Informe de Laboratorio 06

Tema: Git y GitHub

Nota

Estudiante	Escuela	Asignatura
Auccacusi Conde Brayan Carlos bauccacusic@unsa.edu.pe	Escuela Profesional de Ingenieria de Sistemas	Programación Web I Semestre: II Código: 20211196
Palma Apaza Santiago Enrique spalmaa@unsa.edu.pe	Escuela Profesional de Ingenieria de Sistemas	Programación Web I Semestre: II Código: 20240689
Pamo Condori Benjamin Andre bpamoc@unsa.edu.pe	Escuela Profesional de Ingenieria de Sistemas	Programación Web I Semestre: II Código: 20233480
Huaynacho Mango Jerry Anderson jhuaynacho@unsa.edu.pe	Escuela Profesional de Ingenieria de Sistemas	Programación Web I Semestre: II Código: 20142322

Laboratorio	Tema	Duración
Trabajo final	Git y GitHub	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - B	10 diciembre 2024	18 diciembre 2024

1. Introducción

Objetivo del Informe

Presentar el desarrollo de una aplicación web que permite gestionar finanzas mediante el uso de sesiones y bases de datos, utilizando scripts CGI escritos en **Perl**, **HTML**, **CSS** y bases de datos en **MariaDB**. La aplicación fue desplegada y ejecutada en un contenedor **Docker** y se uso **GitHub** para trabajar de manera colaborativa.

Importancia del Proyecto

- Utilización de **Git** y **GitHub** para el control de versiones y desarrollo de la aplicación en equipo.
- Uso de bases de datos para almacenar y recuperar datos.

2. Equipos, Materiales y Temas Utilizados

Equipos

- Computadoras con capacidad para ejecutar Docker.
- Conexión a Internet.

Materiales, Software y herramientas:

- **Docker:** Para crear la imagen y ejecutar el contenedor.
- **Git:** Control de versiones local.
- **GitHub:** Repositorio remoto.
- **Perl y JavaScript:** Para implementar la funcionalidad de la página web.
- **HTML y CSS:** Para la interfaz del usuario.
- **MariaDB:** Para gestionar la base de datos
- **Navegador web:** Para pruebas de la aplicación.
- VIM 9.0.

Temas trabajados

- Uso de Git y GitHub.
- Conexión a bases de datos mediante CGI.
- Manejo de datos a través de AJAX.
- Desarrollo de interfaces web interactivas con CGI, HTML y CSS.

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- https://github.com/santiagopalma12/pWeb_Trabajo_Final.git

4. Actividades con el repositorio GitHub

4.1. Creando e inicializando repositorio GitHub

- Se creó el repositorio GitHub para gestionar el proyecto.
- Se realizaron los siguientes comandos en la computadora para crear el repositorio local y conectarlo al repositorio remoto:

Listing 1: Creando directorio de trabajo

```
$ mkdir -p $HOME/pWeb_Trabajo_Final/
```

Listing 2: Dirigiéndonos al directorio de trabajo

```
$ cd $HOME/pWeb_Trabajo_Final/
```

Listing 3: Inicializando directorio para el repositorio GitHub

```
$ git init
$ git add database/setup.sql
$ git commit -m "crear setup.sql para inicializar la base de datos finance_manager"
$ git branch -M main
$ git remote add origin https://github.com/santiagopalma12/pWeb_Trabajo_Final.git
$ git push -u origin main
```

4.2. Commits

- El archivo **echo.pl** fue inicialmente creado para ser la página principal desde la cual se pueden registrar transacciones, ver nuestro saldo actual y los últimos movimientos.

```
1  #!/usr/bin/env perl
2
3  use strict;
4  use warnings;
5
6  print "Content-type: text/html\n\n";
7  print <<EOF;
8  <!DOCTYPE html>
9  <html lang="en">
10 <head>
11     <meta charset="UTF-8">
12     <meta name="viewport" content="width=device-width, initial-scale=1.0">
13     <title>Gestión Financiera</title>
14     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
15     <script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/js/bootstrap.bundle.min.js"></script>
16     <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
17     <link href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap/5.3.0/css/bootstrap.min.css" rel="stylesheet">
```

```
</head>
    <div class="container">
        <div class="header">
            <div class="image-header">
                <img src="../imagenes/image_finanzas.png" alt="Gestión Financiera">
            </div>
            <h2 class="text-center mb-4">Gestión Financiera</h2>
        </div>

        <div class="main-content">
            <!-- Saldo Actual -->
            <div class="saldo">
                <h3>Saldo Actual</h3>
                <div id="saldoContenido" style="margin-top: 20px;">
                    <!-- El saldo será mostrado aquí después de hacer la solicitud -->
                </div>

                <!-- Gráfico de evolución de saldo -->
                <canvas id="saldoChart" width="400" height="200"></canvas>
            </div>

            <!-- Formulario para Registrar Transacción -->
            <div class="formulario">
                <h3>Registrar Transacción</h3>
                <form id="formInsertar" method="POST">
                    <div class="mb-3">
                        <label for="tipo">Tipo:</label>
                        <select class="form-select" id="tipo" name="tipo" required>
                            <option value="ingreso">Ingreso</option>
                            <option value="gasto">Gasto</option>
                        </select>
                    </div>
                    <div class="mb-3">
                        <label for="descripcion">Descripción:</label>
                        <input type="text" class="form-control" id="descripcion" name="descripcion" required>
                    </div>
                    <div class="mb-3">
                        <label for="cantidad">Cantidad:</label>
                        <input type="number" step="0.01" class="form-control" id="cantidad" name="cantidad" required>
```

El archivo myScriptAjax.pl se encarga de registrar las operaciones en la base de datos.

```
# Conectar a la base de datos
my $dbh;
eval {
    $dbh = DBI->connect($dsn, $db_user, $db_pass, { RaiseError => 1, AutoCommit => 1 });
    warn "Conexión a la base de datos exitosa.\n";
};
if ($@) {
    print encode_json({
        success => 0,
        message => "Error al conectar a la base de datos: $@"
    });
    exit;
}

# Verificar si la base de datos se ha conectado correctamente
if (!$dbh) {
    print encode_json({
        success => 0,
        message => "No se pudo conectar a la base de datos."
    });
    exit;
}

# Preparar e insertar datos en la base de datos
my $sth;
eval {
    $sth = $dbh->prepare("INSERT INTO transacciones (tipo, cantidad, descripcion, fecha) VALUES (?, ?, ?, ?)");
    $sth->execute($tipo, $cantidad, $descripcion, $fecha_actual);
    warn "Inserción de datos exitosa.\n";
};
if ($@) {
    print encode_json({
        success => 0,
        message => "Error al insertar los datos en la base de datos: $@"
    });
    $dbh->disconnect if $dbh;
    exit;
}
```

- El archivo verSaldo.pl recupera el saldo desde la base de datos

```perl
# Conectar a la base de datos
my $dbh;
eval {
    $dbh = DBI->connect($dsn, $db_user, $db_pass, { RaiseError => 1, AutoCommit => 1 });
};
if ($@) {
    print encode_json({ success => 0, message => "Error al conectar a la base de datos: $@" });
    exit;
}

# Preparar y ejecutar la consulta
my $sth = $dbh->prepare('SELECT SUM(cantidad) FROM transacciones');
$sth->execute();

# Recuperar el saldo
my ($saldo) = $sth->fetchrow_array;

# Si no hay saldo, asignar 0
$saldo //= 0;

# Devolver el saldo en formato JSON
print encode_json({ success => 1, saldo => $saldo });
```

- El archivo verUltimosMovimientos.pl hace lo mismo con las transacciones recientes.

```perl
# Realizar la consulta para los últimos 5 movimientos
my $sth = $dbh->prepare('SELECT tipo, descripcion, cantidad, fecha FROM transacciones ORDER BY fecha DESC LIMIT 1000');
$sth->execute();

my @movimientos;

# Variable para el saldo acumulado
my $saldo_acumulado = 0;

while (my @row = $sth->fetchrow_array) {
    # Actualizar el saldo acumulado según el tipo de transacción
    if ($row[0] eq 'ingreso') {
        $saldo_acumulado += $row[2];
    } elsif ($row[0] eq 'gasto') {
        $saldo_acumulado -= $row[2];
    }

    # Agregar movimiento y saldo a la lista
    push @movimientos, {
        tipo => $row[0],
        descripcion => $row[1],
        cantidad => $row[2],
        fecha => $row[3],
        saldo => $saldo_acumulado
    };
}

# Devolver la respuesta en formato JSON
print encode_json({
    success => 1,
    movimientos => \@movimientos
});
```

- Los archivos script.js , script2.js y script3.js trabajan en conjunto para actualizar y mostrar los datos en la página principal

```javascript
$(document).ready(function () {
    // Mostrar el saldo automáticamente al cargar la página
    $.getJSON('verSaldo.pl', function (response) {
        if (response.success) {
            $('#saldoContenido').html(`<p><strong>Saldo Actual:</strong> ${response.saldo}</p>`);
        } else {
            $('#saldoContenido').html(`<p>Error: ${response.message}</p>`);
        }
    }).fail(function () {
        $('#saldoContenido').html('<p>Error al cargar el saldo.</p>');
    });

    // Registrar transacción usando AJAX
    $('#submitAJAXInsertar').on('click', function () {
        $.post('myScriptAjax.pl', $('#formInsertar').serialize(), function (response) {
            $('#respInsertar').removeClass('alert-danger').addClass('alert-success').text(response.message).show();

            // Actualizar el saldo automáticamente después de registrar una transacción
            $.getJSON('verSaldo.pl', function (response) {
                if (response.success) {
                    $('#saldoContenido').html(`<p><strong>Saldo Actual:</strong> ${response.saldo}</p>`);
                } else {
                    $('#saldoContenido').html(`<p>Error: ${response.message}</p>`);
                }
            });
        }).fail(function () {
            $('#respInsertar').removeClass('alert-success').addClass('alert-danger').text('Error al registrar la transacción.').show();
        });
    });
});
```

```javascript
$(document).ready(function() {
    // Función para actualizar los últimos movimientos automáticamente al cargar la página
    function cargarUltimosMovimientos() {
        $.getJSON('verUltimosMovimientos.pl', function(response) {
            if (response.success) {
                let movimientosHTML = '<table border="1"><thead><tr><th>Tipo</th><th>Descripción</th><th>Cantidad</th><th>Fecha</th></tr></thead><tbody>';
                response.movimientos.forEach(function(movimiento) {
                    movimientosHTML += `<tr>
                        <td>${movimiento.tipo}</td>
                        <td>${movimiento.descripcion}</td>
                        <td>${movimiento.cantidad}</td>
                        <td>${movimiento.fecha}</td>
                    </tr>`;
                });
                movimientosHTML += '</tbody></table>';
                $('#ultimosMovimientosContenido').html(movimientosHTML);
            } else {
                $('#ultimosMovimientosContenido').html('<p>Error al cargar los últimos movimientos.</p>');
            }
        }).fail(function() {
            $('#ultimosMovimientosContenido').html('<p>Error al cargar los últimos movimientos.</p>');
        });
    }

    // Llamar a la función para cargar los últimos movimientos al cargar la página
    cargarUltimosMovimientos();

    // Registrar transacción usando AJAX y luego actualizar los últimos movimientos
    $('#submitAJAXInsertar').on('click', function() {
        $.post('myScriptAjax.pl', $('#formInsertar').serialize(), function(response) {
            $('#respInsertar').removeClass("alert-danger").addClass("alert-success").text(response.message).show();

            // Actualizar los últimos movimientos automáticamente después de registrar una transacción
            cargarUltimosMovimientos();
        }).fail(function() {
            $('#respInsertar').removeClass("alert-success").addClass("alert-danger").text("Error al registrar la transacción.").show();
        });
    });
});
```

```javascript
// Función para crear el gráfico de evolución del saldo
function crearGraficoSaldo(movimientos) {
    const ctx = document.getElementById('saldoChart').getContext('2d');

    // Preparar los datos del gráfico (fechas y saldos acumulados)
    const fechas = movimientos.map(mov => mov.fecha);
    const saldos = movimientos.map(mov => parseFloat(mov.saldo));

    // Crear o actualizar el gráfico de línea
    if (window.chart) {
        // Si ya existe el gráfico, actualízalo
        window.chart.data.labels = fechas;
        window.chart.data.datasets[0].data = saldos;
        window.chart.update();
    } else {
        // Si no existe el gráfico, crea uno nuevo
```

```javascript
// Función para obtener los últimos movimientos y actualizar el gráfico
function obtenerUltimosMovimientos() {
    $.get("verUltimosMovimientos.pl", function(response) {
        if (response.success) {
            // Crear o actualizar el gráfico con los movimientos recibidos
            crearGraficoSaldo(response.movimientos);
        } else {
            console.error("Error al obtener los movimientos.");
        }
    });
}

// Llamar a la función de actualización cada 5 segundos (5000 ms)
setInterval(obtenerUltimosMovimientos, 2000);

// Llamar a la función al cargar la página
window.onload = obtenerUltimosMovimientos;
```

4.3. Estructura del proyecto

- El contenido del proyecto que se entregará en este laboratorio es el siguiente:

```
pWeb_Trabajo_Final/
|---html
|     |---imagenes
|     |     |---fondo_finanzas.jpeg
|     |     |---image_finanzas.png
|     |     |---test.png
|     |---echo.pl
|     |---myScriptAjax.pl
|     |---script.js
|     |---script2.js
|     |---script3.js
|---mariadb-data
|     |---gestor_finanzas
|     |     |---db.opt
|     |     |---transacciones.frm
|     |     |---transacciones.ibd
|---docker-compose.yml
|---Dockerfile
|---gestor_finanzas.sql
|---leeme.txt
|---README.md
```

4.4. Despliegue con Docker

- El despliegue de la aplicación se realiza con 2 contenedores Docker utilizando el archivo **Dockerfile** y el archivo **docker-compose.yml**.
- El archivo **Dockerfile** se encarga de crear la imagen del servidor apache2 e instala los módulos necesarios para que la aplicación pueda funcionar correctamente.

```
FROM ubuntu:20.04

# Configurar el entorno
ENV DEBIAN_FRONTEND=noninteractive

# Actualizar el sistema e instalar dependencias necesarias
RUN apt-get update && apt-get install -y \
    apache2 \
    perl \
    libapache2-mod-perl2 \
    mariadb-client \
    cpanminus \
    libmariadb-dev \
    build-essential \
    libssl-dev \
    libcurl4-openssl-dev \
    && rm -rf /var/lib/apt/lists/*

# Instalar módulos Perl adicionales con cpanm
RUN cpanm --notest CGI DBI JSON DBD::MariaDB
```

- El archivo **docker-compose.yml** contiene las especificaciones para poder crear y correr ambos contenedores simultáneamente, tanto del servidor apache2 como del servidor MariaDB donde se encuentra la base de datos, usando solo el comando **docker-compose.**

```
services:
  apache:
    build: .
    container_name: apache-container
    ports:
      - "8080:80"
    volumes:
      - ./html:/var/www/html
      - ./apache-config:/etc/apache2/sites-enabled
    depends_on:
      - mariadb
    networks:
      - webnet
    restart: always

  mariadb:
    image: mariadb:10.5
    container_name: mariadb-container
    environment:
      MYSQL_ROOT_PASSWORD: santiago81
      MYSQL_DATABASE: gestor_finanzas
      MYSQL_USER: root
      MYSQL_PASSWORD: santiago81
    volumes:
      - ./mariadb-data:/var/lib/mysql
      - ./gestor_finanzas.sql:/docker-entrypoint-initdb.d/gestor_finanzas.sql
    networks:
      - webnet
    restart: always

networks:
  webnet:
    driver: bridge
```

Listing 12: Comandos para despliegue

```
$ docker-compose build
$ docker-compose up
```

5. Pregunta: ¿Qué se aprendió del trabajo colaborativo en GitHub con cuatro integrantes en este proyecto?

En este proyecto, trabajamos de manera colaborativa con un equipo de cuatro integrantes utilizando GitHub como herramienta principal para gestionar el desarrollo y la coordinación del proyecto. A lo largo del proceso, aprendimos varias lecciones importantes sobre trabajo en equipo y control de versiones:

- **Importancia del control de versiones:** GitHub nos permitió gestionar los cambios en el código de manera eficiente. Cada miembro del equipo pudo trabajar de forma independiente en su parte del proyecto y, al final, realizar integraciones sin riesgo de sobrescribir el trabajo de los demás. La creación de ramas (branches) y la posterior fusión (merge) nos permitió trabajar simultáneamente en distintas funcionalidades sin conflictos.

- **Trabajo en equipo y comunicación:** El uso de GitHub facilitó la comunicación y la organización del trabajo entre los integrantes del equipo. Las issues y pull requests fueron esenciales para llevar un seguimiento de las tareas y para realizar revisiones de código antes de integrar las modificaciones al proyecto final. Esto mejoró la calidad del código y permitió que todos los miembros estuvieran al tanto de los avances y problemas.

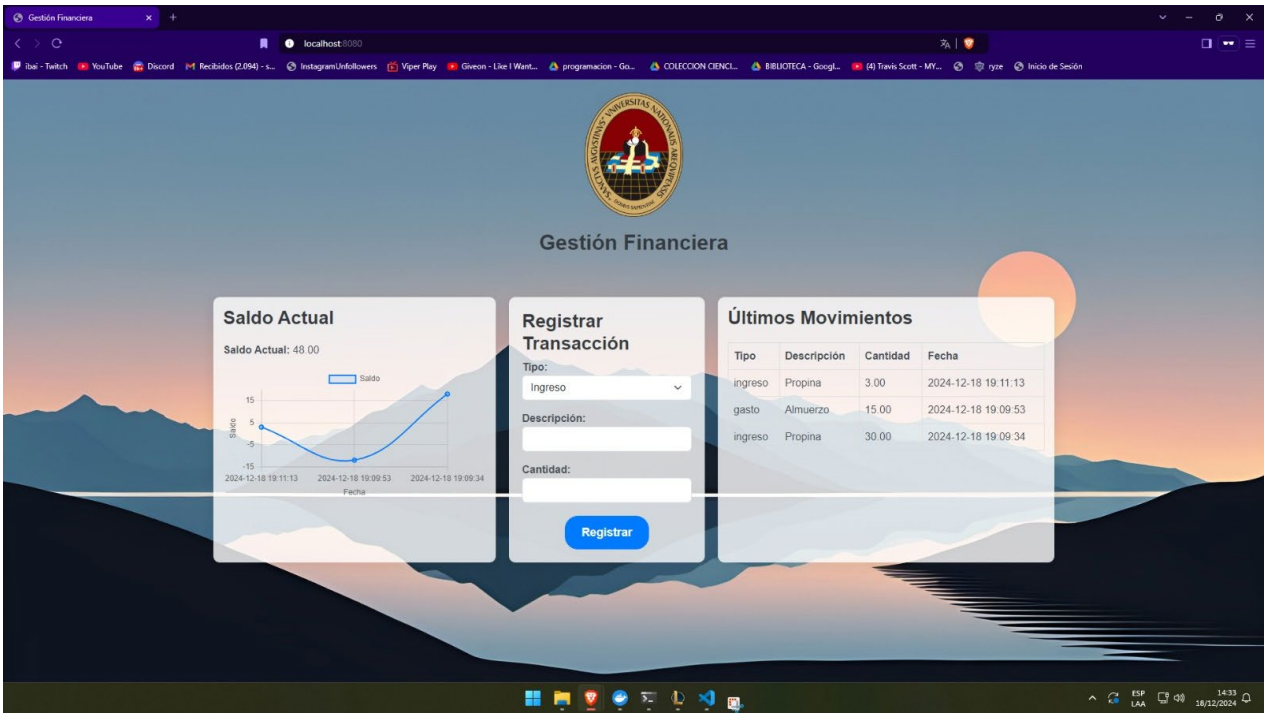
- **Resolución de conflictos:** En ocasiones surgieron conflictos de código durante la fusión de ramas, lo que nos obligó a colaborar directamente para resolverlos. Este proceso nos ayudó a mejorar nuestras habilidades para manejar problemas técnicos en equipo y encontrar soluciones rápidamente.

  Mejora de la organización: Al crear una estructura de carpetas clara y un archivo README.md bien documentado, pudimos mantener el proyecto organizado y accesible para todos. Esto facilitó el que todos supieran cómo testear y contribuir al proyecto.

  Gestión de tareas y asignación de responsabilidades: GitHub también nos permitió asignar tareas específicas mediante issues, lo que mejoró la asignación de responsabilidades y la planificación del trabajo. Esto nos permitió cumplir con los plazos establecidos y dividir el proyecto en tareas manejables para cada miembro.

En resumen, el trabajo colaborativo en GitHub nos permitió coordinar eficientemente nuestras tareas y mejorar la calidad del proyecto mediante una mejor organización, comunicación y manejo de versiones. Aprendimos a trabajar de forma más efectiva como equipo y a resolver problemas de manera conjunta, lo que fue esencial para el éxito del proyecto.

6. Capturas de pantalla



Index

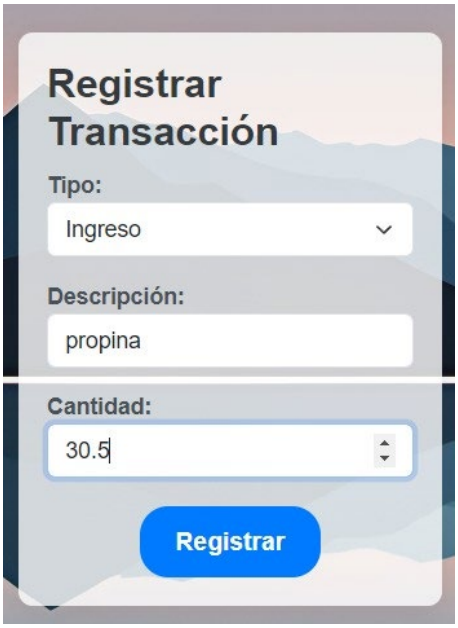


Figura 1

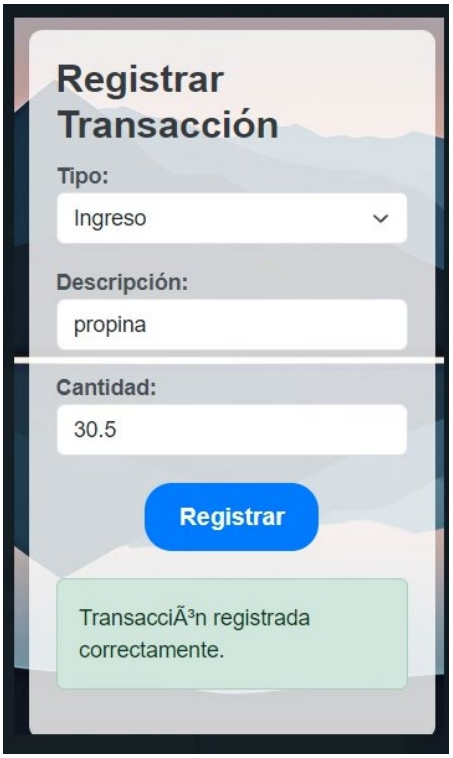


Figura 2

Figura 3

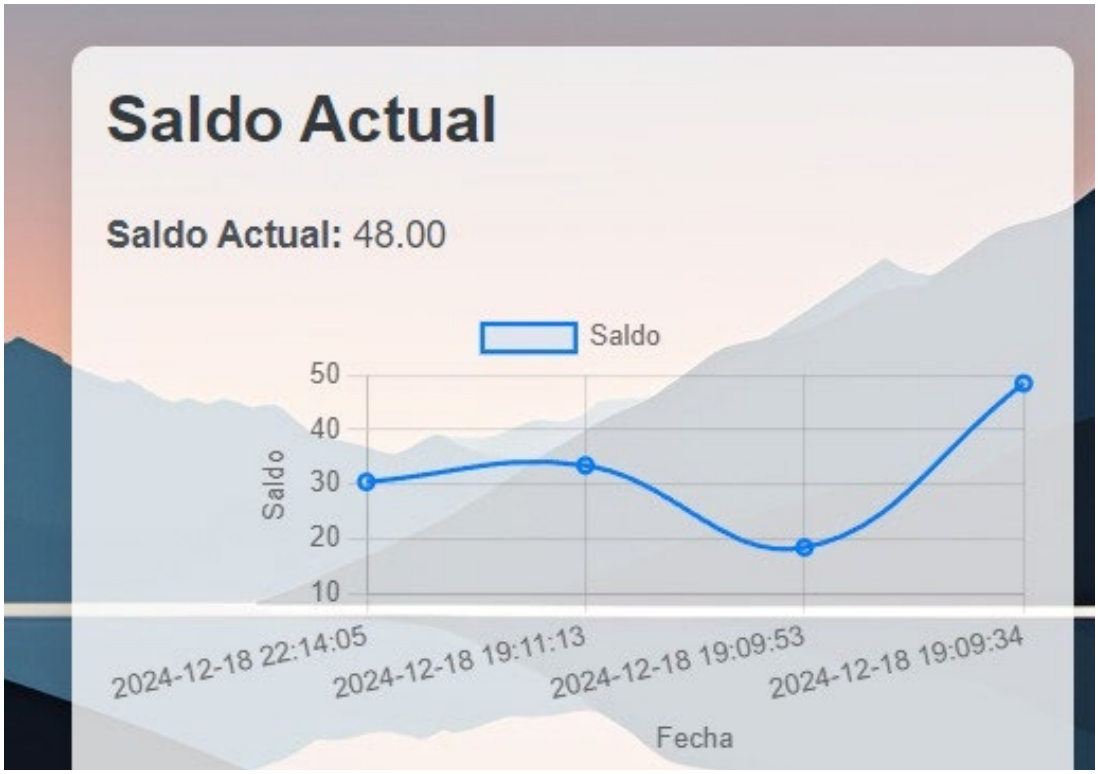

Saldo Actual
Saldo Actual: 48.00
Saldo
50
40
30
20
10
Saldo
2024-12-18 22:14:05
2024-12-18 19:11:13
2024-12-18 19:09:53
2024-12-18 19:09:34
Fecha

Figura 4

7. Rubricas de calificaciones

Tabla 1: Estudiante **Auccacusi Conde Brayan Carlos**

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
3. Codigo fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografia	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	3	X	0	
Total		20		14	

Tabla 2: Estudiante **Palma Apaza Santiago Enrique**

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
3. Codigo fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografıa	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	0	
Total		20		16	

Tabla 3: Estudiante **Pamo Condori Benjamin Andre**

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
3. Codigo fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografıa	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	0	
Total		20		14	

Tabla 4: Estudiante **Huaynacho Mango Jerry Anderson**

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	3	
3. Codigo fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	1	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografıa	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	0	
Total		20		14	

8. Referencias

- https://www.w3schools.com/java/default.asp
- https://www.geeksforgeeks.org/insertion-sort/
- https://www.w3schools.com/java/default.asp
- https://www.geeksforgeeks.org/insertion-sort/ https://docs.github.com/en/github/collaborating-with-issues-and-
- pull-requests/about-issues https://www.git-scm.com/book/en/v2/Getting-Started-About-Version-Control
- https://www.atlassian.com/git/tutorials/comparing-workflows
- https://www.educative.io/edpresso/what-are-pull-requests-in-git
- https://www.gitkraken.com/learn/git/tutorials/git-merge-conflict
- https://dev.to/rammyblog/how-to-merge-and-resolve-conflicts-in-git-2dgp
- https://www.turing.com/kb/what-is-git-and-why-is-it-important
- https://stackabuse.com/understanding-and-resolving-git-merge-conflicts/ https://www.learngitbranching.js.org/
- https://www.geeksforgeeks.org/git-merge-conflict/
- https://opensource.com/article/19/11/why-use-git