Análisis de Resultados

Primera Tarea Programada (Prueba de Concepto)

Brian Esteban Ramírez Arias

Escuela de Ingeniería en Computación

Instituto Tecnológico de Costa Rica (ITCR)

Base de Datos 1

Profesor: Franco Quiros

Fecha de Entrega: 24 de marzo de 2025

Correo de contacto: brianramirez01arias@gmail.com

Resumen:

Este documento analiza los resultados obtenidos en la primera tarea programada, enfocada en la conexión de una base de datos a una aplicación web. Se detallan las tecnologías empleadas, la arquitectura de red y los flujos de datos dentro del sistema.

ÍNDICE

ÍNDICE DE FIGURAS

I. Introducción

Este documento evalúa el desarrollo de la primera tarea programada, que consistió en la creación de una aplicación web conectada a una base de datos para operaciones de consulta e inserción de datos. Se implementaron procedimientos almacenados y validaciones en la capa de presentación para garantizar la integridad de los datos.

Recursos del proyecto:

- Aplicacion web
- Endpoint "GET Sorted Employees" del API
- Repositorio en GitHub
- Bitácoras en Medium

II. Descripción del Ambiente de Desarrollo

II-A. Arquitectura de Red

Se utilizó **Azure SQL Database** en su versión gratuita para estudiantes, lo que permitió disponer de **100,000 segundos vCore mensuales y 32 GB de almacenamiento sin costo**. Para fines didácticos, la base de datos no tiene restricciones de acceso por IP, permitiendo la conexión desde cualquier ubicación con autenticación SQL.

II-B. Arquitectura de la Aplicación

La aplicación sigue una arquitectura cliente-servidor, desplegada en **Vercel**:

- **Base de Datos:** Azure SQL Database con procedimientos almacenados para inserción y consulta.
- **API:** Implementada con Express.js y desplegada en Vercel, facilitando el acceso global a los datos.
- **Front-end:** Desarrollado con React y Next.js, también alojado en Vercel.

II-C. Flujo de la Aplicación - Consulta de Empleados

El siguiente diagrama ilustra el flujo de datos para la consulta de empleados:

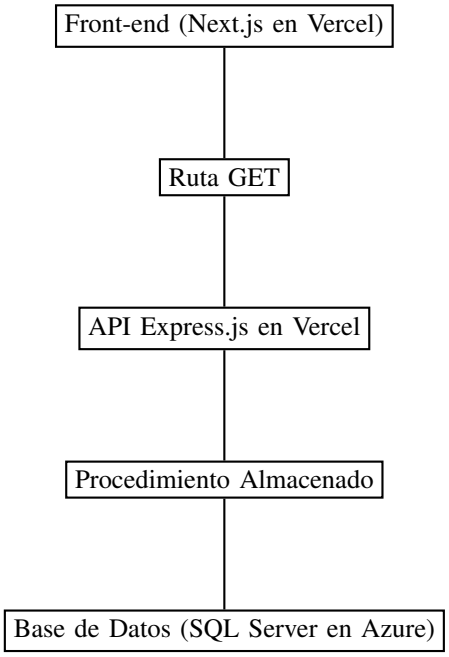

Figura 1. Flujo de consulta de empleados

II-D. Flujo de la Aplicación - Inserción de Empleados

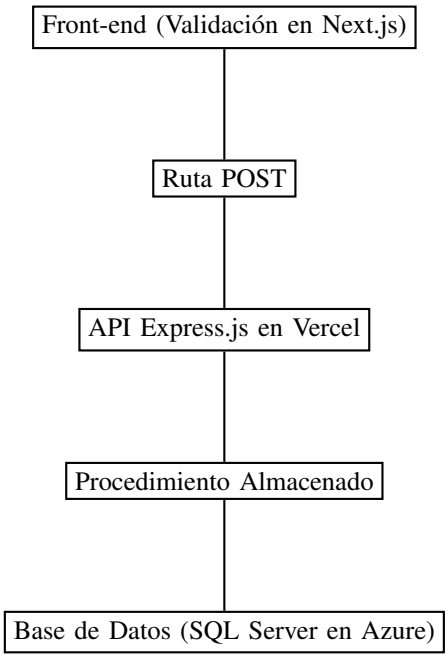

Figura 2. Flujo de inserción de empleados

III. Tecnologías Utilizadas

III-A. Front-end

- **React:** Biblioteca de JavaScript para la construcción de interfaces de usuario interactivas basadas en componentes reutilizables.
- **Next.js:** Framework para React que optimiza el renderizado del lado del servidor (SSR) y facilita el enrutamiento.
- **Tailwind CSS:** Framework de diseño basado en utilidades que permite crear estilos de manera eficiente y responsiva.

III-B. Back-end

- **Express.js:** Framework minimalista para Node.js que facilita la creación de servidores y APIs REST.
- **mssql.js:** Librería para la conexión con bases de datos SQL Server en Node.js.

III-C. Base de Datos

- **Microsoft SQL Server (MSSQL):** Implementado en Azure con acceso remoto.
- **Lenguaje de Programación: NodeJS/TypeScript** para la aplicación web; **SQL** para la base de datos.

IV. Análisis de Resultados

IV-A. Evaluación de Requerimientos

Cuadro I
Evaluación de Requerimientos

Elemento	Valoración	% Implementación	Comentarios
Base de Datos Creada	Exitosa	100 %	Se creó correctamente.
Procedimientos Almacenados	Exitosa	100 %	Implementados con éxito.
Interfaz de Usuario	Exitosa	100 %	Funcional y validada.
Validación de Datos al Crear	Exitosa	100 %	Completas en front-end y backend.
Actualización del Grid	Exitosa	100 %	Se refresca correctamente.
Manejo de errores	Exitosa	100 %	Se manejan los errores y se informa correctamente al desarrollador y usuario.
Creacion de API(Adicional)	Exitosa	100 %	RESTful API funcional que conecta la DB con el cliente.

REFERENCIAS

[1] Registro 1 - Bitácora. Disponible en: https://medium.com/@bracr10/registro-1-inicio-de-la-bitÃącora-tarea-1-db-poc-4a2b79380172
[2] Registro 2 - Bitácora. Disponible en: https://medium.com/@bracr10/registro-2-inicio-de-la-bitÃącora-tarea-1-db-poc-d4b82c1345f6
[3] Registro 3 - Bitácora. Disponible en: https://medium.com/@bracr10/registro-3-creaciÃşn-del-front-end-tarea-1-db-poc-4c1a105e3044
[4] Repositorio en GitHub. Disponible en: https://github.com/BraCR10/POC-DB1-Homework1
[5] Documentación Next.js. Disponible en: https://nextjs.org/docs/app
[6] Video Tutorial Next.js. Disponible en: https://youtu.be/7iobxzd_2wY
[7] Creación de Base de Datos en Azure. Disponible en: https://learn.microsoft.com/es-es/azure/azure-sql/database/single-database-create-quickstart
[8] Errores en SQL Server. Disponible en: https://learn.microsoft.com/en-us/sql/relational-databases/errors-events/database-engine-error-severities
[9] Video sobre MSSQL. Disponible en: https://youtu.be/tVBb79WLScc
[10] Video Express.js. Disponible en: https://youtu.be/h3FuIshjjuk
[11] Video API MSSQL. Disponible en: https://youtu.be/YuhKhkQqtP8
[12] Documentación mssql.js. Disponible en: https://www.npmjs.com/package/mssql
[13] Documentación Express.js. Disponible en: https://expressjs.com/