

Primera tarea programada (Prueba de concepto).

Uno. Objetivos.

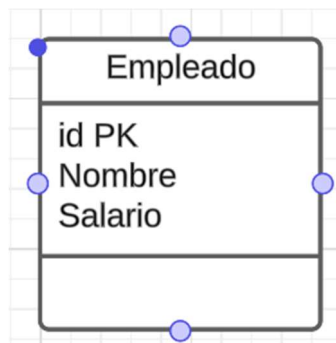
Implementar una prueba de concepto para conectar una base de datos a un programa que implementa funcionalidades simples (consulta e inserción) y que corre en un web browser.

Objetivos específicos:

- Implementar un ambiente de desarrollo que permite a equipos de trabajo de 2 personas, conectarse a un recurso compartido en donde estará instalado el servidor de BD y la BD, tal que los miembros del equipo puedan accederla desde sus estaciones de trabajo.
- Construir una aplicación web sencilla que acceda a la BD para consultar y hacer una operación de actualización.
- Conectar la BD a la aplicación.
- 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.

Dos. Descripción.

El modelo conceptual en el que se basa la tarea es una sola tabla con la siguiente estructura:



En el modelo físico (en la base de datos) se implementará la siguiente tabla:

Empleado			
	Column Name	Data Type	Allow Nulls
🔑	Id	int	<input type="checkbox"/>
	Nombre	varchar(64)	<input type="checkbox"/>
	Salario	money	<input type="checkbox"/>
			<input type="checkbox"/>

Lo anterior puede declararse, de manera no gráfica, mediante un script.

```
CREATE TABLE dbo.Empleado
(
    id INT IDENTITY (1, 1) PRIMARY KEY
    , Nombre VARCHAR(128) NOT NULL
    , Salario MONEY NOT NULL
);
```

La tarea programada consiste en:

Crear la base de datos.

Crear la tabla Empleado.

Cargar datos en la tabla manualmente mediante el estatuto insert, al menos 40 filas.

```
INSERT dbo.Empleado (Nombre, Salario)
VALUES ('Juan Perez', 200000.00);
```

Construir una aplicación que corra en un browser en internet

Hacer una conexión a la base de datos desde la aplicación.

Una vez que la aplicación “sube” se muestra un grid que muestra los empleados en orden alfabético ascendente por nombre, mostrando una interfaz similar a la siguiente.

Interfaz inicial que es también la principal.

Lista de Empleados

id	Nombre	Salario
5	Luis Chaves	200000
3	Juan Perez	150000
7	Ana Ruiz	350000
⋮	⋮	⋮

Insertar

El botón de "Insertar Empleado" levanta una interfaz de usuario que captura nombre y salario de un nuevo empleado, en la cual se valida que el nombre del Empleado contenga solamente valores alfabéticos o un guion y que el salario sea un valor monetario bien formado.

Interfaz de "Insertar Empleado"

Insertar Empleado

Nombre:

Salario:

En la interfaz de "Insertar Empleado" el botón de "Regresar" regresa a la interfaz de usuario inicial, que muestra el grid con la lista de empleados.

En la interfaz de “Insertar Empleado” el botón de “Insertar” valida que los campos de nombre y salario de un nuevo Empleado no sean vacíos, que el nombre contenga solamente valores alfabéticos o un guion y que el salario sea un valor monetario bien formado. Se deben mostrar mensajes de error al usuario en caso de que la validación falle. Si la validación es correcta, invocará un procedimiento almacenado que intentará la inserción. Algunas validaciones, tal como que el campo de entrada no sea vacío o que sus valores de entrada sean correctos deben hacerse en capa de presentación.

El procedimiento almacenado que inserta validará manualmente si ya existe un nombre y apellidos de Empleado en la BD con el nombre y apellidos capturado en la interfaz y si es así, retornará un código de error, si no, hará una inserción en la tabla Empleado con el nombre y salario capturados.

Si la ejecución del procedimiento almacenado es correcta, se mostrará un mensaje de “inserción exitosa” y luego se regresa a la interfaz de usuario inicial, el grid debe ser refrescado mostrando la fila recientemente insertada. Si hubo error, se mostrará el mensaje “Nombre de Empleado ya existe.” y se vuelve a la interfaz de usuario de insertar.

Tres. ¿Qué se pide?

El código en capa lógica para el funcionamiento de las páginas.

La base de datos creada, las tablas con información cargada y los procedimientos almacenados para realizar las funcionalidades solicitadas, deben ser al menos dos procedimientos, el que obtiene la lista de empleados y el que inserta.

La documentación que son dos documentos, una bitácora y un análisis de resultados.

La bitácora

Debe ser escrita en una herramienta para hacer blogs, por ejemplo www.blogger.com, para cada entrada se indica la hora de inicio y la hora de fin respecto de una sesión de trabajo, la cantidad de horas trabajadas, se hace un relato de los avances acerca de los problemas encontrados, como fueron resueltos; dudas, divergencias de criterio, forma en que trabajó el equipo de trabajo, problemas con la instalación del software, problemas de aprendizaje del framework, investigaciones, pruebas de concepto, experiencias, moralejas, ayuda recibida, consejos a dar, buenas prácticas descubiertas, incluir preguntas que se hagan al profe en el foro o en comunicaciones privadas, desahogarse, etc.; La entrada en el blog describe el proceso de solución de la tarea programada; una descripción sincera y detallada será bien evaluada. Incluya referencias externas a recursos utilizados en internet para solventar dudas o resolver

problemas, debe incluir los mensajes de error y como fueron resueltos. Agregar versiones del código con un error y la versión con el arreglo.

Debe escribir las entradas en la bitácora teniendo en mente que serán la prueba de que el equipo de trabajo laboró regularmente (o sea que no hizo la tarea la noche previa), que muestre el proceso de aprendizaje (paulatino e incremental), en otras palabras, que sea evidencia de que la tarea NO fue copiada.

Análisis de resultados

Es un documento que debe ser formal o sea profesionalmente bien presentado, su objetivo principal es indicar que funciona o que no, de los requerimientos y mostrar las métricas del proyecto.

Los contenidos del documento deben ser:

- Portada,
- Índice de contenido,
- Índice de figuras,
- Introducción,
- Descripción del ambiente de desarrollo: (Arquitectura de red y arquitectura de aplicación, con una explicación de las tecnologías utilizada),
- Análisis de resultados descritos mediante una tabla donde para cada elemento que se evalúa de la tarea, según la plantilla (rúbrica) de evaluación, se muestra el nombre del elemento (requerimiento, artefacto de diseño, etc.), una valoración de si esta implementado exitosamente; o si no; un % de implementación, y si no es un 100% un comentario que explique que falta para que ese elemento esté a un 100%; y
- Métricas del proyecto, a través de una tabla se indican las métricas del proyecto, por ejemplo: horas trabajadas, cantidad de sesiones, cantidad de líneas de código, cantidad de entradas en el git hub, cantidad de datos de pruebas procesados, cantidad de pruebas realizadas, tiempo de duración de las pruebas, cantidad de tablas creadas, cantidad de procedimientos almacenados, cantidad de funciones, etc., etc., todas las métricas que se puedan derivar del proyecto.

Se pueden agregar gráficos que ofrezca el github, o cualquier elemento que mejore la calidad del documento.

Toda tabla o gráfico debe tener una referencia textual. Por ejemplo: “En la siguiente tabla se indica el porcentaje de implementación de cada uno de los elementos de evaluación del proyecto y se comenta su estado final.”

El uso del github es obligatorio, será una prueba de trabajo constante y colaborativo.

Cuatro. Reglas.

Todo el código de programación referido a base de datos, debe ser un procedimiento almacenado. No puede haber SQL incrustado en capa lógica, lo único permitido es invocar un procedimiento almacenado.

Grupos de 2 personas. Motor de base de datos: MS SQL cualquier versión superior a 2014. Código en capa lógica, en el lenguaje o framework de su preferencia.

Fecha de entrega: 24 de Marzo.