

Documentación del Proyecto - NovaBank

Enlace al repositorio

<https://github.com/JCCutino/NovaBank>

Descripción del Proyecto

IMPORTANTE: Para utilizar el admin el correo es admin@admin.com y la contraseña admin2023

El proyecto NovaBank es una plataforma bancaria en línea que ofrece a los usuarios la capacidad de gestionar sus cuentas, realizar transacciones, solicitar préstamos y comunicarse con el banco de manera eficiente y segura. La aplicación se ha desarrollado utilizando tecnologías web estándar, como HTML, CSS, SASS, JavaScript y PHP. Además, se han empleado herramientas de diseño como Figma, Gimp y Canva para crear una interfaz intuitiva y atractiva.

Objetivo

El objetivo principal de NovaBank es proporcionar a los usuarios una experiencia bancaria completa y accesible en línea. Los objetivos específicos incluyen:

Registro e Identificación: Implementar un proceso de registro seguro y acceso fácil a las cuentas mediante HTML, CSS y JavaScript.

Gestión de Perfil: Permitir a los usuarios ver y editar su información personal utilizando formularios dinámicos y funciones de PHP.

Consulta de Saldo: Mostrar el saldo de los usuarios en la moneda seleccionada mediante PHP y JavaScript, con redondeo a dos decimales.

Registro de Transacciones: Registrar cada transacción realizada por el usuario utilizando PHP y almacenar la información en la base de datos.

Préstamos: Facilitar a los usuarios la solicitud de préstamos, gestionados por PHP y JavaScript, con aprobación o rechazo por parte del administrador.

Mensajería: Implementar un servicio de mensajería para la comunicación entre usuarios y el administrador mediante PHP y JavaScript.

Diseño Personalizado: Desarrollar una interfaz atractiva utilizando HTML, CSS y SASS. Se utilizan herramientas como Figma, Gimp y Canva para el diseño visual.

Tecnologías Utilizadas

Frontend: HTML, CSS, SASS, JavaScript

Backend: PHP

Herramientas de Diseño: Figma, Gimp, Canva, Inkscape

Control de Versiones: Git

Requisitos del Sistema

Página de Inicio/Log In/Registro: Implementada con HTML, CSS y JavaScript para proporcionar un acceso y registro seguro a las cuentas de los usuarios. Deberá incluir formularios de inicio de sesión y registro con validación de datos.

Página de Bienvenida: Personalizada con información relevante. Incluirá la foto de perfil del usuario, el saldo y el nombre del usuario.

Página para Ver/Editar Perfil: Utilizando formularios dinámicos y funciones PHP para gestionar la información personal del usuario. Los usuarios podrán ver y editar su nombre, apellido, dirección, fecha de nacimiento, foto, etc.
Los cambios realizados se reflejarán en la base de datos.

Página de Chat: Implementada con HTML, CSS y JavaScript para proporcionar una interfaz interactiva de chat. Los usuarios podrán enviar y recibir mensajes en tiempo real. Se mostrará la lista de contactos y la capacidad de seleccionar usuarios para iniciar un chat.

Página de Pagos: Permitirá a los usuarios realizar transacciones a otros usuarios y así retirar su dinero.

Página de Préstamos: Proporcionará información sobre los préstamos del usuario y la capacidad de solicitar nuevos préstamos.

Página de Transacciones: (Mostrará un registro detallado de las transacciones realizadas por el usuario.

Usuarios y Perfiles

Administrador: Acceso a datos de todos los usuarios y funciones de gestión.

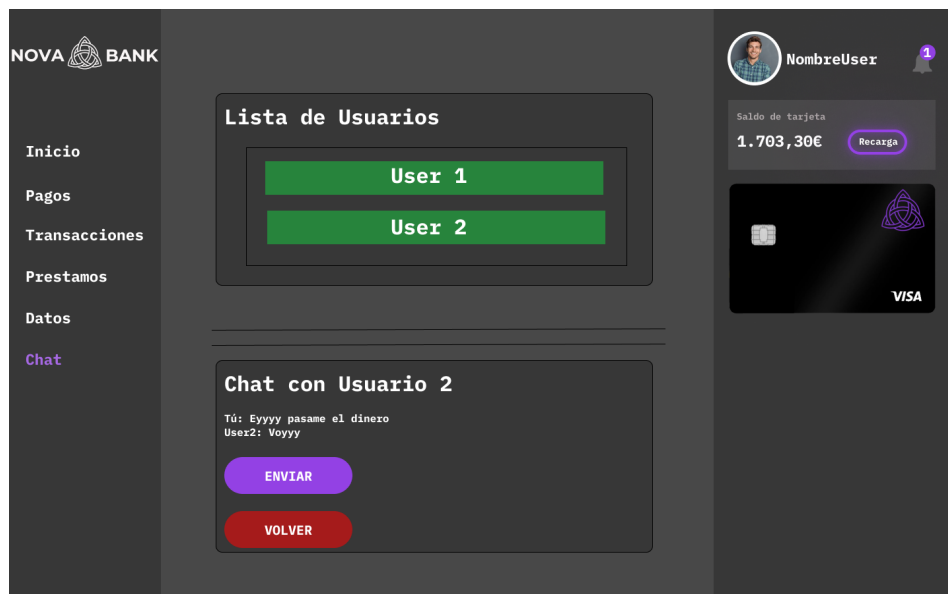
Usuario: Capacidad para gestionar su perfil, realizar transacciones y solicitar préstamos.

Mockup

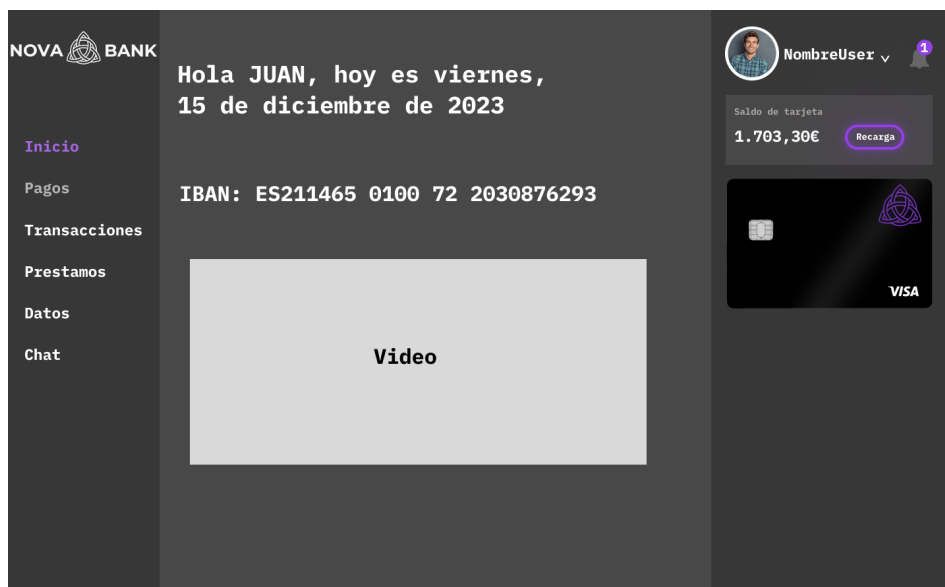
En el proceso de crear el mockup para NovaBank, he utilizado herramientas prácticas y efectivas. Figma fue la elección principal para diseñar la interfaz, permitiéndome planificar cada parte de la aplicación. Para el tratamiento de imágenes, Gimp e Inkscape fueron herramientas clave, ayudándome a refinar y mejorar gráficos y elementos visuales. Este enfoque individual ha permitido crear una representación detallada y precisa de la experiencia que los usuarios encontrarán en NovaBank.

A continuación incluiré algunas vistas, en este repositorio podemos ver todas ellas: <https://drive.google.com/drive/folders/1-AFRFvQWD0YeX3mXJUOSJlcvGjaPmGit?usp=sharing>

Chat



Inicio



Partes del código a resaltar

Pantalla Responsive

Hemos utilizado mayoritariamente Bootstrap para la realización de la página web, pero para realizarlo bien hemos tenido que hacer una función con JS que cada vez que se redimensione la pantalla te ejecute una función que comprueba el ancho y ejecuta a su vez otras acciones diferentes.

```
function ajustarVentana() {  
  let anchoVentana = window.innerWidth;  
  
  if (anchoVentana <= anchoLimite && !accionSuperiorEjecutada) {  
    quitarClase();  
    cerrarOffcanvas();  
    accionSuperiorEjecutada = true;  
    accionInferiorEjecutada = false;  
    ocultarBotonFijo(false);  
  
  } else if (anchoVentana > anchoLimite && !accionInferiorEjecutada) {  
    añadirClase();  
    abrirOffcanvas();  
    accionInferiorEjecutada = true;  
    accionSuperiorEjecutada = false;  
    ocultarBotonFijo(true);  
  
  }  
}  
  
window.addEventListener('resize', ajustarVentana);
```

```
function añadirClase() {  
  let elemento = document.getElementById("saltoSeccion1");  
  let elemento2 = document.getElementById("saltoSeccion2");  
  
  let elemento3 = document.getElementById("container-responsive");  
  if (!elemento) {  
    console.error("Elemento no encontrado con ID: " + "saltoSeccion1");  
    return;  
  }  
  
  elemento.classList.remove("col-lg-6");  
  elemento2.classList.remove("col-lg-6");  
  elemento3.classList.remove("col-lg-12");  
  
  elemento.classList.add("col-lg-12");  
  elemento2.classList.add("col-lg-12");  
  elemento3.classList.add("col-lg-6");  
}
```

Mostrar modales de error / éxito

Para que el usuario tenga una experiencia agradable al usar nuestra página necesito tener estímulos cada vez que interactúa con algo. Decidí que la mejor forma de hacerlo era mediante el uso de modales que indican que sucede, asignamos una sesión con una variable que nos indica el tipo de fallo, usando JavaScript y un switch llamamos a una función cada vez que la página se recarga. De esta manera podemos mostrar el modal adecuado haciendo uso de JavaScript y PHP. Después de mostrarlo volvemos a poner la sesión en null

```
if ($decision === "aceptar") {
    if (aceptarPrestamo($idPrestamo, $tipoInteres, $plazoPagar, $iban, $cantidad)) {
        $_SESSION['resultadoPrestamoAdmin'] = "aceptarExito";
        header("location: " . $_SERVER['HTTP_REFERER']);
    } else {
        $_SESSION['resultadoPrestamoAdmin'] = "aceptarFallo";
        header("location: " . $_SERVER['HTTP_REFERER']);
    }
} elseif ($decision === "rechazar") {
    if (rechazarPrestamo($idPrestamo)) {
        $_SESSION['resultadoPrestamoAdmin'] = "rechazarExito";
        header("location: " . $_SERVER['HTTP_REFERER']);
    } else {
        $_SESSION['resultadoPrestamoAdmin'] = "rechazarFallo";
        header("location: " . $_SERVER['HTTP_REFERER']);
    }
}
```

```
<script>
document.addEventListener('DOMContentLoaded', function () {
    <?php
    $resultadoPrestamoAdmin = isset($_SESSION['resultadoPrestamoAdmin']) ? $_SESSION['resultadoPrestamoAdmin'] : '';
    ?>

    var resultadoPrestamoAdmin = "<?php echo $resultadoPrestamoAdmin; ?>";
    console.log(resultadoPrestamoAdmin);
    switch (resultadoPrestamoAdmin) {
        case 'aceptarExito':
            mostrarModal('aceptarExitoModal');
            break;
        case 'aceptarFallo':
            mostrarModal('aceptarFalloModal');
            break;
        case 'rechazarExito':
            mostrarModal('rechazarExitoModal');
            break;
        case 'rechazarFallo':
            mostrarModal('rechazarFalloModal');
            break;
    }
});

function mostrarModal(modalId) {
    var modal = new bootstrap.Modal(document.getElementById(modalId));
    modal.show();
    <?php $_SESSION['resultadoPrestamoAdmin'] = null; ?>
}
```