



Instituto tecnológico de Costa Rica

Escuela de Ing. en Computación

Campus: Tecnológico de Cartago

Curso: Base de Datos I [IC-4301]

Profesor: Franco Quirós Ramirez

Estudiante(s):

Fernando Andrés Sánchez Hidalgo [2022218688]

Análisis de resultados

Tarea programada 1: Prueba de concepto

Semestre I

2024

## Índice de contenidos

<b>Introducción.....</b>	<b>3</b>
<b>Ambiente utilizado.....</b>	<b>3</b>
<b>Tabla de evaluación.....</b>	<b>3</b>
<b>Métricas del proyecto.....</b>	<b>4</b>
<b>Figuras y Gráficos.....</b>	<b>5</b>

## Índice de figuras

<a href="#"><u>Figura 1: Tabla de empleados .....</u></a>	<a href="#"><u>5</u></a>
<a href="#"><u>Figura 2: Procedimiento almacenado: ListarEmpleados .....</u></a>	<a href="#"><u>5</u></a>
<a href="#"><u>Figura 3: Procedimiento almacenado: InsertarEmpleado .....</u></a>	<a href="#"><u>6</u></a>
<a href="#"><u>Figura 4: Gráfico de Github: Commits .....</u></a>	<a href="#"><u>6</u></a>
<a href="#"><u>Figura 5: Gráfico de GitHub: Frecuencia de código .....</u></a>	<a href="#"><u>7</u></a>
<a href="#"><u>Figura 6: link al blog .....</u></a>	<a href="#"><u>7</u></a>
<a href="#"><u>Figura 7: link al repositorio .....</u></a>	<a href="#"><u>7</u></a>

## Introducción

La primera tarea programada se basa en utilizar Microsoft SQL server para crear una base de datos simple con una sola tabla de empleados. Se requiere mostrar los datos de la base de datos en una página web, conectando SQL server con HTML por medio de un lenguaje intermediario que manejara el traspaso de datos. La página web debe no solamente mostrar los empleados en la base de datos en tiempo real, sino también habilitar una opción para insertar un nuevo empleado a la base de datos.

Esta primera tarea programada funciona como una manera simple de evaluar conceptos vistos en el curso, como la creación de procedimientos almacenados y las normas básicas para programarlas. Igualmente, nos proporciona una manera sencilla de analizar la conexión entre los datos de una página web junto con una base de datos, proporcionando conocimientos que vamos a utilizar para las siguientes asignaciones en el curso.

## Ambiente utilizado

La base de datos se basaba en Microsoft SQL server, utilizando SQL server management studio para la creación de la base de datos y sus scripts. Ya que este proyecto se realizó sin un grupo de trabajo, no fue necesario utilizar una herramienta para conectar la base de datos a varios computadores, de lo contrario se utilizará una aplicación como "hamachi" o una base de datos en la nube gratuita como el servicio "AWS" de amazon.

La conexión entre la base de datos y la capa de presentación (Página web) se realizó por medio de python, utilizando la librería 'PyODBC' como conector entre python y sql server. Para la generación de la página web y sus funcionalidades dinámicas (insertar y mostrar datos) se utilizó el framework de 'Django', el cual permite generar formularios e información dinámica en python, similar a como se utiliza javascript.

La página web está hospedada por medio de un servidor local, el cual se tiene que activar e ingresar a la dirección habilitada para utilizar la página web.


## Tabla de evaluación

Requerimiento	Valor propuesto	Comentarios
Documentación	20/20	
BD y diseño	10/10	
SP creados	9/10	El SP de listar empleados realiza un SELECT 'Error' en vez de utilizar una variable de salida para mostrar errores
Datos de prueba	5/5	
Conexión a BD	18/20	Pueden existir conexiones y cursores sin cerrar luego de realizar los procedimientos de SQL
Listar empleados	15/15	Se muestran los empleados en orden alfabético ascendente
insertar empleado	20/20	Se realizan todas las validaciones solicitadas
Total	97/100	

### Métricas del proyecto

Métrica	Valor	Comentarios
horas trabajadas	14 horas y 16 minutos	1h 25 2h 25 3h 58 1h 03 + 2h 03 1h 25 + 1h 58
cantidad de sesiones	5	
cantidad de líneas de código	315	124 SQL 133 Python 58 HTML
cantidad de entradas en github	18 commits	7 main 11 django-server
cantidad de datos de prueba	6	<ul style="list-style-type: none"> <li>• Fernando Sanchez</li> <li>• Fernando-Sanchez</li> <li>• " " (espacio blanco)</li> <li>• 21</li> <li>• F3nando Sanchez</li> <li>• -2</li> </ul>
cantidad de pruebas realizadas	10	
duración de las pruebas	30 min	
cantidad de tablas creadas	1	<ul style="list-style-type: none"> <li>• Empleado</li> </ul> Ver figura 1
cantidad de procedimientos almacenados	2	<ul style="list-style-type: none"> <li>• ListarEmpleados</li> <li>• InsertarEmpleado</li> </ul> Ver figuras 2 y 3
cantidad de funciones	6	<ul style="list-style-type: none"> <li>• sistema</li> <li>• insertarUsuario</li> <li>• validarInserción</li> <li>• insertar</li> <li>• thanks</li> <li>• error</li> </ul>

## Figuras y Gráficos

Empleado			
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	Nombre	varchar(128)	<input type="checkbox"/>
	Salario	money	<input type="checkbox"/>
			<input type="checkbox"/>

(Figura 1. Tabla empleado)

La figura 1 muestra el diseño de la base de datos dentro de SQL. Se pueden verificar los tipos de datos y nombres de los campos de la tabla. Es importante notar que el 'id' es llave primaria y está configurado con autoincremento.

```
CREATE PROCEDURE [dbo].[ListarEmpleados]
AS
BEGIN
    BEGIN TRY
        SET NOCOUNT ON;
        SELECT
            [id],
            [Nombre],
            [Salario]
        FROM [TareaP1].[dbo].[Empleado]
        ORDER BY 2
        SET NOCOUNT OFF;
    END TRY
    BEGIN CATCH
        SELECT 'Error'
    END CATCH
END
```

(Figura 2. SP ListarEmpleados)

La figura 2 muestra el script para crear el procedimiento almacenado de para listar los empleados en la base de datos. Esta tabla está ordenada por el nombre de los empleados de manera alfabética ascendente.

```

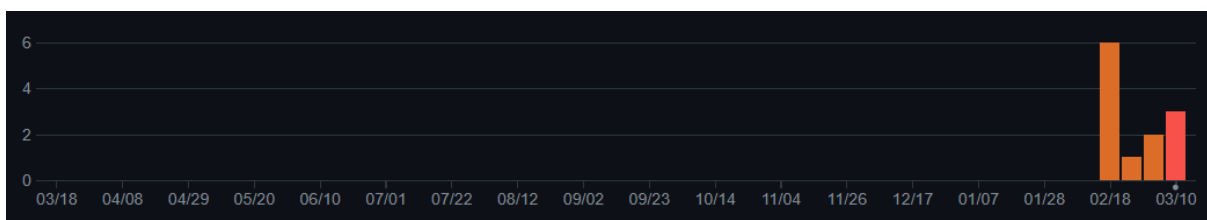
CREATE PROCEDURE [dbo].[InsertarEmpleado]
    @inNombre VARCHAR(128),
    @inSalario MONEY,
    @outResultCode INT OUTPUT
AS
BEGIN
    SET @outResultCode = 0
    BEGIN TRAN
    BEGIN TRY
        SET NOCOUNT ON;
        INSERT INTO [dbo].[Empleado] ([Nombre], [Salario]) VALUES (@inNombre, @inSalario)

        IF (SELECT COUNT([Nombre]) --verifica si hay repetidos luego de insertar un nombre
            FROM [dbo].[Empleado]
            GROUP BY [Nombre]
            HAVING COUNT([Nombre]) > 1) > 1
        BEGIN
            SET @outResultCode = 50001
            ROLLBACK
        END
    ELSE
    BEGIN
        COMMIT
    END
    SET NOCOUNT OFF;
    END TRY
    BEGIN CATCH
        SET @outResultCode = 50002
        ROLLBACK
    END CATCH
END
GO;

```

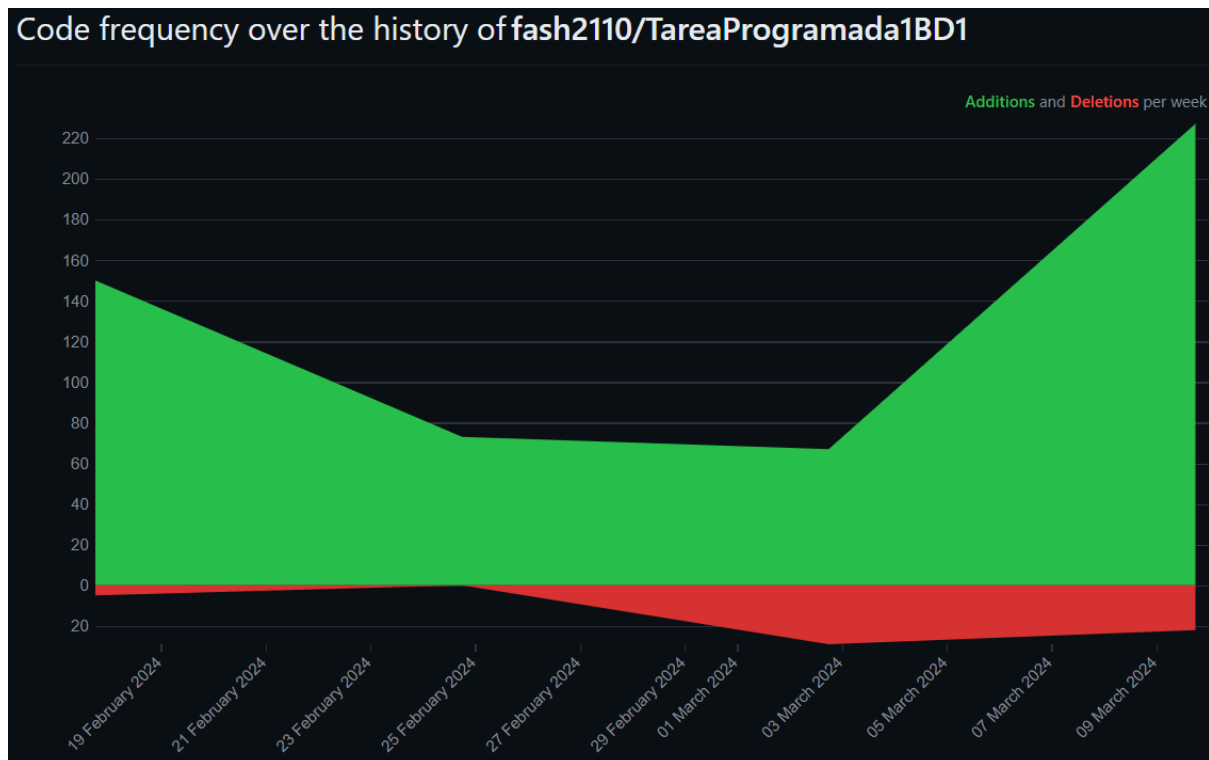
(Figura 3. SP insertarEmpleado)

La figura 3 muestra el script para generar el procedimiento almacenado que inserta un empleado nuevo a la base de datos. Este procedimiento inserta al empleado y luego revisa si se repite el nombre en la tabla. En caso de que exista un empleado con el mismo nombre, se realiza un rollback y no ocurre la inserción.



(Figura 4. Gráfico de commits)

La figura 4 demuestra los commits realizados en el repositorio de github, este gráfico demuestra una idea de la cantidad de trabajo realizado por medio de los commits realizados cada semana.



(Figura 5. Gráfico de frecuencia de código )

La figura 5 es un gráfico de frecuencia de código a través de las semanas. La figura verde representa la cantidad de líneas nuevas agregadas a los diferentes archivos del repositorio, mientras que la figura roja representa la cantidad de líneas eliminadas de los archivos existentes en el repositorio. El gráfico ayuda a demostrar el desarrollo del trabajo durante la semana, además de dar una idea de los cambios realizados mientras transcurre el proyecto.

<https://bdupcfrrnd.blogspot.com>

(Figura 6. Blog)

La figura 6 es un hipervínculo al blog del proyecto, donde se pueden encontrar las entradas de cada una de las sesiones. Estas incluyen fecha, horas de inicio y fin, objetivos propuestos y realizados, problemas encontrados y su solución y anotaciones de cada sesión.

<https://github.com/fash2110/TareaProgramada1BD1>

(Figura 7. Repositorio Github)

La figura 7 es el hipervínculo al repositorio de github donde se encuentra todo el código utilizado para el proyecto. Dentro se puede ver un historial de los archivos de código, junto a los gráficos que demuestran la realización del proyecto.