electrónico** (input obligatorio)
 - Validar formato estándar, convertir a minúscula
7. **Teléfono** (input opcional)
 - Formato: +506 #####-####
8. **Contraseña** (input obligatorio, tipo password)
 - Mínimo 8 caracteres
 - Al menos 1 mayúscula, 1 minúscula, 1 dígito
9. **Confirmación de contraseña** (input obligatorio)
 - Debe coincidir con la contraseña

10. Aceptación de términos y condiciones (checkbox obligatorio)

- El sistema mostrará un **enlace o botón** “Ver términos y condiciones” que abrirá un **PDF en ventana modal**.
- Solo tras aceptar la casilla se podrá habilitar el botón “Registrar”.

Validaciones adicionales

- **Username:** único
- **Fecha de nacimiento:** cálculo de edad mínima de 18 años.
- **Correo:** regex estándar, error si no es válido.
- **Contraseña:** regex sugerida $\rightarrow ^{(?=[a-z])(?=[A-Z])(?=\d)\{8,\}}\$$.

Estados de la UI

- **Loading:** durante el envío del formulario.
- **Error:** mostrar mensajes claros (ej. “El correo no es válido”).
- **Éxito:** pantalla de confirmación y redirección a login.

Inicio de sesión

El sistema deberá permitir que el usuario se autentique mediante un **formulario de inicio de sesión** compuesto por su nombre de usuario y contraseña.

- Si los datos son correctos, se iniciará una sesión y el usuario será dirigido a la pantalla principal de su perfil.
- Si las credenciales son incorrectas, el sistema mostrará un mensaje de error genérico para no exponer detalles sensibles (ej. “Usuario o contraseña incorrectos”).
- Se deberá incluir la opción “**Mostrar/Ocultar contraseña**”, de manera que el campo de password pueda alternar entre texto visible y enmascarado.
- El formulario también debe incluir un enlace a “**¿Olvidó su contraseña?**”, que llevará al flujo de recuperación (definido en otra sección).

Estados de la UI

- **Loading:** al enviar credenciales.
- **Error:** credenciales inválidas \rightarrow “Usuario o contraseña incorrectos”.
- **Éxito:** redirección al dashboard.

Recuperación de contraseña

El sistema deberá proveer un **flujo de recuperación de contraseña** accesible desde el inicio de sesión.

1. El usuario selecciona la opción “**¿Olvidó su contraseña?**”.
2. Se le solicitará su **username** o **correo electrónico registrado**.
3. El sistema realizara el **envío de un código de verificación** a dicho correo
4. En la interfaz, el usuario ingresará el **código de verificación** recibido.
5. Una vez validado el código, se mostrará el formulario para ingresar la **nueva contraseña** y su confirmación.
6. Tras el cambio exitoso, el sistema mostrará una **pantalla de confirmación** y redirigirá a **login**.

Validaciones adicionales

- No permitir avanzar al siguiente paso sin haber completado el anterior.
- Reglas de contraseña iguales a las del registro
- Confirmación debe ser idéntica a la nueva contraseña.

Elementos adicionales de la UI

- Botón reenviar código.
- Indicador de progreso paso a paso: *Identificación → Verificación → Nueva contraseña → Confirmación*.

Estados de la UI

- **Loading**: mientras se “envía” el código o se procesa el cambio.
- **Error**: código inválido, expirado o contraseñas no coinciden.
- **Éxito**: mensaje “Tu contraseña ha sido restablecida con éxito. Ahora puedes iniciar sesión”.

Módulos de producto - Cuentas

Al iniciar sesión, el usuario accede al **Dashboard** donde se muestra un resumen por cada cuenta disponible. Cada cuenta presenta: **número, moneda, saldo disponible** y un botón “**Ver detalles**”.

Al seleccionar una cuenta, se abre la **vista de detalle**, que incluye encabezado con datos de la cuenta y una **lista de movimientos**. El usuario puede **filtrar** por **tipo de movimiento** (crédito/debito), **búsqueda por texto** (Descripción). La interfaz debe manejar **estados** (cargando, vacío, error).

Campos mínimos de una cuenta

- account_id: Identificador único en formato **IBAN**:
 - CR01-{4 banco}-{4 sucursal}-{12 cuenta}
- Alias: Nombre amigable definido por el usuario. Ej.: "Ahorros Principal"
- Tipo: Ahorro | Corriente.
- Moneda: Valores permitidos: CRC | USD.
- Saldo: Número decimal (2 decimales) Ej: 1523400.50.
- Propietario: customer_id del usuario.

Campos mínimos de un movimiento

- Id: ID de transacción en estándar:
- account_id: Cuenta asociada.
- Fecha: ISO 8601 (YYYY-MM-DDTHH:mm:ssZ).
- Tipo: Valores permitidos: CREDITO | DEBITO.
- Descripción: Texto corto (ej. "Pago servicios", "Depósito nómina").
- Moneda: Valores permitidos: CRC | USD.
- Saldo: Número decimal (2 decimales) Ej: 1523400.50.

Módulo de producto - Tarjetas de Crédito

El sistema deberá mostrar en el **Dashboard** un carrusel o grilla de tarjetas de crédito del usuario. Cada tarjeta tendrá un **diseño visual** acorde con su tipo (Gold/Platinum/Black) e incluirá información clave: numero de tarjeta enmascarado, fecha de vencimiento y nombre. Al seleccionar una cuenta, se abre la **vista de detalle**, que incluye encabezado con datos de la tarjeta y una **lista de movimientos**. El usuario puede **filtrar** por **tipo de movimiento** (compra, pago), **búsqueda por texto** (Descripción). La interfaz debe manejar **estados** (cargando, vacío, error).

Campos mínimos de una cuenta

- Tipo: Gold | Platinum | Black.
- Numero de tarjeta: Se deberá presentar enmascarado al usuario 1234 **** * 1234.
- Exp: MM/AA.
- PIN
- CVV
- Titular
- Moneda: Valores permitidos: CRC | USD.
- Límite: límite total de crédito.
- Saldo: consumido actual

Campos mínimos de un movimiento

- Id: ID de transacción en estándar:
- Card_id: tarjeta asociada.
- Fecha: ISO 8601 (YYYY-MM-DDTHH:mm:ssZ).
- Tipo: Valores permitidos: COMPRA | PAGO.
- Descripción: Texto corto (ej. “Pago servicios”, “Depósito nómina”).
- Moneda: Valores permitidos: CRC | USD.
- Saldo: Número decimal (2 decimales) Ej: 1523400.50.

Consultar PIN - Tarjeta de crédito

Desde el detalle de una tarjeta de débito, el usuario podrá seleccionar la opción “**Consultar PIN**”.

1. El sistema abrirá una **ventana modal** de verificación en dos pasos:
 - **Paso 1 (Verificación de identidad):** ingresar **código de verificación** enviado por correo/SMS.
 - **Paso 2 (Confirmación):** mostrar un **resumen** de la tarjeta (tipo, últimos 4 dígitos, CVV Y PIN) y **por un máximo de 8–12 segundos** y luego se **auto-oculta**.
2. El usuario puede **copiar** el PIN con un botón (opcional) y cerrar la ventana.

Estados de la UI

- **Loading:** al validar contraseña/OTP y al “generar” PIN.

- **Error:** OTP inválido/expirado “El código no es válido o ha expirado.”
- **Éxito:** PIN mostrado por tiempo limitado, luego oculto

Transferencias - Cuentas propias y a terceros (mismo banco)

Permitir que el usuario realice **transferencias internas** dentro del banco: entre **sus propias cuentas** y hacia **terceros** registrados.

Desde el **Dashboard** debe existir el módulo **Transferencias**, donde el usuario podrá:

1. Elegir **tipo de transferencia**: *Propias* o *Terceros (mismo banco)*.
2. Completar un **formulario** con origen, destino, monto, moneda, descripción
3. Ver una **pantalla de confirmación** (modal) con el **resumen** de la operación.
4. Confirmar para generar un **comprobante** con fecha/hora y detalles.
5. Ver el **resultado** (éxito/error) y la opción de **descargar/compartir** el comprobante

Formularios y campos

A. Transferencia entre cuentas propias

- **Cuenta origen** (select obligatorio)
 - Listar solo cuentas del usuario.
- **Cuenta destino** (select obligatorio)
 - Listar otras cuentas propias \neq origen.
- **Moneda** (auto o select; se recomienda la moneda de origen)
- **Monto** (input numérico con 2 decimales)
- **Descripción** (texto opcional, máx. 140–255)

B. Transferencia a terceros (mismo banco)

- **Cuenta origen** (select obligatorio)
 - Listar solo cuentas del usuario.
- **Cuenta destino** (editable se ingresa directo el número de cuenta)
 - Al continuar deberá mostrar validaciones de existencia de la cuenta
 - En el resumen se deberá mostrar el nombre de la persona de la cuenta destino
- **Moneda** (auto o select)
- **Monto** (input numérico)
- **Descripción** (opcional)

Elementos comunes de UI

- **Botón Continuar/Confirmar** (deshabilitado hasta validaciones básicas).
- **Modal de confirmación** con resumen (origen, destino, monto, moneda, descripción, fecha).
- **Comprobante** tras confirmar **Descargar**.

Alcance y criterios técnicos

El Proyecto 1 corresponde al **desarrollo de un prototipo visual** que simula la interfaz y los flujos principales de una plataforma bancaria en línea. Este prototipo será programado en HTML, CSS y JavaScript, con la posibilidad de utilizar un **framework de elección del estudiante** (ejemplo: React, Vue, Angular); sin embargo, los **componentes gráficos deben ser construidos desde cero por los estudiantes**, sin apoyarse en librerías de diseño preconstruidas como **Material UI, Bootstrap u otros equivalentes**.

El alcance de este proyecto se limita a:

- **Diseño visual y flujos de interacción** (mockups funcionales).
- **Estructura semántica correcta del HTML** siguiendo buenas prácticas vistas en clase.
- Aplicación de **principios de accesibilidad y usabilidad**, considerando estándares internacionales (WCAG, WAI-ARIA).
- Implementación de un diseño **mobile-first**, asegurando que la experiencia primaria esté optimizada para dispositivos móviles y luego escale a pantallas más grandes (tablet y desktop).
- Inclusión de validaciones en formularios y estados de UI (loading, error, vacío, éxito).
- Aplicación de **buenas prácticas de CSS**: selectores, box model, flexbox/grid, posicionamiento, responsive con media queries.
- Navegación e interacción simuladas entre las pantallas principales (registro, login, dashboard, cuentas, transferencias, etc.), sin lógica de negocio real.

Este prototipo servirá como **base para el Proyecto 2**, donde se implementarán las funcionalidades y la lógica de negocio real. Por tanto, **cualquier requerimiento omitido o incompleto en el Proyecto 1 deberá ser corregido y extendido en el Proyecto 2**.

En resumen, el Proyecto 1 tiene como objetivo consolidar los conocimientos de **estructura, accesibilidad, usabilidad y diseño responsive** vistos en el curso, además de establecer un estándar técnico de desarrollo que garantice consistencia y escalabilidad en fases posteriores.

Rubrica de evaluación

La evaluación del laboratorio se realizará bajo los siguientes criterios:

Módulos del sistema (70%)

Módulo	1 – Insuficiente	3 – Aceptable	5 – Excelente	%
Registro	No implementado o con errores graves (falta de campos obligatorios, sin validaciones).	Formulario presente con la mayoría de los campos, validaciones parciales y mensajes básicos.	Formulario completo con todos los campos, validaciones correctas, mensajes claros y flujo de confirmación con Términos y Condiciones.	10%
Inicio de sesión	No implementado o no funcional.	Implementado, pero sin feedback claro en estados de error/éxito.	Login completo con feedback adecuado, opción mostrar/ocultar contraseña y enlace a recuperación.	10%
Recuperación de contraseña	No implementado o sin flujos claros.	Flujo parcial (ej. ingreso de usuario pero sin validación de código).	Flujo completo: ingreso de usuario, validación de código, nueva contraseña, mensajes de éxito/error y estados UI.	10%
Cuentas	No implementado o listado vacío.	Listado presente con datos mínimos y vista de detalle básica.	Listado completo, detalle con movimientos, filtros aplicables y manejo de estados (cargando, vacío, error).	10%

Tarjetas de crédito	No implementado o sin detalle.	Módulo presente pero incompleto (ej. solo vista general).	Módulo completo con diseño custom por tipo (Gold, Platinum, Black), detalle de movimientos, filtros y estados UI.	10%
Consulta de PIN (tarjeta débito/crédito)	No implementado.	Flujo parcial: se abre modal pero sin validaciones claras o tiempo de visualización.	Flujo completo: validación (OTP/contraseña), PIN visible por tiempo limitado, auto-ocultamiento y mensajes claros.	10%
Transferencias (propias y terceros)	No implementado.	Flujo parcial (ej. propias pero no terceros, o sin comprobante).	Flujo completo: propias y terceros, formulario con validaciones, confirmación y comprobante simulado.	10%

Criterios transversales (30%)

Criterio	1 – Insuficiente	3 – Aceptable	5 – Excelente	%
Estructura semántica (HTML5)	Uso excesivo de div sin propósito, jerarquía incorrecta.	Uso parcial de etiquetas semánticas, jerarquía incompleta.	Estructura semántica coherente, jerarquía clara y consistente en todo el proyecto.	10%
Accesibilidad y usabilidad	Sin atributos de accesibilidad, problemas de contraste o navegación.	Uso parcial de alt, aria, tabindex; errores menores de foco o contraste.	Atributos completos, navegación por teclado fluida, contraste AA, mensajes de error comprensibles.	10%
Responsive (mobile-first)	Sitio solo usable en desktop, scroll horizontal en móviles.	Diseño responsive parcial (algunos breakpoints básicos).	Diseño mobile-first completo, escalado correcto en móvil, tablet y desktop, sin scrolls innecesarios.	10%

Entrega

Cada grupo deberá entregar un archivo comprimido (.zip) con el proyecto completo, siguiendo estas instrucciones:

1. Una carpeta con el repositorio completo
2. Un archivo denominado enlaces.txt que contenga
 - a. URL del repositorio público en GitHub:
 - b. URL del sitio desplegado (GitHub Pages o Netlify)

Equipos de trabajo: 2 personas

Fecha límite: Domingo 28 de septiembre a las 11:55 p.m.

Forma de entrega: El archivo .zip debe subirse a la plataforma TEC Digital, en el espacio correspondiente al curso.

Portafolio del curso: Este proyecto forma parte del portafolio individual del curso.