



Universidad Politécnica Internacional

Programación II

Proyecto I

Integrantes:

Melanny Víquez Saborío

Iriel Vargas Pinell

Anthony Castro Muñoz

Prof. Alexander Benjamín Curling.

Investigación y Análisis:

El programa de cursos de conversación de la Escuela de Lenguas Modernas de la Universidad de Costa Rica, ofrece a los costarricenses la posibilidad de llevar cursos de idiomas con profesores certificados de calidad. Estos cursos se ofrecen en la Casa de Idiomas y su matrícula se realiza en línea mediante el sitio web de la Fundación UCR, los idiomas que se ofrecen para matricular son los siguientes: japonés, portugués, inglés, francés e italiano. Para matricular los estudiantes deben de acceder al link de matrícula e identificarse, seguidamente se les muestra la oferta de cursos disponibles por idioma, allí se puede ver el nombre del curso (nivel), horario y precio. Luego el estudiante confirma su matrícula y procede con el pago y de esta forma queda matriculado en el curso.

En el Centro Educativo Cooperativo San Carlos Borromeo se ofrece educación privada para preescolar, primaria y secundaria. Esta institución se compromete a una educación bilingüe con un profesorado de calidad, manteniendo precios accesibles para las familias de la Zona Norte. Además, cuenta con formación en tecnología y robótica, lógica matemática y deportes. Para el proceso de matrícula, se debe realizar un formulario de preinscripción y presentar documentos en un folio A4. Entre los requisitos necesarios para este proceso son cédula y partida de nacimiento, realizar el pago de matrícula ₡167 000 y el valor de la mensualidad de ₡135 000.

Análisis del Caso

Con base en lo mencionado anteriormente es requerido generar un sistema de matrícula para El programa de cursos de conversación de la Escuela de Lenguas Modernas de la Universidad de Costa Rica, en el cual los usuarios puedan ingresar y realizar su proceso de matrícula completo, ingresando la información del estudiante hasta llegar al pago del curso. Todo debe realizarse de la manera más amigable y confiable para los usuarios, dicho sistema debe contener las validaciones necesarias para que los errores realizados por los usuarios puedan ser solventados e informados al usuario para que el sistema pueda seguir funcionando sin caerse y que el usuario tenga claro que debe de hacer y que debe cambiar en caso de errores.

Justificación de Decisiones del Sistema de Matrícula

Se estructuró el sistema en las clases Estudiante, Matrícula y Pago. Esta decisión se tomó para seguir el principio de responsabilidad única, donde cada clase se encarga de una parte específica del proceso de matrícula. El realizar la organización de esta forma facilita la

comprensión, el mantenimiento y la expansión del sistema. Cada clase puede ser desarrollada y probada de manera independiente, mejorando la robustez del sistema y permitiendo modificaciones sin afectar otras partes del código.

También se decidió incluir la solicitud de confirmación del usuario antes de completar la matrícula y el pago. La confirmación final es un paso crítico que permite a los usuarios revisar y corregir cualquier error antes de finalizar el proceso. Esto asegura que todos los datos ingresados sean correctos y completos, mejorando la precisión del sistema y la satisfacción del usuario. La posibilidad de revisión final reduce la probabilidad de errores y garantiza que el estudiante tenga la oportunidad de verificar toda la información antes de proceder.

Cabe mencionar el uso de bucles y sentencias de control, en la cual se utilizaron bucles while y sentencias switch para gestionar las interacciones del usuario. Estas estructuras de control permiten manejar de manera eficiente las entradas del usuario, proporcionando retroalimentación inmediata y la posibilidad de reintentar en caso de errores. Esto asegura que el usuario complete cada paso correctamente antes de avanzar al siguiente, mejorando la robustez y la usabilidad del sistema. La capacidad de gestionar múltiples interacciones del usuario sin interrupciones garantiza un flujo continuo y efectivo del proceso de matrícula.

A continuación se explica a grandes rasgos cada una de las clases utilizadas en el sistema de matrícula:

Clase Estudiante

Se incluyen atributos como NombreCompleto, NumeroDeCedula, Telefono y Email, y métodos para la identificación (Identificarse) y para mostrar información (MostrarInformacion). La identificación del estudiante es el primer paso crítico en el proceso de matrícula. Capturar y gestionar la información personal asegura que solo los usuarios registrados puedan continuar con el proceso de matrícula. Además, el método MostrarInformacion permite verificar que los datos ingresados sean correctos, evitando errores posteriores y asegurando que los estudiantes puedan corregir cualquier dato antes de finalizar su inscripción.

Clase Matricula

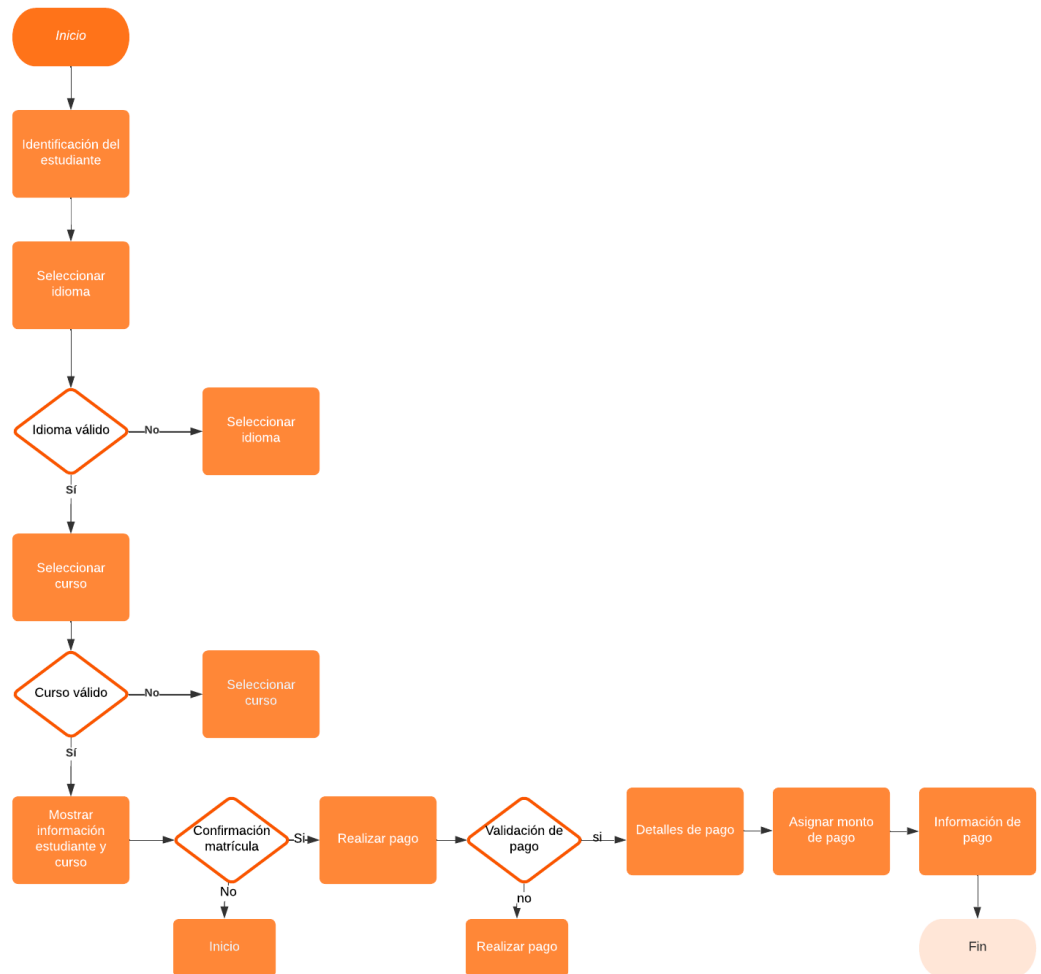
Se incluyeron atributos como Idioma y Curso, y métodos para seleccionar el idioma (EscogerIdioma) y el curso (EscogerCurso). La selección del idioma y del curso es fundamental para personalizar la experiencia de aprendizaje del estudiante. Ofrecer menús

interactivos para la selección de idiomas y cursos permite a los estudiantes elegir entre las opciones disponibles de manera clara y sencilla. Esto reduce la posibilidad de errores y mejora la satisfacción del usuario al asegurar que el estudiante se inscriba en el curso correcto según su preferencia y disponibilidad.

Clase Pago

Se incluyeron atributos como Monto, MetodoPago, NumeroConfirmacion, NumeroTarjeta, CVV y FechaCaducidad, y métodos para realizar el pago (RealizarPago) y mostrar la información del pago (MostrarInformacion). El proceso de pago es una parte crítica del sistema de matrícula. Ofrecer múltiples métodos de pago (tarjeta de crédito o depósito bancario) asegura que el sistema sea accesible para una mayor variedad de usuarios. Capturar y validar todos los detalles de pago garantiza que las transacciones se realicen de manera segura y confiable. El método MostrarInformacion permite a los estudiantes verificar los detalles del pago antes de finalizar la transacción, reduciendo errores y disputas potenciales.

Diagrama de Flujo



Pseudocódigo

A continuación se presenta el pseudocódigo de la solución realizada, el mismo se basa en la sintaxis de la herramienta Pseint:

Proceso Principal

```

Mientras Verdadero Hacer
    Definir estudiante Como Estudiante
    Llamar estudiante.Identificarse()

    Definir matricula Como Matricula
    Llamar matricula.EscogerIdioma()
    Llamar matricula.EscogerCurso()

    LimpiarPantalla()
  
```

Escribir "Información del estudiante:"
Llamar estudiante.MostrarInformacion()
Escribir ""
Escribir "Curso a matricular:"
Llamar matricula.MostrarInformacion()
Escribir ""
Escribir "¿Está seguro de su confirmación de matrícula? (S/N): "
Leer confirmacion

Si Mayusculas(confirmacion) = "S" Entonces
Definir pago Como Pago
Llamar pago.RealizarPago()
SiNo
Escribir "Matrícula cancelada. Presione cualquier tecla para continuar..."
Leer respuesta
FinSi
FinMientras

FinProceso

Proceso Estudiante

// Atributos
Definir NombreCompleto Como Cadena
Definir NumeroDeCedula Como Cadena
Definir Telefono Como Cadena
Definir Email Como Cadena

SubProceso Identificarse()
LimpiarPantalla()

Escribir "Bienvenido al sistema de matrícula de la Casa de Idiomas de la Universidad de Costa Rica"

Escribir "Por favor identifíquese:"
Escribir "Nombre Completo: "
Leer NombreCompleto
Escribir "Número de Cédula: "
Leer NumeroDeCedula
Escribir "Teléfono: "
Leer Telefono

Escribir "Email: "

Leer Email

FinSubProceso

SubProceso MostrarInformacion()

Escribir "Nombre: ", NombreCompleto

Escribir "Cédula: ", NumeroDeCedula

Escribir "Teléfono: ", Telefono

Escribir "Email: ", Email

FinSubProceso

FinProceso

Proceso Matricula

// Atributos

Definir Idioma Como Cadena

Definir Curso Como Cadena

SubProceso EscogerIdioma()

Mientras Verdadero Hacer

LimpiarPantalla()

Escribir "Seleccione el idioma que desea matricular:"

Escribir "1. Francés"

Escribir "2. Inglés"

Escribir "3. Japonés"

Escribir "4. Italiano"

Escribir "5. Portugués"

Escribir "Seleccione una opción: "

Leer opcionIdioma

Segun opcionIdioma Hacer

Caso "1":

Idioma = "Francés"

SalirSegun

Caso "2":

Idioma = "Inglés"

SalirSegun

Caso "3":

Idioma = "Japonés"

SalirSegun

Caso "4":

Idioma = "Italiano"

SalirSegun

Caso "5":

Idioma = "Portugués"

SalirSegun

DeOtroModo:

Escribir "Opción no válida. Presione cualquier tecla para continuar..."

Leer respuesta

FinSegun

FinMientras

FinSubProceso

SubProceso EscogerCurso()

Mientras Verdadero Hacer

LimpiarPantalla()

Escribir "Seleccione el curso de ", Idioma, ":"

Escribir "1. ", Idioma, " Nivel 1 L 6pm a 9pm"

Escribir "2. ", Idioma, " Nivel 2 K 6pm a 9pm"

Escribir "3. ", Idioma, " Nivel 3 M 6pm a 9pm"

Escribir "4. ", Idioma, " Nivel 4 J 6pm a 9pm"

Escribir "Seleccione una opción: "

Leer opcionCurso

Segun opcionCurso Hacer

Caso "1":

Curso = Idioma + " Nivel 1"

SalirSegun

Caso "2":

Curso = Idioma + " Nivel 2"

SalirSegun

Caso "3":

Curso = Idioma + " Nivel 3"

SalirSegun

Caso "4":

Curso = Idioma + " Nivel 4"

SalirSegun

DeOtroModo:

Escribir "Opción no válida. Presione cualquier tecla para continuar..."

Leer respuesta

FinSegun

FinMientras

FinSubProceso

SubProceso MostrarInformacion()

Escribir "Idioma: ", Idioma

Escribir "Curso: ", Curso

FinSubProceso

FinProceso

Proceso Pago

// Atributos

Definir Monto Como Real

Definir MetodoPago Como Cadena

Definir NumeroConfirmacion Como Cadena

Definir NumeroTarjeta Como Cadena

Definir CVV Como Cadena

Definir FechaCaducidad Como Cadena

SubProceso RealizarPago()

Mientras Verdadero Hacer

LimpiarPantalla()

Escribir "Método de Pago:"

Escribir "1. Con Tarjeta"

Escribir "2. Con Numero de Confirmación de Depósito"

Escribir "Seleccione una opción: "

Leer opcionPago

Segun opcionPago Hacer

Caso "1":

MetodoPago = "Tarjeta"

Escribir "Ingrese el número de tarjeta: "

Leer NumeroTarjeta

Escribir "Ingrese el CVV: "

Leer CVV

Escribir "Ingrese la fecha de caducidad (MM/AAAA): "

Leer FechaCaducidad

SalirSegun

Caso "2":

MetodoPago = "Deposito"

Escribir "Ingrese el número de confirmación del depósito: "

Leer NumeroConfirmacion

SalirSegun

DeOtroModo:

Escribir "Opción no válida. Presione cualquier tecla para continuar..."

Leer respuesta

FinSegun

FinMientras

Monto = 100000 // Costo fijo para el curso

LimpiarPantalla()

Escribir "Información del Pago:"

Llamar MostrarInformacion()

Escribir ""

Escribir "Todo fue satisfactorio. Presione cualquier tecla para finalizar..."

Leer respuesta

FinSubProceso

SubProceso MostrarInformacion()

Escribir "Monto: ", Monto

Escribir "Método de Pago: ", MetodoPago

Si MetodoPago = "Tarjeta" Entonces

Escribir "Número de Tarjeta: ", NumeroTarjeta

Escribir "CVV: ", CVV

Escribir "Fecha de Caducidad: ", FechaCaducidad

FinSi

Si MetodoPago = "Deposito" Entonces

Escribir "Número de Confirmación: ", NumeroConfirmacion

FinSi

FinSubProceso

FinProceso

Justificación del Uso de Estructuras de Datos:

El código se estructura de manera que cada clase se encarga de un aspecto del proceso de matrícula, se detalla de la siguiente forma donde contiene las clases Estudiante que maneja la información personal del estudiante, Matrícula maneja la selección del idioma y el curso, Pago maneja la información de pago y la realiza, y Program con el Main para ejecutar el programa.

1. Clase Estudiante:

A. Atributos:

- i. `public string NombreCompleto { get; set; }`
- ii. `public string NumeroDeCedula { get; set; }`
- iii. `public string Telefono { get; set; }`
- iv. `public string Email { get; set; }`

B. Métodos:

- i. `public void Identificarse():` Solicita al usuario que ingrese sus datos personales y los almacena en los atributos.
- ii. `public void MostrarInformacion():` Muestra la información almacenada del estudiante.

2. Clase Matrícula:

A. Atributos:

- i. `public string Idioma { get; set; }`
- ii. `public string Curso { get; set; }`

B. Métodos:

- i. `public void EscogerIdioma():` Muestra el menú para seleccionar un idioma y la almacena en el atributo Idioma.
- ii. `public void EscogerCurso():` Muestra el menú para seleccionar un curso basado en el idioma elegido y lo almacena en el atributo Curso.
- iii. `public void MostrarInformacion():` Muestra la información del idioma y el curso seleccionado.

3. Clase Pago:

A. Atributos:

- i. `public float Monto { get; set; };` Monto del pago ya establecido como 100000.
- ii. `public string MetodoPago { get; set; };` Método de pago seleccionado, tarjeta o depósito.
- iii. `public string NumeroConfirmacion { get; set; };` Número de confirmación del depósito, solo para método de pago depósito.
- iv. `public string NumeroTarjeta { get; set; };` Número de tarjeta, solo para método de pago tarjeta.
- v. `public string CVV { get; set; };` CVV de la tarjeta, para método de pago tarjeta.
- vi. `public string FechaCaducidad { get; set; };` Fecha de caducidad de la tarjeta, para método de pago Tarjeta.

B. Métodos:

- i. `public void RealizarPago();` Solicita al usuario que seleccione un método de pago y muestra los detalles según el método seleccionado.
- ii. `public void MostrarInformacion();` Muestra la información del pago realizado.

4. Clase Program:

A. . Métodos:

- i. `static void Main(string[] args);` Método principal que ejecuta el programa. Donde se realiza lo siguiente:
 - Crea una instancia de Estudiante y llama a Identificarse.
 - Crea una instancia de Matrícula, llama a EscogerIdioma y EscogerCurso.
 - Muestra la información del estudiante y la matrícula.
 - Pregunta al usuario si desea confirmar la matrícula.
 - Si se confirma, crea una instancia de Pago y llama a RealizarPago.
 - Si no se confirma, muestra un mensaje de cancelación y reinicia el proceso.

Justificación de la Eficiencia y Uso de la Aplicación de Sistema de Matrículas

En este apartado encontramos como bien el título una explicación de la eficiencia y el uso de la aplicación.

Eficiencia

El haber realizado el sistema con una estructura de tres clases principales (Estudiante, Matricula y Pago) permite una mejor organización del código y facilita el mantenimiento. Cada clase se encarga de una parte específica del proceso de matrícula, lo que hace que el código sea más fácil de entender y actualizar. La modularidad no solo mejora la claridad del código, sino que también permite una reutilización efectiva de los métodos para no saturar el programa con código innecesario. Esto significa que cualquier modificación o mejora se puede realizar en un solo lugar sin afectar al resto del sistema, mejorando así la eficiencia del desarrollo y mantenimiento del software.

El utilizar bucles while y sentencias switch asegura que el usuario pueda interactuar con el sistema de manera continua y fluida hasta que se ingresen opciones válidas. Esto minimiza la posibilidad de errores y mejora la experiencia del usuario al garantizar que solo se procesen datos correctos, ya que sabemos que sin validaciones el sistema es muy propenso a tener fallos incluso llegar a detenerse o caerse. Además, el manejo adecuado de entradas no válidas mediante mensajes de error y la posibilidad de reintentar la operación, asegura que el sistema funcione de manera robusta y sin interrupciones, contribuyendo a la eficiencia general del sistema.

Facilidad de Uso

El diseño de la interfaz de usuario del sistema de matrículas es intuitivo, utilizando mensajes claros y consistentes para guiar al usuario a través del proceso de matrícula, intentando que el usuario tenga lo más claro posible los datos que debe de ingresar o los pasos que debe seguir. La limpieza de la consola entre pantallas utilizando el método "Console.Clear" ayuda a mantener la atención del usuario en la tarea actual, evitando distracciones y posibles confusiones. Este uso del método mencionado anteriormente garantiza que el usuario siempre tenga una vista limpia de la tarea que está realizando, mejorando así la usabilidad del sistema.

El proceso de matrícula está dividido en pasos secuenciales, lo que simplifica la interacción para el usuario y reduce la posibilidad de errores, a parte de esto el utilizar a nivel validaciones permite al usuario entender cuál fué el fallo que cometió en caso. Antes de completar la matrícula, el sistema solicita una confirmación del usuario, lo que permite verificar y corregir cualquier error antes de finalizar el proceso. Esta verificación final es

crucial para garantizar que los datos ingresados sean correctos y completos. Además, el sistema ofrece la posibilidad de métodos de pago, permitiendo al usuario elegir entre tarjeta de crédito o depósito. Esta flexibilidad mejora la accesibilidad y conveniencia para diferentes tipos de usuarios, asegurando que el sistema sea útil para una amplia variedad de personas con diferentes necesidades y preferencias.