Proyecto gestión de reservas, hotel Estrella del Mar.

Paula Camacho

Programacion Dos

2024

Contenido

[2 Declaración de autoría. 2](#_Toc181824237)

[3 Introducción: 3](#_Toc181824238)

[4 Abstract: 4](#_Toc181824239)

[5 Diccionario de clases 5](#_Toc181824240)

[6 Funciones que destaco de la Clase Program 11](#_Toc181824241)

[7 Metodos de la Clase Hotel a destacar 13](#_Toc181824242)

[8 Herencia: 15](#_Toc181824243)

[9 Diagrama de Capas UML: 17](#_Toc181824244)

[10 Cronograma de trabajo: 18](#_Toc181824245)

# Declaración de autoría.

Yo, Paula Camacho, declaro que el trabajo que se presenta es de mi propia mano. Puedo asegurar que la obra fue producida mientras cursaba la materia Programación Dos con el profesor Carlos Rodríguez. He tenido en cuenta las clases dictadas por el Prof. Carlos Rodríguez, quien además proporcionó el material. Cuando he consultado el trabajo publicado por otros, lo he atribuido con claridad, y cuando he citado obras de otros, he indicado las fuentes. Con excepción de estas citas, la obra es enteramente mía. En la obra, he acusado recibo de las ayudas recibidas y cuando la obra se basa en trabajo realizado conjuntamente con otros, he explicado claramente qué fue construido por otros, y qué fue contribuido por mí. Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Firma:

Paula Camacho

# Introducción:

El presente documento describe el desarrollo del Sistema de Gestión de Reservas del Hotel Estrella del Mar, una aplicación de consola creada en C# (.NET 6.0) que tiene como objetivo la gestión eficiente y automatizada de las reservas, consultas de habitaciones y pagos dentro del hotel. El sistema está diseñado para permitir tanto a los usuarios como al personal administrativo llevar a cabo operaciones clave, como la consulta de información sobre el hotel, la realización de reservas, la modificación o cancelación de las mismas, y la gestión de pagos relacionados con las estadías.

El Hotel Estrella del Mar se encuentra en una ciudad costera turística, ofreciendo una experiencia única para sus huéspedes mediante instalaciones modernas y una variedad de opciones de habitaciones adaptadas a diversas necesidades. Con este sistema, se busca mejorar la eficiencia operativa del hotel, optimizando la gestión de reservas y la experiencia del cliente, asegurando que cada huésped tenga un acceso rápido y sencillo a los servicios que se le ofrecen, además de proporcionar al personal del hotel herramientas efectivas para la administración de las operaciones diarias.

A lo largo de este documento se detalla la estructura del proyecto, sus componentes principales, y los procedimientos de implementación de cada funcionalidad clave, como la creación y gestión de reservas, la modificación de datos de los huéspedes, y la generación de reportes de las actividades realizadas en el hotel.

# Abstract:

El Hotel Estrella del Mar enfrenta el desafío de gestionar eficientemente las reservas, las habitaciones disponibles y los pagos de sus huéspedes de manera manual, lo que puede resultar en procesos lentos, errores humanos y dificultad para acceder rápidamente a la información crucial. Actualmente, el sistema de gestión no permite realizar reservas de forma automatizada, lo que genera confusión y posibles conflictos de disponibilidad de habitaciones, como reservas duplicadas o en fechas no disponibles. Además, el personal del hotel debe llevar un control manual de las reservas, lo que aumenta el riesgo de errores y dificulta la gestión de la ocupación del hotel.

El sistema propuesto busca automatizar y mejorar estos procesos, permitiendo a los usuarios realizar reservas de forma rápida y sencilla a través de una interfaz de consola, gestionando automáticamente la disponibilidad de las habitaciones, evitando duplicados y asegurando que se respeten las fechas de reserva. Además, el sistema permitirá al personal administrativo realizar modificaciones, cancelaciones y consultas de reservas de manera eficiente y sin riesgo de error.

Tecnología Utilizada:

La solución está desarrollada utilizando C# como lenguaje de programación, en la IDE Visual Studio, sobre la plataforma .NET 6.0, aprovechando la facilidad de mantenimiento y escalabilidad. La aplicación es de tipo consola, lo que permite su implementación rápida y fácil, además de ser una opción ligera para su uso en el entorno de un hotel pequeño o mediano.

El sistema utiliza una arquitectura orientada a objetos (POO), lo que facilita la organización del código, la reutilización de componentes y el mantenimiento a largo plazo. Además, se implementa el uso de colecciones de datos como listas, lo que permite gestionar de forma eficiente las reservas y las habitaciones del hotel.

En la aplicación de consola, he creado un menú que presenta al usuario una lista de opciones para interactuar con el sistema. A través de este menú, se pueden visualizar los métodos y la información que el usuario puede agregar, tales como gestionar reservas, consultar disponibilidad de habitaciones y realizar pagos.

Para simplificar las pruebas, he incluido un usuario de prueba predefinido con las siguientes credenciales:

- Email: Juancho@gmail.com

- Contraseña: UwU

Este usuario harcodeado permite acceder al sistema y probar las funcionalidades sin necesidad de registrarse, facilitando el proceso de desarrollo y demostración del sistema.

# Diccionario de clases

Tabla de atributos de la Clase Usuario:

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo de dato | Descripción |
| Email | string | Email del usuario |
| Contrasenia | object | Contraseña del usuario |
| Usuario | Usuario | Lista de usuario |
|  |  |  |

Tabla de métodos de la Clase Usuario:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Devuelve | Recibe(Parametros) | Descripción |
| ToString | Nada | Niguno | Devuelve la instancia en string |

Tabla atributos Clase Pago

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo de dato | Descripcion |
| idPago | Int | Id del pago |
| fechaPago | DateTime | Fecha del pago |
| Monto | decimal | Cantidad a pagar |
| tipoPago | string | Qué clase de pgo |

Tabla Clase PagoConTarjeta

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo de dato | Descripcion |
| NumeroTarjeta | int | Número de la tarjeta con la que se paga |
| TipoTarjeta | string | Tipo de tarjeta, si es Credito o Debito |

Tabla atributos Pago Efectivo

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo de dato | Descripcion |
| CambioEntregado | decimal | Vuelto del pago Efectivo |

Tabla Clase Huesped

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo de dato | Descripcion |
| idHuesped | Int | Id del huésped |
| nombreHuesped | string | Nombre del huésped |
| apellidoHuesped | string | Apellido del huésped |
| tipoDocHuesped | string | Tipo documento del huésped |
| documentoNHues | double | Numero de documento |
| fechaNacimientoHues | DateTime | Fecha de nacimiento del huésped |
| numeroCelHues | int | Celular del huésped |
| emailHues | sting | Mail del huésped |
| paisHues | strin | País de ubicación del huésped |
| Usuario | List<Usuario> | Lista del tipo Usuario, refiriendo al Ususario |

Tabla métodos Huesped

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Devuelve | Recibe(Parametros) | Descripción |

|  |  |  |  |
| --- | --- | --- | --- |
| ToString | String | Nada | Devuelve una string del objeto seleccionado |

Tabla Clase Habitacion

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo de dato | Descripcion |
| idNumeroHabitacion | Int | Id de la habitacion |
| tipoHabitacion | string | Tipo dehabitacion |
| cantidadHuesped | Int | Cantidad de huespedes |
| precioDiario | Int | precio diario de la habitacion |
| disponible | bool | Si esta disponible o no, true or false |

Tabla Metodos Habitacion

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Devuelve | Recibe(Parametros) | Descripción |
| ToString | string | No | Devuelve una lista de habitaciones en string |

Tabla Clase Hotel

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo de dato | Descripcion |
| hotelId | Int | Id del hotel |
| hotelName | string | Nombre del hotel |
| ubicHotel | string | Ubicación del hotel |
| categoria | int | Categoria del hotel |
| servicio | string | Servicio del hotel |
| habitaciones | List<Habitacion> | Lista del tipo Habitacion(clase) |
| personas | List<Huesped> | Lista del tipo Huesped(clase) |
| reservas | List<Reserva> | Lista del tipo Reserva(clase) |

Tabla Metodos Hotel

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Devuelve | Recibe(Parametros) | Descripción |
| AgregarHabitacion | void | Habitacion habitacion | Agrega una instancia habitacion a la lista habitacion |
| ListarHabitaciones | void | No | Recorre cada habitación de la lista habistaciones |
| ListarReseras | void | Int id | Busca las reservas especificadas, en una lista de reservas y muestra el id en consola |
| AgregarHuesped | void | No |  |
| ListarHuesped | void | No | Agrega huésped a una lista |
| ListarHabitacionesDisponibles | List | No | si está disponible la habitación la lista |
| ToString | string | No | Da la string del objeto |

Tabla Metodo GestionDeHuesped

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Devuelve | Recibe(Parametros) | Descripción |
| CargarHuesped | Lista | No | Lista de Huespedes precargados |

Tabla Metodos GestionDeUsusario

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Devuelve | Recibe(Parametros) | Descripción |
| CargarUsuario | Lista | No | Tiene una lista de tres usuarios precargados, en la cual se usan dos |

Tabla Metodos GestionDeHabitaciones

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Devuelve | Recibe(Parametros) | Descripción |
| CargarHabitaciones | Lista habitaciones precargadas | No | Lista habitaciones precargadas con un loop foreach y ternarios |

Tabla Metodos PrecargaReserva

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Devuelve | Recibe(Parametros) | Descripción |
| PrecargaReserva | reservas | No | Es una lista de Reservas precargadas |

# Funciones que destaco de la Clase Program

PrecargarUsuario()

- Descripción: Carga y selecciona un usuario específico de una lista de usuarios precargados.

- Función en el sistema: Utilizada para simular un usuario ya registrado que inicia sesión en el sistema.

PrecargarReserva()

-Descripción: Utiliza una lista de usuarios y habitaciones precargadas para crear y agregar reservas iniciales al hotel.

- Función en el sistema: Facilita la carga de reservas predeterminadas al sistema para realizar pruebas o inicializar datos sin la necesidad de ingresarlos manualmente cada vez.

IngresarNuevaReserva()

- Descripción: Solicita al usuario seleccionar una habitación y las fechas de entrada y salida para crear una nueva reserva. La función verifica que la duración no exceda 30 días y llama a métodos para listar habitaciones disponibles y agregar la reserva.

- Función en el sistema: Permite al usuario hacer reservas personalizadas, manejando tanto la selección de habitaciones como la validación de fechas, asegurando que la duración de las reservas sea adecuada.

ListarHabitacionesDisponibles()

- Descripción: Muestra todas las habitaciones disponibles en el hotel al usuario.

- Función en el sistema: Proporciona una lista de opciones de alojamiento, permitiendo que los usuarios seleccionen una habitación específica al reservar.

. EliminarReserva()

- Descripción: Permite al usuario eliminar una reserva existente, siempre y cuando la fecha de inicio de la reserva sea posterior a la fecha actual.

- Función en el sistema: Ofrece una funcionalidad de gestión de reservas al permitir cancelarlas si aún no han comenzado, mejorando la flexibilidad y control del cliente sobre su reserva.

AltaHabitacion()

- Descripción: Solicita datos para una nueva habitación y la agrega al inventario del hotel.

- Función en el sistema: Facilita la expansión de habitaciones disponibles, permitiendo al administrador del hotel agregar habitaciones según sea necesario.

AltaHuesped()

- Descripción: Solicita y registra la información de un nuevo huésped en el sistema.

.

BuscarHabitacionPorId(int id)

-Descripción: Busca una habitación específica por su ID en la lista de habitaciones del hotel y la retorna.

- Función en el sistema: Ayuda a ubicar y seleccionar una habitación específica, lo cual es esencial para procesar reservas y verificar la disponibilidad de habitaciones exactas.

IdReserva()

- Descripción: Genera un nuevo ID para cada reserva, incrementándolo en base a la cantidad actual de reservas.

- Función en el sistema: Asegura que cada reserva tenga un identificador único, evitando conflictos y manteniendo el sistema de reservas organizado.

# Metodos de la Clase Hotel a destacar

AgregarHabitacion(Habitacion habitacion)

Descripción: Añade una nueva habitación a la lista de habitaciones del hotel.

.

ListarHabitaciones()

Descripción: Muestra la información de todas las habitaciones en el hotel.

Función en el sistema: Ofrece una vista general de las habitaciones, útil tanto para los administradores como para los clientes que consultan opciones.

ListarHabitacionesDisponibles()

Descripción: Devuelve una lista de habitaciones que están actualmente disponibles para reservar.

Función en el sistema: Permite identificar fácilmente las habitaciones que pueden ser asignadas a nuevas reservas, mejorando la eficiencia en la gestión de disponibilidad.

AgregarHuesped(Huesped huesped)

Descripción: Añade un nuevo huésped a la lista de personas en el sistema del hotel.

ListarHuesped()

Descripción: Muestra la información de todos los huéspedes registrados en el hotel.

Función en el sistema: Ofrece una lista completa de los clientes actuales y pasados, útil para consultas o reportes de la gestión hotelera.

AgregarReserva(Reserva reserva)

Descripción: Agrega una nueva reserva a la lista de reservas del hotel.

Función en el sistema: Permite registrar reservas de clientes, facilitando la gestión del historial de reservas y la asignación de habitaciones.

ListarReservas(int id)

Descripción: Muestra información sobre una reserva específica, utilizando su ID.

Función en el sistema: Proporciona confirmación visual de que una reserva se ha registrado correctamente y permite consultar la información de la reserva.

ToString()

Descripción: Genera una cadena de texto con la información del hotel, incluyendo su nombre, ubicación, categoría y servicios ofrecidos.

Función en el sistema: Sirve para presentar una vista general del hotel, ya sea para mostrarla en la interfaz o en reportes, proporcionando una descripción completa y rápida del establecimiento

# 

# 8 Herencia:

Clase padre: Pago

Clases hijas: PagoEfectivo y PagoConTarjeta

En mi proyecto de gestión de hotel, he diseñado una estructura de clases para manejar los pagos realizados por los clientes. Para ello, defino una clase base llamada Pago, que almacena la información general de un pago, y luego creo clases derivadas que representan diferentes tipos de pago. Esto me permite reutilizar la lógica común y extenderla para casos específicos, siguiendo los principios de herencia en Programación Orientada a Objetos.

namespace ProyectoHotel

{

internal class Pago // Clase padre

{

public int idPago { get; set; }

public DateTime fechaPago { get; set; }

public decimal monto { get; set; }

public string tipoPago { get; set; }

public Pago(int idPago, int idReserva, DateTime fechaPago, decimal monto, string tipoPago)

{

this.idPago = idPago;

this.fechaPago = fechaPago;

this.monto = monto;

this.tipoPago = tipoPago;

}

}

}

namespace ProyectoHotel

{

internal class PagoConTarjeta : Pago //Clase hija

{

public int NumeroTarjeta { get; set; }

public string TipoTarjeta { get; set; } // Credito o Debito

// Constructor que llama al constructor base y agrega detalles específicos de la tarjeta

public PagoConTarjeta(int idPago, int idReserva, DateTime fechaPago, decimal monto, string tipoPago, int numeroTarjeta, string tipoTarjeta)

: base(idPago, idReserva, fechaPago, monto, tipoPago)

{

NumeroTarjeta = numeroTarjeta;

TipoTarjeta = tipoTarjeta;

}

}

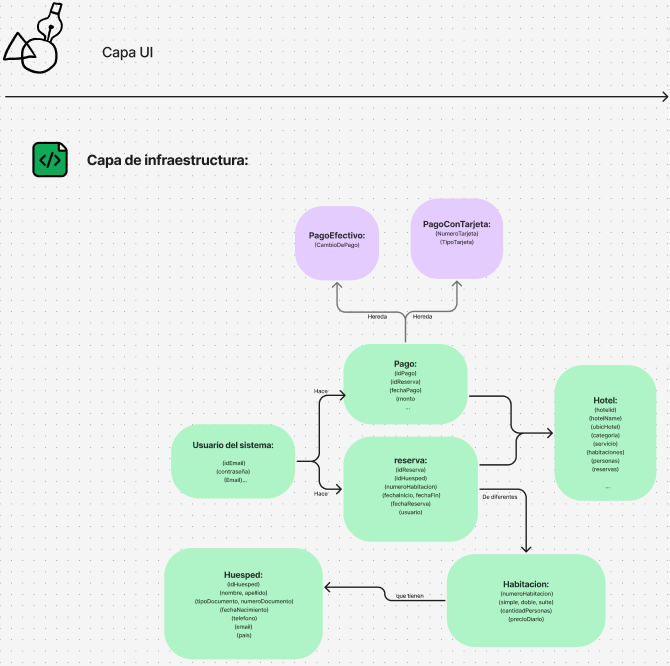
}

# Diagrama de Capas UML:

Adjunto link debido a que *figma* se utiliza de forma online.

Link: https://www.figma.com/board/6ONaJZhmjsb1m78zU62yWb/DiagramaDeCapas?node-id=0-1&t=4aOBpKBNf3SshTlU-1

Vista previa:



# Cronograma de trabajo:

|  |  |
| --- | --- |
| Actividad realizada: | Tiempo: |
| Diagrama de Capas. | 2 hs. |
| Control 1 Diagrama de Capas. | 2hs Entrega 15 de noviembre. |
| Actualizacion Diagrama de Capas | 1h. |
| Creación de Clases. | 2 hs. |
| Altas, bajas y modificaciones. | 10 hs Entrega 13 de noviembre. |
| Reservas | 16 hs |
| Menú | 4 hs |
| Diagrama de Capas UML | 2 hs |
| Testeo | 2hs. |
| Documentación | 6hs |