**Instituto Tecnológico Colonia**

**CTC**

**Evaluación: Programación II**

Entregado como requisito para la obtención del título de Analista Programador

Leonardo Fernández

Matias Erramouspe

Tutor: Alen López

Año: 2021

Contenido

[Declaración de autoría 2](#_Toc90304046)

[Lectura 2](#_Toc90304047)

[Cronograma 3](#_Toc90304048)

[Análisis 4](#_Toc90304049)

[Diagrama de clases 6](#_Toc90304050)

[Diagrama de 3 capas 7](#_Toc90304051)

[Diccionario de clases 8](#_Toc90304052)

[Documentación de la base de datos 13](#_Toc90304053)

[Pseudocódigo 13](#_Toc90304054)

[Consulta 1 13](#_Toc90304055)

[Consulta 2 14](#_Toc90304056)

[Consulta 3 14](#_Toc90304057)

[Consulta 4 15](#_Toc90304058)

[Consulta 5 17](#_Toc90304059)

[Consulta 6 18](#_Toc90304060)

[Consulta 7 19](#_Toc90304061)

[Manual de usuario y prueba del programa 20](#_Toc90304062)

[Alta de autor 20](#_Toc90304063)

[Alta lector 21](#_Toc90304064)

[Alta genero y pais 22](#_Toc90304065)

[Alta de un libro 22](#_Toc90304066)

[Realizar venta y devolución 23](#_Toc90304067)

[Bajas y Modificaciones 24](#_Toc90304068)

[Estadísticas 25](#_Toc90304069)

Declaración de autoría

Nosotros, Leonardo Fernández y Matías Erramouspe, declaramos que el trabajo que se presenta en esta obra es de nuestra propia mano. Podemos asegurar que:

* La obra fue producida en su totalidad mientras cursábamos la materia Programación 2 a cargo del Prof. Alen López, quien además nos proporcionó material;
* Los algoritmos utilizados fueron obra de los conocimientos adquiridos en clase y en base a un trabajo grupal. Además, todo nuestro desarrollo se apoya en los ejemplos trabajados durante el semestre.
* Ninguna parte de nuestroprograma se encuentra publicado en ningún sitio o plataforma, la totalidad de su contenido es propia de nuestra autoría.

Lectura

Texto, Carta

Descripción generada automáticamenteParte 1

Parte 2

Texto, Carta

Descripción generada automáticamente

Cronograma

|  |  |
| --- | --- |
| Actualizar Cronograma | 10 minutos |
| Actualizar diagrama de clases | 10 minutos |
| Actualizar Seudocódigo | 10 minutos |
| Mudar ABM de libros | 40 minutos |
| Mudar ABM de autores | 35 minutos |
| Mudar ABM de lectores | 20 minutos |
| Agregar ABM Países | 10 minutos |
| Agregar ABM Genero | 10 minutos |
| Mudar Alta y baja de ventas | 20 minutos |
| Clase controladora | 75 minutos |
| Clase libro | 10 minutos |
| Clase autor | 10 minutos |
| Clase lector | 10 minutos |
| Clase venta | 10 minutos |
| Clase País | 10 minutos |
| Clase Genero | 10 minutos |
| Formulario Inicio | 10 minutos |
| Formulario Autor y Lector | 15 minutos |
| Formulario Ventas | 10 minutos |
| Formulario Estadísticas | 20 minutos |
| Estadística 1 | 30 minutos |
| Estadística 2 | 30 minutos |
| Estadística 3 | 45 minutos |
| Estadística 4 | 35 minutos |
| Estadística 5 | 30 minutos |
| Estadística 6 | 20 minutos |
| Estadística 7 | 15 minutos |
| Testing | 20 minutos |
| Manual de usuario | 35 minutos |
| Diccionario de clases | 30 minutos |
| Documentación | 50 minutos |
| Diagrama de 3 capas | 20 minutos |
| Creación base de datos | 10 minutos |
|  |  |
| Total de horas trabajadas | 11.9 Horas |

Análisis

El problema planteado consiste en migrar a ASP.net el sistema para una feria del libro previamente desarrollado en C# con Windws Form. La solución permitirá registrar, modificar y dar de baja libros, autores, géneros, países y lectores. También registrar y dar de baja ventas. Para resolver estos puntos se decidió realizar un ABM para cada entidad a excepción de la venta que requiere alta y baja.

Al momento de realizar el lector y el autor se utiliza herencia, ya que ambos comparten atributos tales como nombre y apellido. A la hora de dar de alta a los autores se tendrá en cuenta que este puede haber fallecido. A diferencia de la entrega anterior se separarán ambos formularios para facilitar el uso del sistema.

A la hora de mostrar las estadísticas todas se visualizarán en un formulario a excepción de la lista de libros ordenada alfabéticamente que se realizará en el mismo formulario de ABM libros.

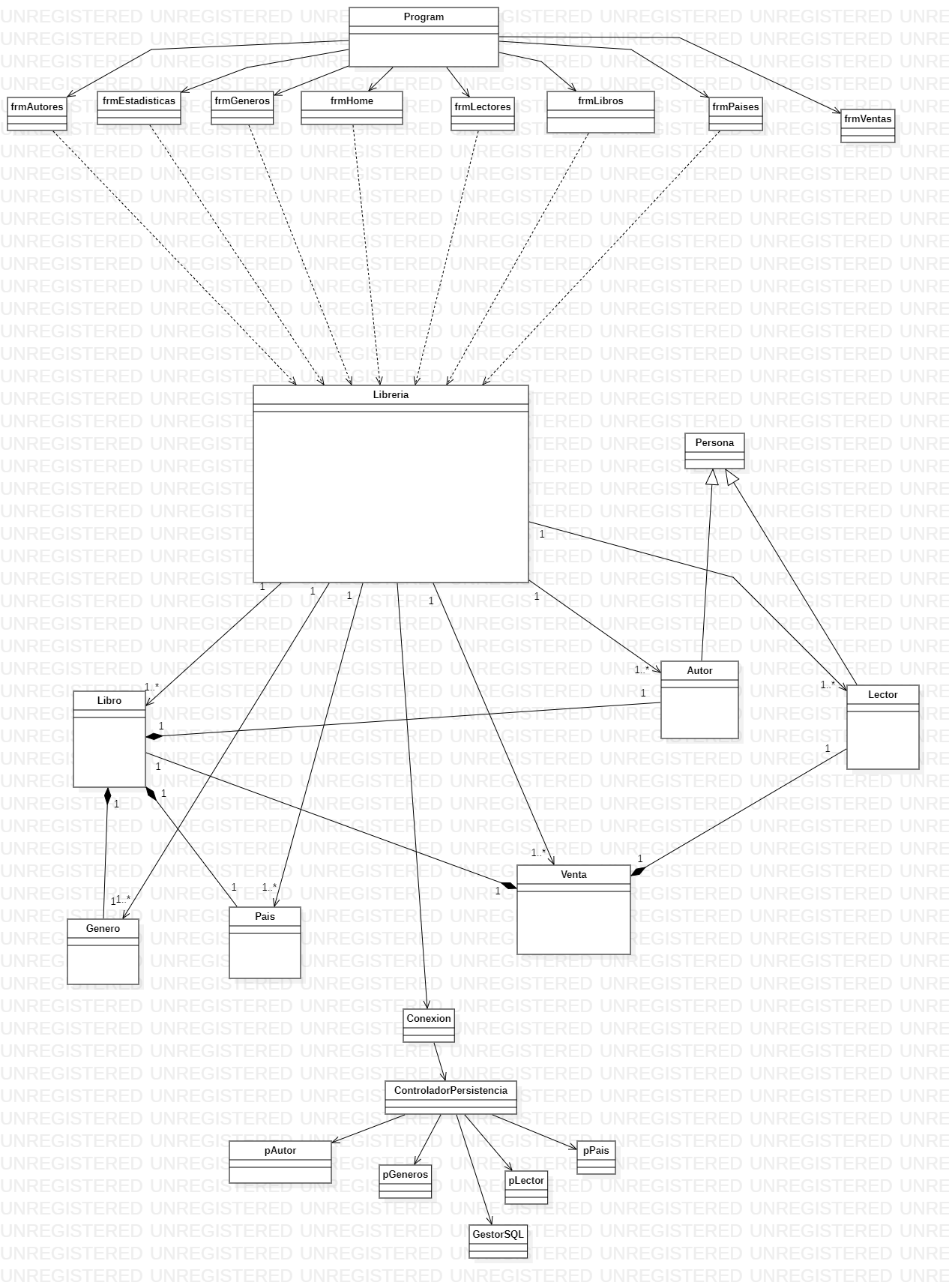
Para mostrar el/los libros más vendidos y el/los autores que más vendieron se utilizará un atributo numérico, en el cual a la hora de realizar la consulta se contaran las veces que el articulo se registra, por ejemplo, en autores se cuenta la cantidad de veces que se encuentra el autor en venta y a autor se suma 1, se compara y se da el autor que más libros vendió. De este modo resulta más eficiente hacer el calculo y mostrar esa estadística.

A demás los datos de autores, lectores, géneros y países se guardarán una base de datos utilizando SQL server.

Interfaz de usuario gráfica

Descripción generada automáticamenteDiagrama de clases

Diagrama de 3 capas



Diccionario de clases

|  |  |  |
| --- | --- | --- |
| **Nombre** | Libro | |
| **Descripción** | Administar libros | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| \_id | short | private |
| \_titulo | string | private |
| \_genero | string | private |
| \_ano | DateTime | private |
| \_autor | class Autor | private |
| \_pais | Class Pais | private |
| \_precio | Double | private |
| \_comentario | string | private |
| \_cantVend | int | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Persona | |
| **Descripción** | Clase base | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| \_id | int | private |
| \_nombre | string | private |
| \_apellido | string | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Persona: Autor | |
| **Descripción** | Administra autores heredando los atributos de la clase Persona | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| \_fechaDeN | String | private |
| \_fechaDeF | String | private |
| \_nacionalidad | string | private |
| \_librosVend | int | private |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | Persona: Lector | | |
| **Descripción** | Administra lectores heredando los atributos de la clase Persona | | |
|  | | | |
| **Nombre** | **Tipo de dato** |  | **Visibilidad** |

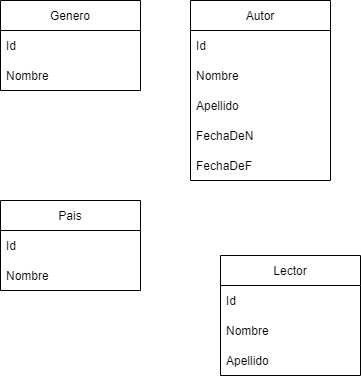
|  |  |  |
| --- | --- | --- |
| **Nombre** | País | |
| **Descripción** | Administrar países | |
| **Atributos** | | |
|  | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| \_id | int | private |
| \_nombre | string | private |
| \_cantvend | int | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Genero | |
| **Descripción** | Administrar Géneros | |
| **Atributos** | | |
|  | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| \_id | int | private |
| \_nombre | string | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Venta | |
| **Descripción** | Administrar ventas | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| \_id | short | private |
| \_FechaVen | DateTime | private |
| \_libro | class Libro | private |
| \_lector | class Lector | private |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | Municipalidad | | |
| **Descripción** | Clase controladora | | |
| **Atributos** | | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** | |
| \_listaVentas | List<Venta> | Private static | |
| \_listaLibros | List<Libro> | Private static | |
| \_listaLectores | List<Lector> | Private static | |
| \_listaAutores | List<Autor> | Private static | |
| \_listaGeneros | List<Genero> | private static | |
| \_listaPaises | List<Paises | Private static | |
| \_persistencia | ControladorPersistencia | Private static | |
| **Metodos** | | | |
| **Nombre** | **Tipo de dato** | **Descripción** | **Visibilidad** |
| ListaVentas | List<Venta> | Devuelve el listado de ventas | Public |
| ListaLibros | List<Libro> | Devuelve el listado de libros | Public |
| ListaLectores | List<Lector> | Devuelve el listado de lectores | Public |
| ListaAutores | List<Autor> | Devuelve el listado de autores | Public |
| ListaGeneros | List<Genero> | Devuelve el listado de géneros | Public |
| ListaPaises | List<Pais> | Devuelve el listado de países | Public |
| Municipalidad | List<> | Actualiza las listas de datos con la listas guardadas en SQL | Public |
| BuscarVenta | Venta | Devuelve la venta en caso de que exista, sino retorna null | Public |
| Altavent | bool | Si la venta no existe se agrega la venta a la lista | Public |
| Bajavent | bool | Si la venta existe se elimina de la lista | Public |
| BuscarLibro | Libro | Devuelve el libro en caso de que exista, sino retorna null | Public |
| AltaLibro | bool | Si el libro no existe se agrega el libro a la lista | Public |
| BajaLibro | bool | Si el libro existe se elimina de la lista | Public |
| ModificarLibro | bool | Si el libro existe se reemplaza todos los atributos con los nuevos | Public |
| ListaOrdenadaLibro | List<Libro> | Devuelve los libros ordenados por nombre alfabéticamente | Public |
| BuscarLector | Lector | Devuelve el lector en caso de que exista, sino retorna null | Public |
| AltaLector | bool | Si el lector no existe se agrega el lector a la lista | Public |
| BajaLector | bool | Si el lector existe se elimina de la lista | Public |
| ModificarLector | bool | Si el lector existe se reemplaza todos los atributos con los nuevos | Public |
| BuscarAutor | Autor | Devuelve el autor en caso de que exista, sino retorna null | Public |
| AltaAutor | bool | Si el autor no existe se agrega el autor a la lista | Public |
| BajaAutor | bool | Si el autor existe se elimina de la lista | Public |
| ModificarAutor | bool | Si el autor existe se reemplaza todos los atributos con los nuevos | Public |
| BuscarGenero | Genero | Devuelve el género en caso de que exista, sino retorna null | Public |
| AltaGenero | bool | Si el género no existe se agrega el género a la lista | Public |
| BajaGenero | bool | Si el género existe se elimina de la lista | Public |
| ModificarGenero | Bool | Si el género existe se reemplaza todos los atributos con los nuevos | Public |
| BuscarPais | Pais | Devuelve el país en caso de que existe, sino retorna null | Public |
| AltaPais | bool | Si el país no existe se agrega el país a la lista | Public |
| BajaPais | bool | Si el autor existe se elimina de la lista | Public |
| ModificarPais | bool | Si el país existe se reemplaza todos los atributos con los nuevos | Public |
| Recuadacion | double | Devuelve la suma de todas las ventas | Public |
| IngresoCant | bool | Suma una unidad a determinado autor por cada vez que este se encuentre en la lista de ventas | Public |
| AutorConMasLibrosVend | List<Autor> | Retorna el autor que vendió más libros | Public |
| IngresoCantLib | bool | Suma una unidad a determinado libro por cada vez que este se encuentre en la lista de ventas | Public |
| LibroConMasVent | List<Libro> | Retorna el libro que más se vendió | Public |
| LectorDadoNom | List<Venta> | Retorna la lista de ventas que realizo un lector | Public |
| BuscarLectorNom | Lector | Retorna un lector si dado un nombre | Public |
| IngresoCantPais | bool | Suma una unidad a determinado país por cada vez que este se encuentre en la lista de ventas | Public |
| PaisConMasPub | List<Pais> | Retorna el país que más publicación tiene | Public |
| LibroDadoXGenero | List<Libro> | Retorna una Libro/s usando de parámetro un genero | Public |
| BuscarGeneroNom | Genero | Retorna un género dado un nombre | Public |

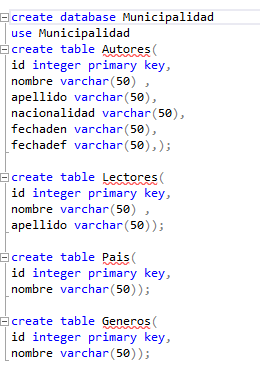
Documentación de la base de datos



Genero(**Id**, Nombre)

Pais(**Id,** Nombre)

Lector(**Id**, Nombre, Apellido)

Autor(**Id**, Nombre, Apellido, FechaDeN FechaDeF)

Pseudocódigo

## Consulta 1

Recaudación total

Inicio

Variables: unVenta = Venta resultado =0;

Por cada (Venta Unventa en \_listaVentas)

Resultado = restultado + unVenta.Libro.Precio;

Fin Por Cada

Devolver Resultado;

Fin

## Consulta 2

Ordenar Libro por titulo

Inicio

Variables: LibrosOrdenados = \_listaLibros auxLibro =”” int I,J

para (int i = 0; i < LibrosOrdenados.Count; i++)

para (int j = 0; j < LibrosOrdenados.Count - 1; j++)

si(LibrosOrdenados[j].Titulo.ToUpper().CompareTo(LibrosOrdenados[j + 1].Titulo.ToUpper()) > 0)

auxLibro = LibrosOrdenados[j];

LibrosOrdenados[j] = LibrosOrdenados[j + 1];

LibrosOrdenados[j + 1] = auxLibro;

finSi

FinPara

FinPara

Devolver LibrosOrdenados

Fin

## Consulta 3

Ingresa cantidad de libros vendidos dado un autor

Inicio

Variables unaVenta =\_listaVentas unAutor = \_listaAutores

Porcada (Venta unaVenta in \_listaVentas)

PorCada(Autor unAutor in \_listaAutores)

Si((unAutor.Id == unaVenta.Libro.Autor.Id)

unAutor.LibrosVend++;

fin si

fin Porcada

fin Porcada

Fin

Autor con mas libros vendidos

Inicio

Variables: AutorVentas = \_listaAutores mayor=0 Autror = unAutor;

Por cada (Autor unAutor in \_listaAutores)

Si(unAutor.LibrosVend > 0 && unAutor.LibrosVend > mayor)

AutorVentas.Clear();

AutorVentas.Add(unAutor);

mayor = unAutor.LibrosVend;

Sino (unAutor.LibrosVend> 0 && unAutor.LibrosVend ==mayor)

AutorVentas.Add(unAutor);

mayor = unAutor.LibrosVend;

FinSI

FinParaCada

Devolver AutorVentas

Fin

## Consulta 4

Ingresa cantidad de libros vendidos

Inicio

Variables unaVenta =\_listaVentas unLibro = \_listaLibros

Porcada (Venta unaVenta in \_listaVentas)

PorCada (Libro unLibro in \_listaLibros)

Si (unLibro.Id == unaVenta.Libro.Id)

unLibro.CantVend++;

fin si

fin Porcada

fin Porcada

Fin

Libros Mas Vendidos

Inicio

Variables: LibroVentas =\_listaLibros mayor =0 Libro = unLibro;

PorCada (Libro unLibro in \_listaLibros)

SI (unLibro.CantVend > 0 && unLibro.CantVend > mayor)

LibroVentas.Clear();

LibroVentas.Add(unLibro);

mayor = unLibro.CantVend;

SiNo (unLibro.CantVend > 0 && unLibro.CantVend == mayor)

LibroVentas.Add(unLibro);

mayor = unLibro.CantVend;

FinSi

FinaParaCada

Devoler LibroVentas;

Fin

## Consulta 5

Lector Dado Nombre

Inicio

Variables = unLecNom = this.BuscarLectorNom(pNombre) LectVen = new List<Venta> unVenta = \_listaVentas

Porcada (Venta unaVenta in \_listaVentas)

Leo unLecNom y Ventas

Si(unLecNom != null)

Si (unaVenta.Lector.Nombre == unLecNom.Nombre)

LectVen.Add(unaVenta);

FinSi

FinSI

FinPorCada

Devolver LectVen

Fin

Buscar Lector Dado nombre

Inicio

Variables = Lector = lector String pNombre lector = \_listaLectores

porcada (Lector lector in \_listaLectores)

Si (lector.Nombre.== pNombre)

Devolver lector;

Finsi

FinPorCada

ReturnNull

Fin

## Consulta 6

Ingresa cantidad de libros publicados dado país

Inicio

Variables unLibro =\_listaLibros unPais = \_listaPaises

Porcada (Libro unLibro in \_listaLibros)

PorCada (Pais unPais in \_listaPaises)

Si (unPais.Id == unLibro.Pais.Id)

unPais.Cantvend++

finSI

fin Porcada

fin Porcada

Fin

País con más Libros publicados

Inicio

Variables: PaisVentas = new List<Pais> (\_listaPaises) mayor=0 unPais = \_listaPaises

Por cada (Autor unAutor in \_listaAutores)

Si (unPais.Cantvend > 0 && unPais.Cantvend > mayor)

PaisVentas.Clear();

PaisVentas.Add(unPais);

mayor = unPais.Cantvend;

Sino (unPais.Cantvend > 0 && unPais.Cantvend == mayor)

PaisVentas.Add(unPais);

mayor = unPais.Cantvend;FinSI

FinParaCada

Devolver PaisVentas

Fin

## Consulta 7

Inicio

Libro Dado un Genero

Variables unGenero= BuscarGeneroNom(pGenero) LibroDadoGen = new List<Libro> unLibro = ListaOrdenadaLibro()

Porcada (Libro unLibro in ListaOrdenadaLibro())

Si(unGenero != null)

Si (unGenero.Nombre == unLibro.Genero.Nombre)

LibroDadoGen.Add(unLibro);

finSi

finSi

FinPorcada

Devolver LibroDadoGen  
Fin

Buscar Genero Nombre

Inicio

Variables: unGenero = \_listaGeneros

Porcada (Genero unGenero in \_listaGeneros)

Si (unGenero.Nombre == pNombre)

return unGenero;

finSi

FinPorCada

return null;

Fin

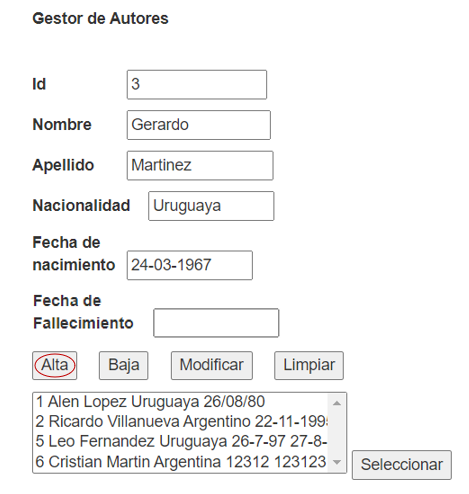
Manual de usuario y prueba del programa

Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

Descripción generada automáticamente

Al iniciar el sistema se mostrará una página con este encabezado. Desde esta barra de navegación se accede a todo el sistema haciendo clic sobre la pestaña a la que desea ingresar.

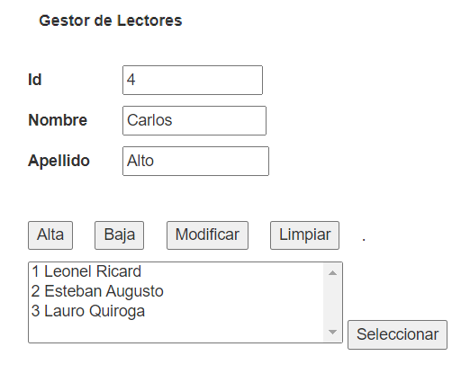
## Alta de autor



Una vez ingresada a la pestaña “ABM Autores” se deberá completar todos los campos requeridos, el único opcional es Fecha de fallecimiento

Una vez rellenado los campos hacer clic en el botón “Alta”. En caso de que uno o más campos estén sin completar se mostrará un mensaje “Faltan datos”. De igual manera se alertará si se ingresa un “id” existente.

## Alta lector



Una vez ingresado en el gestor de lectores se procede a completar los campos requeridos para la creación del lector (Id, nombre y apellido). Posteriormente hacer clic en el botón “Alta”.

## Alta genero y pais

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Para realizar el alta de un genero es necesario indicar su Id y el nombre del género, una vez ingresados los datos clic en el botón “Alta.

En la siguiente imagen se muestra el alta de un País, al igual que en el género, se ingresa el Id y el nombre correspondiente. Acto seguido presionar el botón “Alta”.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Alta de un libro

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Para realizar el alta se deberán completar todos los campos con excepción del campo “Comentario” que es opcional.

El género, autor y país deben seleccionarse mediante un clic de la lista que sucede al título del dato.

Los libros ingresados se pueden ordenar por orden de ingreso (por defecto) o por orden alfabético. Esto se puede seleccionar en la parte superior de la lista donde se muestran los mismos.

# Realizar venta y devolución

Interfaz de usuario gráfica

Descripción generada automáticamente

Para realizar una venta se deben rellenar los campos de “id”, “Fecha de venta” que por defecto se encuentra al día de hoy. Se selecciona el libro y luego el “Lector” que va a realizar la compra.

# Bajas y Modificaciones

Para realizar una baja o una modificación de cualquier ABM se debe seleccionar de la lista que se muestra en la parte inferior (en el ejemplo se muestra la lista de libro). Una vez seleccionado con el clic se procede a presionar el botón “Seleccionar” ubicado a la derecha del listado. Observará que los campos se autocompletan permitiendo modificar los datos que se deseen presionando el botón “Modificar”.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Para realizar una baja de igual forma que en la modificación se selecciona la línea que se desea eliminar de la base de datos y se presiona el botón “Seleccionar”. Una vez se autocompleten los campos presionar el botón “Baja”.

# Estadísticas

La pestaña estadísticas nos muestra datos útiles sobre el sistema.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Como se muestra en la imagen de arriba al presionar el botón “Consulta 1” muestra el total de dinero recaudado con la venta de libros.

Al cliquear “Consulta 3” se autocompleta la lista con los autores que vendieron más libros

Texto, Aplicación

Descripción generada automáticamente

La consulta 4 nos muestra el listado de libros que mayor cantidad se han vendido.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

En la estadística “5” se deberá escribir el nombre de un Lector, si este no concuerda con ninguno o no se ha rellenado ese campo se alertará en la correspondiente.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

La estadística 6 nos muestra el nombre de los países que mas publicaciones han presentado en la feria.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Por último, la consulta 7 presenta un buscador de libros por género. En la caja de texto que se muestra a la derecha del botón se escribe el género que se desea listar y al presionar el botón “Consulta 7” se despliegan en la lista.