

**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**(UNI)**



**Materia: Matemática para computación II**

**Ing. En computación.**

**Tema:** Arreglos ordenados

Laboratorio #2

**Integrantes:**

Erick Hazael Morales Jarquin 2023-0862U

Denis Josué Wong S. 2019-0919U

**Grupo:** 2M1-CO.

Programa, Registro de Feligreses hecho por: Erick Morales.

Programa, Registro de Consultas hecho por: Denis Wong.



## Laboratorio #2: Registro de Feligreses y sus diezmos en un rango de meses

### 1-Clase objeto de tipo feligrense y sus atributos

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace AEDLaboratorio2.Entities.Church
8  {
9      15 referencias
10     public class Parishioner
11     {
12         11 referencias
13         public int ID { get; set; }
14         6 referencias
15         public string? Name { get; set; }
16         3 referencias
17         public string? Address { get; set; }
18         3 referencias
19         public string? PhoneNumber { get; set; }
20         3 referencias
21         public string? PeriodStart { get; set; }
22         3 referencias
23         public string? PeriodEnd { get; set; }
24         5 referencias
25         public decimal Amount { get; set; }
26         3 referencias
27         public DateTime DateRegister { get; set; }
28     }
29 }
30
31 /*
32  * Entidades en la cual estaremos trabajando durante el manejo del arreglo de objetos
33  */
```

2 referencias

```
public class Validations
{
```

```
    private ErrorProvider errorProvider = new ErrorProvider();
```

18 referencias

```
    public bool TextNullOrEmpty(Control control)
    {
```

```
        errorProvider.SetError(control, "");
```

```
        if (control is ComboBox comboBox)
        {
```

```
            if (comboBox.SelectedIndex == -1)
```

```
            {
```

```
                errorProvider.SetError(comboBox, "Cannot be null or empty");
```

```
                return false;
```

```
            }
```

```
        }
```

```
        if (control is TextBox textBox)
```

```
        {
```

```
            if (string.IsNullOrEmpty(textBox.Text))
```

```
            {
```

```
                errorProvider.SetError(textBox, "Cannot be null or empty");
```

```
                return false;
```

```
            }
```

```
        }
```

```
        if (control is MaskedTextBox maskedTextBox)
```

```
        {
```

```
            if (!maskedTextBox.MaskFull)
```

```
            {
```

```
                errorProvider.SetError(maskedTextBox, "Cannot be null or empty");
```

```
                return false;
```

```
            }
```

```
        }
```

```
        return true;
```

```
    }
```

```
}
```

## 2-En formulario de ParishionerView se crean los metodos a utilizar

```
public partial class ParishionerView : Form
{
    private readonly Validations validations;
    private Parishioner? _parishioner;
    /*
     * Instancia unica de la clase validations para aplicar el metodo situado.
     * Instancia del objeto que guardaremos en el arreglo
     */
    1 referencia
    public ParishionerView()
    {
        InitializeComponent();
        validations = new Validations();
        _parishioner = new Parishioner();
        //Pasamos las instancias en el constructor para su uso
    }

    int autoIncrementID = 0; //variable autoincrementable para tener el conteo de registros y mostrarlos en el datagridview y su previa busueda
    4 referencias
    private void PrintParishioners()
    {
        dgPrintParishioner.DataSource = null;
        dgPrintParishioner.DataSource = ParishionerModel.arrayLogic.GetParisioner(); //metodo para actualizar el datagriview en funcion del arreglo
    }

    1 referencia
    private void SaveParishioner()
    {
        _parishioner = new Parishioner
        {
            ID = ++autoIncrementID,
            Name = TxtName.Text,
            Address = TxtAddress.Text,
            PhoneNumber = TxtMaskPhone.Text,
            Amount = decimal.Parse(textCant.Text),
            PeriodStart = CmbPStart.Text,
            PeriodEnd = cmbPEnd.Text,
            DateRegister = DateTime.Now
        };
        ParishionerModel.arrayLogic.AddParishioner(_parishioner); //metodo para guardar a los feligreses inicializando un nuevo feligrense y usadndo el metodo en models
    }
    1 referencia
}
```

```
1 referencia
private void UpdateParishioner(Parishioner parishioner)
{
    var _parishionerUpdate = new Parishioner
    {
        ID = parishioner.ID,
        Name = TxtName.Text,
        Address = TxtAddress.Text,
        PhoneNumber = TxtMaskPhone.Text,
        Amount = decimal.Parse(textCant.Text),
        PeriodStart = CmbPStart.Text,
        PeriodEnd = cmbPEnd.Text,
        DateRegister = DateTime.Now
    };
    ParishionerModel.arrayLogic.UpdateParishioner(_parishionerUpdate);
    /*
     * Similar al metodo guardar pero este recibe el mismo id del usuario seleccionado y por medio del uso models
     * se busca en el arreglo el indice con ese id y se actualiza la informacion
     */
}

2 referencias
public void ValidationsTools()
{
    bool isValid = true;

    isValid &= validations.TextNullOrEmpty(TxtAddress);
    isValid &= validations.TextNullOrEmpty(TxtName);
    isValid &= validations.TextNullOrEmpty(TxtMaskPhone);
    isValid &= validations.TextNullOrEmpty(textCant);
    isValid &= validations.TextNullOrEmpty(CmbPStart);
    isValid &= validations.TextNullOrEmpty(cmbPEnd);

    if (!isValid)
    {
        MessageBox.Show("Information null or empty", "Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

/*
 * Validacion para los campos de texto nulos o invalidos, &= si el metodo retorna false, el booleano isValid se vuelve false y salta la alerta
 */
4 referencias
```

### 3-Crear eventos de botones y formulario para el funcionamiento de la aplicacion

```
*/  
4 referencias  
private void CleanBox()  
{  
    TxtName.Text = textCant.Text = TxtMaskPhone.Text = TxtAddress.Text = string.Empty;  
    CmbPStart.SelectedIndex = -1;  
    cmbPEnd.SelectedIndex = -1; //funcion para limpiar los campos de texto  
}  
1 referencia  
private void ParishionerView_Load(object sender, EventArgs e)  
{  
    PrintParishioners(); //uso de una funcion  
}  
  
1 referencia  
private void BtnSave_Click(object sender, EventArgs e)  
{  
    ValidationsTools(); //aplicacion del metodo validationstools y si uno falla no permite avanzar  
  
    if (!validations.TextNullOrEmpty(TxtAddress) || !validations.TextNullOrEmpty(TxtName) ||  
        !validations.TextNullOrEmpty(TxtMaskPhone) || !validations.TextNullOrEmpty(textCant) ||  
        !validations.TextNullOrEmpty(CmbPStart) || !validations.TextNullOrEmpty(cmbPEnd))  
    {  
        return; // No proceder si hay errores de validación  
    }  
  
    SaveParishioner();  
    PrintParishioners(); //uso de funciones  
    CleanBox();  
}
```

```
1 referencia  
private void BtnClearForm_Click(object sender, EventArgs e)  
{  
    CleanBox(); //uso de funciones  
}  
  
1 referencia  
private void BtnPrevious_Click(object sender, EventArgs e)  
{  
  
    if (dgPrintParishioner.CurrentRow == null) return;  
  
    var currentIndex = dgPrintParishioner.CurrentRow.Index;  
  
    if (currentIndex > 0)  
    {  
        dgPrintParishioner.ClearSelection();  
  
        dgPrintParishioner.Rows[currentIndex - 1].Selected = true;  
  
        dgPrintParishioner.CurrentCell = dgPrintParishioner.Rows[currentIndex - 1].Cells[0];  
    }  
}  
/*  
* Verifica si el indice es nulo y si el indice actual es mayor a 0, si es mayor a 0 retrocede una celda en el data gridview  
*/
```

```

1 referencia
private void BtnDelete_Click(object sender, EventArgs e)
{
    if (dgPrintParishioner.SelectedRows.Count == 0)
    {
        MessageBox.Show("No row selected. Please select a row to update.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    var parishionerSelected = dgPrintParishioner.SelectedRows[0].DataBoundItem as Parishioner;
    if (parishionerSelected == null)
    {
        MessageBox.Show("Option selected doesn't exist or is invalid.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    var dlgTest = MessageBox.Show("Are you sure delete: " + parishionerSelected.Name + " with ID : " + parishionerSelected.ID, "Error", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (dlgTest == DialogResult.No) return;
    var proveDelete = ParishionerModel.arrayLogic.DeleteParishioner(parishionerSelected.ID);
    if (!proveDelete)
    {
        MessageBox.Show("Cannot delete parishioner : " + parishionerSelected.Name + " with ID : " + parishionerSelected.ID + "?", "Alert", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
    }

    PrintParishioners();
    CleanBox();
    btnSaveUpdate.Visible = false;
    btnSave.Visible = true;
}

/*
 * Si no se ha seleccionado una celda muestra el mensaje de error, si no se obtiene la informacion de esa celda de tipo objeto pariishioner respectivamente se usa su id para buscar en el arreglo el objeto con ese id
 * y eliminarlo
 */
1 referencia
private void BtnShowAll_Click(object sender, EventArgs e)
{
    PrintParishioners();//uso de funciones
}

```

```

1 referencia
private void BtnNext_Click(object sender, EventArgs e)
{
    if (dgPrintParishioner.CurrentRow == null) return;
    var currentIndex = dgPrintParishioner.CurrentRow.Index;
    if (currentIndex < dgPrintParishioner.Rows.Count - 1)
    {
        dgPrintParishioner.ClearSelection();
        dgPrintParishioner.Rows[currentIndex + 1].Selected = true;
        dgPrintParishioner.CurrentCell = dgPrintParishioner.Rows[currentIndex + 1].Cells[0];
    }
}

/*
 * se repite el mismo proceso pero esta vez se verifica si el indice es menor a la cantidad de celdas en el dg, si esta
 * hasta el tope no avanza, se caso contrario avanza una celda
 */
1 referencia

```

1 referencia

```
private void btnSaveUpdate_Click(object sender, EventArgs e)
{
    if (dgPrintParishioner.SelectedRows.Count == 0)
    {
        MessageBox.Show("No row selected. Please select a row to update.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    var parishionerSelected = dgPrintParishioner.SelectedRows[0].DataBoundItem as Parishioner;
    if (parishionerSelected == null)
    {
        MessageBox.Show("Option selected doesn't exist or is invalid.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    BtnPrevious.Enabled = true;
    BtnNext.Enabled = true;
    btnSaveUpdate.Visible = false;
    BtnSave.Visible = true;
    ValidationsTools();

    if (!validations.TextNullOrEmpty(TxtAddress) || !validations.TextNullOrEmpty(TxtName) ||
        !validations.TextNullOrEmpty(TxtMaskPhone) || !validations.TextNullOrEmpty(textCant) ||
        !validations.TextNullOrEmpty(CmbPStart) || !validations.TextNullOrEmpty(cmbPEnd))
    {
        return; // No proceder si hay errores de validación
    }
    UpdateParishioner(parishionerSelected);
    CleanBox();
    MessageBox.Show("Parishioner update sucessfully", "Success", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}

/*
 * Se vuelve a validar si la celda seleccionada es valida y se manda a llamar el metodo actualizar, ahora con la informacion actualizada en los campos de text,
 * (Antes se habian rellenado los campos de texto con los datos del feligrense ubicado por su id para su respectiva edicion) se envia el id del feligrense y el metodo hace su accion
 */
```

1 referencia

```
private void BtnUpdate_Click(object sender, EventArgs e)
{
    if (dgPrintParishioner.SelectedRows.Count == 0)
    {
        MessageBox.Show("No row selected. Please select a row to update.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    var parishionerSelected = dgPrintParishioner.SelectedRows[0].DataBoundItem as Parishioner;
    if (parishionerSelected == null)
    {
        MessageBox.Show("Option selected doesn't exist or is invalid.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    var dlgTest = MessageBox.Show($"Are you sure you want to update: {parishionerSelected.Name} with ID: {parishionerSelected.ID}?", "Alert", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (dlgTest == DialogResult.No) return;

    TxtName.Text = parishionerSelected.Name;
    TxtAddress.Text = parishionerSelected.Address;
    TxtMaskPhone.Text = parishionerSelected.PhoneNumber;
    dateTimeRegister.Value = parishionerSelected.DateRegister;
    CmbPStart.Text = parishionerSelected.PeriodStart;
    cmbPEnd.Text = parishionerSelected.PeriodEnd;
    textCant.Text = parishionerSelected.Amount.ToString();

    BtnPrevious.Enabled = false;
    BtnNext.Enabled = false;
    btnSaveUpdate.Visible = true;
    BtnSave.Visible = false;
}

/*
 * Se hace el mismo proceso que Delete pero cuando se encuentra al objeto con el id , se actualiza los campos de texto con la infromacion de ese usuario y se bloquean los botones
 * de avanzar y retroceder para evitar editar otro feligrense
 */
```



```

1 referencia
private void BtnSearchById_Click(object sender, EventArgs e)
{
    int idParishioner = (int)TxtNumSearchID.Value;
    var specificParishioner = ParishionerModel.arrayLogic.SearchParishioner(idParishioner);
    if (specificParishioner == null)
    {
        MessageBox.Show("ID Doesn't Exist", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    dgPrintParishioner.DataSource = new Parishioner[] { specificParishioner };
}
/*
 * Se convierte a entero el valor recibido de numericuptodown debido que se requiere buscar el feligrense con ese id e imprimirlo en el datagridview, si no se
 * encuentra se envia un mensaje de error
 */
1 referencia
private void TxtCant_KeyPress(object sender, KeyPressEventArgs e)
{
    char index = e.KeyChar;
    if (index == 8)
    {
        e.Handled = false; //permite backspace
        return;
    }
    if (!char.IsDigit(index))
    {
        e.Handled = true; //Permite solo numeros
        return;
    }
}

1 referencia
private void TxtName_KeyPress(object sender, KeyPressEventArgs e)
{
    char index = e.KeyChar;
    if (index == 8)
    {
        e.Handled = false; //permite backspace
        return;
    }
    if (char.IsDigit(index))
    {
        e.Handled = true; //permite solo letras
        return;
    }
}

```

## 4-Creacion de la carpeta models con los procesos para guardar, listar, eliminar y actualizar en el orden de mayor monto de diezmo a menor monto de diezmo.

```
1 referencia
public Parishioner[] GetParishioner() { return parishionerArray; } //devuelve el arreglo de objetos para su visualizacion

1 referencia
public bool DeleteParishioner(int id) //usando el id se ubica al objeto que se desea eliminar
{
    var index = Array.FindIndex(parishionerArray, p => p.ID.Equals(id) == true);
    if (index < 0) return false;

    for (int i = index; i < Quantity - 1; i++)
    {
        parishionerArray[i] = parishionerArray[i + 1]; //desplaza todos los objetos hacia la izquierda menos al deseado dejandolo de ultimo a la derecha y se disminuye el tamaño para desechar el objeto eliminado
    }
    --Quantity;
    Array.Resize(ref parishionerArray, Quantity);
    return true;
}

1 referencia
public Parishioner? SearchParishioner(int id)
{
    return Array.Find(parishionerArray, p => p.ID.Equals(id)); //metodo para filtrar un feligrense por su id
}

1 referencia
public bool UpdateParishioner(Parishioner? parishioner)
{
    if (parishioner == null) return false;
    var index = Array.FindIndex(parishionerArray, p => p.ID.Equals(parishioner?.ID) == true); //Recibe el nuevo objeto con la informacion actualizada y se guarda en la misma posicion
    if (index < 0) return false;
    parishionerArray[index] = parishioner;
    return true;
}

5 referencias
public static readonly ParishionerModel InstanceParishioner = new(); //Se crea una instancia unica, mas que todo para evitar errores a futuro y asegurar un mejor manejo de la clase
public static ParishionerModel arrayLogic { get { return InstanceParishioner; } } //se vuelve publica la instancia
public Parishioner[] parishionerArray; //arreglo donde se guardaran los objetos
int Size = 0, Quantity = 0; //Cantidad de registros y tamaño del arreglo

1 referencia
public ParishionerModel()
{
    parishionerArray = new Parishioner[Size]; //se le asigna el tamaño al arreglo
}

1 referencia
public void AddParishioner(Parishioner parishioner) //Se verifica si el tamaño es menor a la cantidad de registro, de caso contrario se redimensiona el arreglo y se almeneca el objeto y la cantidad aumenta
{
    if (Size <= Quantity)
    {
        Size = Size == 0 ? 1 : Size + 1;
        Array.Resize(ref parishionerArray, Size);
    }
    parishionerArray[Quantity] = parishioner;
    Array.Sort(parishionerArray, (f, s) => SecondParishion?.Amount.CompareTo(firstParishion?.Amount) ?? 0);
    //compara los índices del arreglo (parámetro Parishioner parishioner)
    MessageBox.Show("Parishioner sa " + parishioner + " no es NULL aquí" + "=> SecondParishion?.Amount.CompareTo(firstParishion?.Amount) ?? 0);
    //si el primer es mayor se mueve hacia arriba de caso contrario se mueva hacia abajo, si son iguales no hace su accion y sigue comparando otros índices
    ++Quantity;
}
```



## 5-Diseño de la aplicaicion con validaciones de entrada.

ParishionerView

## San Cristobal's Parishioners

### Parishioner Information

Name:

Erick Morales

Address:

San Judas

Number:

2131-4234

### Tithe Information

Amount:

C\$ 322

Months:

January

August

DateRegister:

Sunday, 22 de septiembre de 2024

Save

Clean

Success

Parishioner save successfully

Aceptar

Search By ID:

1

Search

ID	Name	Address	PhoneNumber	PeriodStart	PeriodEnd	Amount	DateRegister
----	------	---------	-------------	-------------	-----------	--------	--------------

Edit

Delete

Show Details

Back

Next

ParishionerView

## San Cristobal's Parishioners

### Parishioner Information

Name:

Address:

Number: #####-####

### Tithe Information

Amount: C\$

Months:

Date:

+ Save - Clean

Search By ID:

ID	Name	Address	PhoneNumber	PeriodStart	PeriodEnd	Amount	DateRegister
▶ 1	Erick Morales	San Judas	2131-4234	January	August	322	22/9/2024 13:...

Edit

Delete

Show Details

◀ Back

Next ▶

## 7-Usuario guardado si el proceso es valido





**San Cristobal's Parishioners****Parishioner Information****Name:** **Address:** **Number:** #####-#### **Tithe Information****Amount:** C\$ **Months:**  **Date:** domingo, 22 de septiembre de 2024 Search By ID: 

	ID	Name	Address	PhoneNumber	PeriodStart	PeriodEnd	Amount	DateRegister
▶	2	Bridget Stewart	San Judas	4324-3243	January	December	2000	22/9/2024 13:...





arishionerView

### San Cristobal's Parishioners

#### Parishioner Information

**Name:**

**Address:**

**Number:** #####-####

#### Tithe Information

**Amount:** C\$

**Months:**

septiembre de 2024

Save

Clean

Alert

Are you sure you want to update: Bridget Stewart with ID: 2?

Sí

No

Search By ID: 2

	ID	Name	Address	PhoneNumber	PeriodStart	PeriodEnd	Amount	DateRegister
▶	2	Bridget Stewart	San Judas	4324-3243	January	December	2000	22/9/2024 13:...
	1	Erick Morales	San Judas	2131-4234	January	August	322	22/9/2024 13:...
	4	Denis Wong	Dmitrov	2343-2434	January	March	300	22/9/2024 13:...
	5	Luis Largaespa...	Jinotega	3243-4324	April	May	200	22/9/2024 13:...
	3	Juan Gutierrez	Camilo Ortega	2332-3353	January	March	30	22/9/2024 13:...

Edit

Delete

Show Details

← Back

Next →

9-Actualizar seleccionando la celda

ParishionerView

### San Cristobal's Parishioners

#### Parishioner Information

**Name:**

**Adress:**

**Number:**

#### Tithe Information

**Amount:** **C\$**

**Months:**

**Date:**

Search By ID:

	ID	Name	Address	PhoneNumber	PeriodStart	PeriodEnd	Amount	DateRegister
▶	2	Bridget Stewart	San Judas	4324-3243	January	December	2000	22/9/2024 13:...
	1	Erick Morales	San Judas	2131-4234	January	August	322	22/9/2024 13:...
	4	Denis Wong	Dmitrov	2343-2434	January	March	300	22/9/2024 13:...
	5	Luis Largaespa...	Jinotega	3243-4324	April	May	200	22/9/2024 13:...
	3	Juan Gutierrez	Camilo Ortega	2332-3353	January	March	30	22/9/2024 13:...

**10-Se rellenan los campos con la información del feligrese seleccionado**



ParishionerView

### San Cristobal's Parishioners

#### Parishioner Information

**Name:**

**Adress:**

**Number:** #####-####

#### Tithe Information

**Amount: C\$**

**Months:**

2 de septiembre de 2024

Error

Are you sure delete: Luis Largaespada with ID : 5

SíNo

Search By ID: 2

	ID	Name	Address	PhoneNumber	PeriodStart	PeriodEnd	Amount	DateRegister
	2	Bridget Stewart	San Judas	4324-3243	January	December	1000	22/9/2024 13:...
	1	Erick Morales	San Judas	2131-4234	January	August	322	22/9/2024 13:...
	4	Denis Wong	Dmitrov	2343-2434	January	March	300	22/9/2024 13:...
▶	5	Luis Largaespa...	Jinotega	3243-4324	April	May	200	22/9/2024 13:...
	3	Juan Gutierrez	Camilo Ortega	2332-3353	January	March	30	22/9/2024 13:...

12-Se elimina el feligrés seleccionado

## San Cristobal's Parishioners

### Parishioner Information

**Name:**

**Address:**

**Number:** #####-####

### Tithe Information


**Amount:** C\$

**Months:**

**Date:**

 Save

 Clean

Search By ID:   Search

	ID	Name	Address	PhoneNumber	PeriodStart	PeriodEnd	Amount	DateRegister
▶	2	Bridget Stewart	San Judas	4324-3243	January	December	1000	22/9/2024 13:...
	1	Erick Morales	San Judas	2131-4234	January	August	322	22/9/2024 13:...
	4	Denís Wong	Dmitrov	2343-2434	January	March	300	22/9/2024 13:...
	3	Juan Gutierrez	Camilo Ortega	2332-3353	January	March	30	22/9/2024 13:...

 Edit

 Delete

 Show Details

 Back

Next 



