

DOCUMENTACION EXPLICATIVA

NOTA:

En esta solución, es importante destacar que algunas funcionalidades están incompletas o no funcionan según lo esperado. A continuación, detallaré los pasos necesarios para mejorar y completar la aplicación, así como las áreas que aún están en desarrollo y las limitaciones actuales que enfrentamos durante el proceso de desarrollo.

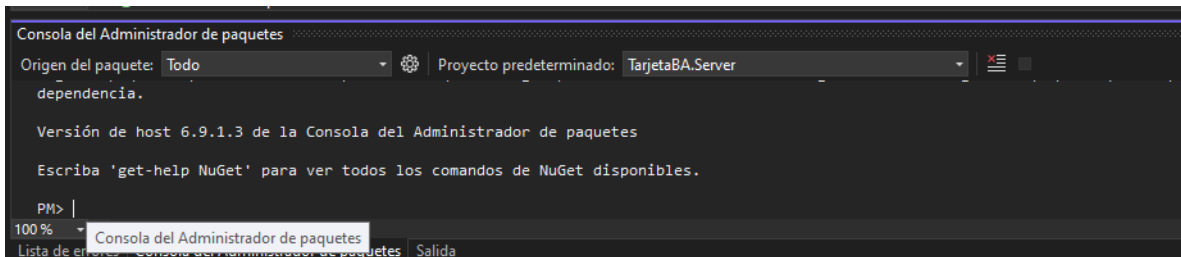
PASO PARA HACER FUNCIONAR LA APLICACIÓN

Conectar a la Base de datos

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "DefaultConnection": "Server=LAPTOP-7SKAH6B6;Database=TarjetasCredito;User Id=sa;Password=salva;MultipleActiveResultSets=true;Encrypt=true"
  }
}
```

La cadena de conexión debe contener el nombre de usuario y la contraseña del servidor SQL Server. Asegúrate de configurar correctamente esta cadena para establecer la conexión con la base de datos.

Migraciones



Consola del Administrador de paquetes

Origen del paquete: Todo Proyecto predeterminado: TarjetaBA.Server

dependencia.

Versión de host 6.9.1.3 de la Consola del Administrador de paquetes

Escriba 'get-help NuGet' para ver todos los comandos de NuGet disponibles.

PM> |

100 % Consola del Administrador de paquetes

Lista de errores: Consola del Administrador de paquetes Salida

Para migrar debe colocar los siguientes comandos

Para borrar la migración existente en el proyecto de API

Remove-migration

Para crear una nueva migración

add-migration Tarjetas

Para digitarlo al base de datos

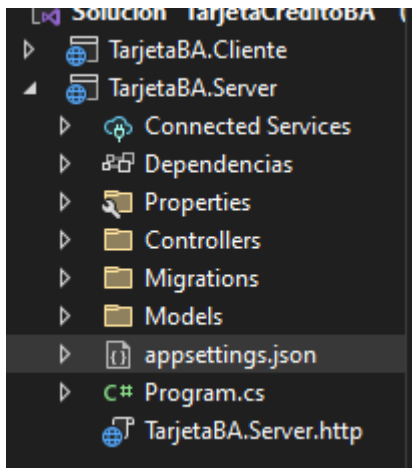
update-database

Y listo ya se conecta la tablas con su SQL SERVER conectarse con la el comando use **TarjetasCredito** para usar la base de datos y sus tablas

La migración incluye tres tablas relacionales con claves primarias y foráneas. Originalmente, se había diseñado utilizando tablas no relacionales, lo que causó una reestructuración para satisfacer los requisitos de la API. Esta reestructuración fue una de las limitaciones que enfrente durante el desarrollo y ha requerido tiempo adicional para su implementación.

Antes de implementar la aplicación digite los scripts almacenado con PLSQL para la obtención y creación de listas , seguido le comparto colección de POSTMAN ya que funciona la parte de API mas que la de MVC.

He realizado buena estructura de MVC y API



La creación de relacionar llaves primarias y foráneas en ASP.Net me ha llevado tiempo en desarrollarlo .

```

modelBuilder.Entity<Tarjeta>(entity =>
{
    entity.HasKey(e => e.IdTarjeta)
        .HasName("PK__TARJETA__A3C02A1033A1C444");

    entity.ToTable("Tarjetas");

    entity.Property(e => e.Cuenta)
        .HasMaxLength(50)
        .IsUnicode(false);

    entity.Property(e => e.Nombre)
        .HasMaxLength(50)
        .IsUnicode(false);

    entity.Property(e => e.Apellido)
        .HasMaxLength(50)
        .IsUnicode(false);

});

```

Se hace uso de PL/SQL , AutoMapper y FLUENTValidation

```

[HttpPost]
[Route("crear")]
0 referencias
public async Task<ActionResult> PostCompra(Compra compra)
{
    //Se llama al procedimiento almacenado en PLSQL de CrearCompra
    await _context.Database.ExecuteSqlInterpolatedAsync($"EXEC CrearCompra @IdTarjeta={compra.IdTarjeta},@Cuenta={compra.Cuenta}");

    return Ok();
}

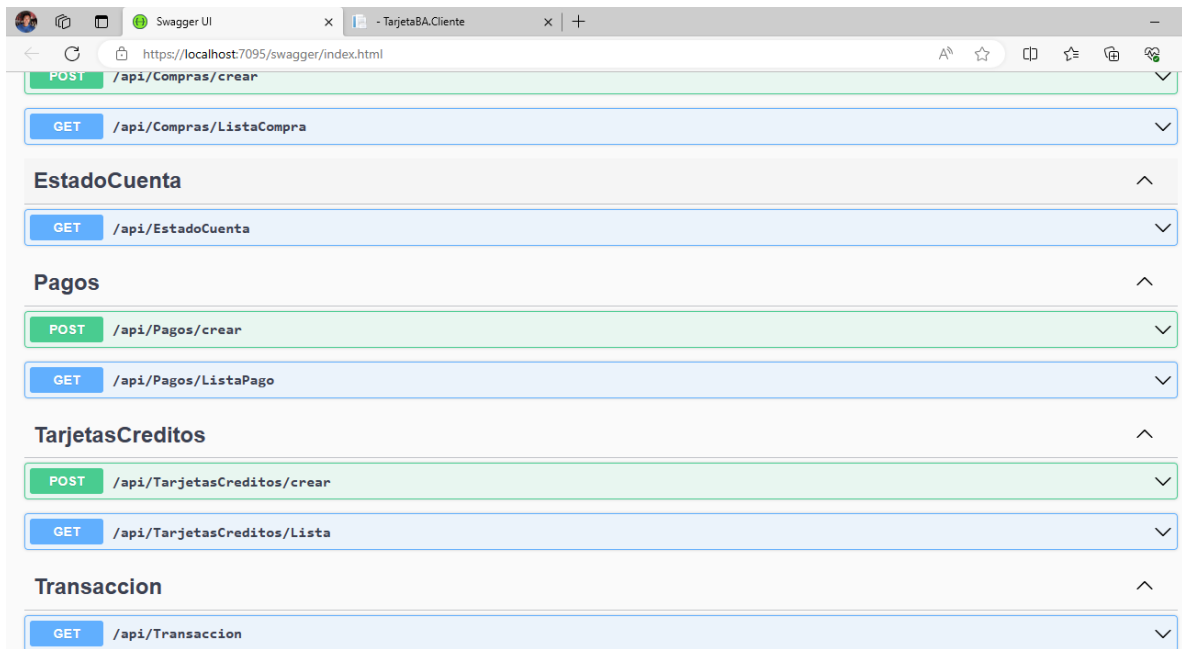
[HttpGet]
[Route("ListaCompra")]
0 referencias
public async Task<ActionResult<IEnumerable<Compra>>> ListaCompra()
{
    //Se llama al procedimiento almacenado en PLSQL de ObtenerListaCompras
    var compras = await _context.Compras.FromSqlRaw("EXEC ObtenerListaCompras").ToListAsync();

    return Ok(compras);
}

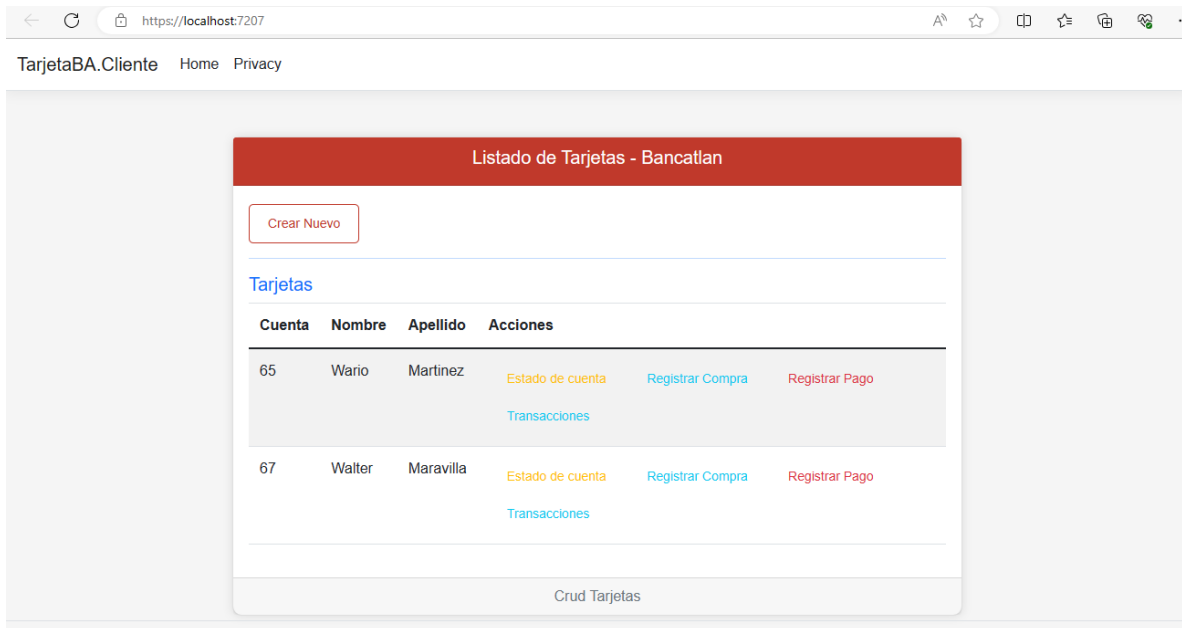
```

En API he usado Modelos y Contolladores en MVC al consumo de Api he usado Modelos,Vistas y Contrllador, es que mi gran dominio de códigos , buenas practicas , unos procedimiento quedaron a implementar como el editar , eliminar y ver por id para cada tarjeta crédito seleccionada , en el mismo proyecto encontrara la documentación de que hace cada método, controlador , modelos etc.

Vista esperado de api



De parte MVC



Erro en Estado Cuenta

An unhandled exception occurred while processing the request.

NullReferenceException: Object reference not set to an instance of an object.

AspNetCoreGeneratedDocument.Views_Shared_Error.ExecuteAsync() in Error.cshtml, line 9

Stack Query Cookies Headers Routing

NullReferenceException: Object reference not set to an instance of an object.

```
AspNetCoreGeneratedDocument.Views_Shared_Error.ExecuteAsync() in Error.cshtml
9. @if (Model.ShowRequestId)
Microsoft.AspNetCore.Mvc.Razor.RazorView.RenderPageCoreAsync(RazorPage page, ViewContext context)
Microsoft.AspNetCore.Mvc.Razor.RazorView.RenderPageAsync(RazorPage page, ViewContext context, bool invokeViewStarts)
Microsoft.AspNetCore.Mvc.Razor.RazorView.RenderAsync(ViewContext context)
Microsoft.AspNetCore.Mvc.ViewFeatures.ViewExecutor.ExecuteAsync(ViewContext viewContext, string contentType, Nullable<int> statusCode)
Microsoft.AspNetCore.Mvc.ViewFeatures.ViewExecutor.ExecuteAsync(ViewContext viewContext, string contentType, Nullable<int> statusCode)
Microsoft.AspNetCore.Mvc.ViewFeatures.ViewExecutor.ExecuteAsync(ActionContext actionContext, IView view, ViewDataDictionary viewData, ITempDataDictionary TempData, string contentType, Nullable<int> statusCode)
Microsoft.AspNetCore.Mvc.ViewFeatures.ViewResultExecutor.ExecuteAsync(ActionContext context, ViewResult result)
Microsoft.AspNetCore.Mvc.ViewResult.ExecuteResultAsync(ActionContext context)
Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.<InvokeNextResultFilterAsync>g__Awaited|30_0<TFilter, TFilterAsync>(ResourceInvoker invoker, Task lastTask, State next, Scope scope, object state, bool isCompleted)
Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.Rethrow(ResultExecutedContextSealed context)
Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.ResultNext<TFilter, TFilterAsync>(ref State next, ref Scope scope, ref object state, ref bool isCompleted)
Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.InvokeResultFilters()
Microsoft.AspNetCore.Mvc.Infrastructure.ResourceInvoker.<InvokeNextResourceFilter>g__Awaited|25_0(ResourceInvoker invoker, Task lastTask, State next, Scope scope, object state, bool isCompleted)
```

Registra Compra

TarjetaBA.Cliente Home Privacy

Registro de compra - Bancatlán

dd/mm/aaaa --:--

Descripcion

Monto

Guardar

Volver a la Lista

© 2024 - TarjetaBA.Cliente - [Privacy](#)

Registrar Pago

Registro de Pago TC - Bancatlán

dd/mm/aaaa --:--

Abono

Guardar

Volver a la Lista

© 2024 - TarjetaBA.Cliente - [Privacy](#)

Obtiene lista de Compra y Pago

Transaccion - Banco Atlandida

Titular

Número de Tarjeta

Wario Martinez

65

Fecha	Descripción	Cargo	Abono
10/05/2024 9:33:25	hhh	76,00	
10/05/2024 9:33:25	tttt	56,00	
10/05/2024 9:33:25	aaa	33,00	
10/05/2024 9:33:25	hhj	56,00	
10/05/2024 9:33:25	hhj	56,00	
10/05/2024 9:33:25	hhj	56,00	
10/05/2024 9:33:25	hhj	56,00	

Asegurarse que este la bibliotecas usada en MVC

```

<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net8.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="AutoMapper" Version="12.0.1" />
    <PackageReference Include="AutoMapper.Extensions.Microsoft.DependencyInjection" Version="12.0.1" />
    <PackageReference Include="EPPlus" Version="7.1.2" />
    <PackageReference Include="FluentValidation.AspNetCore" Version="11.3.0" />
    <PackageReference Include="Microsoft.VisualStudio.Web.CodeGeneration.Design" Version="8.0.2" />
  </ItemGroup>

</Project>

```

Asegurarse que este la bibliotecas en Api

```

<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net8.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="AutoMapper.Extensions.Microsoft.DependencyInjection" Version="12.0.1" />
    <PackageReference Include="EPPlus" Version="7.1.2" />
    <PackageReference Include="FluentValidation.AspNetCore" Version="11.3.0" />
    <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="7.0.11" />
    <PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="7.0.11">
      <PrivateAssets>all</PrivateAssets>
      <IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>
    </PackageReference>
    <PackageReference Include="Swashbuckle.AspNetCore" Version="6.4.0" />
  </ItemGroup>

</Project>

```

Como persona con discapacidad, estoy comprometido a contribuir positivamente en cualquier entorno de trabajo. Tengo la determinación y la capacidad para aprender y mejorar constantemente, y estoy emocionado por la oportunidad de formar parte de proyectos desafiantes y significativos. Agradezco la consideración y la oportunidad de participar en este proceso de selección.

En resumen, aunque algunas funcionalidades de la aplicación no están completas, he aprendido mucho durante este proceso y estoy comprometido a seguir mejorando mis habilidades. Agradezco la oportunidad de participar en este proyecto y estoy dispuesto a continuar aprendiendo y contribuyendo en futuros proyectos.