Tarea 1 Prueba de concepto

Andrés Jiménez Valverde
Helena Vargas Quiros

Tecnológico de Costa Rica

IC 4301 : Bases de datos 1

Franco Quirós Ramírez

25 de Marzo del 2025

Índice de contenido

Índice de figuras

Capítulo 1. Introducción

El presente documento tiene como objetivo presentar el contexto, ambiente, resultados y métricas de la Tarea Programada 1 del curso de Bases de Datos 1 del I semestre del 2025. Esta primera tarea consiste en una prueba de concepto en la que se deben realizar funcionalidades simples SQL y correrlas en un web browser. Esto con el fin de familiarizarse con el lenguaje y el ambiente de desarrollo que se utilizará a lo largo del curso, así como los diferentes lineamientos y buenas prácticas de programación, los cuales son necesarios para un desarrollo de calidad en la vida profesional.

Entre los objetivos y requerimientos del proyecto se encuentran: Implementar un ambiente de desarrollo que permite a equipos a un servidor de BD. Construir una aplicación web sencilla que acceda a la BD para consultar y hacer una operación de actualización. Ejecutar desde la aplicación procedimientos almacenados para consultar una tabla o actualizarla mediante una operación. “Subir” un dataset (o recordset) que es resultado en un SP y mostrarlo en la interfaz de usuario. Obtener el resultado de ejecución de un SP para saber si corrió correctamente o no.

En las siguientes secciones, se detalla información importante sobre el ambiente de desarrollo, el proceso de programación de la Tarea, los resultados obtenidos y las métricas del proyecto, explicado por medio de tablas y gráficos que reflejan la realidad del ambiente de desarrollo y los resultados.

Capítulo 2. Descripción del ambiente de desarrollo

En esta sección de ambiente de desarrollo se definen las principales herramientas y tecnologías utilizadas en la implementación de esta Tarea Programada, así como la forma en que los componentes se conectan e interactúan entre sí.

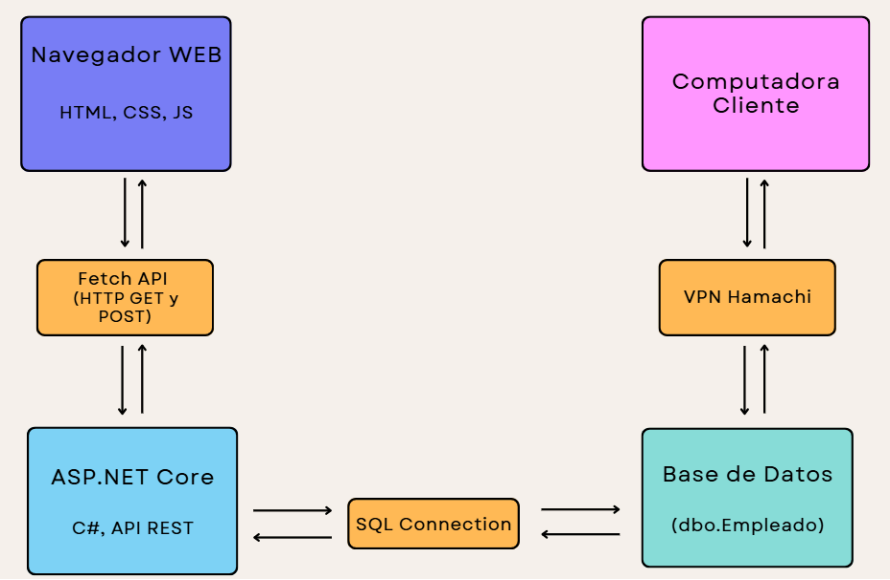
Herramientas y tecnologías utilizadas

- **IDE:** Visual Studio Community (Para desarrollo en C# y ASP.NET Core)

- **Base de Datos:** Microsoft SQL Server Community Edition
- **Lenguaje de capa lógica (backend):** C# con ASP.NET Core
- **Lenguajes de capa usuario (frontend):** HTML, CSS Y JavaScript
- **Cliente y Servidor:** El servidor de la base de datos se encuentra en la máquina de Helena Vargas. Para que la computadora cliente pueda acceder al servidor, se utiliza el VPN Hamachi, que permite establecer una red virtual privada.
- **Método de Comunicación:** El navegador web se comunica con el servidor a través de Fetch API llamando a endpoints HTTP de operaciones GET y POST. El backend se conecta a la base de datos usando SqlConnection y ejecuta consultas mediante SqlCommand.

El siguiente diagrama Figura 1 “Diagrama de ambiente de desarrollo” representa cómo los elementos del sistema interactúan en el desarrollo y ejecución de la aplicación:

Figura 1.
Diagrama de ambiente de desarrollo



Nota: Esta figura muestra las diferentes conexiones en el ambiente de desarrollo.

Arquitectura de Aplicación

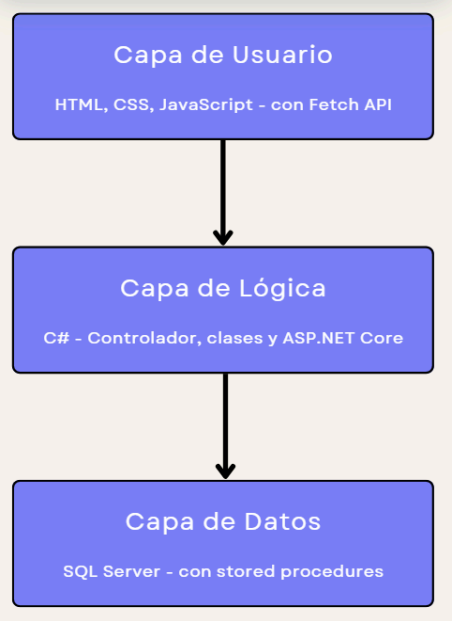
En esta subsección se describe cómo está estructurado el código y cómo interactúan sus diferentes componentes. Este proyecto sigue una arquitectura en capas, donde cada capa tiene una responsabilidad específica.

- **Capa de Datos (BD y acceso a BD):**
    - La BD contiene tabla dbo.Empleado con id, Nombre y Salario.
    - Se usan stored procedures para las operaciones principales: INSERT para agregar empleados y SELECT para recuperar la lista de empleados.
    - La clase AccesarBD.cs se encarga de la conexión a la base de datos y la ejecución de los stored procedures, así como la recuperación del dataset.
- **Capa de Lógica (Backend y API en ASP.NET Core):**
    - El backend crea dos APIs mediante la clase BDController.cs, GET para obtener la lista de empleados y POST para insertar un nuevo empleado.
    - Se utiliza SqlConnection para conectarse a la base de datos, y SqlCommand para ejecutar las stored procedures.
    - La API devuelve respuestas en formato json donde se encuentran los datos o los códigos de error dependiendo de la solicitud.
- **Capa de Usuario (Frontend . HTML, CSS, JavaScript):**
    - La interfaz de usuario se encuentra en la carpeta wwwroot, donde hay dos archivos principales: VistaUsuario.html (se despliega la lista) y InsertarEmpleado.html (formulario para insertar empleados).
    - El archivo css FontsYEstilos.css se encarga del diseño y posicionamiento de elementos estáticos.
    - Un archivo JavaScript maneja la comunicación con el backend usando Fetch API. GET muestra la lista de empleados, POST envía los datos del formulario y

muestra éxito o error. Además se incluyen las acciones de los botones, y validaciones de forma del nombre y salario.

Nota: No se utiliza MVC, solo una arquitectura por capas que se comunican mediante APIs. El backend expone APIs REST y el frontend las recibe mediante Fetch API.

Figura 2.
Gráfico de arquitectura de aplicación



Nota: Esta figura muestra las capas de arquitectura de aplicación.

Capítulo 3. Análisis de resultados

En este capítulo mostramos y comentamos los porcentajes obtenidos de satisfacción a la hora de completar cada métrica del proyecto, con comentarios de si faltaron funciones, errores sin terminar, funcionalidad y los porcentajes correspondientes obtenidos según los resultados de las pruebas.

La siguiente tabla “Tabla 1. Tabla de resultados” nos permite ver la evaluación de cada una de las partes del proyecto y el porcentaje atribuido a cada una según el funcionamiento observado en la tarea

Tabla 1.

Tabla de resultados

Item	**Resultado**
Documentación	% 20 / 20% Completa de acuerdo a la plantilla
BD y diseño	%10 / 10% Base de datos está creada de manera correcta siguiendo los estándares vistos en clase
SP creados	% 10 / 10% Mostrar e insertar empleado funciona correctamente con validaciones
Datos de prueba	% 5 / 5% La tabla de empleados se prueba con hasta un máximo de 40 empleados
Conexion a BD	% 20 / 20% La conexión existe y logra movilizar los datos de manera eficiente
Listar empleados	% 15 / 15% Listar empleados funciona correctamente y muestra todos los datos
Insertar empleados	% 20 / 20% Insertar esta validado lo que permite evitar ingresar datos erróneos además de hacer su función y añadir al SQL

Nota: Esta es una tabla que contiene los resultados de la métrica de cada parte del proyecto su % y un comentario del porque el valor correspondiente, utiliza los valores de % por cada ítem

Capítulo 4. Métricas de proyecto

En este capítulo veremos las mediciones de tiempo, pruebas, entradas, horas trabajadas y otros que nos permitan medir el esfuerzo y tiempo que se le dedicó al proyecto desde el momento en el que fue asignado. En la siguiente tabla, “Tabla 2. Tabla con las métricas del proyecto”

Tabla 2.

Tabla con las métricas de la tarea

Métrica	Valor	Comentarios
Fecha de Inicio	10/03/25	Fecha inicial
Total de horas proyecto	17 horas y media	Aproximadamente 2 horas al día
Consultas al profesor	3, 2 virtuales y 1 presencial	1 presencial sobre qué era lo necesario a compartirle del proyecto, 2 virtuales sobre si podíamos utilizar tablas HTML en vez de grids y si era necesario tener la implementación de tablas de errores
Entradas de blog	7	Repartidas equitativamente durante las 3 semanas de trabajo
Actualizaciones de github	13	Repartidas a los largo de 3 semanas, con la mayor actividad en la primera semana. En las figuras 3, 4, 5 y 6 se muestran las estadísticas de github.
Cantidad de tablas	1	dbo.Empleados
Cantidad de pruebas	20 al día	Se hacían pruebas para verificar funcionalidades específicas continuamente, y pruebas que abarcaban todas las funcionalidades el último día.
Tiempo de ejecución del script	4s Helena y 33s Andrés	Helena es la computadora host y Andrés la

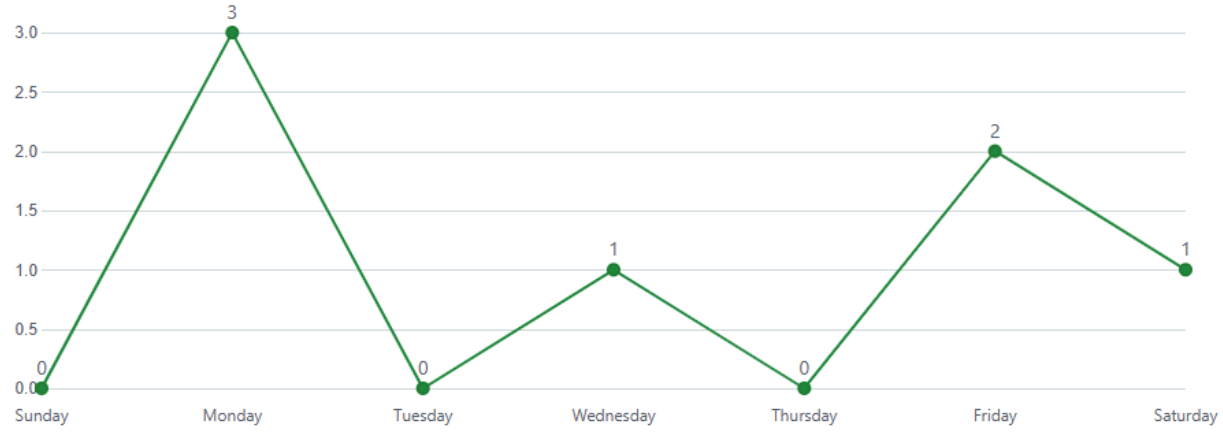
		computadora cliente
Cantidad de datos procesados	20~40	Variaba según se iban agregando más datos a la tabla
Cantidad líneas de código (SQL)	110	Divididos en Insertar Empleado y Mostrar Empleados
Cantidad líneas de código (C#)	219	Se encuentra dividido en 4 códigos los de acceso a la BD,controladores y programa principal
Cantidad líneas de código (JS)	152	Funcionalidad de botones y conexión con Fetch API
Cantidad líneas de código (HTML)	48+	Se divide en 2 archivos html para la página de insertar y vista, el + corresponde a que se posee la tabla dinámica
Cantidad líneas de código (CSS)	78	Posicionamiento y colores
Cantidad líneas de código (Total)	607	

Nota: Aquí se indican varias de las cantidades medibles que se tengan que tratar del proyecto ya sean horas o cantidades de pruebas y comentarios acerca de cada una

Las siguientes figuras “Figura 3. Cantidad de entradas en la semana del 9 de marzo en github”, “Figura 4. Cantidad de entradas en la semana del 16 de marzo en github” y “Figura 5. Cantidad de entradas en la semana 23 de marzo en github” nos permiten tener una visualización más detallada de los días en los que se trabajó y se subió la información al repositorio, esto permite que podamos ver la actividad del proyecto y el progreso realizado de manera semanal para la realización de la tarea

Figura 3.

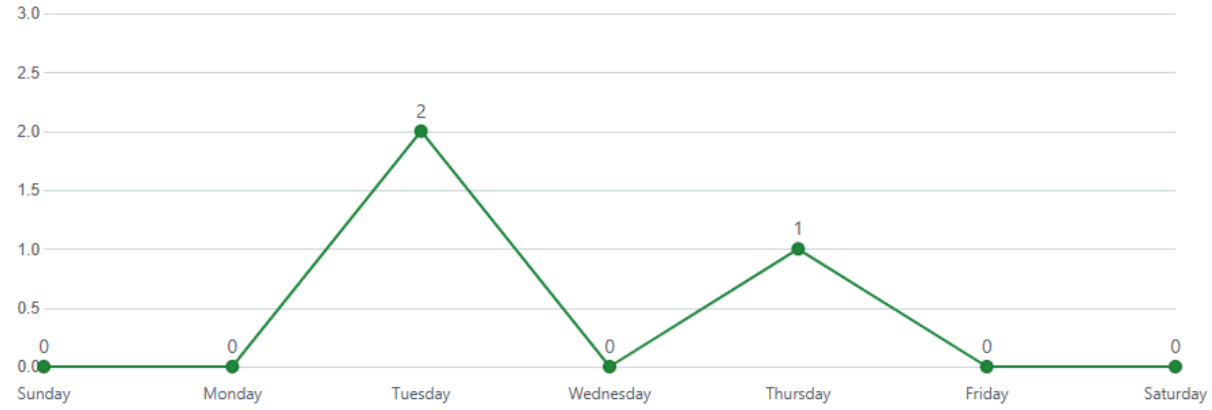
Cantidad de entradas en la semana del 9 de marzo en github



Nota: Esta figura muestra los días en los que se actualizó la base de datos de github con nuevas entradas en la semana del 9 de marzo, la tabla viaja del 9 de marzo hasta el 15 por lo cual el primer dia mostrado es 10 de marzo

Figura 4.

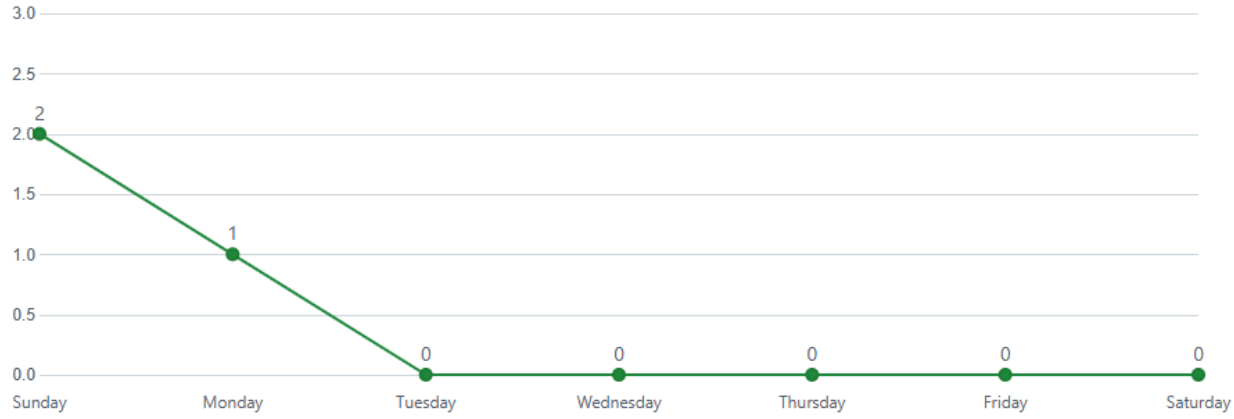
Cantidad de entradas en la semana del 16 de marzo en github



Nota: Esta figura muestra los días en los que se actualizó la base de datos de github con nuevas entradas en la semana del 16 de marzo hasta el 22 de marzo

Figura 5.

Cantidad de entradas en la semana del 23 de marzo en github

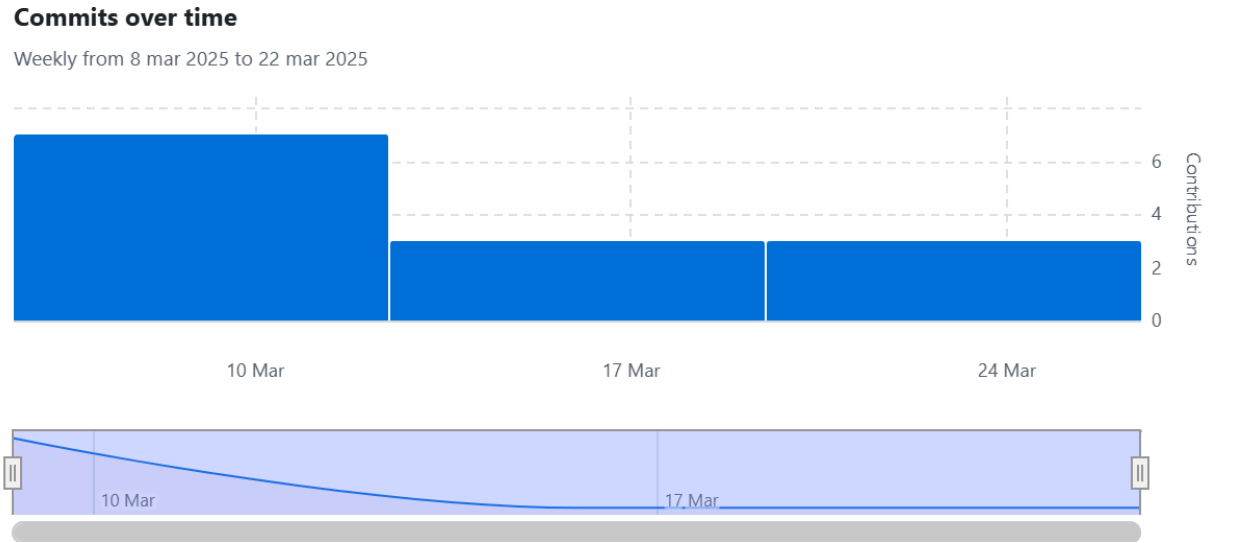


Nota: Esta figura muestra los días en los que se actualizó la base de datos de github con nuevas entradas en la semana del 23 de marzo hasta la entrega del proyecto

Mientras tanto la siguiente figura “Figura 6. *Contribuciones totales a lo largo del proyecto de manera semanal desde el 8 de marzo hasta el 22 de marzo” es una figura resumen que nos permite ver de manera compacta el progreso realizado semanalmente en github y la cantidad de entradas realizadas en cada lapso de tiempo*

Figura 6.

Contribuciones totales a lo largo del proyecto de manera semanal desde el 8 de marzo hasta el 22 de marzo



Nota: Esta figura muestra de manera compresa las entradas al github a lo largo del tiempo según las semanas de trabajo

Links adicionales

Link al repositorio GitHub:
https://github.com/LilyJmz/Tarea-1-prueba-de-concepto

Link al blogger:
tarea1bases12025.blogspot.com