



Universidad Politécnica Internacional

Carrera: Ingeniera en sistemas

Materia: Programación II

Proyecto 1 - Desarrollo de Aplicación de Consola en C# para Matrícula
Escolar

Profesor: Alexander Benjamin Curling

Integrantes:

Jarol Roberto Chaverri Sanabria

Johan Espinoza Alfaro

Jonathan Segura Méndez

II Cuatrimestre 2024

Contenido

1. Investigación Previa:	3
Investiga cómo se realiza el proceso de matrícula en diferentes centros educativos	3
Identifica los datos necesarios para la matrícula de estudiantes.....	8
Define los requisitos específicos para el formulario de matrícula de estudiantes.....	9
2. Análisis de Caso y Justificación	9
Realiza un análisis del caso de matrícula para el colegio que elijas.....	9
b) Justifica las decisiones tomadas en cuanto al diseño del formulario y la funcionalidad requerida	10
3. Diagrama de Flujo	12
4.Pseudocódigo	13
Pseudocódigo 1: Inicio y Menú Principal	13
Pseudocódigo 2: Mostrar Menú	13
Pseudocódigo 3: Seleccionar Opción del Menú	14
Pseudocódigo 4: Registrar Estudiante	15
Pseudocódigo 5: Editar Nombre del Estudiante	15
Pseudocódigo 6: Ver Matrículas Registradas	16
Pseudocódigo 7: Seleccionar Materias	16
Pseudocódigo 8: Mostrar Materias Disponibles	17
Pseudocódigo 9: Mostrar Resumen de Matrícula	18
6. Justificación del Uso de Estructuras de Datos	18
Biografía	21

1. Investigación Previa:

Investiga cómo se realiza el proceso de matrícula en diferentes centros educativos.

Este proyecto de investigación propone una solución informática, un sistema de programación, para solucionar las necesidades de una entidad educativa universitario, tales como la gestión y administración de datos de los estudiantes que desean realizar la matrícula en la institución con la finalidad de poder inscribirse y llevar el curso o carrera a fin con normalidad

La solución propuesta, contará con múltiples clases, métodos, atributos, ciclos y demás técnicas que programación en C# que hemos venido desarrollando en clases, los cuales se construirán a partir de los requerimientos necesarios para llevar a cabo una matrícula para estudiante, esto para el correcto diseño del software, logrando así, la correcta funcionalidad.

Para la investigación de este proyecto se tomaron en cuenta 2 universidades:

En la UNED se puede ingresar desde www.UNED.AC.CR, donde se puede ingresar mediante un usuario y contraseña que se realiza después de un proceso de admisión



001

Luego de ingresar desplegara una serie de opciones como se muestran en la imagen 002, de la cual se debe seleccionar “Matricula”.



002

Una vez seleccionado matrícula se deberá proceder a ingresar nuevamente, pero esta vez ingresando el numero de Cedula y una contraseña que debieron entregar cuando se realizó el proceso de admisión, además se debe seleccionar el periodo a matricular

003

Al ingresar los datos la aplicación realizara una validación de los datos, para luego realizar la matrícula, la cual debe ser cancelada por el monto correspondiente el mismo de seleccionadas las materias, la UNED tiene diferentes tipos de matrícula por lo tanto se deberá seleccionar el periodo cuatrimestral y el periodo de matrícula ya sea para Ordinario, Posgrado, cursos libres y suficiencia

La página muestra una serie de datos, que indica los datos personales del estudiante, como numero de cedula, Nombre completo, número de teléfono, sede, los cuales se pueden modificar según se requiera necesario.

004

En la pestaña 2 mostrara las asignaturas a matricular, en las cuales se seleccionan las materias a matricular mediante un código, lo mismo para cada asignatura

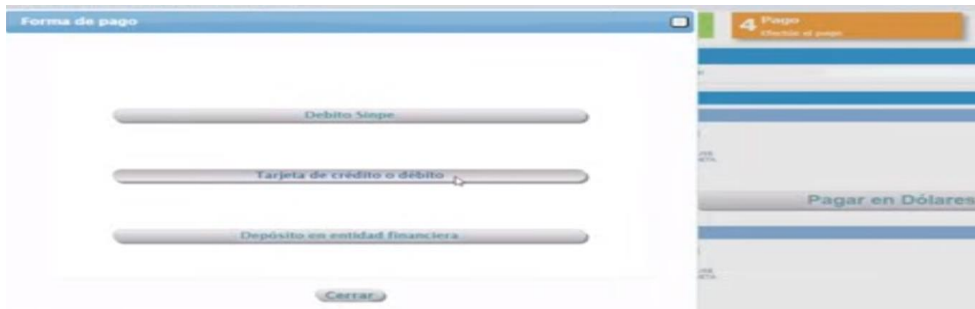
005

En la pestaña 3 mostrara los costos, ósea el monto total de las materias y matricula a cancelar

Rubro	Colones
Cuota Estudiantil	0.00
Matrícula	0.00
Costo de Asignaturas	96,500.00
Cena Estudiantil	0.00
Fianza Estudiantil	0.00
Fondo Solidario	0.00
Deudas Pendientes	0.00
Comprobante de Pago	24,625.00
Total a Cancelar:	73,875.00

006

En la pestaña 4 se despliega un formulario a llenar por el estudiante para añadir una cuenta para realizar el pago de matricula y su respectivo comprobante



007

En la segunda Universidad que realizamos la investigación es la UPI, en la cual también fue basado nuestro proyecto, esto por que nos parece que estamos mas familiarizados con el sistema de matrícula, las 2 aplicaciones son bastante similares entre sí, pero al final nos decidimos por la de la Universidad Politécnica internacional, ya que se nos hace un poco simple de utilizar.

Para la matricula se debe ingresar al avatar desde la página

<https://politecnica.avatarsys.io/avatar/log/menu.jsp>, con un usuario y contraseña



00

Luego de ingresar los datos de usuario y contraseña se ingresa al avatar de la universidad, la cual despliega 2 opciones, Automatricula para el debido proceso de selección de las asignaturas a matricular y Servicio al estudiante las cuales muestran un histórico de las materias matriculas por periodo y notas, ausencias, entre otras



009

Seleccionada la opción Automatricula desplegara los datos personales del estudiante para verificar o modificar según se considere

 This screenshot shows the 'Automatricula' system interface. The 'Personal' tab is selected, displaying a form titled 'Verifique por favor sus datos personales' and 'Datos Personales'. The form includes fields for: Carnet, Nombre, Teléfono Habitación, Teléfono Celular, Fax, Correo Electrónico, Apartado Postal, Dirección de Domicilio, and Exacta.

010

Luego de a ver verificado todos los datos del estudiante, seleccionamos siguiente para que se despliegue la opción parámetros en la cual se deberá llenar el formulario para la debida matricula

 This screenshot shows the 'Automatricula' system interface with the 'Parametros' tab selected. The form is titled 'Parámetros de su sesión de automatricula:'. It includes fields for: Sede (dropdown), Carnet (401990608), Nombre del Estudiante (SEGURA MENDEZ JONATHAN FRANCISCO), Año (dropdown), Periodo Lectivo (dropdown), Modalidad de Matrícula (Normal), Carrera (dropdown), and Cita de matrícula.

011

En la Opción de matrícula, nos permite seleccionar los cursos o asignaturas que vayamos a matricular según el periodo lectivo

012

Luego de realizar la debida matricula, se deberá realizar la confirmación de la misma después de visualizar la boleta de matrícula, los montos, el horario entre otros.

Posterior a esto, debe darle botón de Financiero

En la siguiente pantalla, usted debe seleccionar la boleta y la letra de cambio (en caso de que seleccionará un financiamiento), y luego darle clic en siguiente, Cuando le da el botón de formalizar, le va a salir un mensaje., indicando que la matrícula ha sido procesada, solo debe darle el botón de Aceptar.

	Total	Por Cancelar
Colones	0.00	
Dolares	0.00	

013

Identifica los datos necesarios para la matrícula de estudiantes.

Entre los diferentes aspectos necesarios para poder realizar la matrícula de los estudiantes en la institución educativa, identificamos los siguientes datos como esenciales para llevar a cabo la debida gestión:

1. Id o número Cedula de identidad
2. Nombre del estudiante
3. Edad

4. Materia y costo de la misma
5. Matricula y costo de la misma

Define los requisitos específicos para el formulario de matrícula de estudiantes.

Para el formulario de matricula para los estudiantes de la Universidad Tecno_CR Internacional, definimos los siguientes puntos

- Los datos de los estudiantes (Nombre, edad, ID, etc.)
- Materias a matricular
- Registrar y modificar los documentos de cada estudiante de ser necesario
- Mostrar resumen de matricula
- Guardar matricula de materias seleccionadas

2. Análisis de Caso y Justificación

Realiza un análisis del caso de matrícula para el colegio que elijas.

Para este proyecto se realizó una investigación, para definir una propuesta del diseño y análisis para la posterior construcción de un sistema de programación, creando así los pilares para el desarrollo de este software, con la finalidad de administrar y gestionar los procesos específicos de la respectivas matriculas de los estudiantes en una Universidad.

Teniendo la finalidad de integrar los requisitos de la institución donde se implementará a futuro este sistema, con una solución diseñada a medida, para la correcta funcionalidad de los procesos requeridos y la escalabilidad de este sistema.

b) Justifica las decisiones tomadas en cuanto al diseño del formulario y la funcionalidad requerida

Para el desarrollo de nuestro código de programación, trataremos de mostrar un pequeño resumen en cuanto al software desarrollado, describiendo brevemente la funcionalidad del mismo.

Class: creamos nuestras tres clases principales las cuales van a ser Estudiantes, Materia y Matricula.

public class Estudiante: Para el ID, nombre y edad del estudiante

public class Materia: Para el nombre de cada materia y el costo de la misma

public class Matricula: Registra un estudiante y una lista de materias. Incluye métodos para calcular el costo total y para proporcionar un resumen de la matrícula.

Métodos: Se implementan los métodos para que la universidad pueda realizar la matricula o hacer cambios según consideren necesario

static void MostrarMenu(): Desplega todas las opciones de nuestro programa

static void RegistrarEstudiante(): Permite registrar un nuevo estudiante solicitando ID, nombre y edad.

static void EditarNombreEstudiante(): Permite editar el nombre del estudiante, actualizando así sus datos

static void VerMatriculas(): Muestra todas las matrículas registradas por los estudiantes.

static void SeleccionarMaterias(): Permite seleccionar las materias que el estudiante quiera llevar en el periodo lectivo.

static void MostrarMateriasDisponibles(): Indica que materias están disponible según el récord académico del estudiante.

static void MostrarResumen(): Muestra un resumen detallado de la matrícula actual.

static void GuardarMatricula(): Guarda el resumen de la matrícula actual en un archivo de texto

Variables: Utilizamos diferentes variables para almacenar información que puede ser modificada durante la ejecución de un programa.

string Nombre: Guarda el nombre del estudiante

decimal Costo: Para calcular las operaciones matemáticas de la suma de cantidad de materias matriculadas (Costo x materia 76200 colones) y el precio de la matricula (63.400 colones)

int ID: Registra el numero de cedula de cada estudiante

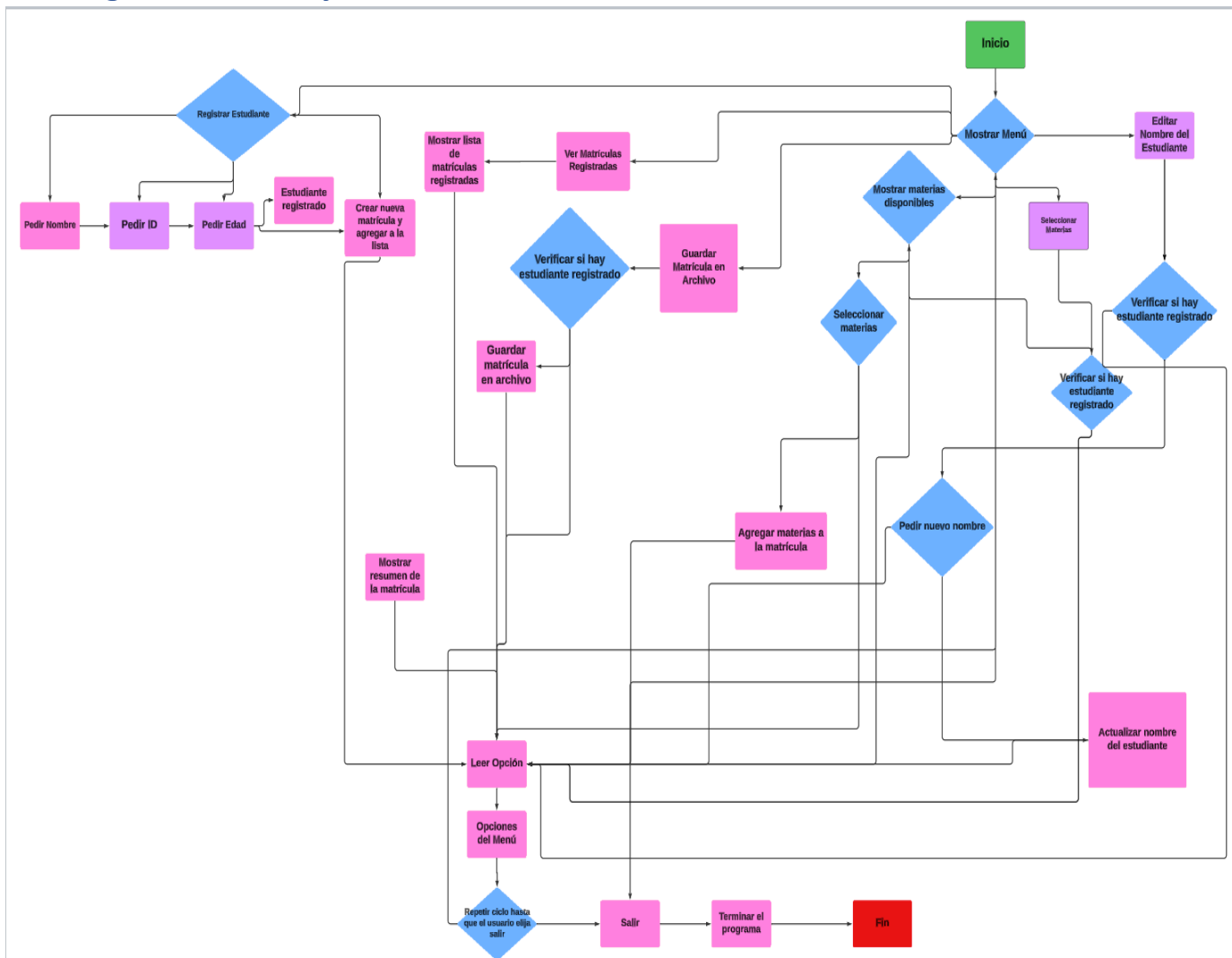
Int Edad: Guarda la edad de cada estudiante

Ciclos **While**: Nos permiten evaluar ejecutar repetidamente un bloque de código mientras se cumpla una condición específica.

int edad;

```
while (!int.TryParse(Console.ReadLine(), out edad) || edad <= 0)
{
    Console.WriteLine("Edad inválida. Ingresar un número entero positivo.");
    Console.Write("Edad: ");
}
```

3. Diagrama de Flujo



4.Pseudocodigo

Pseudocódigo 1: Inicio y Menú Principal

- Descripción: Inicializa el programa y muestra el menú principal.

Inicio

MostrarMenú()

continuar = Verdadero

Mientras continuar

 opción = LeerOpción()

 SeleccionarOpción(opción)

FinMientras

Fin

Pseudocódigo 2: Mostrar Menú

- Descripción: Muestra las opciones disponibles en el menú principal.

Procedimiento MostrarMenú

 Imprimir "Menú:"

 Imprimir "1. Registrar Estudiante"

 Imprimir "2. Editar Nombre del Estudiante"

 Imprimir "3. Ver Matrículas Registradas"

 Imprimir "4. Seleccionar Materias"

 Imprimir "5. Mostrar Resumen de Matrícula"

 Imprimir "6. Guardar Matrícula en Archivo"

 Imprimir "7. Salir"

 Imprimir "Elige una opción: "

FinProcedimiento

Pseudocódigo 3: Seleccionar Opción del Menú

- Descripción: Ejecuta la acción correspondiente según la opción seleccionada por el usuario.

Procedimiento SeleccionarOpción(opción)

Si opción == 1 Entonces

 RegistrarEstudiante()

SinoSi opción == 2 Entonces

 EditarNombreEstudiante()

SinoSi opción == 3 Entonces

 VerMatriculas()

SinoSi opción == 4 Entonces

 SeleccionarMaterias()

SinoSi opción == 5 Entonces

 MostrarResumen()

SinoSi opción == 6 Entonces

 GuardarMatricula()

SinoSi opción == 7 Entonces

 continuar = Falso

FinSi

FinProcedimiento

Pseudocódigo 4: Registrar Estudiante

- Descripción: Registra un nuevo estudiante solicitando su ID, nombre y edad.

Procedimiento RegistrarEstudiante

Imprimir "Registro de Estudiante:"

Imprimir "ID: "

Leer id

Imprimir "Nombre: "

Leer nombre

Imprimir "Edad: "

Leer edad

nuevaMatricula = CrearMatricula(id, nombre, edad)

AgregarMatricula(nuevaMatricula)

estudianteRegistrado = Verdadero

Imprimir "Estudiante registrado correctamente."

FinProcedimiento

Pseudocódigo 5: Editar Nombre del Estudiante

- Descripción: Permite editar el nombre del último estudiante registrado.

Procedimiento EditarNombreEstudiante

Si estudianteRegistrado == Falso Entonces

Imprimir "Primero debes registrar un estudiante."

Sino

Imprimir "Nombre actual: ", UltimaMatricula().nombre

Imprimir "Ingresa el nuevo nombre: "

Leer nuevoNombre

UltimaMatricula().nombre = nuevoNombre

Imprimir "Nombre del estudiante actualizado correctamente."

FinSi

FinProcedimiento

Pseudocódigo 6: Ver Matrículas Registradas

- Descripción: Muestra una lista de todas las matrículas registradas.

Procedimiento VerMatriculas

Si NumeroDeMatriculas() == 0 Entonces

Imprimir "No hay matrículas registradas."

Sino

Para cada matricula en matriculas

Imprimir "ID: ", matricula.id, ", Nombre: ", matricula.nombre

FinPara

FinSi

FinProcedimiento

Pseudocódigo 7: Seleccionar Materias

- Descripción: Permite seleccionar materias para el último estudiante registrado.

Procedimiento SeleccionarMaterias

Si estudianteRegistrado == Falso Entonces

Imprimir "Primero debes registrar un estudiante."

Sino

Imprimir "Selecciona las materias:"

MostrarMateriasDisponibles()

Mientras opcion != FinalizarSeleccion

Imprimir "Elige una materia (1-", NumeroDeMateriasDisponibles() + 1, "): "

Leer opcion

Si opcion <= NumeroDeMateriasDisponibles Entonces

materiaSeleccionada = ObtenerMateria(opcion)

Si MateriaNoSeleccionada(materiaSeleccionada) Entonces

AgregarMateria(UltimaMatricula(), materiaSeleccionada)


```
        Imprimir "", materiaSeleccionada.nombre, "" seleccionada correctamente."
    Sino
        Imprimir "La materia ", materiaSeleccionada.nombre, "" ya ha sido seleccionada."
    FinSi
FinSi
FinMientras
Imprimir "Selección de materias finalizada."
FinSi
FinProcedimiento
```

Pseudocódigo 8: Mostrar Materias Disponibles

- Descripción: Muestra las materias disponibles para selección.

Procedimiento MostrarMateriasDisponibles

```
    Para i = 1 Hasta NumeroDeMateriasDisponibles()
        Imprimir i, ". ", ObtenerMateria(i).nombre, " - Costo: ", ObtenerMateria(i).costo, " colones"
    FinPara
FinProcedimiento
```

Pseudocódigo 9: Mostrar Resumen de Matrícula

- Descripción: Muestra el resumen de la matrícula del último estudiante registrado.

Procedimiento MostrarResumen

Si estudianteRegistrado == Falso Entonces

 Imprimir "Primero debes registrar un estudiante."

SinoSi NumeroDeMateriasSeleccionadas(UltimaMatricula()) == 0 Entonces

 Imprimir "No se han seleccionado materias."

Sino

 Imprimir "Resumen de Matrícula:"

 Imprimir UltimaMatricula().Resumen()

FinSi

FinProcedimiento

Pseudocódigo 10: Guardar Matrícula en Archivo

- Descripción: Guarda la información de la matrícula del último estudiante en un archivo.

Procedimiento GuardarMatricula

Si estudianteRegistrado == Falso Entonces

 Imprimir "No hay información de matrícula para guardar."

SinoSi NumeroDeMateriasSeleccionadas(UltimaMatricula()) == 0 Entonces

 Imprimir "No se han seleccionado materias."

Sino

 nombreArchivo = "matricula_" + UltimaMatricula().id + ".txt"

 EscribirArchivo(nombreArchivo, UltimaMatricula().Resumen())

 Imprimir "Matrícula guardada en el archivo '", nombreArchivo, "'."

FinSi

FinProcedimiento

Justificación del Uso de Estructuras de Datos

Justificación

Las listas se utilizaron en este código debido a su flexibilidad, permitiendo manejar colecciones de elementos cuyo tamaño puede variar durante la ejecución del programa. Proporcionan métodos integrados para agregar, eliminar y verificar elementos fácilmente, lo que simplifica la manipulación de datos como las materias y matrículas de estudiantes. Además, las listas permiten un acceso rápido y eficiente a los elementos mediante índices y soportan iteración directa, facilitando operaciones como mostrar y procesar todos los elementos de una colección. Su compatibilidad con C# asegura que el código sea mantenible y entendible.

Flexibilidad:

Motivo: Las listas son una estructura de datos dinámica, lo que significa que su tamaño puede cambiar durante la ejecución del programa. Esto es especialmente útil en escenarios donde no se conoce de antemano el número exacto de elementos que se manejarán.

Aplicación en el Código: En el contexto de la matrícula universitaria, el número de materias que un estudiante puede seleccionar no está predeterminado y puede variar ampliamente. Las listas permiten agregar y eliminar materias según las necesidades de cada estudiante sin requerir una nueva estructura de datos o una reasignación de memoria.

Facilidad de Uso:

Motivo: Las listas en C# proporcionan una variedad de métodos integrados para manipular los elementos de la colección. Métodos como Add (para agregar elementos), Remove (para eliminar elementos), Contains (para verificar la existencia de un elemento) y Count (para obtener el número de elementos) simplifican enormemente la gestión de colecciones.

Aplicación en el Código: La facilidad de agregar y eliminar materias de la lista de materias seleccionadas por un estudiante mejora la eficiencia y la legibilidad del código. Del mismo modo, la capacidad de contar el número de materias seleccionadas permite calcular costos totales de manera sencilla.

Acceso Rápido y Eficiente:

Motivo: Las listas permiten acceder a sus elementos mediante índices, lo que facilita operaciones rápidas de lectura y escritura. El acceso indexado es particularmente útil cuando se necesita acceder frecuentemente a elementos específicos dentro de la colección.

Aplicación en el Código: En el programa de matrícula, es crucial poder acceder rápidamente a las materias disponibles para mostrarlas al usuario y permitir su selección. El acceso eficiente a los elementos asegura que el programa funcione de manera rápida y sin demoras perceptibles.

Consistencia y Estabilidad:

Aplicación en el Código: La elección de listas garantiza que el código sea mantenible y entendible para otros desarrolladores que puedan trabajar en el proyecto en el futuro, aprovechando una estructura de datos familiar y bien documentada.

Biografía

Pagina Web donde se realizaron investigaciones y el diagrama de flujo

<http://www.uned.ac.cr/>

https://lucid.app/lucidchart/0bab4890-5da4-4734-87a4-43f511312bc5/edit?viewport_loc=-86%2C429%2C5572%2C2750%2C0_0&invitationId=inv_bc6a8423-4d0d-4b32-b730-ebfc6b8c4275

<https://politecnica.avatarsys.io/avatar/log/menu.jsp>

<https://learn.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/tutorials/arrays-and-collections>