

Cruceros El Dorado

Viajes a todo el mundo.

Manual de uso

Última actualización 4 de octubre del 2022

Sistema de administración de servicios de viajes

En el presente manual se explicará el funcionamiento del programa con ilustraciones para el uso, luego se explicará el funcionamiento con diagramas de clases y capturas de pantalla de código, y finalmente se señalará dónde se aplicaron los conceptos de los temas 01 al 09.



¡Atención!

El manual de uso se actualiza después de actualizar a la última versión del programa. Puede que el contenido de este manual difiera levemente del funcionamiento y código del repositorio.

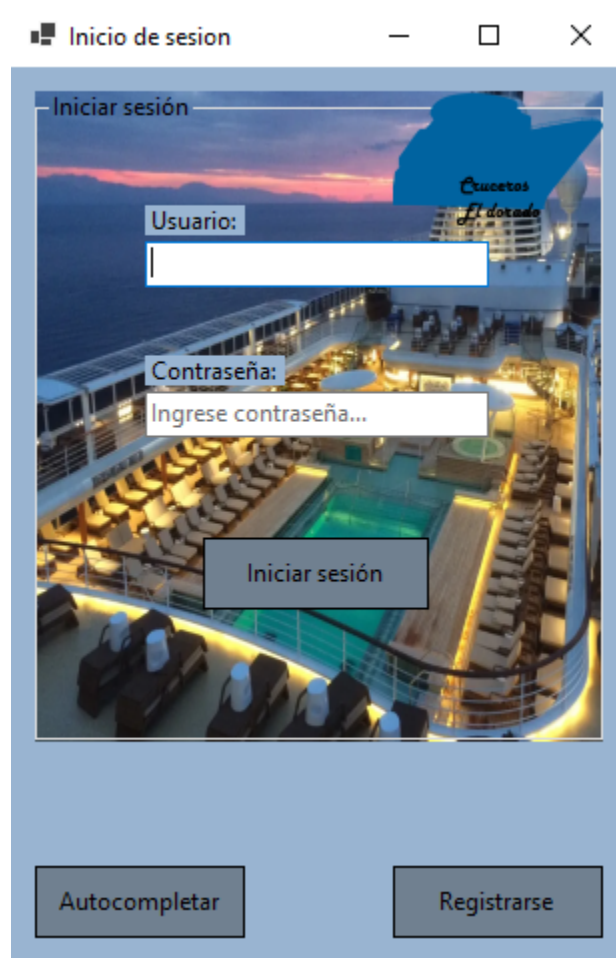
Docente: NEINER, Maximiliano.

Alumno: LUCIANO, Mauro.

Uso del sistema

Login

Inicio de sesión

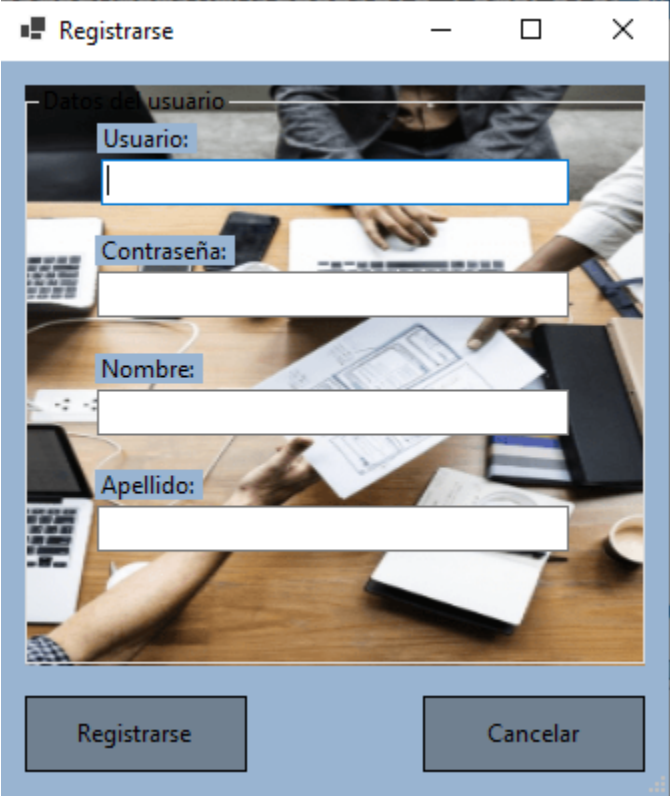


The screenshot shows a web application window titled "Inicio de sesion". The background image depicts a cruise ship deck at night with a pool and lounge chairs. The login form is overlaid on this image. It includes a title "Iniciar sesión", a "Usuario:" label with a text input field, a "Contraseña:" label with a password input field containing the placeholder "Ingrese contraseña...", and a central "Iniciar sesión" button. At the bottom of the window are two buttons: "Autocompletar" and "Registrarse". A blue cloud-like graphic is positioned in the upper right corner of the login area.

Para iniciar sesión, ingrese sus credenciales registradas en el sistema y a continuación presione la tecla "Enter" o haga click en el botón "Iniciar sesión". Se provee un usuario por defecto para el inicio de sesión, con nombre de usuario "maurnl" y contraseña "contrasenia123". Estas credenciales serán cargadas por defecto presionando el botón "Autocompletar". Para registrar un nuevo usuario presione "Registrarse".

4 - Manual de uso

Registro de nuevo usuario



Registrarse

Datos del usuario

Usuario:

Contraseña:

Nombre:

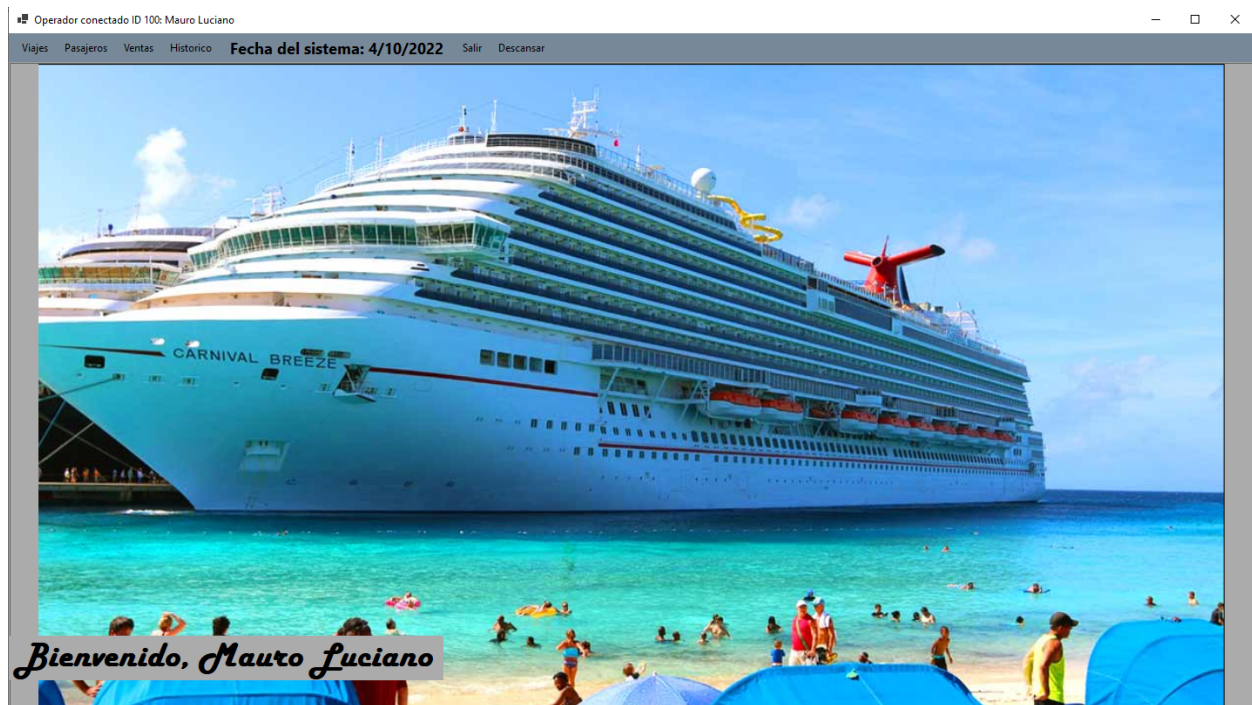
Apellido:

Registrarse

Cancelar

Cree un nuevo usuario ingresando los datos nombre de usuario, contraseña, nombre y apellido. Los campos de nombre y apellido no podrán contener caracteres numéricos. Para completar el registro presione "Registrarse" y se debería ver un mensaje de éxito en el formulario de Login. Para cancelar el registro presione "Cancelar".

Panel principal



Una vez conectado con su usuario y contraseña registrados podrá acceder al menú principal del sistema. Aquí tendrá las siguientes opciones: Ver listados de viajes, de pasajeros, de ventas e historial de viajes y estadísticas.

Listados

Listado de viajes

Operador conectado ID 100: Mauro Luciano

Viajes Pasajeros Ventas Historico **Fecha del sistema: 4/10/2022** Salir Descansar

Agregar pasajero... Editar viaje... Eliminar filtros

☐ Filtrar cruceros con gimnasio
☐ Filtrar cruceros con piscina

	Id	EstadoDeViaje	Origen	Destino	Crucero	PrecioTurista	PrecioPremium	DuracionEnHoras	PasajerosABordo	PasajerosTuristas	PasajerosPremiur	FechaSalida	FechaLlegada	OfreceGimna
	5000	Lleno	Regional: Buen...	Regional: Puert...	[OCUPADO] 'P...	11742	14090,4	206	100	64	36	4/10/2022 14:59	13/10/2022 04:59	SI
	5001	Abordando	Regional: Buen...	Regional: Ushu...	[OCUPADO] 'Ba...	9063	10875,6	159	15	5	10	4/10/2022 14:59	11/10/2022 05:59	NO
	5002	Abordando	Regional: Buen...	Regional: Recif...	[OCUPADO] 'La...	12255	14706	215	53	41	12	4/10/2022 14:59	13/10/2022 13:59	SI
	5003	Abordando	Regional: Buen...	Regional: Santi...	[OCUPADO] 'AL...	8094	9712,8	142	25	3	22	4/10/2022 14:59	10/10/2022 12:59	SI
	5004	Abordando	Regional: Buen...	Regional: Recif...	[OCUPADO] 'El ...	7980	9576	140	0	0	0	4/10/2022 14:59	10/10/2022 10:59	NO
	5005	Abordando	Regional: Buen...	Regional: Puert...	[OCUPADO] 'Tr...	5814	6976,8	102	0	0	0	4/10/2022 14:59	8/10/2022 20:59	NO

En el listado de viajes, se podrá ver todos los viajes en sistema y, se podrá seleccionar un viaje desde la primera columna de la izquierda (que contiene un cursor) y hacer presionar “Agregar pasajero...” para dar de alta un pasajero o grupo familiar al vuelo actualmente seleccionado.

Para editar un viaje, debe seleccionar el viaje y presionar “Editar viaje...”. Se podrá cambiar el destino y la fecha de llegada del viaje.

Para filtrar según la necesidad del cliente, tilde las casillas de filtro necesarias. Para eliminar los filtros aplicados, presione “Eliminar filtros”.

7 - Manual de uso

Pasajeros

Operador conectado ID 100: Mauro Luciano

Viajes Pasajeros Ventas Historico **Fecha del sistema: 4/10/2022** Salir Descansar

Ver información detallada... Base de datos de pasajeros Eliminar filtros Filtrar por: DNI Ingrese valor: Mostrando viaje... Viaje Id 5000. De Regional: BuenosAires, Argentina a Regional: PuertoMadr

Id	Dni	Edad	Genero	FechaNacimiento	Pasaporte	TipoPasajero	CantidadEquipaje	EstaViajando	CantidadViajesRe	Equipaje	NombreCompleto
1030	3460993	29	Masculino	30/12/1992	Fecha de venci...	Premium	2	<input checked="" type="checkbox"/>	1	Peso: 17 kg. Es ...	Hilliard Drewell
1031	3678368	32	Femenino	3/8/1990	Fecha de venci...	Premium	1	<input checked="" type="checkbox"/>	1	Peso: 17 kg. Es ...	Frankie Drewell
1032	3816009	23	Femenino	3/6/1999	Fecha de venci...	Premium	2	<input checked="" type="checkbox"/>	1	Peso: 17 kg. Es ...	Keri Godsmark
1033	2098206	50	Masculino	29/5/1972	Fecha de venci...	Premium	1	<input checked="" type="checkbox"/>	1	Peso: 17 kg. Es ...	Roxanne Enser
1034	3017386	49	Femenino	23/3/1973	Fecha de venci...	Premium	2	<input checked="" type="checkbox"/>	1	Peso: 17 kg. Es ...	Daphne Enser
1035	2754770	38	Masculino	23/9/1984	Fecha de venci...	Premium	1	<input checked="" type="checkbox"/>	1	Peso: 17 kg. Es ...	Roxanne Owers
1037	3115937	38	Masculino	13/9/1984	Fecha de venci...	Turista	2	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Daniella Benne...
1038	2326919	51	Femenino	12/2/1971	Fecha de venci...	Turista	1	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Roxanne McCal...
1039	2043421	40	Femenino	19/9/1982	Fecha de venci...	Turista	2	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Arnold Drewell
1040	2410500	31	Masculino	12/12/1990	Fecha de venci...	Turista	1	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Alfie Denison
1041	2799267	40	Femenino	14/11/1981	Fecha de venci...	Turista	2	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Roxanne Denison
1042	2461815	21	Femenino	7/1/2001	Fecha de venci...	Turista	1	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Hilliard Denison
1043	2082666	47	Masculino	22/6/1975	Fecha de venci...	Turista	2	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Arnold McCalli...
1044	4463098	40	Masculino	16/10/1981	Fecha de venci...	Turista	1	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Corilla Enser
1045	2171091	29	Masculino	5/7/1993	Fecha de venci...	Turista	2	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Tuck Benneywo...
1046	3120570	20	Femenino	25/9/2002	Fecha de venci...	Turista	1	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Corilla Wickstead
1047	4211241	49	Masculino	5/2/1973	Fecha de venci...	Turista	2	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Pepillo Benney...
1048	3574201	50	Masculino	25/11/1971	Fecha de venci...	Turista	1	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Alfie Drewell
1049	4965147	30	Femenino	3/11/1991	Fecha de venci...	Turista	2	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Frankie Denison
1050	4293747	33	Femenino	19/8/1989	Fecha de venci...	Turista	1	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Daphne Denison
1051	4447807	29	Masculino	12/2/1993	Fecha de venci...	Turista	2	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Pepillo Godsm...
1052	4020798	30	Femenino	10/11/1991	Fecha de venci...	Turista	1	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Daphne Enser
1053	4564744	32	Masculino	13/10/1989	Fecha de venci...	Turista	2	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Garvy Drewell
1054	2803030	35	Femenino	12/5/1987	Fecha de venci...	Turista	1	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Frankie Makiver
1055	2867959	33	Masculino	4/9/1989	Fecha de venci...	Turista	2	<input checked="" type="checkbox"/>	1	Peso: 18 kg. Es ...	Pepillo Enser

En el listado de pasajeros, se podrá seleccionar de cada vuelo, los pasajeros que están a bordo.

En el listado de opciones en la esquina superior derecha, seleccione en vuelo a informar pasajeros.

8 - Manual de uso

Operador conectado ID 100: Mauro Luciano

Viajes Pasajeros Ventas Historico Fecha del sistema: 4/10/2022 Salir Descansar

Calcular recaudacion Mostrar destino mas demandado Eliminar filtros

Mostrando ventas del viaje... Viaje Id 5000. De Regional: BuenosAires, Argentina a Regional: PuertoMadrin, A

Id	CantidadPasajeros	Clientes	Fecha	PrecioBruto	PrecioNeto	EsServicioRegion	Viaje
0	1	Nombre: Daph...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
1	1	Nombre: Ethel...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
2	1	Nombre: Ethel...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
3	1	Nombre: Davin...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
4	1	Nombre: Alfie E...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
5	1	Nombre: Ethel...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
6	1	Nombre: Arnol...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
7	1	Nombre: Roxan...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
8	1	Nombre: Alfie ...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
9	1	Nombre: Ethel...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
10	1	Nombre: Pepill...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
11	1	Nombre: Corill...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
12	1	Nombre: Keri ...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
13	1	Nombre: Roxan...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
14	1	Nombre: Arnol...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
15	1	Nombre: Franki...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
16	1	Nombre: Arnol...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
17	1	Nombre: Davin...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
18	1	Nombre: Ethel...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
19	1	Nombre: Jethro...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
20	1	Nombre: Gany ...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
21	1	Nombre: Ethel...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
22	1	Nombre: Roxan...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
23	1	Nombre: Davin...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...
24	1	Nombre: Keri ...	4/10/2022	18057,6	21849,695	<input checked="" type="checkbox"/>	Viaje Id 5000. D...

En el listado de ventas, se podrán ver todas las ventas realizadas en el sistema, ordenadas por ID.

9 - Manual de uso

Estadísticas históricas

Operador conectado ID 100: Mauro Luciano

Viajes Pasajeros Ventas Historico **Fecha del sistema: 19/10/2022** Salir Descansar

Ver viajes finalizados Ver recaudacion por destinos Calcular recaudacion del viaje Ver ganancias por servicio

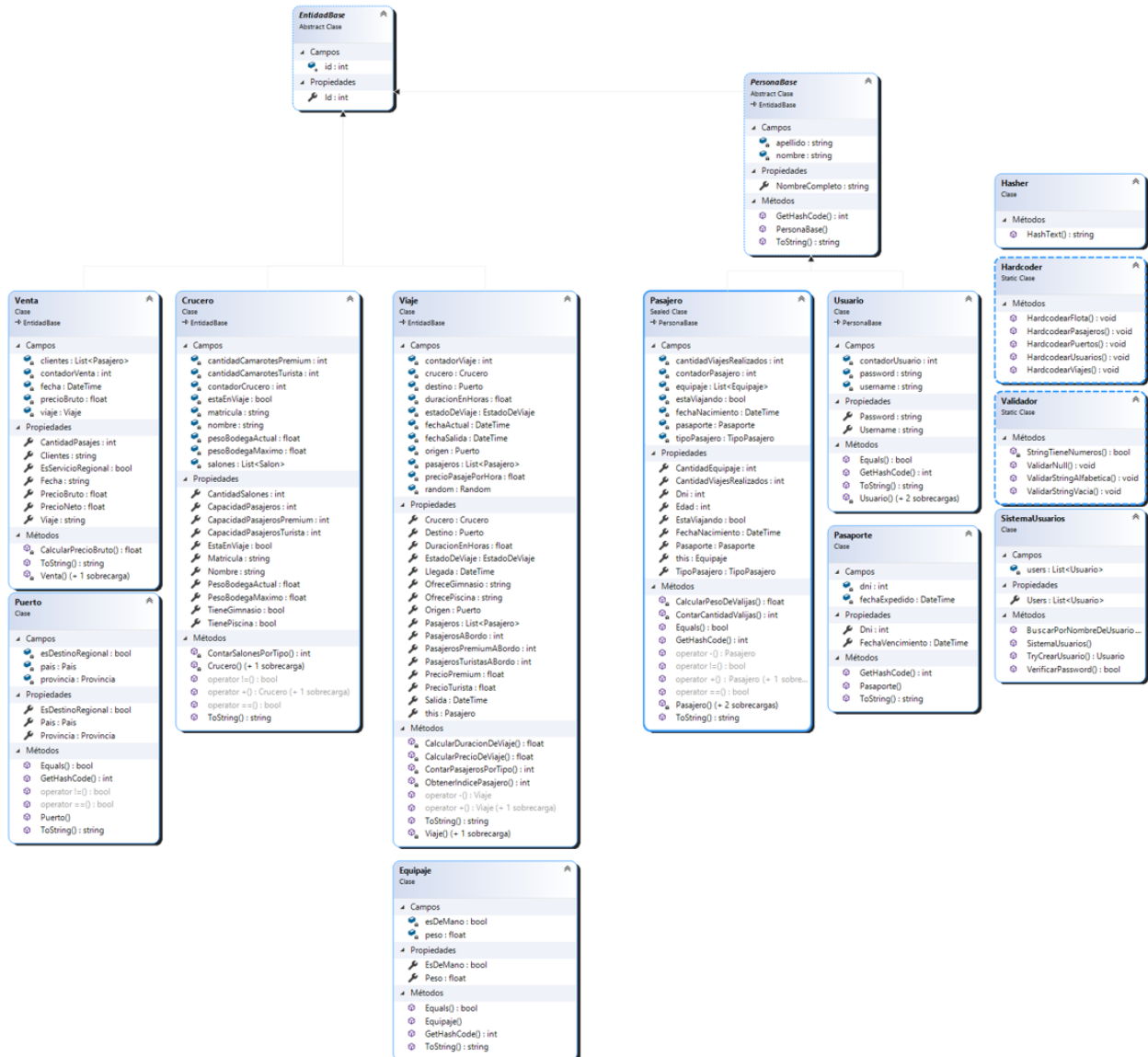
	Id	EstadoDelViaje	Origen	Destino	Crucero	PrecioTurista	PrecioPremium	DuracionEnHoras	PasajerosABordo	PasajerosTuristas	PasajerosPremiur	FechaSalida	FechaLlegada	OfreceGimna
▶	5004	Finalizado	Regional: Buen...	Regional: Recif...	[DISPONIBLE] '...	8550	10260	150	0	0	0	4/10/2022 16:02	10/10/2022 22:02	NO
	5003	Finalizado	Regional: Buen...	Regional: Santi...	[DISPONIBLE] '...	7011	8413,2	123	25	3	22	4/10/2022 16:02	9/10/2022 19:02	SI
	5001	Finalizado	Regional: Buen...	Regional: Ushu...	[DISPONIBLE] '...	11514	13816,8	202	15	5	10	4/10/2022 16:02	13/10/2022 02:02	NO
	5005	Finalizado	Regional: Buen...	Regional: Puert...	[DISPONIBLE] '...	17841	21409,2	313	0	0	0	4/10/2022 16:02	17/10/2022 17:02	NO
	5002	Finalizado	Regional: Buen...	Regional: Recif...	[DISPONIBLE] '...	19551	23461,2	343	53	41	12	4/10/2022 16:02	18/10/2022 23:02	SI
	5000	Finalizado	Regional: Buen...	Regional: Puert...	[DISPONIBLE] '...	15048	18057,6	264	71	35	36	4/10/2022 16:02	15/10/2022 16:02	SI

En este listado, se mostrarán solo los viajes finalizados. Se podrá presionar en “Ver viajes finalizados” para entrar a este listado. Luego, se podrá seleccionar un viaje y calcular su recaudación, presionando en “Calcular recaudación del viaje”. Se podrá también ver la recaudación por destinos presionando en “Ver recaudación por destinos”. Finalmente se podrá ver la recaudación por tipo de servicio presionando en “Ver ganancias por servicio”.

Estructura del programa

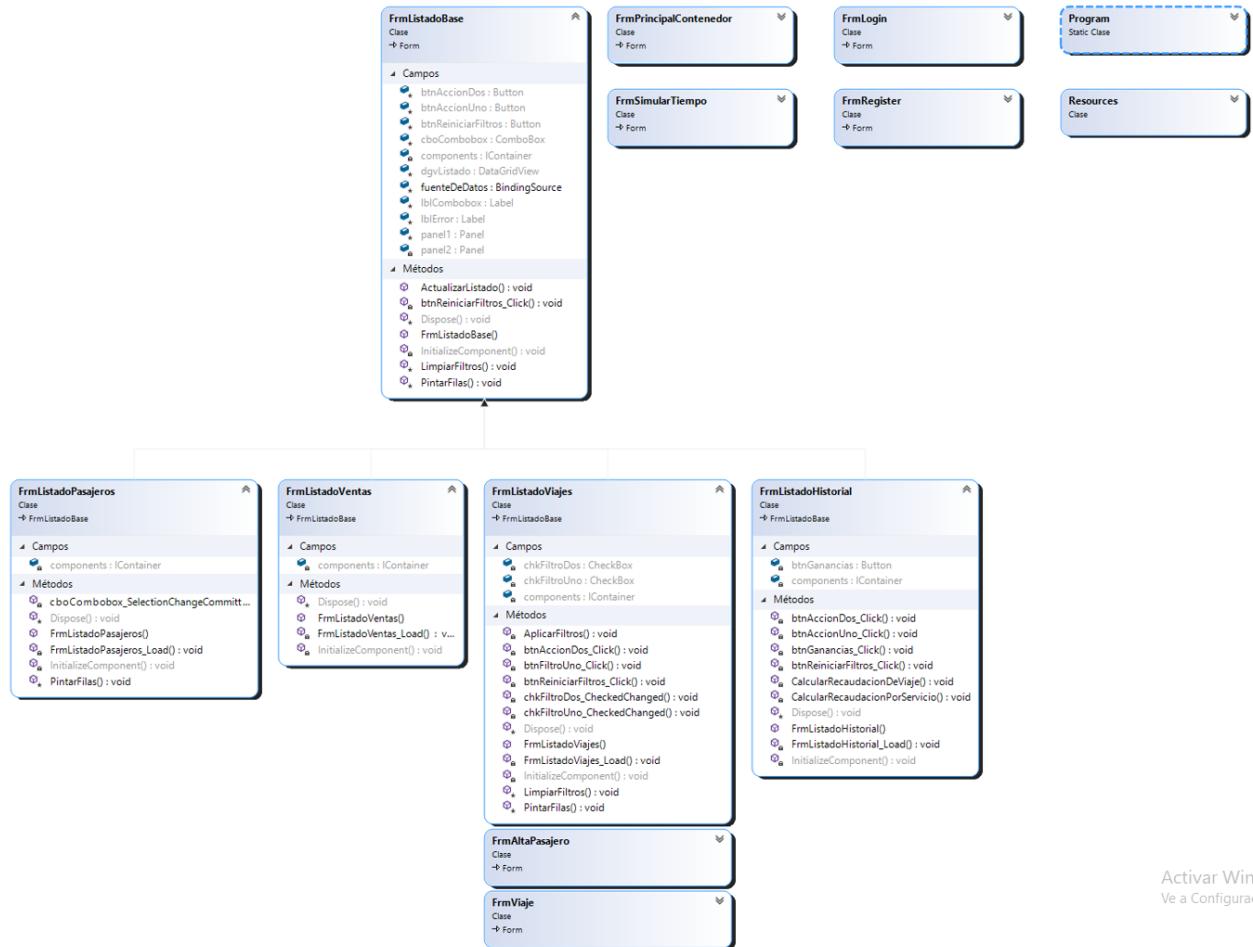
Biblioteca de clases

Diagrama de clases



Formularios

Diagrama de clases



Justificación técnica

Tema 01: C# y .NET

La aplicación se desarrolló sobre el framework .NET 5.0, utilizando el lenguaje de programación C# ya que es la exigencia de la cátedra.

Tema 02: Clases y métodos estáticos

Se decidió utilizar una clase estática **SistemaCruceros** a fin de:

- Tener una sola clase no instanciable, y no heredable que maneje el sistema.
- Poder acceder desde cualquier Formulario a la clase, sin necesidad de tener la referencia pasada por el constructor.
- Controlar la fecha del sistema desde una única instancia DateTime

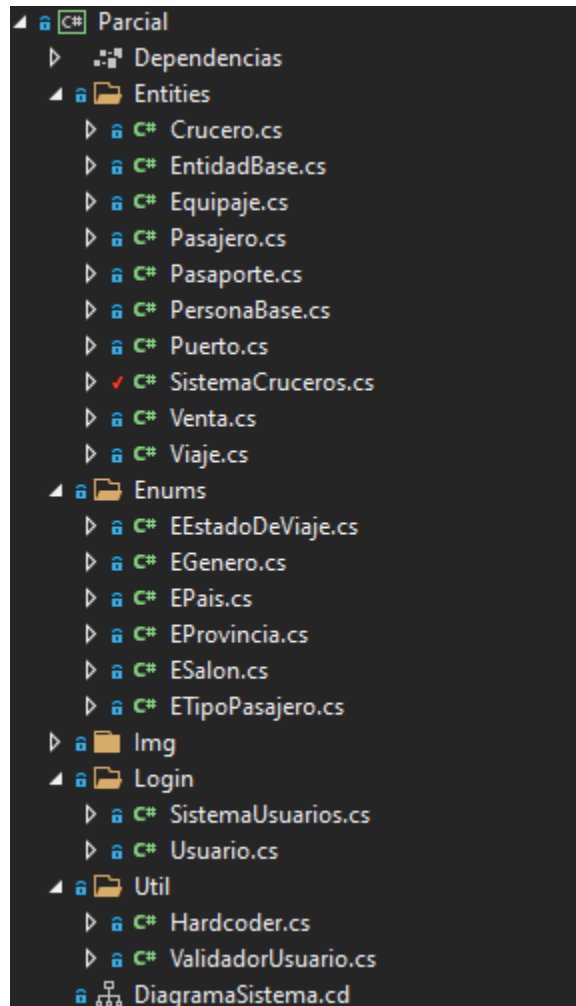
```
0 referencias
static SistemaCruceros()...

8 referencias
public static DateTime FechaDelSistema...
1 referencia
public static List<Crucero> Flota...
2 referencias
public static List<Venta> Ventas...
2 referencias
public static List<Puerto> Puertos...
10 referencias
public static List<Viaje> Viajes...
3 referencias
public static List<Viaje> HistorialViajes...
1 referencia
public static Dictionary<Puerto, float> DiccionarioRecaudacionPorDestino...
2 referencias
public static List<Pasajero> BaseDeDatosPasajeros...

2 referencias
public static void ActualizarViajesActivos()...
2 referencias
public static Pasajero ObtenerPasajeroEnSistema(int dni)...
1 referencia
public static Puerto ObtenerDestinoMasDemandado()...
2 referencias
public static Venta AltaDeVenta(Viaje viaje, List<Pasajero> pasajerosPosibles)...
1 referencia
public static Viaje AltaDeViaje(Puerto origen, Puerto destino, Crucero crucero, DateTime fechaSalida)...
```

Tema 03: Programación Orientada a Objetos

De acuerdo con las exigencias de la carrera, se adoptó el paradigma de POO para desarrollar la solución, modelando el negocio en objetos de la vida real.



Tema 04: Sobrecarga

Sobrecarga de operadores de conversión implícitos:

Se generó la sobrecarga de operadores con el objetivo de crear un objeto Pasaporte a partir del DNI de un pasajero implícitamente, encapsulando el constructor y sólo exponiendo la creación de instancia a este método.

```
/// <summary>
/// Convierte implícitamente un valor entero a un objeto Pasaporte.
/// </summary>
/// <param name="numeroPasaporte">Valor de numero del pasaporte a crear</param>
public static implicit operator Pasaporte(int numeroPasaporte)
{
    return new Pasaporte(numeroPasaporte, DateTime.Now);
}
```

Sobrecarga de operadores

Se realizó la sobrecarga de operadores para encapsular las colecciones pertenecientes a las clases, no dando acceso a la colección directamente y dando formas de realizar operaciones sobre ellas. Además, se utilizó para comparar los tipos generados para la solución de manera customizada.


```

/// <summary>
/// Agrega el Salon sumado a la lista de salones del crucero.
/// </summary>
/// <param name="crucero"></param>
/// <param name="salon">El Salon a agregar</param>
/// <returns></returns>
24 referencias
public static Crucero operator +(Crucero crucero, Salon salon)
{
    crucero.salones.Add(salon);
    return crucero;
}

```

```

1 referencia
public static Viaje operator +(Viaje viaje, Pasajero pasajero)
{
    if (viaje == pasajero)
    {
        throw new Exception("Este pasajero ya esta a bordo.");
    }

    if (pasajero.EstaViajando)
    {
        throw new Exception("Este pasajero ya esta en algun viaje.");
    }

    if (viaje.PasajerosABordo + 1 > viaje.crucero.CapacidadPasajeros)
    {
        viaje.estadoDeViaje = EstadoDeViaje.Lleno;
        throw new Exception("Este crucero esta lleno.");
    }

    if (pasajero.TipoPasajero == TipoPasajero.Premium
        && viaje.PasajerosPremiumABordo + 1 > viaje.crucero.CapacidadPasajerosPremium)
    {
        throw new Exception("No hay mas asientos Premium");
    }

    if (pasajero.TipoPasajero == TipoPasajero.Turista
        && viaje.PasajerosTuristasABordo + 1 > viaje.crucero.CapacidadPasajerosTurista)
    {
        throw new Exception("No hay mas asientos Turista");
    }

    for (int i = 0; i < pasajero.CantidadEquipaje; i++)
    {
        viaje.crucero += pasajero[i].Peso;
    }

    pasajero += 1;
    viaje.pasajeros.Add(pasajero);
    pasajero.EstaViajando = true;
    return viaje;
}

```

Sobrecarga de constructores

Se utilizó la sobrecarga de constructores para mejorar la legibilidad del código al momento de cargar los ID de las entidades separando este proceso de la inicialización de los demás atributos de la clase. Además, se implementó para la reutilización de formular

```
static Pasajero()
{
    Pasajero.contadorPasajero = 1000;
}

private Pasajero(string nombreCompleto) : base(nombreCompleto)
{
    base.id = Pasajero.contadorPasajero;
    Pasajero.contadorPasajero++;
}
```

```
2 referencias
public FrmViaje()
{
    InitializeComponent();
    this.estaEditando = false;
}

1 referencia
public FrmViaje(Viaje viaje) : this()
{
    this.estaEditando = true;
    this.cboOrigen.Enabled = false;
    this.viajeDelForm = viaje;
    this.cboOrigen.SelectedItem = viaje.Origen;
    this.cboDestino.SelectedItem = viaje.Destino;
    this.cboCrucero.SelectedItem = viaje.Crucero;
    this.cboCrucero.Enabled = false;
    this.dtpFechaSalida.Value = viaje.FechaSalida;
    this.viajeDelForm.Crucero.EstaEnViaje = false;
    this.btnAceptar.Text = "Aplicar cambios";
}
```

Tema 05: Windows Forms

La aplicación se desarrolló en Windows Forms, utilizando el lenguaje de programación C# ya que es la exigencia de la cátedra.

Tema 06: Colecciones

Se decidió implementar dos tipos de colecciones genéricas: List y Dictionary. Para la primera, se utilizó con el fin de guardar una colecciones manteniendo la seguridad de tipo. Para la segunda, se utilizó un diccionario para guardar colecciones de tipo clave-valor, con el objetivo de guardar recaudaciones clasificados por destinos.

92 referencias

```
public static class SistemaCruceros
{
    private static DateTime fechaDelSistema;
    private static List<Crucero> flota;
    private static List<Venta> ventas;
    private static List<Puerto> puertos;
    private static List<Viaje> viajes;
    private static List<Viaje> historialViajes;
    private static Dictionary<Puerto, float> diccionarioRecaudacionPorDestino;
```

Tema 07: Encapsulamiento

Se ha aplicado el encapsulamiento en las clases principales del programa, con el objetivo de sólo dar acceso a comportamientos específicos de las colecciones, y no a la colección entera.

21 - Manual de uso

```
/// <summary>
/// Indexador para Pasajero. Retorna el equipaje correspondiente al indice de la lista de equipajes.
/// </summary>
/// <param name="indice">Indice</param>
/// <returns></returns>
5 referencias
public Equipaje this[int indice]
{
    get
    {
        Equipaje equipajeRetorno = null;
        if(indice >= 0 && indice < this.equipaje.Count)
        {
            equipajeRetorno = this.equipaje[indice];
        }
        return equipajeRetorno;
    }
    set
    {
        if (indice >= 0 && indice < this.equipaje.Count)
        {
            this.equipaje[indice] = value;
        }
    }
}
```

```
/// <summary> Agrega el Equipaje a la lista de equipajes del pasajero.
3 referencias
public static Pasajero operator +(Pasajero pasajero, Equipaje equipaje)
{
    if (!equipaje.EsDeMano && (pasajero.TipoPasajero == TipoPasajero.Turista && pasajero.ContarCantidadValijas() >= 1
    || pasajero.TipoPasajero == TipoPasajero.Premium && pasajero.ContarCantidadValijas() >= 2))
    {
        throw new Exception($"Se ha excedido el maximo de valijas permitidos (MAX {(pasajero.tipoPasajero == TipoPasajero.Turista ? '1' : '2')})");
    }
    if (equipaje.EsDeMano && pasajero.equipaje.Count - pasajero.ContarCantidadValijas() != 0)
    {
        throw new Exception("Este pasajero ya tiene un bolso de mano. (MAX 1)");
    }
    if (!equipaje.EsDeMano)
    {
        if (pasajero.tipoPasajero == TipoPasajero.Turista && pasajero.CalcularPesoDeValijas() + equipaje.Peso > 25
        || pasajero.tipoPasajero == TipoPasajero.Premium && pasajero.CalcularPesoDeValijas() + equipaje.Peso > 50)
        {
            throw new Exception("Se ha excedido del peso maximo permitido");
        }
    }

    pasajero.equipaje.Add(equipaje);
    return pasajero;
}
```

```
/// <summary>
/// Elimina el equipaje de la lista de equipajes del pasajero.
/// </summary>
/// <param name="pasajero"></param>
/// <param name="equipaje">Equipaje a eliminar</param>
/// <returns></returns>
1 referencia
public static Pasajero operator -(Pasajero pasajero, Equipaje equipaje)
{
    pasajero.equipaje.Remove(equipaje);
    return pasajero;
}
```

Tema 08: Herencia

Se ha optado por crear dos jerarquías de herencia: La primera en las clases de la Biblioteca, teniendo a **PersonaBase** como clase base, heredando hacía **Usuario** y **Pasajero**; y la segunda en los Formularios, heredando de **FrmListadoBase** hacia **FrmListadoVentas**, **FrmListadoViajes**, etc. Se estructuró así, con el fin de reutilizar la mayor cantidad de código y elementos visuales posibles (Ver diagramas de clases).

Tema 09: Polimorfismo y clases abstractas

Se aplicó polimorfismo en los formularios principalmente, para utilizar el método **PintarFilas** con un comportamiento por defecto desde la clase padre, y una implementación distinta para sus derivados.

[3 referencias](#)

```
protected virtual void PintarFilas()
{
    if(this.dgvListado.DataSource == null)
    {
        return;
    }

    for (int i = 0; i < this.dgvListado.RowCount; i++)
    {
        DataGridViewRow filaActual = this.dgvListado.Rows[i];
        {
            if (i%2 == 0)
            {
                filaActual.DefaultCellStyle.BackColor = Color.LightGray;
            }
        }
    }
}
```

2 referencias

```
protected override void PintarFilas()
{
    for (int i = 0; i < SistemaCruceros.Viajes.Count; i++)
    {
        DataGridViewRow filaActual = base.dgvListado.Rows[i];
        if (filaActual.DataBoundItem is Viaje viaje)
        {
            if (viaje.EstadoDeViaje == EstadoDeViaje.Lleno)
            {
                filaActual.DefaultCellStyle.BackColor = Color.LightSalmon;
            }
            else if (viaje.EstadoDeViaje == EstadoDeViaje.EnCurso)
            {
                filaActual.DefaultCellStyle.BackColor = Color.Yellow;
            }
            else
            {
                filaActual.DefaultCellStyle.BackColor = Color.LightGreen;
            }
        }
    }
}
```

2 referencias

```
protected override void PintarFilas()
{
    for (int i = 0; i < this.dgvListado.RowCount; i++)
    {
        DataGridViewRow filaActual = base.dgvListado.Rows[i];
        if (filaActual.DataBoundItem is Pasajero pasajero)
        {
            filaActual.DefaultCellStyle.BackColor = Color.LightBlue;
            if (pasajero.TipoPasajero == TipoPasajero.Premium)
            {
                filaActual.DefaultCellStyle.BackColor = Color.DarkGoldenrod;
            }
        }
    }
}
```


7 referencias

```
public void ActualizarListado()
{
    this.fuenteDeDatos.ResetBindings(false);
    this.PintarFilas();
}
```

7 referencias

```
private void MostrarListado(string nombreListado)
{
    FrmListadoBase formMostrado;
    this.lblBienvenido.Visible = false;
    switch (nombreListado)
    {
        case "viajes":
            formMostrado = formListadoViajes;
            break;
        case "pasajeros":
            formMostrado = formListadoPasajeros;
            break;
        case "ventas":
            formMostrado = formListadoVentas;
            break;
        case "historico":
            formMostrado = formListadoHistorial;
            break;
        default:
            formMostrado = formListadoViajes;
            break;
    }
    formMostrado.RemoveSeleccionDatagrid();
    formMostrado.Show();
    formMostrado.Activate();
    formMostrado.ActualizarListado();
}
```

Tema 10: Excepciones

Adicionalmente, se utilizaron excepciones genéricas para la validación de datos, no dependiendo de Windows Forms para ello, con el fin de la mantenibilidad, flexibilidad y robustez del código.