-El sitio debe contar con una sección de bienvenida con inmuebles destacados.

-Debe contener una sección de “quienes somos”y “nuestra trayectoria”.

-Se necesita una sección de “contacto”para poder ver los datos de contacto del negocio.

-Ver un listado del total de los inmuebles con los filtros necesarios para poder ver especificando: cantidad de ambientes, país, ciudad, metros cuadrados mínimos y máximos, rango de precio, y tipo de contratación(alquilero compra)+ESTADOS+ booleano DESTACADO. CATALOGO

-Se debe poder acceder al detalle de cada inmueble y poder acceder a un link para contactar a la inmobiliaria enviando el código de referencia de ese inmueble en particular. DETALLE

-La inmobiliaria debe contar con un panel privado para poder gestionar el listado de inmuebles para hacer la carga, edición y eliminación de cada uno.- PRODUCTOS PRODUCTO

Los inmuebles pueden estar en diferentes estados como “disponibles”, “reservado”, “alquilado”, o “vendido”.

Tabla de usuarios y tabla de inmuebles

create database Inmobiliaria

go

use Inmobiliaria

create table Inmueble(

IdInmueble int primary key identity,

Titulo varchar(50),

Nombre varchar(50),

TipoInmueble varchar (50),

Detalles varchar(1000),

Pais varchar(50),

Ciudad varchar(50),

Ambientes int,

Metros int,

Antigüedad int,

Precio decimal(10,2),

Expensas decimal (10,2),

PublicadoPara varchar (50),

Estado varchar (50),

Destacado bool,

FechaCreacion datetime default getdate(),

Imagen varchar(max),

)

go

use Inmobiliaria

create table Usuario(

IdUsuario int primary key identity,

NombreCompleto varchar(50),

Correo varchar(50),

Clave varchar(50),

Rol varchar(50),

FechaCreacion datetime default getdate()

)

Create table Imagen

IdImagen int primary key identity,

IdInmueble int references Inmueble (IdInmueble),

EsPrincipal Bit

--insertamos un usuario para poder iniciar sesion

insert into Usuario(NombreCompleto,Correo,Clave,Rol) values

('administrador','admin@example.com'

DDBB Posta

create database Inmobiliaria

go

use Inmobiliaria

create table Inmueble(

IdInmueble int primary key identity,

Titulo varchar(50),

Nombre varchar(50),

TipoInmueble varchar (50),

Detalles varchar(1000),

Pais varchar(50),

Ciudad varchar(50),

Ambientes int,

Metros int,

Antigüedad int,

Precio decimal(10,2),

Expensas decimal (10,2),

PublicadoPara varchar (50),

Estado varchar (50),

Destacado Bit,

FechaCreacion datetime default getdate(),

Imagen varchar(max),)

use Inmobiliaria

create table Usuario(

IdUsuario int primary key identity,

NombreCompleto varchar(50),

Correo varchar(50),

Clave varchar(50),

Rol varchar(50),

FechaCreacion datetime default getdate()

)

Scaffold-DbContext "Server=(local); DataBase= Inmobiliaria; Trusted\_Connection=True; TrustServerCertificate=True;" Microsoft.EntityFrameworkCore.SqlServer

Arquitectura

<https://www.youtube.com/watch?v=UsEx2L-E6-Y&t=2166s>

public partial class InmobiliariaContext : DbContext

{

public InmobiliariaContext()

{

}

public InmobiliariaContext(DbContextOptions<InmobiliariaContext> options)

: base(options)

{

}

public virtual DbSet<Inmueble> Inmuebles { get; set; }

public virtual DbSet<Usuario> Usuarios { get; set; }

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Inmueble>(entity =>

{

entity.HasKey(e => e.IdInmueble).HasName("PK\_\_Inmueble\_\_6B8E7D3E16F8F6E1");

entity.ToTable("Inmueble");

entity.Property(e => e.Ciudad)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Detalles)

.HasMaxLength(1000)

.IsUnicode(false);

entity.Property(e => e.Estado)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Expensas).HasColumnType("decimal(10, 2)");

entity.Property(e => e.FechaCreacion)

.HasDefaultValueSql("(getdate())")

.HasColumnType("datetime");

entity.Property(e => e.Imagen).IsUnicode(false);

entity.Property(e => e.Pais)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Precio).HasColumnType("decimal(10, 2)");

entity.Property(e => e.PublicadoPara)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.TipoInmueble)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Titulo)

.HasMaxLength(50)

.IsUnicode(false);

});

modelBuilder.Entity<Usuario>(entity =>

{

entity.HasKey(e => e.IdUsuario).HasName("PK\_\_Usuario\_\_5B65BF97C63A90CC");

entity.ToTable("Usuario");

entity.Property(e => e.Clave)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Correo)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.FechaCreacion)

.HasDefaultValueSql("(getdate())")

.HasColumnType("datetime");

entity.Property(e => e.NombreCompleto)

.HasMaxLength(50)

.IsUnicode(false);

entity.Property(e => e.Rol)

.HasMaxLength(50)

.IsUnicode(false);

});

OnModelCreatingPartial(modelBuilder);

}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);



Agregar tipo de moneda

Quitar decimales.

Agregar dirección.