

Actividad 2 - Conexión y Tablas

Lenguajes de Programación II

Ingeniería en Desarrollo de Software

Tutor: Miguel Ángel Rodríguez Vega.

Alumno: Uziel Abisai Martinez Oseguera.

Fecha: 28/11/2023.

Índice

INTRODUCCIÓN	1
DESCRIPCIÓN	2
JUSTIFICACIÓN	3
DESARROLLO	4
CONEXIÓN	4
TABLAS	5
CÓDIGO	6
CONCLUSIÓN	7
REFERENCIAS	8

Introducción

En esta actividad, se aborda la creación de una estructura de clases en C++ que permita a una empresa gestionar información sobre sus empleados. Se busca establecer una conexión con una base de datos, así como la generación de tablas que almacenen datos relevantes de los trabajadores, como su identificación, detalles personales, roles y detalles específicos para directivos. Este proceso se enmarca en la necesidad actual de las empresas de sistematizar y organizar eficientemente la información de sus empleados para una gestión más efectiva de recursos humanos.

Descripción

La actividad se centra en la implementación de una solución informática que permite a la empresa UNI controlar y gestionar la información de sus empleados mediante clases y herencia en C++. Los requisitos detallados definen los datos a capturar y cómo se relacionan entre sí, incluyendo la diferenciación entre empleados directivos y normales. El objetivo es crear una estructura de datos que refleje con precisión la información necesaria para la gestión de recursos humanos, abarcando desde datos personales hasta detalles específicos del puesto laboral.

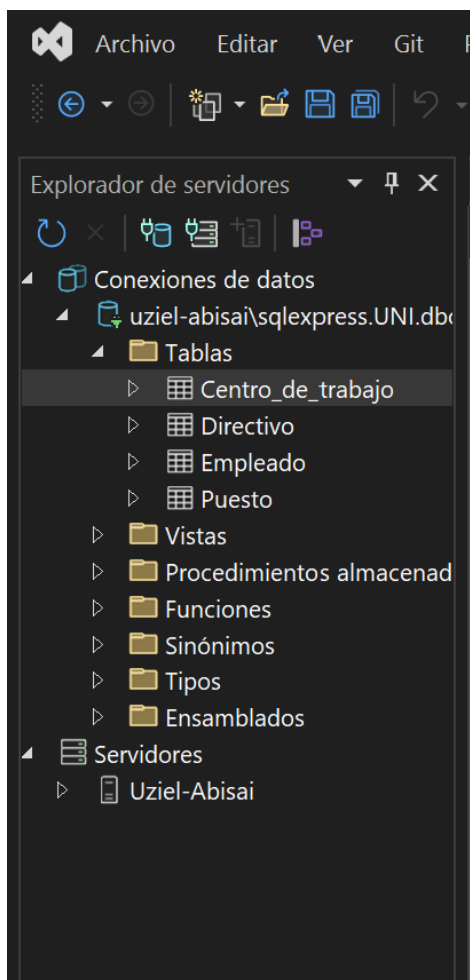
Justificación

Este enfoque de desarrollo en C++ y el uso de bases de datos ofrece una solución eficiente y escalable para la gestión de empleados. La adopción de esta metodología permite una mejor organización y accesibilidad de la información de los empleados de la empresa UNI, facilitando la toma de decisiones informadas en cuanto a recursos humanos. Además, el uso de bases de datos proporciona robustez y seguridad en el almacenamiento y acceso a la información, cumpliendo con estándares de seguridad y privacidad requeridos en entornos empresariales.

Desarrollo


Conexión


En este apartado realizamos la conexión de la base datos SQL Sever a Visual Studio.



Tablas

En este apartado le mostramos las tablas de la base datos.

	Nombre	Tipo de datos	Permitir valores NULL
	Número_de_centro	int	<input type="checkbox"/>
	Nombre_de_centro	varchar(100)	<input type="checkbox"/>
	Ciudad	varchar(100)	<input type="checkbox"/>
			<input type="checkbox"/>

	Nombre	Tipo de datos	Permitir valores NULL
	Número_de_empleado	int	<input type="checkbox"/>
	Número_del_centro_que_s	int	<input type="checkbox"/>
	Prestación_de_combustible	bit	<input type="checkbox"/>
			<input type="checkbox"/>

	Nombre	Tipo de datos	Permitir valores NULL
	Número_de_empleado	int	<input type="checkbox"/>
	Nombre	varchar(100)	<input type="checkbox"/>
	Apellido_paterno	varchar(100)	<input type="checkbox"/>
	Apellido_materno	varchar(100)	<input type="checkbox"/>
	Fecha_de_nacimiento	datetime	<input type="checkbox"/>
	RFC	varchar(13)	<input type="checkbox"/>
	Nombre_de_centro	int	<input type="checkbox"/>
	Descripción_del_puesto	varchar(200)	<input type="checkbox"/>
	Es_directivo	bit	<input type="checkbox"/>
			<input type="checkbox"/>

Código

En este apartado le mostramos los códigos de la base de datos.

```
1 CREATE TABLE [dbo].[Centro_de_trabajo] (
2     [Número_de_centro] INT NOT NULL,
3     [Nombre_de_centro] VARCHAR (100) NOT NULL,
4     [Ciudad] VARCHAR (100) NOT NULL,
5     PRIMARY KEY CLUSTERED ([Número_de_centro] ASC)
6 );
```

```
CREATE TABLE [dbo].[Directivo] (
    [Número_de_empleado] INT NOT NULL,
    [Número_del_centro_que_supervisa] INT NOT NULL,
    [Prestación_de_combustible] BIT NOT NULL,
    PRIMARY KEY CLUSTERED ([Número_de_empleado] ASC),
    CONSTRAINT [fk_Número_de_empleado] FOREIGN KEY ([Número_de_empleado]) REFERENC
    CONSTRAINT [fk_Número_del_centro_que_supervisa] FOREIGN KEY ([Número_del_cen
);
```

```
CREATE TABLE [dbo].[Empleado] (
    [Número_de_empleado] INT NOT NULL,
    [Nombre] VARCHAR (100) NOT NULL,
    [Apellido_paterno] VARCHAR (100) NOT NULL,
    [Apellido_materno] VARCHAR (100) NOT NULL,
    [Fecha_de_nacimiento] DATETIME NOT NULL,
    [RFC] VARCHAR (13) NOT NULL,
    [Nombre_de_centro] INT NOT NULL,
    [Descripción_del_puesto] VARCHAR (200) NOT NULL,
    [Es_directivo] BIT NOT NULL,
    PRIMARY KEY CLUSTERED ([Número_de_empleado] ASC),
    CONSTRAINT [fk_Nombre_de_centro] FOREIGN KEY ([Nombre_de_centro]) REFERENC
    CONSTRAINT [fk_Descripción_del_puesto] FOREIGN KEY ([Descripción_del_puesto]
```

```
1 CREATE TABLE [dbo].[Puesto] (
2     [Descripción_del_puesto] VARCHAR (200) NOT NULL,
3     [Directivo] BIT NOT NULL,
4     PRIMARY KEY CLUSTERED ([Descripción_del_puesto] ASC)
5 );
6
7
```


Conclusión

La creación de una estructura de clases en C++ y la conexión con una base de datos SQL para gestionar la información de empleados se revela como una solución crucial en entornos empresariales modernos. Esta actividad evidencia la importancia de la organización eficiente de datos para la toma de decisiones fundamentadas en la gestión de recursos humanos. La habilidad para gestionar y acceder rápidamente a información detallada y estructurada de empleados potencia la eficiencia operativa y la capacidad de adaptación de las empresas ante cambios internos. La combinación de lenguajes de programación como C++ con bases de datos sólidas ofrece un enfoque poderoso y versátil para abordar las necesidades empresariales en la actualidad.

Referencias

Link del archivo y código en [GitHub](#).

Link del archivo zip en [Drive](#).