Tarea 1

SC-701 Programacion Avanzada Web - Miercoles 6-9

Entrega, Semana 04

Consideraciones

- Tomar en cuenta de que el proyecto necesita:
- Solucion
- Script de SQL
- Enlace y commandos para EF

Indicaciones

- Verificar que la solucion compile
- Correr el script productdb.sql
- Implementar los pasos en el enlace: EF Scaffolding
- Seguir los pasos del enlace: EF Scaffolding

Puede pedir a google o AI que traduzca la pagina

- Scripts. Para correr los scripts dentro del proyecto lo puede hacer en Tools > NuGet Pakcage Manager > Package Manager Console.

```
dotnet add package Microsoft.EntityFrameworkCore.SqlServer
dotnet add package Microsoft.EntityFrameworkCore.Design
dotnet add package Microsoft.EntityFrameworkCore.Tools

# Toda la Database
dotnet ef dbcontext scaffold
"Server=<<servername>>;Database=<<DBNAME>>;Trusted_Connection=True;TrustServerCertificate=True;" Microsoft.EntityFrameworkCore.SqlServer -o Models

# Unico Modelo
dotnet ef dbcontext scaffold
"Server=<<servername>>;Database=CatalogDB;Trusted_Connection=True;TrustServerCertificate=True;" Microsoft.EntityFrameworkCore.SqlServer -o Models -t <<DB_TABLE>> -f

dotnet tool install --global dotnet-ef
```

dotnet tool update --global dotnet-ef

- El scaffolding the EF creara unos modelos en el proyecto por lo que debe hacerlo dentro un assembly llamado APW.Data | PAW.Data | etc
- El scaffolding creara un Context class
- Debe crear y utilizar el patron de Repository (unicamente repository) para la clase product [Repository Pattern Link]
- Registrar la interfaz como un servicio en la clase de Program.cs (i.e. builder.Services.AddScoped<Interfaz, Implementacion>)
- Esta implementacion debe ser inyectada en el controller del ProductApiController en el proyecto de APW.API
- Asegurese de crear un ProductViewModel compatible con el Entity creado por EntityFramework
- Debe mostrar la lista de productos y modificar el controller de APW.API para que lea desde la dependencia de productRepository en lugar de leer un metodo que genera productos aleatorios
- Una vez que tiene acceso a los productos debe implementar lo siguiente en la vista:
- if statement

```
@if (EvaluarPrecio(item.price))
{
   algo
}
else
{
   algo
}
```

- switch statement

```
@switch (product.Category)
{
   case "opcion1":

algo


      break;

   case "opcion2":

algo


      break;

   default:

Otra categoría


      break;
}
```

- foreach statement

```
@foreach (var product in Model)
   {


         algo


   }
```

- while utilice el mod para calcular el par y el impar

```
@while (indice < productos.Count && totalAcumulado <= limite)
{
   var producto = productos[indice];
   totalAcumulado += producto.Precio;

      @producto.Nombre
      $@producto.Precio

   indice++;
}
```

- function utilice esta funcion como evaluacion dentro del IF

```
@functions {
   public string EvaluarPrecio(decimal price)
   {
      if (price > 100)
      {
         return "Producto caro";
      }
      else if (price > 50)
      {
         return "Producto moderado";
      }
      else
      {
         return "Producto económico";
      }
   }
}
```

Rúbrica de Evaluación (Calificación del 1 al 20)

Rubro	Descripción	Puntaje
1 punto por cada indicacion		1 (*18)
2 puntos funcionando correctamente		2
Total		**20**