

Cruceros El Dorado

Viajes a todo el mundo.

Manual de uso

Última actualización 27 de septiembre 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



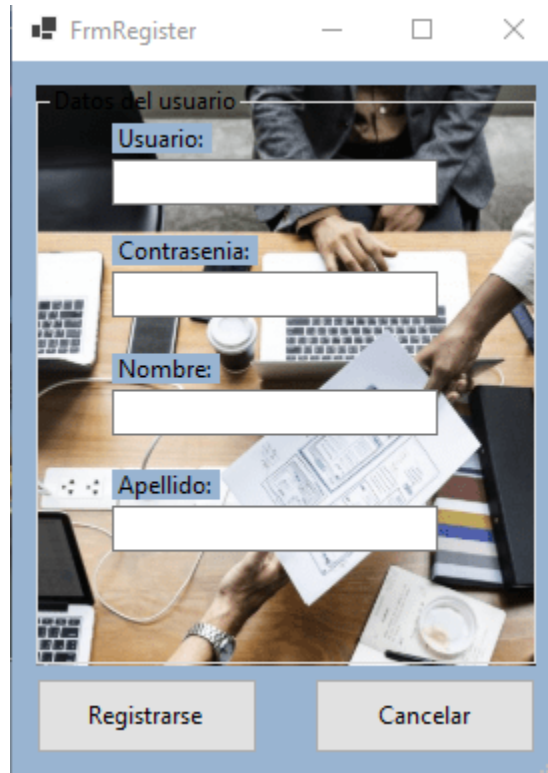
The screenshot shows a web application window titled "Inicio de sesion". The background image depicts the deck of a cruise ship at night, with a swimming pool and lounge chairs illuminated. The ship's name "Cruceiros El Dorado" is visible in the upper right. The login form includes the following elements:

- A header "Iniciar sesion" at the top left of the form area.
- A label "Usuario:" followed by a text input field.
- A label "Contraseña:" followed by a text input field containing the placeholder text "Ingrese contraseña...".
- A button labeled "Iniciar sesion" positioned below the password field.
- At the bottom of the window, two buttons: "Autocompletar" on the left and "Registrarse" on the right.

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



The image shows a screenshot of a Windows application window titled "FrmRegister". The window contains a registration form with the following fields and buttons:

- Datos del usuario** (User Data)
- Usuario:** (Username) - Text input field
- Contraseña:** (Password) - Password input field
- Nombre:** (Name) - Text input field
- Apellido:** (Last Name) - Text input field
- Registrarse** (Register) - Button
- Cancelar** (Cancel) - Button

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 - [Listado de viajes]

Agregar pasajero...
 Editar viaje...
 Eliminar filtros
 ☐ Filtrar cruceros con gimnasio
 ☐ Filtrar cruceros con piscina
 Ordenar por...

	EstadoDeViaje	Origen	Destino	Crucero	PrecioTurista	PrecioPremium	DuracionEnHoras	PasajerosABordo	PasajerosTuristas	PasajerosPremiur	Salida	Llegada	OfreceGimnasio	OfrecePiscina	Id
▶	Abordando	Regional: Buen...	Regional: Puert...	[OCUPADO] Cr...	12198	14637,6	214	20	10	10	27/9/2022 16:04	6/10/2022 14:04	SI	NO	5000
	Abordando	Regional: Buen...	Regional: Ushu...	[OCUPADO] Cr...	11514	13816,8	202	40	30	10	27/9/2022 16:04	6/10/2022 02:04	NO	SI	5001
	Abordando	Regional: Buen...	Regional: Recif...	[OCUPADO] Cr...	18525	22230	325	0	0	0	27/9/2022 16:04	11/10/2022 05:04	SI	SI	5002
	Abordando	Regional: Buen...	Regional: Santi...	[OCUPADO] Cr...	9405	11286	165	0	0	0	27/9/2022 16:04	4/10/2022 13:04	NO	NO	5003
	Abordando	Regional: Buen...	Regional: Recif...	[OCUPADO] Cr...	9234	11080,8	162	0	0	0	27/9/2022 16:04	4/10/2022 10:04	NO	NO	5004
	Abordando	Regional: Buen...	Regional: Puert...	[OCUPADO] Cr...	7467	8960,4	131	0	0	0	27/9/2022 16:04	3/10/2022 03:04	NO	NO	5005

Viajes Pasajeros Ventas Informar 27/9/2022 16:04:34

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 - [Listado de pasajeros]

Mostrando viaje...aje Id 5000. De Regional: BuenosAires, Argen

Boton 1

Boton 2

Eliminar filtros

Dni	Edad	FechaNacimiento	Pasaporte	TipoPasajero	CantidadEquipaje	EstaViajando	CantidadViajesRe	NombreCompleto	Id
2468959	52	12/7/1970	Fecha de venci...	Premium	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1000
3192987	52	12/7/1970	Fecha de venci...	Premium	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1001
3264000	52	12/7/1970	Fecha de venci...	Premium	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1002
4442904	52	12/7/1970	Fecha de venci...	Premium	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1003
2097587	52	12/7/1970	Fecha de venci...	Premium	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1004
3174720	52	12/7/1970	Fecha de venci...	Premium	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1005
4525870	52	12/7/1970	Fecha de venci...	Premium	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1006
3490934	52	12/7/1970	Fecha de venci...	Premium	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1007
3408417	52	12/7/1970	Fecha de venci...	Premium	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1008
4097320	52	12/7/1970	Fecha de venci...	Premium	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1009
2163060	52	12/7/1970	Fecha de venci...	Turista	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1010
4816747	52	12/7/1970	Fecha de venci...	Turista	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1011
3925490	52	12/7/1970	Fecha de venci...	Turista	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1012
3480120	52	12/7/1970	Fecha de venci...	Turista	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1013
4648832	52	12/7/1970	Fecha de venci...	Turista	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1014
3611487	52	12/7/1970	Fecha de venci...	Turista	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1015
3223988	52	12/7/1970	Fecha de venci...	Turista	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1016
2514461	52	12/7/1970	Fecha de venci...	Turista	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1017
4682975	52	12/7/1970	Fecha de venci...	Turista	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1018
3196237	52	12/7/1970	Fecha de venci...	Turista	0	<input checked="" type="checkbox"/>	1	Pepito Pepe	1019

Viajes Pasajeros Ventas Informar

27/9/2022 16:04:34

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

Operator conectado ID 100: Mauro Luciano - [Listado de ventas]

Boton 1 Boton 2 Eliminar filtros Ordenar por...

Cientes	Fecha	PrecioBruto	PrecioNeto	Viaje	EsServicioRegion	CantidadPasajes	Id
Nombre: Pepit...	27/9/2022	14637,6	17711,496	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	0
Nombre: Pepit...	27/9/2022	14637,6	17711,496	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	1
Nombre: Pepit...	27/9/2022	14637,6	17711,496	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	2
Nombre: Pepit...	27/9/2022	14637,6	17711,496	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	3
Nombre: Pepit...	27/9/2022	14637,6	17711,496	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	4
Nombre: Pepit...	27/9/2022	14637,6	17711,496	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	5
Nombre: Pepit...	27/9/2022	14637,6	17711,496	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	6
Nombre: Pepit...	27/9/2022	14637,6	17711,496	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	7
Nombre: Pepit...	27/9/2022	14637,6	17711,496	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	8
Nombre: Pepit...	27/9/2022	14637,6	17711,496	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	9
Nombre: Pepit...	27/9/2022	12198	14759,58	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	10
Nombre: Pepit...	27/9/2022	12198	14759,58	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	11
Nombre: Pepit...	27/9/2022	12198	14759,58	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	12
Nombre: Pepit...	27/9/2022	12198	14759,58	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	13
Nombre: Pepit...	27/9/2022	12198	14759,58	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	14
Nombre: Pepit...	27/9/2022	12198	14759,58	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	15
Nombre: Pepit...	27/9/2022	12198	14759,58	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	16
Nombre: Pepit...	27/9/2022	12198	14759,58	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	17
Nombre: Pepit...	27/9/2022	12198	14759,58	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	18
Nombre: Pepit...	27/9/2022	12198	14759,58	Viaje Id 5000. D...	<input checked="" type="checkbox"/>	1	19
Nombre: Pepit...	27/9/2022	13816,8	16718,328	Viaje Id 5001. D...	<input checked="" type="checkbox"/>	1	20
Nombre: Pepit...	27/9/2022	13816,8	16718,328	Viaje Id 5001. D...	<input checked="" type="checkbox"/>	1	21
Nombre: Pepit...	27/9/2022	13816,8	16718,328	Viaje Id 5001. D...	<input checked="" type="checkbox"/>	1	22
Nombre: Pepit...	27/9/2022	13816,8	16718,328	Viaje Id 5001. D...	<input checked="" type="checkbox"/>	1	23
Nombre: Pepit...	27/9/2022	13816,8	16718,328	Viaje Id 5001. D...	<input checked="" type="checkbox"/>	1	24
Nombre: Pepit...	27/9/2022	13816,8	16718,328	Viaje Id 5001. D...	<input checked="" type="checkbox"/>	1	25
Nombre: Pepit...	27/9/2022	13816,8	16718,328	Viaje Id 5001. D...	<input checked="" type="checkbox"/>	1	26

Viajes Pasajeros Ventas Informar 27/9/2022 16:04:34

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

9 - Manual de uso

Historial

Operador conectado ID 100 Mauro Luciano - [Informes]

Ver viajes finalizados Ver recaudación por destinos Calcular recaudación del viaje Ver ganancias por servicio Ordenar por...

	EstadoDeViaje	Origen	Destino	Crucero	PrecioTurista	PrecioPremium	DuracionEnHoras	PasajerosABordo	PasajerosTuristas	PasajerosPremiur	Salida	Llegada	OfreceGimnasio	OfrecePiscina	Id
▶	Finalizado	Regional: Buen...	Regional: Puert...	[DISPONIBLE] C...	12198	14637,6	214	10	5	5	27/9/2022 16:04	6/10/2022 14:04	SI	NO	5000
	Finalizado	Regional: Buen...	Regional: Ushu...	[DISPONIBLE] C...	11514	13816,8	202	20	15	5	27/9/2022 16:04	6/10/2022 02:04	NO	SI	5001
	Finalizado	Regional: Buen...	Regional: Santi...	[DISPONIBLE] C...	9405	11286	165	0	0	0	27/9/2022 16:04	4/10/2022 13:04	NO	NO	5003
	Finalizado	Regional: Buen...	Regional: Recif...	[DISPONIBLE] C...	9234	11080,8	162	0	0	0	27/9/2022 16:04	4/10/2022 10:04	NO	NO	5004
	Finalizado	Regional: Buen...	Regional: Puert...	[DISPONIBLE] C...	7467	8960,4	131	0	0	0	27/9/2022 16:04	3/10/2022 03:04	NO	NO	5005

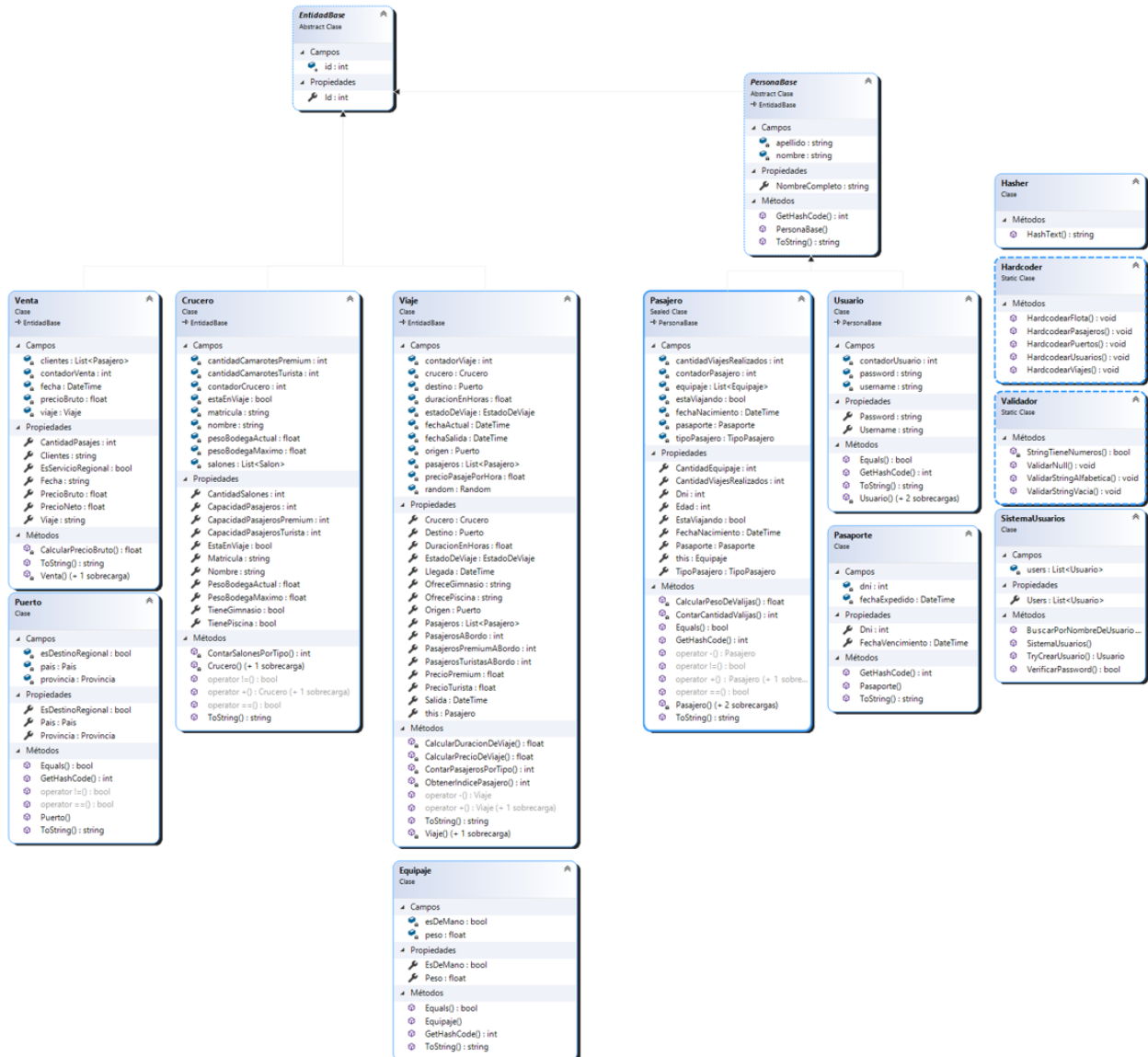
Viajes Pasajeros Ventas Informar 7/10/2022 16:04:34

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

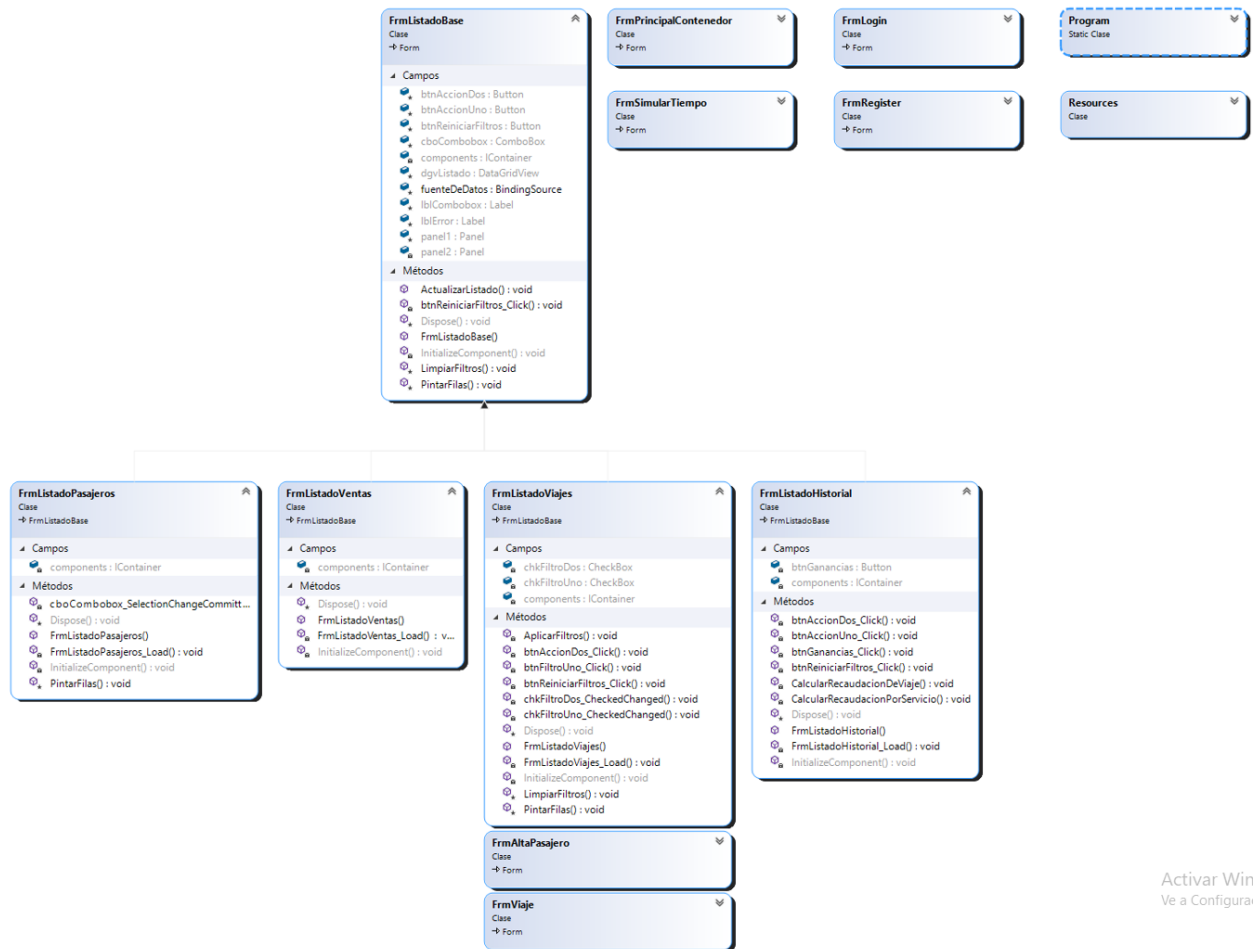
Biblioteca de clases

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 Forma a la clase, sin necesidad de tener la referencia pasada por el constructor.
- Controlar la fecha del sistema desde una única instancia DateTime

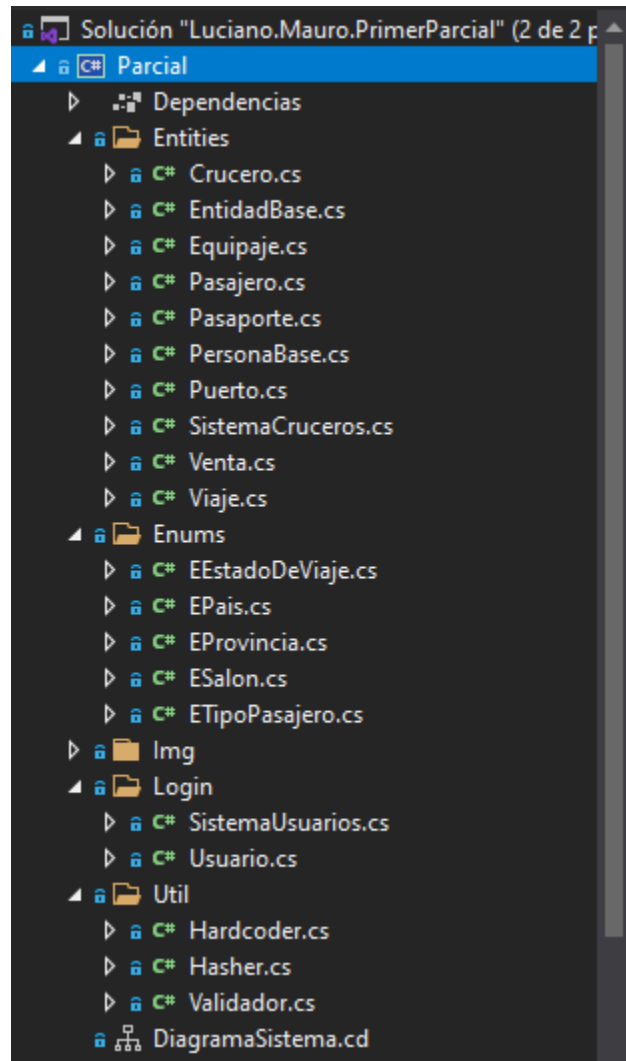
```
51 referencias
public static class SistemaCruceros
{
    public static DateTime fechaDelSistema;
    public static List<Crucero> flota;
    public static List<Viaje> historialViajes;
    public static List<Viaje> viajes;
    public static List<Venta> ventas;
    public static List<Puerto> puertos;
    public static Dictionary<Puerto, float> contadorRecaudacionDestinos;
    1 referencia
    public static List<Pasajero> BaseDeDatosPasajeros[...]

0 referencias
    static SistemaCruceros() [...]

2 referencias
    public static Pasajero ObtenerPasajeroEnSistema(int dni)
    {
        Pasajero pasajeroEncontrado = null;
        foreach (Pasajero pasajero in SistemaCruceros.BaseDeDatosPasajeros)
        {
            if(pasajero.Dni == dni)
            {
                pasajeroEncontrado = pasajero;
                break;
            }
        }
        return pasajeroEncontrado;
    }
}
```

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.

```
1 referencia
private Pasaporte(int dni, DateTime fechaExpedido)
{
    this.dni = dni;
    this.fechaExpedido = fechaExpedido;
}

public static implicit operator Pasaporte(int dni)
{
    return new Pasaporte(dni, 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.

4 referencias

```
public static Crucero operator +(Crucero crucero, Salon salon)
{
    crucero.salones.Add(salon);
    return crucero;
}
```

2 referencias

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

    if(viaje.PasajerosABordo + 1 > viaje.crucero.CapacidadPasajeros)
    {
        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);

    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

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

2 referencias
public Pasajero(string nombreCompleto, Pasaporte pasaporte, DateTime fechaNacimiento, TipoPasajero tipoPasajero) : this(nombreCompleto)
{
    this.pasaporte = pasaporte;
    this.fechaNacimiento = fechaNacimiento;
    this.tipoPasajero = tipoPasajero;
    this.estaViajando = true;
    this.equipaje = new List<Equipaje>();
}
```

```
2 referencias
public FrmViaje()
{
    InitializeComponent();
    this.estaEditando = false;
    this.Text = "Registro de viaje";
    this.lblError.ForeColor = Color.Red;
    this.lblError.Text = "";
    this.cboCrucero.DataSource = SistemaCruceros.flota;
    this.cboDestino.DataSource = new List<Puerto>(SistemaCruceros.puertos);
    this.cboOrigen.DataSource = new List<Puerto>(SistemaCruceros.puertos);
}

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

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.

```
51 referencias
public static class SistemaCruceros
{
    public static DateTime fechaDelSistema;
    public static List<Crucero> flota;
    public static List<Venta> ventas;
    public static List<Puerto> puertos;
    public static List<Viaje> viajes;
    public static List<Viaje> historialViajes;
    public static Dictionary<Puerto, float> contadorRecaudacionDestinos;
1 referencia
```

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.

```
2 referencias
public Pasajero this[int indice]
{
    get
    {
        Pasajero pasajero = null;
        if (indice >= 0 && indice < this.pasajeros.Count)
        {
            pasajero = this.pasajeros[indice];
        }
        return pasajero;
    }
}
```

```
1 referencia
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;
}
```

```
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 hacia *Usuario* y *Pasajero*; y la segunda en los Forms, 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.Finalizado)
            {
                filaActual.DefaultCellStyle.BackColor = Color.LightSalmon;
            }
            else if (viaje.EstadoDeViaje == EstadoDeViaje.EnCurso)
            {
                filaActual.DefaultCellStyle.BackColor = Color.Yellow;
            }
            else
            {
                filaActual.DefaultCellStyle.BackColor = Color.LightGreen;
            }
            if (viaje.PasajerosABordo >= viaje.Crucero.CapacidadPasajeros)
            {
                filaActual.DefaultCellStyle.BackColor = Color.OrangeRed;
            }
        }
    }
}
```

```
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 "historial":
            formMostrado = formListadoHistorial;
            break;
        default:
            formMostrado = formListadoViajes;
            break;
    }
    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.