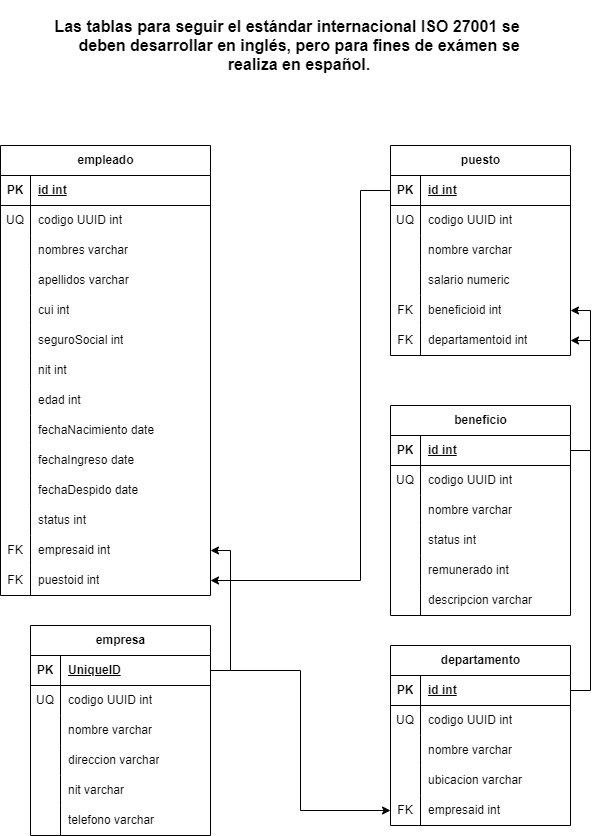
Agregar cada uno de los siguientes ejercicios a un repositorio de git y compartir la URL (opcional)

**Serie I**

1. Crear una aplicación para calcular el área de triángulos, este debe contener las columnas **Nombre, Base, Altura y Área**.

**Formula área = (b\*h) /2 RESUELTA: https://github.com/r1h1/byb-prueba-danielrivas.git**

1. Crear diagrama entidad relación y tablas en la base de datos para almacenar registro de Empleados, determine según sus necesidades las tablas a crear.



1. Crear tabla **Productos** con los campos (Nombre, Precio, Fecha) y realice aplicación para CRUD en C#, desarrollo web o escritorio a su elección.

**https://github.com/r1h1/byb-prueba-danielrivas.git**

1. Crear un procedimiento almacenado para CRUD de tabla **Productos.**

CREATE DATABASE pruebabyb;

USE pruebabyb;

-- Crear tabla Productos con los campos (Nombre, Precio, Fecha) y

-- realice aplicación para CRUD en C#, desarrollo web o escritorio a su elección.

CREATE TABLE productos(

id INT PRIMARY KEY NOT NULL IDENTITY(1,1),

codigoProducto INT UNIQUE,

nombreProducto VARCHAR(100),

precioProducto NUMERIC(10,2),

fechaProducto DATE

);

-- Crear un procedimiento almacenado para CRUD de tabla Productos.

--SELECT

CREATE PROCEDURE selectProductos

AS

SELECT \* FROM productos

GO;

--INSERT

CREATE PROCEDURE insertProductos

@codigoProducto INT,

@nombreProducto VARCHAR(50),

@precioProducto NUMERIC(10,2),

@fechaProducto DATE

AS

BEGIN

INSERT INTO productos(codigoProducto, nombreProducto, precioProducto, fechaProducto)

VALUES (@codigoProducto, @nombreProducto, @precioProducto, @fechaProducto);

END;

--UPDATE

CREATE PROCEDURE updateProductos

@id INT,

@codigoProducto INT,

@nombreProducto VARCHAR(50),

@precioProducto NUMERIC(10,2),

@fechaProducto DATE

AS

BEGIN

UPDATE productos SET codigoProducto = @codigoProducto,

nombreProducto = @nombreProducto, precioProducto = @precioProducto,

fechaProducto = @fechaProducto WHERE id = @id;

END;

--DELETE

CREATE PROCEDURE deleteProductos

@id INT

AS

BEGIN

DELETE FROM productos WHERE id = @id;

END;

1. Crear un algoritmo para el funcionamiento de un reloj (hora, minuto, segundo).

using System;

public class HelloWorld

{

public static void Main(string[] args)

{

for (int hora = 00; hora <= 23; hora++)

{

for (int minutos = 00; minutos <= 59; minutos++)

{

for (int segundos = 00; segundos <= 59; segundos++)

{

Console.WriteLine(hora + ":" + minutos + ":" + segundos);

}

}

}

Console.ReadKey();

}

}**Serie II**

1. Defina que es una clase y que es un objeto: R// Una clase como tal es una plantilla para crear objetos y un objeto es un llamado o instancia a la clase, por ejemplo si se tiene la clase perro y sus métodos alto y bajo a la hora de llamar esa clase en un main u otra clase se pueden utilizar estos métodos sin reescribir toda la estructura pasando parámetros
2. Defina que es un método constructor y realice un ejemplo (pseudo código o C#): R// Un constructor como tal establece valores para que estos sean accesibles a demás datos y para tener una buena estructuración de código, un ejemplo sería:

public string nombre;

public int edad;

public int status;

public Employee(string nombreEmpleado, int edadEmpleado, int statusEmpleado){

nombre = nombreEmpleado;

edad = edadEmpleado;

status = statusEmpleado;

}

1. Defina que es herencia (apoyarse con un ejemplo): R// Herencia es el proceso de heredar como tal métodos y atributos de otras clases como por ejemplo obtener algún parámetro en específico de otra clase
2. Defina que son los modificadores de acceso y cuales son: R// Son las palabras reservadas como protegido, publico, interno, privado y demás que definen cómo actuará esa variable durante todo el código, en este caso protected, public, intern, privated y de tipo file
3. Diferencia entre API REST y SOAP: R// API es una interfaz de programación de aplicación enfocada en el manejo de datos, el api REST como tal maneja interfaz de tipo JSON orientada en entrega de información a través de los protocolos HTTP y manejo de estos mismos, SOAP está enfocado en la estructura XML la cual envía y muestra información como tal referente a temas como facturación electrónica y demás bajo el formato XML W3
4. Defina que es encapsulación (apoyarse con un ejemplo): R// Encapsulación es el término para llamar a ocultar datos como tal para que no sean accesibles desde cualquier medio con métodos y variables de clase privada o protegida siendo accesibles solo por ciertas partes del código como tal
5. Que es polimorfismo: R//Simplemente es una clase accesible para sus hijas o nietas por ejemplo a la hora de que una clase vehículo tenga partes, repuestos etc pueden ser accesibles desde este
6. Defina que es un inner join en bases de datos: R// Es el proceso de unir datos de dos tablas a través de sus llaves foráneas, como tal por ejemplo si se tiene una tabla de tipos de whiskys y una de marcas cada whisky está relacionada con una marca, cada marca cuenta con un ID único y nombre por lo tanto a través de la unión y búsqueda de ese ID se puede obtener el nombre de la máscara, hay varios tipos de JOIN como los left join, inner join, right join y el join simple
7. Defina DDL, DML en base de datos y descríbalos: R// DDL son la lógica de base de datos como creación de esta, tablas, procedimientos, relaciones, etc y DML es el T-SQL en resumen, como el insertado, seleccionado, actualizado y borrado de los datos aunque por buenas prácticas y desarrollo seguro no se debe utilizar nunca el borrado de datos, solamente cambiar estado y no mostrarlos para evitar pérdida de información voluntaria o involuntaria
8. Describa los tipos de dato primitivos de C#: R// Son datos básicos como variables, arreglos, operadores, asignadores y demás que están por defecto en C#