Instituto Tecnológico de Costa Rica

Escuela de Ingenieria en Computación

Curso: Bases de Datos I

Profesor: Franco Quirós Ramirez

Grupo: 02

Trabajo: Segundo Proyecto Programado

Estudiantes:

Erick Manuel Abarca Calderón

2022296303

Jesús Valverde Ureña

2022427462

Entrega: 7/10/2024

II semestre, 2024

Tabla de contenidos

[Tabla de imágenes 2](#_Toc178952186)

[1. Introducción 3](#_Toc178952187)

[2. Ambiente de Desarrollo 4](#_Toc178952188)

[3. Evaluación de los elementos 4](#_Toc178952189)

[4. Métricas del Proyecto 5](#_Toc178952190)

[4.1 Horas trabajadas 5](#_Toc178952191)

[4.2 Distribución de tareas entre los miembros del grupo 5](#_Toc178952192)

[4.3 División de tareas para investigación (pruebas de concepto) 5](#_Toc178952193)

[4.4 Cantidad de sesiones 6](#_Toc178952194)

[4.5 Líneas de código 6](#_Toc178952195)

[4.6 Entradas de GitHub 6](#_Toc178952196)

[4.7 Cantidad de datos de prueba 7](#_Toc178952197)

[4.8 Cantidad de SP 8](#_Toc178952198)

[4.9 Gráficos del GitHub 9](#_Toc178952199)

# 

# Tabla de imágenes

[Imágen 1 7](#_Toc178952140)

[Imágen 2 8](#_Toc178952141)

[Imágen 3 9](#_Toc178952142)

[Imágen 4 10](#_Toc178952143)

[Imágen 5 10](#_Toc178952144)

# 1. Introducción

El documento pretende sintetizar el proceso y los resultados sobre el desarrollo de un sistema básico de gestor de planilla. El sistema permite consultar a cada empleado y a sus salarios, al igual que insertar nuevos empleados al sistema.

# 2. Ambiente de Desarrollo

El sistema básico de gestor de planilla se desarrolló y ejecutó localmente en una computadora. Para la gestión de la base de datos, se utilizó Microsoft SQL Server (MSSQL), un sistema de gestión de bases de datos relacional que permitió almacenar y consultar la información de los empleados y sus movimientos de manera eficiente.

El servidor de la aplicación fue desarrollado utilizando Flask, un microframework web en Python, el cual facilita la creación de aplicaciones ligeras. En este caso, Flask fue configurado para ejecutarse localmente, permitiendo la interacción con la base de datos a través de una interfaz sencilla. La conexión entre Flask y MSSQL se logró utilizando bibliotecas como pyodbc o SQLAlchemy, que permitieron la comunicación directa con la base de datos. Esta a su vez se conecta con la página web enviando archivos JSON utilizando el protocolo AJAX. La página está realizada enteramente con HTML, CSS y JavaScript.

# 3. Evaluación de los elementos

Basados en la rúbrica brindada por el profesor (“Tabla 1. Evaluación elementos según rúbrica”) se realiza la evaluación de avance de las diversas partes del proyecto, así como una valoración de los puntos obtenidos según el avance, en dicha parte.

Tabla 1. Evaluación de los elementos según rúbrica brindada por profesor.

|  |  |  |  |
| --- | --- | --- | --- |
| Parte | Valor | Porcentaje avanzado | Comentarios |
| Documentación | 15 | 15 |  |
| Base de Datos | 7 | 7 |  |
| Script XML | 15 | 13 | Error en el valor NuevoSaldo |
| Código de SP | 23 | 23 |  |
| Funcionalidad |  |  |  |
| Login/Logout | 6 | 5 | No hace la deshabilitación |
| Lista empleado con filtrado | 6 | 6 |  |
| Insertar Empleado | 5 | 5 |  |
| Insertar Movimiento | 7 | 7 |  |
| Lista Movimiento | 5 | 5 |  |
| Trazabilidad | 6 | 6 |  |
| Mensajes y Códigos de error | 5 | 5 |  |
| Total | 100 | 97 |  |

# 4. Métricas del Proyecto

## 4.1 Horas trabajadas

Se trabajaron unas 30 horas y 30 minutos, repartidas durante 13 sesiones de trabajo siendo la ultima el miércoles 2 de octubre.

## 4.2 Distribución de tareas entre los miembros del grupo

La primera tarea, tras asignar el proyecto programado, fue distribuir las partes del proyecto entre los miembros del grupo para que cada uno de ellos pudiera tener un avance correcto durante la realización de este.

## 4.3 División de tareas para investigación (pruebas de concepto)

Para la elaboración del proyecto de manera correcta en el tiempo disponible, se realizó la división de tareas, tal y como se adjunta en la “Tabla 2. Distribución entre miembros del grupo”.

Tabla 2. Distribución entre miembros del grupo.

|  |  |  |
| --- | --- | --- |
| Parte | Encargado | Avance |
| Documentación | Ambos | Listo |
| Base de Datos | Ambos | Listo |
| Script XML | Erick | Listo |
| Código de SP | Ambos | Listo |
| Login/Logout | Erick | Listo |
| Lista empleado con filtrado | Erick | Listo |
| Insertar Empleado | Jesús | Listo |
| Insertar Movimiento | Jesús | Listo |
| Lista Movimiento | Jesús | Listo |
| Trazabilidad | Ambos | Listo |
| Mensajes y Códigos de error | Ambos | Listo |

## 4.4 Cantidad de sesiones

Se realizaron 13 sesiones de trabajo de forma asincrónica, pero, sin embargo, los integrantes del equipo de trabajo viven juntos, por lo que, la mayoría de la comunicación fue verbal o viendo el código desde una computadora de alguno de los integrantes

## 4.5 Líneas de código

Fueron aproximadamente 2596 líneas de código, específicamente:

* Creación de la base: 88 líneas
* SP Cargar XML: 105 líneas
* Stored Procedures: 777 líneas
* Servidor Flask: 555 lineas
* Página web: 1071 lineas

## 4.6 Entradas de GitHub

El GitHub se utilizó en la tarea programada para permitir la sincronización de las diversas versiones del programa. Se realizaron 22 commits de los miembros del grupo, algunos fueron para sincronizar archivos y otros para resolver pequeños errores que veíamos después de hecho el commit.

Imagen 1.

Commits registrados en el Github.

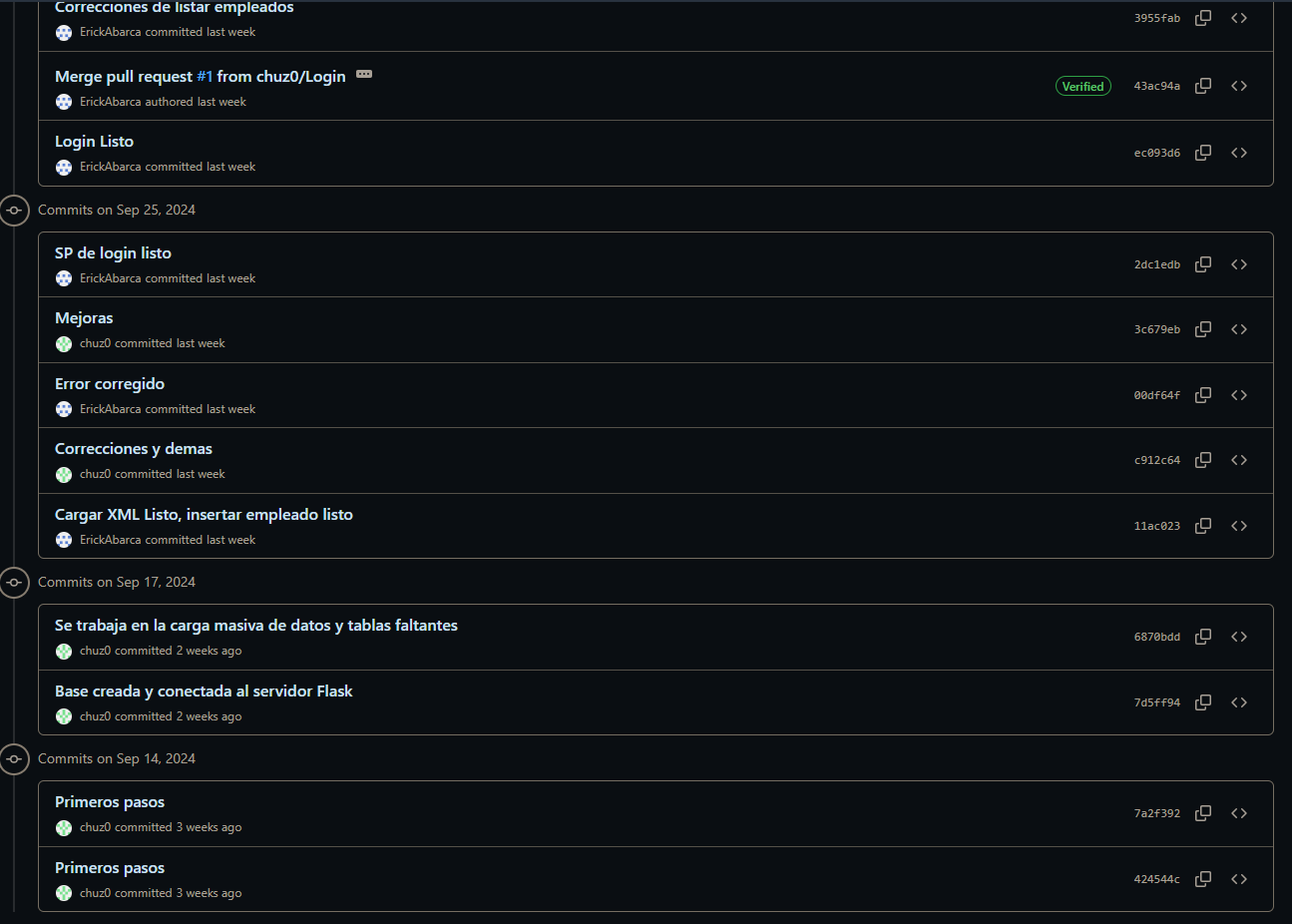


Imagen 1

Imagen 2.

Commits registrados en el Github, segunda parte.

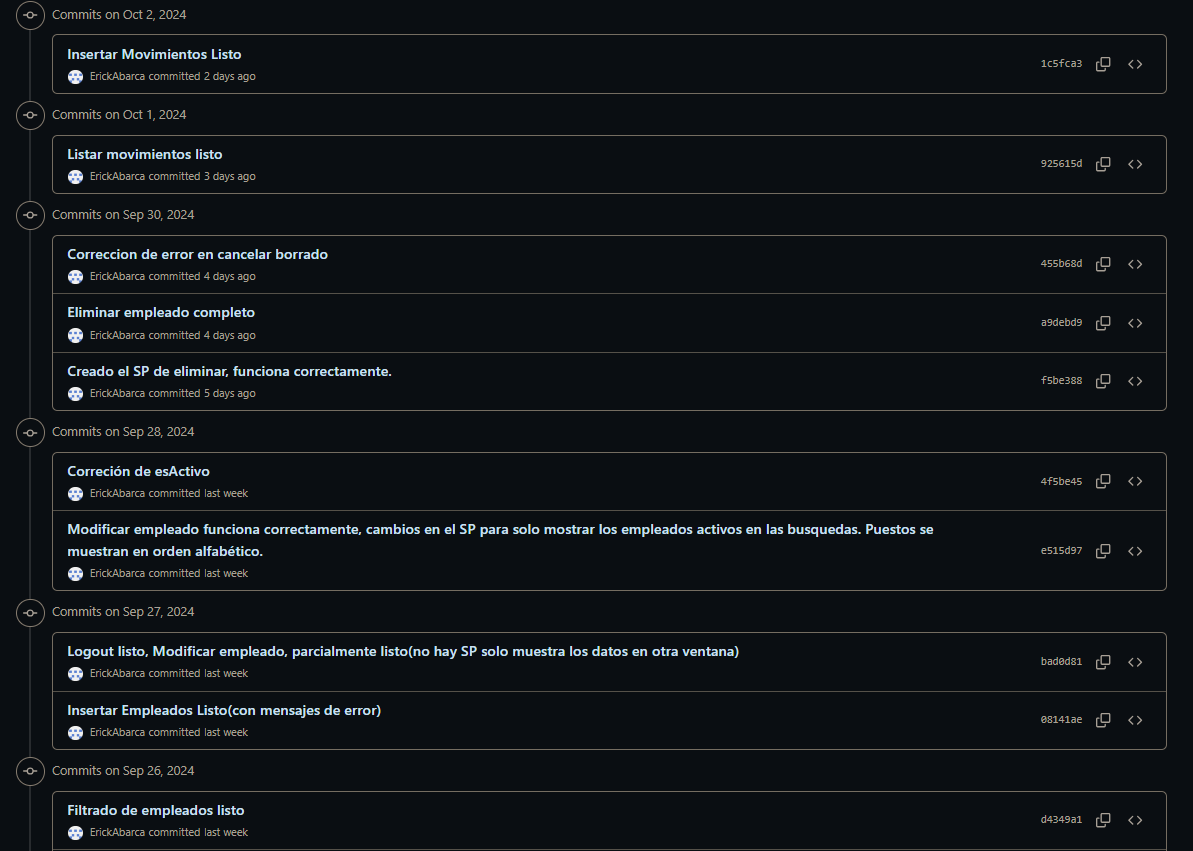


Imagen 2

## 4.7 Cantidad de datos de prueba

Para los datos de prueba, se utilizó el ingreso por medio de una carga masiva desde un archivo XML de:

* 10 puestos de trabajo
* 14 tipos de eventos
* 6 tipos de movimientos
* 30 empleados
* 6 usuarios
* 200 movimientos
* 11 tipos de error

Imagen 3.

Lenguajes utilizados para la realización de la tarea programada.

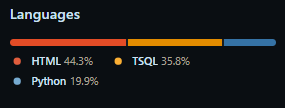


Imagen 3

## 4.8 Cantidad de SP

Para la realización de la tarea programada se crearon 16 Stored Procedure:

* Insertar Empleado
* Listar Empleado
* Validar Credenciales
* Logout
* Filtro por nombre
* Filtro por documento de identidad
* Obtener puestos
* Obtener tipo de movimientos
* Obtener Error
* Obtener empleado por documento de identidad
* Intento de borrado
* Modificar empleado
* Listar Movimientos por Documento de identidad
* Insertar Movimiento
* Cargar XML

## 4.9 Gráficos del GitHub

Imagen 4.

Línea del tiempo de adiciones y eliminaciones por semana de cada integrante.

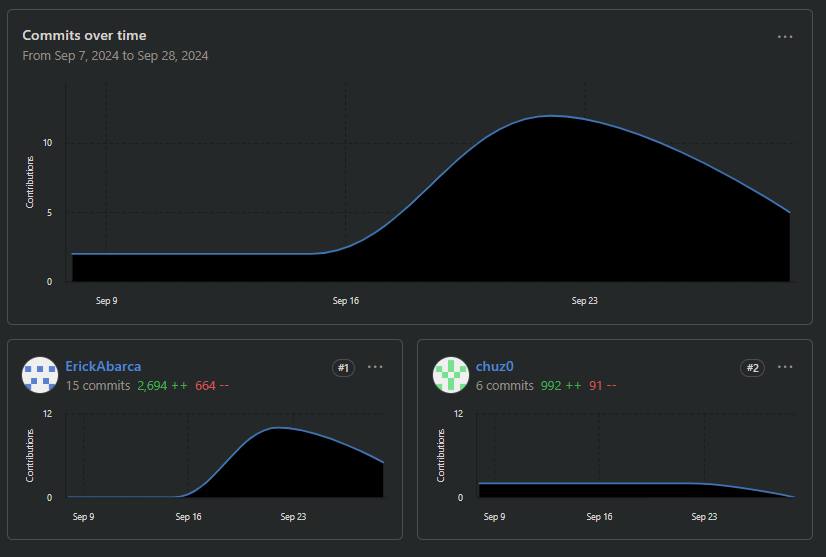


Imagen 4

Imagen 5.

Línea de branches del proyecto (Network)



Imagen 5