

Universidad Don Bosco



Estudiante:

González Pérez, Ernesto José

GP200748

Materia:

Lenguajes Interpretados en el Cliente

Docente:

Ing. Oscar Ariel Pérez Torres

Actividad:

Proyecto Fase 1

Fecha de entrega: domingo 14 de abril de 2024

# Introducción

En la era digital actual, la accesibilidad y la usabilidad son aspectos cruciales para el diseño de cualquier aplicación, especialmente cuando se trata de servicios financieros. En este contexto, el diseño de una interfaz web para un cajero automático adquiere una importancia significativa, ya que se convierte en el punto de contacto principal entre los usuarios y su gestión financiera.

El presente informe detalla el proceso de desarrollo de una interfaz web para un cajero automático, utilizando HTML y CSS como tecnologías principales. Este proyecto se enfoca en crear una experiencia de usuario intuitiva y eficiente, que facilite las transacciones financieras de manera efectiva.

A lo largo del informe, se destacarán las características clave de la interfaz, incluyendo la navegación, la autenticación, la visualización de saldos y la realización de transacciones.

El objetivo final de este proyecto es proporcionar una interfaz web para un cajero automático para la empresa Pokémon Bank que cumpla con los más altos estándares de calidad, brindando a los usuarios una experiencia fluida en la gestión de sus finanzas.

Como primer avance del proyecto final, el presente documento y entregables incluirá únicamente el FrontEnd, es decir, la parte visual del sistema, de forma que el BackEnd se desarrolle para una entrega futura.

# Índice

## Contenido

Introducción .....	2
Índice .....	3
Objetivos Generales y Específicos .....	4
General.....	4
Específicos .....	4
Desarrollo del análisis de proyecto (flujogramas y pantallas) .....	5
Investigaciones y versionado del código .....	11
Conclusiones .....	14

# Objetivos Generales y Específicos

## General

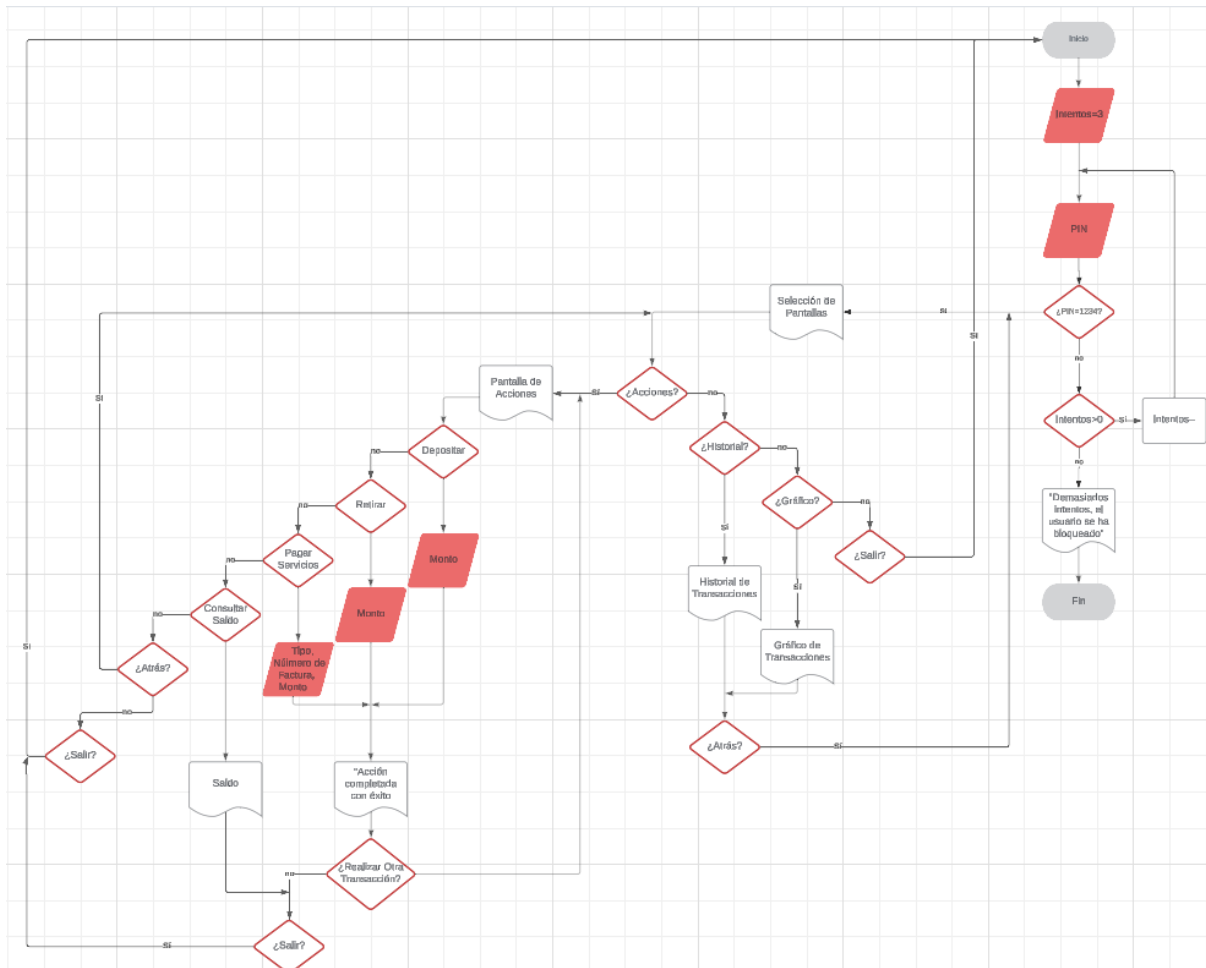
- Diseñar y desarrollar una interfaz web para un cajero automático utilizando HTML y CSS, con el propósito de proporcionar a los usuarios una experiencia intuitiva y eficiente en la gestión de sus transacciones financieras. La interfaz web creada permitirá a los usuarios realizar operaciones como consultar saldos, realizar retiros, realizar depósitos, ver su historial de transacciones y un gráfico representativo de sus transacciones.

## Específicos

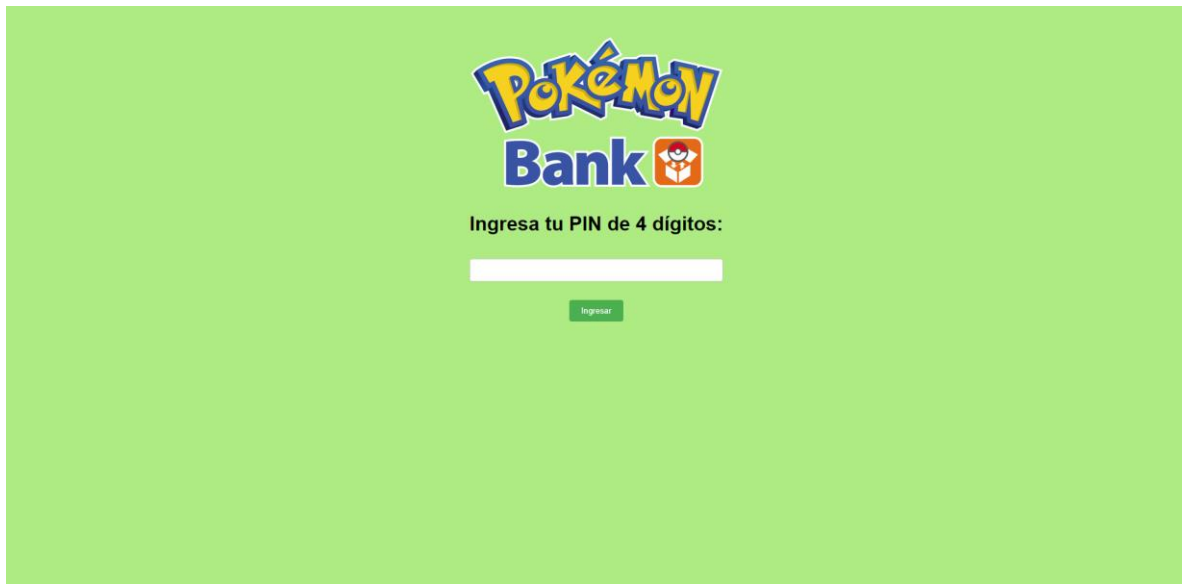
- Diseñar una interfaz web intuitiva y fácil de usar para el cajero automático, que permita a los usuarios navegar sin problemas a través de las diferentes opciones de transacción, utilizando principios de diseño centrados en el usuario.
- Integrar elementos visuales claros y atractivos en la interfaz web del cajero automático, utilizando CSS para estilizar la interfaz de usuario de manera coherente y profesional, con el fin de mejorar la experiencia estética.

# Desarrollo del análisis de proyecto (flujogramas y pantallas)

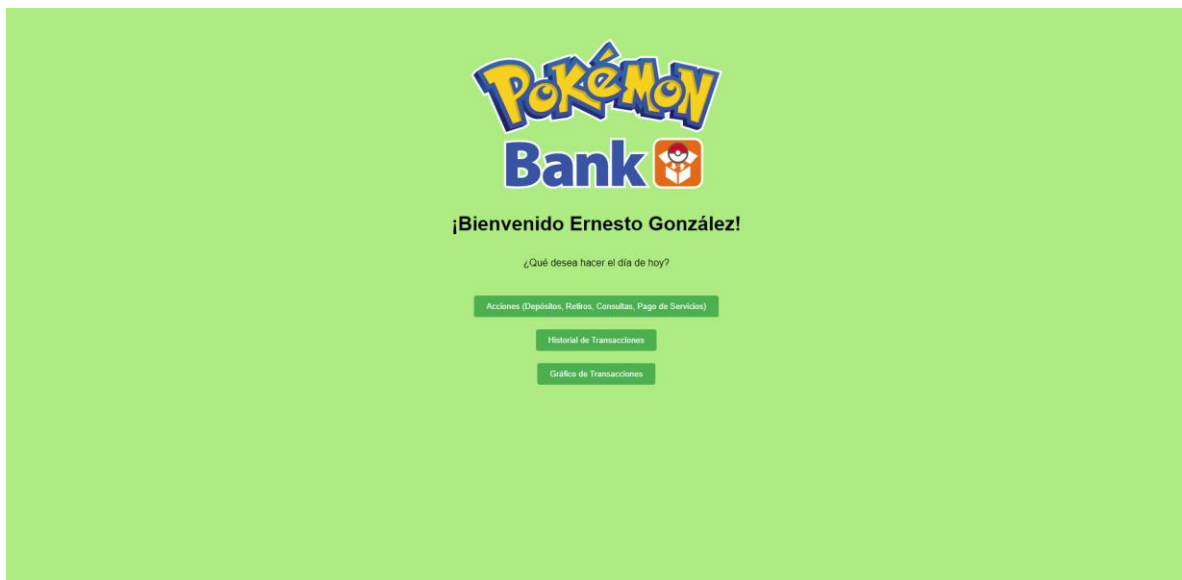
## Diagrama de Flujo



## Login




## Selección de Pantallas



## Acciones

Atás



¡Bienvenido Ernesto González!

¿Qué acción desea hacer el día de hoy?

Depositar

Retirar


Consultar Saldo

Pagar Servicio

Salir

## Depositar

Atás




Ingrese el monto a depositar:

Monto

Realizar acción

Retirar

Atras

Pokémon  
Bank

Ingrese el monto a retirar:

Monto:

Realizar acción

## Consultar SalDOS

The image is a screenshot of the Pokémon Bank app interface. At the top left, there is a green button labeled "Atas". The background is a solid light green. In the center, the "Pokémon Bank" logo is displayed, featuring the word "Pokémon" in its characteristic yellow and blue stylized font, with "Bank" in a blue sans-serif font below it, followed by a red square icon containing a white Poké Ball. Below the logo, the text "Saldo actual:" is written in a bold black font. Underneath this, the value "\$200" is shown in a smaller, regular black font. The bottom of the screen is a solid dark green bar.




Pagar Servicios

Atás

Pokémon

Bank



¡Bienvenido Ernesto González!

¿Qué acción desea hacer el día de hoy?

Depositar

Retirar

Consultar Saldo

Pagar Servicio


Salir

Historial de Transacciones

Atás

Pokémon

Bank

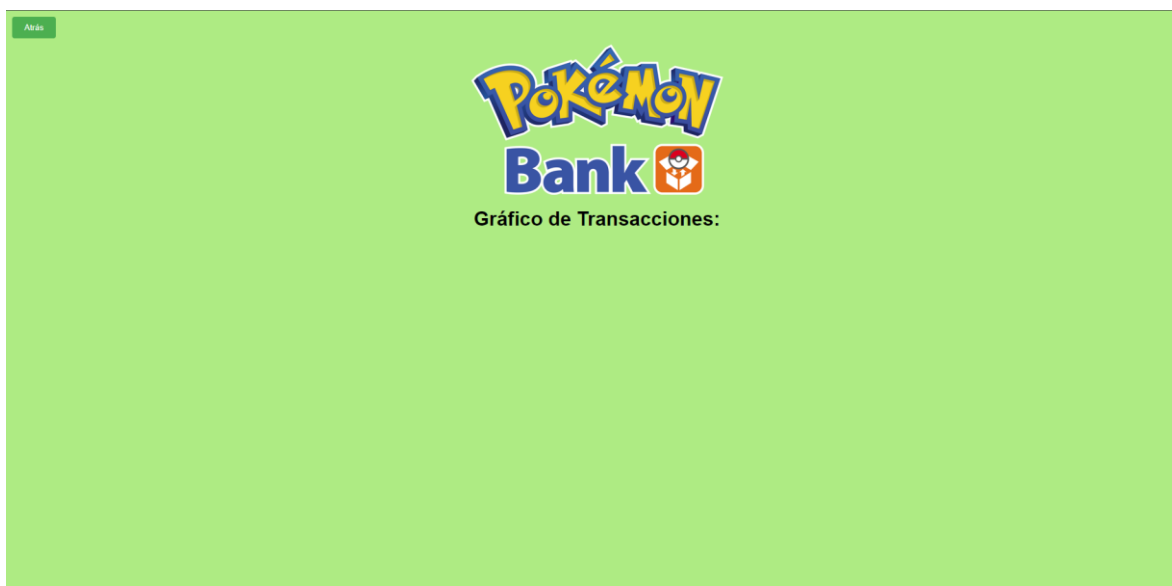


Historial de Transacciones:

Crunchyroll \$4.99

Subway \$3.15

## Gráfico de Transacciones



Imágenes de referencia: Historial y Gráfico de Transacciones se actualizarán por medio de Javascript.

# Investigaciones y versionado del código

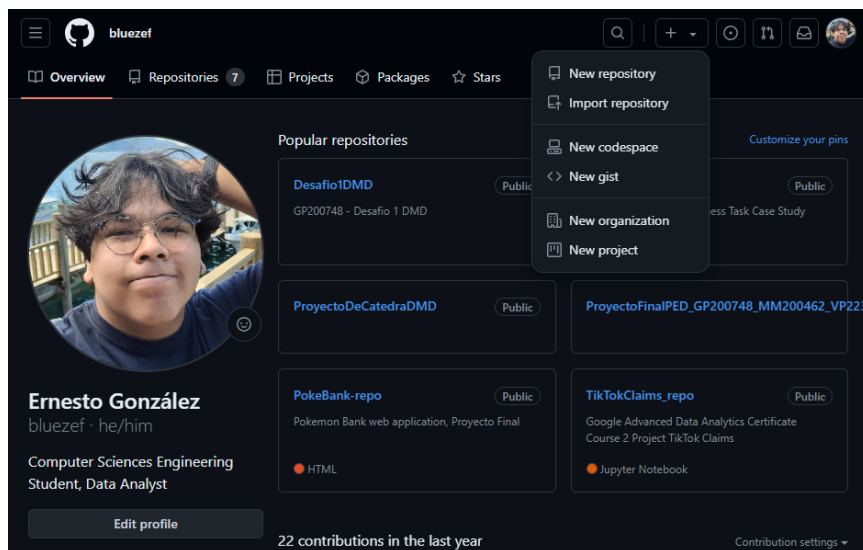
Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. Anteriormente era conocida como Logical Awesome LLC. El código de los proyectos alojados en GitHub se almacena generalmente de forma pública.

Con base en mi investigación para el proyecto, Git y GitHub funcionan de la mano, GitHub nos permite crear un repositorio en la nube para llevar el control de versiones, mientras Git nos permite interactuar con GitHub comunicándose entre nuestra computadora y el repositorio de GitHub.

La ventaja de esto es poder crear un repositorio en donde la colaboración se realice de una manera más intuitiva.

El primer paso para llevar el versionado del código fue crear un repositorio en GitHub, para esto creamos una cuenta en GitHub, y en la página principal nos dirigimos al botón de Create New... (+) >> New repository.



Una vez seleccionada la opción debemos nombrar el repositorio, y agregar una descripción (opcional), para este proyecto el repositorio se ha nombrado PokeBank-repo:


## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---

*Required fields are marked with an asterisk (\*).*

**Owner \*** **Repository name \***


 bluezef / PokeBank-repo


⚠ The repository PokeBank-repo already exists on this account.

Great repository names are short and memorable. Need inspiration? How about potential-sniffle ?

**Description** (optional)

---

☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

---

**Initialize this repository with:**

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

**Add .gitignore**

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

**Choose a license**

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

---

i You are creating a public repository in your personal account.

---

Create repository

Podemos seleccionar si el repositorio será público o privado, para este proyecto se creó un repositorio público bajo el siguiente enlace: <https://github.com/bluezef/PokeBank-repo>

Como siguiente paso, es importante tener instalado Git en nuestra computadora, esto se puede hacer directamente de la página de Git o desde nuestra terminal con el comando `winget install git`.

Para poder conectar nuestra computadora con GitHub se debe abrir la carpeta en la que trabajaremos en la terminal e ingresar los siguientes comandos:

```
git remote add origin https://github.com/bluezef/PokeBank-repo.git
```

```
git branch -M main
```

```
git push -u origin main
```

De esta manera Git al realizar un push lo realizará hacia el enlace del repositorio en la rama principal.

Una vez realizado esta configuración, estamos listos para usar Git, el típico flujo de trabajo de Git es pull, llevar a cabo las modificaciones, status, add, commit y push.

- Pull: Permite descargar los archivos en el repositorio en la nube de la última versión cargada. Durante este proyecto no se hicieron pull requests ya que el proyecto se encontraba en una carpeta de la computadora local.
- Status: Permite ver los cambios que aún no se han registrado con el comando add.
- Add: Permite añadir los archivos para realizar un commit, se puede agregar uno por uno o todos juntos con el comando "add".
- Commit: Permite guardar una versión de los archivos del repositorio, requiere mínimo 2 comentarios, usualmente referentes a qué se realizó y el porqué de los cambios.
- Push: Permite subir los archivos agregados en el commit al repositorio en GitHub.

```
PS C:\Users\netog\OneDrive\Documentos\GP200748_Proyecto_Fase_1> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html
        modified:   pantallas.html
        modified:   script.js

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\netog\OneDrive\Documentos\GP200748_Proyecto_Fase_1> git add pantallas.html
PS C:\Users\netog\OneDrive\Documentos\GP200748_Proyecto_Fase_1> git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   pantallas.html

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html
        modified:   script.js

PS C:\Users\netog\OneDrive\Documentos\GP200748_Proyecto_Fase_1> git commit -m "Added welcome message" -m "Added welcome message deleted in previous version"
[main f01af3b] Added welcome message
 1 file changed, 1 insertion(+)
PS C:\Users\netog\OneDrive\Documentos\GP200748_Proyecto_Fase_1> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 365 bytes | 365.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/bluezef/PokeBank-repo.git
d2aa854..f01af3b  main -> main
```

## Conclusiones

- El desarrollo de la interfaz web para el cajero automático utilizando HTML y CSS ha demostrado ser un proceso que requiere una cuidadosa planificación, diseño y ejecución. A través de este proyecto, se ha logrado crear una interfaz que ofrece una experiencia de usuario intuitiva.
- Durante el desarrollo de la interfaz web del cajero automático, se identificaron y superaron diversos desafíos técnicos y de diseño para las diferentes vistas incluidas en el proyecto. Estos desafíos proporcionaron oportunidades para aprender y mejorar las habilidades en el manejo de tecnologías web.