



Instituto Tecnológico Colonia CTC

Analista programador

Obligatorio solitario, Programación 2

Profesor: Carlos Rodriguez

Julián Silvera

2024

Índice

Abstract.....	2
Cronograma.....	3
Diccionario de clases.....	4
Habitacion.cs.....	4
Huesped.cs.....	5
Reserva.cs.....	6
Pago.cs.....	7
Estadisticas.cshtml.cs.....	8
Habitaciones.cshtml.cs.....	9
Login.cshtml.cs.....	10
MenuUsuario.cshtml.cs.....	11
MisReservas.cshtml.cs.....	12
RealizarReserva.cshtml.cs.....	13
RecuperarContraseña.cshtml.cs.....	14
Registrar.cshtml.cs.....	15
Diagrama de la capa de dominio.....	16
Resultados de las pruebas realizadas:.....	17
Notas Adicionales:.....	20

Abstract

El proyecto "Hotel Estrella del Mar" refleja un avance significativo en mi desarrollo como programador, al pasar de un entorno de consola a un entorno web más complejo, integrando bases de datos para mejorar la funcionalidad del sistema de gestión hotelera. Este cambio ha supuesto un desafío enriquecedor, que me ha permitido expandir mis habilidades en programación y diseño de sistemas web.

El objetivo del sistema es ofrecer una experiencia eficiente y moderna para los usuarios, permitiéndoles registrar o iniciar sesión, registrarse, buscar habitaciones disponibles, realizar reservas, editar reservas, función por si se olvida su contraseña, etc, todo a través de una interfaz web accesible e intuitiva.

La implementación del proyecto ha implicado resolver retos técnicos como la integración con bases de datos, la gestión de sesiones de usuario y la optimización de las operaciones en un entorno web. Cada desafío superado ha fortalecido el sistema, incrementando tanto su funcionalidad como su capacidad de respuesta ante diferentes escenarios.

En resumen, "Hotel Estrella del Mar" es el resultado de un esfuerzo constante y una evolución técnica, que busca mejorar la gestión hotelera mediante un sistema robusto, accesible y diseñado pensando en la experiencia del usuario. Este proyecto representa un paso adelante en mi camino profesional hacia el desarrollo de soluciones tecnológicas más completas y eficientes.

Cronograma

Tarea	Tiempo estimado
Análisis	5 hs
C#	70 hs
Documentación	5 hs
Total	80 horas

Diccionario de clases

Nombre de la Clase: Habitacion.cs

Descripción de la Clase:

La clase Habitacion representa una habitación en el hotel. Contiene atributos como el número de la habitación, su tipo (simple, doble, suite), la capacidad de personas que puede alojar, el precio por noche y su disponibilidad. Los métodos asociados permiten gestionar las características de la habitación, como verificar su disponibilidad o modificar sus detalles.

Atributo	Tipo de Dato	Descripción
Numero	int	Número único de la habitación.
Tipo	string	Tipo de habitación (ej. Simple, Doble, Suite).
Capacidad	int	Capacidad máxima de personas en la habitación.
PrecioDiario	decimal	Precio por noche de la habitación.
EstaDisponible	bool	Indica si la habitación está disponible para reserva.

Tabla de Métodos de la Clase Habitacion.cs:

Nombre	Devuelve	Recibe (Parámetros)	Descripción
Habitacion()	void	Ninguno	Constructor vacío para la inicialización.
Habitacion(int numero, string tipo, int capacidad, decimal precioDiario)	void	int numero, string tipo, int capacidad, decimal precioDiario	Constructor que permite crear una habitación.

Nombre de la Clase: Huesped.cs

Descripción de la Clase:

La clase Huesped representa a un huésped registrado en el sistema del hotel. Contiene atributos como el nombre, apellido, correo electrónico y la contraseña de acceso. Los métodos de esta clase permiten gestionar el registro, la autenticación y la recuperación de la contraseña de los huéspedes.

Tabla de Atributos de la Clase Habitación:

Atributo	Tipo de Dato	Descripción
Id	int	ID único del huésped.
Nombre	string	Nombre del huésped.
Apellido	string	Apellido del huésped.
Email	string	Correo electrónico del huésped.
Contraseña	string	Contraseña del huésped para iniciar sesión.

Tabla de Métodos de la Clase Huesped.cs:

Nombre	Devuelve	Recibe (Parámetros)	Descripción
Huesped()	void	Ninguno	Constructor vacío.
Huesped(int id, string nombre, string apellido, string email, string contraseña)	void	int id, string nombre, string apellido, string email, string contraseña	Constructor para crear un nuevo huésped.

Nombre de la Clase: Reserva.cs**Descripción de la Clase:**

La clase Reserva representa una reserva realizada por un huésped en una habitación del hotel. Incluye atributos como el número de la reserva, la habitación reservada, el huésped que realizó la reserva, y las fechas de inicio y fin de la misma. Los métodos permiten gestionar la creación, modificación y cancelación de reservas en el sistema.

Tabla de Atributos de la Clase Reserva.cs:

Atributo	Tipo de Dato	Descripción
NumeroReserva	int	Número único de la reserva.
NumeroHabitacion	int?	Número de la habitación asociada con la reserva.
Habitacion	Habitacion?	Referencia a la habitación asociada con la reserva.
HuespedId	int	ID del huésped asociado con la reserva.
Huesped	Huesped	Referencia al huésped que realiza la reserva.
FechaInicio	DateTime	Fecha de inicio de la reserva.
FechaFin	DateTime	Fecha de fin de la reserva.

Tabla de Métodos de la Clase Reserva.cs:

Nombre	Devuelve	Recibe (Parámetros)	Descripción
Reserva()	void	Ninguno	Constructor vacío para inicializar una reserva.
Reserva(int numeroReserva, int? numeroHabitacion, Huesped huesped, DateTime fechaInicio, DateTime fechaFin)	void	int numeroReserva, int? numeroHabitacion, Huesped huesped, DateTime fechaInicio, DateTime fechaFin	Constructor que permite crear una reserva.

Nombre de la Clase: Pago.cs**Descripción de la Clase:**

La clase Pago representa un pago realizado por un huésped relacionado con una reserva. Contiene atributos como el número de la reserva, el monto pagado y la fecha en que se realizó el pago. Los métodos asociados permiten registrar un pago, generar comprobantes y gestionar los pagos asociados a las reservas en el hotel.

Tabla de Atributos de la Clase Pago.cs:

Atributo	Tipo de Dato	Descripción
Id	int	ID único del pago.
NumeroReserva	int	Número de la reserva asociada al pago.
Reserva	Reserva	Reserva asociada con el pago.
Monto	decimal	Monto del pago realizado.
FechaPago	DateTime	Fecha en la que se realizó el pago.

Tabla de Métodos de la Clase Pago.cs:

Nombre	Devuelve	Recibe (Parámetros)	Descripción
Pago()	void	Ninguno	Constructor vacío para inicializar un pago.
Pago(int id, int numeroReserva, decimal monto, DateTime fechaPago)	void	int id, int numeroReserva, decimal monto, DateTime fechaPago	Constructor para crear un pago asociado a una reserva.

Nombre de la Clase: Estadisticas.cshtml.cs

Descripción de la Clase:

La clase Estadisticas.cshtml.cs es responsable de gestionar las estadísticas del sistema de reservas, en particular, la habitación con más reservas y el total de reservas realizadas. Esta clase consulta la base de datos para agrupar las reservas por habitación y calcular las estadísticas correspondientes, brindando así información relevante sobre el uso de las habitaciones del hotel.

Tabla de Atributos de la Estadisticas.cshtml.cs:

Atributo	Tipo de Dato	Descripción
HabitacionConMasReservas	string?	Almacena el nombre de la habitación con más reservas.
TotalReservas	int	Almacena el número total de reservas realizadas.

Tabla de Métodos de la Clase Estadisticas.cshtml.cs:

Nombre	Devuelve	Recibe (Parámetros)	Descripción
OnGet()	void	Ninguno	Calcula la habitación con más reservas y el total de reservas.

Nombre de la Clase: Habitaciones.cshtml.cs

Descripción de la Clase:

La clase Habitaciones.cshtml.cs se encarga de mostrar las habitaciones disponibles en el sistema y permitir a los usuarios hacer reservas. Además, gestiona la disponibilidad de las habitaciones y la creación de nuevas reservas, redirigiendo a los usuarios a la página correspondiente para completar el proceso.

Tabla de Atributos de la Habitaciones.cshtml.cs:

Atributo	Tipo de Dato	Descripción
HabitacionesDisponibles	List<Habitacion>	Lista de las habitaciones disponibles en el hotel.

Tabla de Métodos de la Clase Habitaciones.cshtml.cs:

Nombre	Devuelve	Recibe (Parámetros)	Descripción
OnGet()	void	Ninguno	Obtiene las habitaciones disponibles.
OnPostReservar()	ActionResult	int numeroHabitacion	Realiza una reserva de habitación si está disponible.

Nombre de la Clase: Login.cshtml.cs

Descripción de la Clase:

La misma maneja el inicio de sesión del usuario. Permite que los usuarios registrados ingresen al sistema utilizando su correo electrónico y contraseña, y si las credenciales son correctas, establece una sesión y redirige a la página de habitaciones.

Tabla de Atributos de la Clase Login.cshtml.cs:

Atributo	Tipo de Dato	Descripción
Email	string	Almacena el correo electrónico del usuario para iniciar sesión.
Password	string	Almacena la contraseña del usuario para iniciar sesión.

Tabla de Métodos de la Clase Login.cshtml.cs:

Nombre	Devuelve	Recibe (Parámetros)	Descripción
OnGet	ActionResult	Ninguno	Muestra la página de inicio de sesión.
OnPost	ActionResult	Ninguno	Valida las credenciales y establece la sesión para el usuario.

Nombre de la Clase: MenuUsuario.cshtml.cs

Descripción de la Clase:

La clase muestra el menú de usuario con su información y las opciones disponibles, como cerrar sesión. Se asegura de que solo los usuarios logueados puedan acceder a esta página, verificando la sesión activa.

Tabla de Atributos de la Clase MenuUsuario.cshtml.cs:

Atributo	Tipo de Dato	Descripción
Usuario	Huesped?	Almacena los datos del huésped logueado.

Tabla de Métodos de la Clase MenuUsuario.cshtml.cs:

Nombre	Devuelve	Recibe (Parámetros)	Descripción
OnGet	void	Ninguno	Muestra la información del usuario logueado y el menú correspondiente.

Nombre de la Clase: MisReservas.cshtml.cs

Descripción de la Clase:

Gestiona las reservas realizadas por el usuario. Permite al usuario ver sus reservas actuales y modificar o cancelar las existentes.

Tabla de Atributos de la MisReservas.cshtml.cs:

Atributo	Tipo de Dato	Descripción
Reservas	List<Reserva>	Lista de las reservas realizadas por el usuario.

Tabla de Métodos de la Clase MisReservas.cshtml.cs:

Nombre	Devuelve	Recibe (Parámetros)	Descripción
OnGet	void	Ninguno	Muestra las reservas realizadas por el usuario.
OnPostEditar	ActionResult	int id, DateTime nuevaFechaInicio, DateTime nuevaFechaFin	Permite editar una reserva existente.
OnPostCancelar	ActionResult	int id	Permite cancelar una reserva existente.

Nombre de la Clase: RealizarReserva.cshtml.cs

Descripción de la Clase:

Esta permite al usuario realizar una nueva reserva de habitación. Verifica la disponibilidad de la habitación y realiza la reserva si la habitación está disponible.

Tabla de Atributos de la RealizarReserva.cshtml.cs:

Atributo	Tipo de Dato	Descripción
HabitacionId	int	Almacena el número de la habitación que el usuario desea reservar.
ErrorMessage	string	Mensaje de error si la habitación no está disponible o el ID es incorrecto.
FechaInicio	DateTime	Fecha de inicio de la reserva.
FechaFin	DateTime	Fecha de fin de la reserva.

Tabla de Métodos de la Clase RealizarReserva.cshtml.cs:

Nombre	Devuelve	Recibe (Parámetros)	Descripción
OnGet	ActionResult	int habitacionId	Muestra el formulario para realizar la reserva.
OnGet	ActionResult	Ninguno	Procesa la solicitud de reserva y la realiza si la habitación está disponible.

Nombre de la Clase: RecuperarContraseña.cshtml.cs

Descripción de la Clase:

La clase permite a los usuarios recuperar su contraseña mediante su correo electrónico registrado. Si el correo es válido, se muestra la contraseña asociada.

Tabla de Atributos de la Clase RecuperarContraseña.cshtml.cs:

Atributo	Tipo de Dato	Descripción
Email	string	Almacena el correo electrónico del usuario para recuperar la contraseña.
Mensaje	string	Almacena el mensaje que se muestra al usuario (contraseña o error).

Tabla de Métodos de la Clase RecuperarContraseña.cshtml.cs:

Nombre	Devuelve	Recibe (Parámetros)	Descripción
OnGet	void	Ninguno	Muestra el formulario de recuperación de contraseña.
OnGet	ActionResult	Ninguno	Recupera la contraseña asociada al correo electrónico del usuario.

Nombre de la Clase: Registrar.cshtml.cs

Descripción de la Clase:

Permite a un nuevo usuario registrarse en el sistema proporcionando su nombre, apellido, correo electrónico y contraseña.

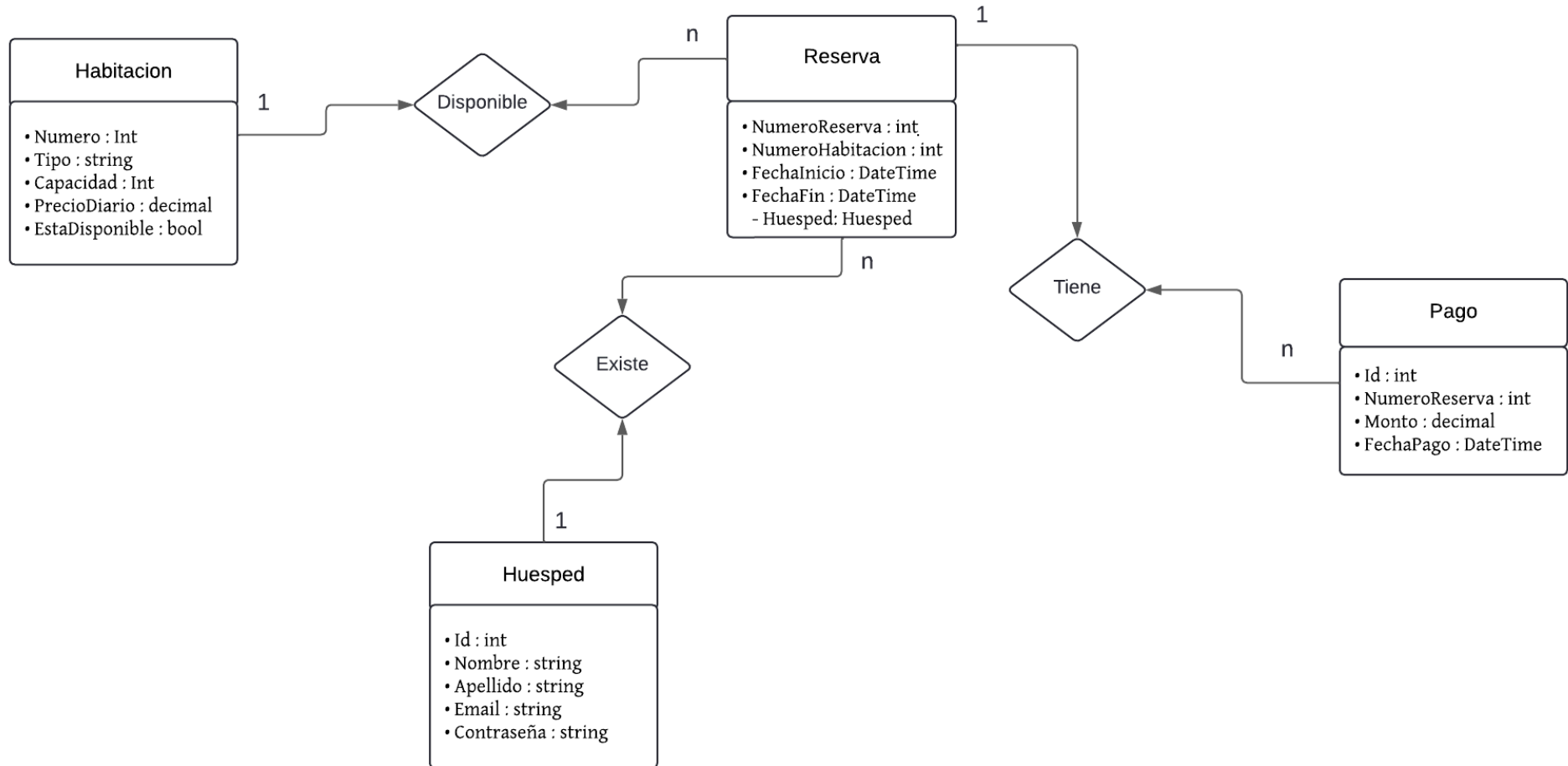
Tabla de Atributos de la Clase Registrar.cshtml.cs:

Atributo	Tipo de Dato	Descripción
Nombre	string	Nombre del usuario a registrar.
Apellido	string	Apellido del usuario a registrar.
Email	string	Correo electrónico del usuario a registrar.
Contraseña	string	Contraseña del usuario a registrar.
Mensaje	string?	Mensaje que se muestra si ya existe el usuario con ese correo.

Tabla de Métodos de la Clase Registrar.cshtml.cs:

Nombre	Devuelve	Recibe (Parámetros)	Descripción
OnPostRegister	IActionResult	Ninguno	Registra un nuevo usuario en el sistema.

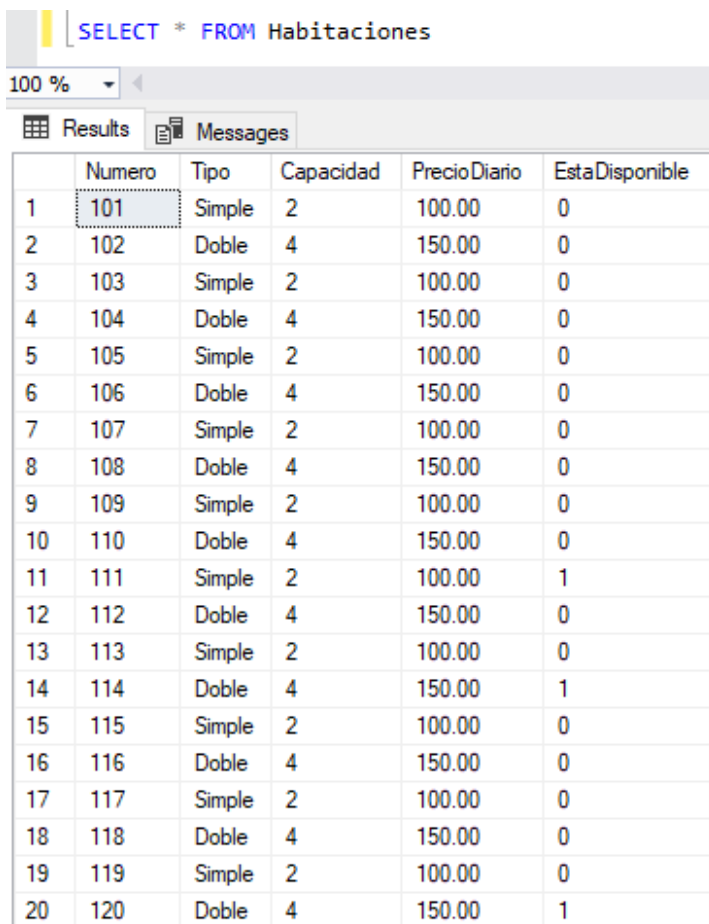
Diagrama de la capa de dominio



Resultados de las pruebas realizadas:

Durante el proceso de desarrollo del sistema, se realizaron una serie de pruebas para asegurar que todas las funcionalidades implementadas estuvieran operando correctamente. Estas pruebas abarcan diversos aspectos del sistema, incluyendo la gestión de reservas, la autenticación de usuarios, la disponibilidad de habitaciones y la correcta ejecución de las funcionalidades clave.

Pruebas en la base de datos:



The screenshot shows a SQL query window with the command `SELECT * FROM Habitaciones`. Below the query, the 'Results' tab is active, displaying a table with 6 columns: 'Numero', 'Tipo', 'Capacidad', 'PrecioDiario', and 'EstaDisponible'. The table contains 20 rows of data, with the first row (101) highlighted. The 'EstaDisponible' column has values 0 for most rows and 1 for rows 111, 114, and 120.

	Numero	Tipo	Capacidad	PrecioDiario	EstaDisponible
1	101	Simple	2	100.00	0
2	102	Doble	4	150.00	0
3	103	Simple	2	100.00	0
4	104	Doble	4	150.00	0
5	105	Simple	2	100.00	0
6	106	Doble	4	150.00	0
7	107	Simple	2	100.00	0
8	108	Doble	4	150.00	0
9	109	Simple	2	100.00	0
10	110	Doble	4	150.00	0
11	111	Simple	2	100.00	1
12	112	Doble	4	150.00	0
13	113	Simple	2	100.00	0
14	114	Doble	4	150.00	1
15	115	Simple	2	100.00	0
16	116	Doble	4	150.00	0
17	117	Simple	2	100.00	0
18	118	Doble	4	150.00	0
19	119	Simple	2	100.00	0
20	120	Doble	4	150.00	1

El enlace entre Visual Studio y la base de datos SQL se confirma con los datos de habitaciones en la consulta. Aunque al principio fue complicado, debido a ser algo nuevo, finalmente se logró subir las habitaciones correctamente a SQL, demostrando que el sistema está funcionando bien.

Pruebas en la página de inicio

Bienvenido al Sistema de Reservas

Por favor, inicia sesión o regístrate para continuar.

Iniciar Sesión

Registrarse

En esta prueba de la página de inicio, se optó por un diseño más minimalista, con botones claros para iniciar sesión o registrarse. Si bien existían varias formas de implementarlo, elegí esta opción porque no era el aspecto principal del sistema. El objetivo principal era permitir a los usuarios realizar reservas de habitaciones, lo cual se priorizó.

Pruebas de inicio de sesión

Iniciar Sesión

Email

Contraseña

Iniciar Sesión

En este caso, la implementación del inicio de sesión fue sencilla, cumpliendo con el propósito de permitir a los usuarios acceder al sistema. Se incluyó un botón para iniciar sesión, así como otro para la recuperación de la contraseña, ubicado en la parte superior derecha. Además, se pre cargaron algunos usuarios para facilitar las pruebas y asegurar que la funcionalidad fuera operativa.

Pruebas sobre inicio de sesión erróneo:

Iniciar Sesión

Credenciales incorrectas. Por favor, intenta de nuevo.

Email

ujujyuj@gmail.com

Contraseña

.....

Iniciar Sesión

En este caso, se verificó la funcionalidad de inicio de sesión con un correo y contraseña incorrectos. Al ingresar datos no válidos, el sistema muestra un mensaje de error en rojo informando que las "Credenciales son incorrectas, por favor, intenta de nuevo", lo cual proporciona al usuario retroalimentación inmediata sobre el error.

Pruebas en recuperación de contraseña:

Recuperar Contraseña

Correo electrónico

Ingresa tu correo

Recuperar Contraseña

Se probaron varias maneras de implementar la recuperación de contraseña, pero al final, lo que facilitó más el proceso fue permitir que el usuario ingrese su correo y recibir automáticamente la contraseña previamente configurada. Esto simplificó la experiencia y ayudó a que fuera más fácil y rápido para el usuario recuperar su acceso al sistema.

Prueba en mis reservas:

Mis Reservas

ID Reserva	Habitación	Tipo	Fecha Inicio	Fecha Fin	Acciones
27	104	Doble	7/12/2024	8/12/2024	<div>01/12/2024 </div> <div> El valor debe ser igual o posterior a 07/12/2024.</div>

Se realizó un testeo para garantizar que al modificar una reserva, el sistema no permite seleccionar una fecha de inicio anterior a la fecha actual. Al intentar seleccionar una fecha anterior al día de la prueba (07/12/2024), el sistema muestra un mensaje de advertencia indicando que la fecha debe ser igual o posterior a la fecha actual. Esto asegura que las reservas siempre se realicen a partir de la fecha actual o en fechas futuras, lo que mejora la lógica de reservas y evita errores al registrar reservas con fechas pasadas.

Conclusión de las Pruebas:

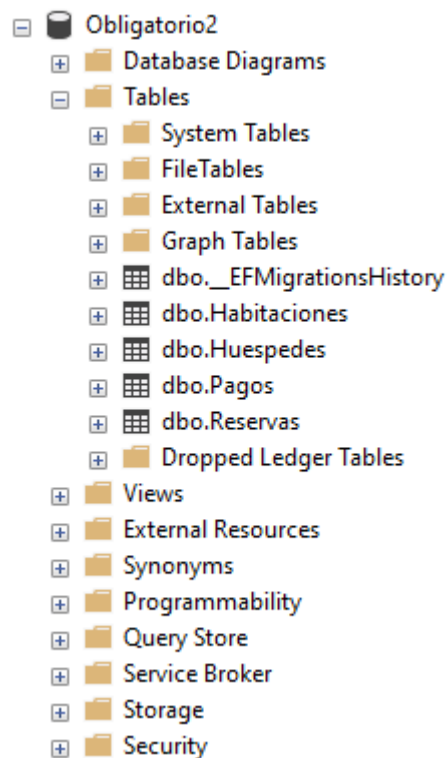
Los resultados obtenidos de las pruebas confirmaron que el sistema funciona de acuerdo a lo esperado. Todas las funcionalidades principales operan de manera efectiva, ofreciendo una experiencia de usuario óptima y fluida. No se encontraron fallos críticos, lo que asegura que el sistema está listo para su implementación en un entorno de producción.

Notas Adicionales:

- **Validación de Fechas al Modificar una Reserva:** Es importante que al modificar la fecha de inicio de una reserva, el sistema asegure que la fecha seleccionada sea igual o posterior a la fecha actual. Si se intenta elegir una fecha pasada, el sistema mostrará un mensaje de advertencia y no permitirá la modificación, garantizando que solo se puedan realizar cambios válidos.
- **Mensaje de Error en Inicio de Sesión:** En caso de que las credenciales de usuario (email y contraseña) sean incorrectas, el sistema ahora muestra un mensaje de error claro, indicando que las credenciales no son válidas, lo que mejora la experiencia del usuario al no dejar la pantalla en blanco sin información.
- **Configurar de la base de datos:**

Asegúrate de configurar o crear la base de datos con el nombre Obligatorio2. Una vez creada, actualiza el Connection String correspondiente.

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=JULI;Database=Obligatorio2;Trusted_Connection=True;MultipleActiveResultSets=True;"
  },
}
```



Después de crear la base de datos y actualizar el Connection String, ejecuta el comando update-database.