

UNIVERSIDAD DE LAS AMÉRICAS

Ingeniería en Ciberseguridad

Programación II

Matías Urgilés

Taller S14: Taller de Diseño Avanzado

Sistema de Matriculación de Automóviles:

Validación de datos: Ingreso de texto en el menú

```
1. Agregar Propietario
2. Asignar Propietario a Auto
3. Asignar Propietario a Moto
4. Buscar Vehículos por Marca
5. Listar Vehículos
6. Listar Propietarios
7. Listar Automóviles
8. Listar Motos por Marca
9. Matricular
10. Salir
nola
Error: Debes ingresar un número válido. Intenta de nuevo.
1. Agregar Propietario
2. Asignar Propietario a Auto
3. Asignar Propietario a Moto
4. Buscar Vehículos por Marca
5. Listar Vehículos
```

oficinaautomoviles > main > menuprincipal > Interfaz > MatriculacionVehiculos

Atrapamos el error, se informa al usuario y regresa al menú principal de forma segura.

Integridad Referencial: Dueño no existe

```
1. Listar Vehículos por Marca
2. Asignar Propietario a Auto
3. Asignar Propietario a Moto
4. Buscar Vehículos por Marca
5. Listar Vehículos
6. Listar Propietarios
7. Listar Automóviles
8. Listar Motos por Marca
9. Matricular
10. Salir
2
Ingrese su cedula:
1234
Propietario No Existe
1. Agregar Propietario
2. Asignar Propietario a Auto
3. Asignar Propietario a Moto
4. Buscar Vehículos por Marca
5. Listar Vehículos
6. Listar Propietarios
7. Listar Automóviles
8. Listar Motos por Marca
9. Matricular
10. Salir
```

Prueba de "Casting y Listado" (Polimorfismo)

Acción: Registra dos autos y una moto. Luego usa la Opción 7 (Listar Automóviles).

```
10. Salir
2
Ingrese su cedula:
1234
Ingrese marca:
Toyota
Ingree modelo:
rta
Ingresar Tipo:
Deportivo
Ingresar traccion:
Normal
Ingresar anio:
2000

10. Salir
3
Escriba la cedula del propietario al que le va a asignar
1234
Ingrese marca:
tundra
Ingresar modelo:
c90
Ingresar año:
2022
Ingresar altura:
1.30
Ingresar arranque:
Fuerte

          Marca: Ford
          Modelo: 100t
          Anio: 2010
          Propietario:
          Cedula: 1234
          Nombre: Matias
          Telefono: 8987
          Traccion: 4x4
          Tipo: Otro
          Marca: Toyota
          Modelo: rta
          Anio: 2000
          Propietario:
          Cedula: 1234
          Nombre: Matias
          Telefono: 8987
          Traccion: Normal
          Tipo: Deportivo
```

Prueba de Búsqueda por Marca

- **Acción:** Registra tres vehículos, dos de marca "Toyota" y uno "Honda". Usa la Opción 4 para buscar "Toyota".

```
Ingrese la marca de vehiculos que quiere buscar:  
Toyota  
[Marca: Toyota  
Modelo: c90  
Anio: 2024  
Propietario:  
Cedula: 1234  
Nombre: Matias  
Telefono: 0987657670  
Traccion: Normal  
Tipo: Deportivo, Marca: Toyota  
Modelo: d100  
Anio: 2000  
Propietario:  
Cedula: 1234  
Nombre: Matias  
Telefono: 0987657670  
Traccion: Normal
```

Sistema de Registro de Personas en una Universidad:

1. Prueba de "Identidad Única" (Validación de Cédula)

- **Acción:** Registra una persona con una cédula cualquiera (ej. 12345). Luego, intenta registrar a otra persona (puede ser de un tipo diferente, como un profesor) usando la misma cédula.
- **Resultado esperado:** El sistema debe mostrar el mensaje: "Error: Ya existe una persona con esa cédula".

```
----- OPCIONES -----  
1.- Mostrar listado general (Todos)  
2.- Ingresar Alumno pregrado  
3.- Ingresar Alumno magister  
4.- Ingresar Profesor hora  
5.- Listar Alumnos Pregrado  
6.- Listar Alumnos Magister  
7.- Listar Profesores  
8.- SALIR  
INGRESE UNA OPCION: 2  
Nombre: Matias  
Cedula: 1234  
Universidad: UdeA  
Carrera: Ciber  
Alumno de Pregrado agregado con éxito.  
  
----- OPCIONES -----  
1.- Mostrar listado general (Todos)  
2.- Ingresar Alumno pregrado  
3.- Ingresar Alumno magister  
4.- Ingresar Profesor hora  
5.- Listar Alumnos Pregrado  
6.- Listar Alumnos Magister  
7.- Listar Profesores  
8.- SALIR  
INGRESE UNA OPCION: 3  
Nombre: Juan  
Cedula: 1234  
Universidad: UdeA  
Tesis: Si  
Error: Ya existe una persona con esa cédula.
```

2. Prueba de "Casting y Filtrado" (Listado por tipo)

Esta prueba demuestra que el programa sabe distinguir quién es quién dentro de la lista general.

- **Acción:** Registra un Alumno de Pregrado, un Alumno de Magíster y un Profesor. Luego, selecciona la Opción 5,6,7 (Listar Alumnos Pregrado, Magister, Profesores).
- **Resultado esperado:** En la consola solo debe aparecer la información de cada una de los tipos de alumnos y del profesor, correspondientes a la opción seleccionada.

```
----- OPCIONES -----  
1.- Mostrar listado general (Todos)  
2.- Ingresar Alumno pregrado  
3.- Ingresar Alumno magister  
4.- Ingresar Profesor hora  
5.- Listar Alumnos Pregrado  
6.- Listar Alumnos Magister  
7.- Listar Profesores  
8.- SALIR  
INGRESE UNA OPCION: 2  
Nombre: Matias  
Cédula: 1234  
Universidad: UdlA  
Carrera: Ciber  
Alumno de Pregrado agregado con éxito.  
  
----- OPCIONES -----  
1.- Mostrar listado general (Todos)  
2.- Ingresar Alumno pregrado  
3.- Ingresar Alumno magister  
4.- Ingresar Profesor hora  
5.- Listar Alumnos Pregrado  
6.- Listar Alumnos Magister  
7.- Listar Profesores  
8.- SALIR  
INGRESE UNA OPCION: 3  
Nombre: Juan  
Cédula: 0987  
Universidad: UdlA  
Tesis: Animales  
Alumno de Magister agregado con éxito.  
  
----- OPCIONES -----  
1.- Mostrar listado general (Todos)  
2.- Ingresar Alumno pregrado  
3.- Ingresar Alumno magister  
4.- Ingresar Profesor hora  
5.- Listar Alumnos Pregrado  
6.- Listar Alumnos Magister  
7.- Listar Profesores  
8.- SALIR  
INGRESE UNA OPCION: 4  
Nombre: Arturo  
Cédula: 4567  
Especialidad: Calculo  
Horas: 48  
Profesor por Hora agregado con éxito.  
  
----- ALUMNOS DE PREGRADO -----  
Nombre='Matias', Cedula='1234'  
Universidad: UdlA  
Carrera='Ciber'  
  
----- PROFESORES POR HORA -----  
Nombre='Arturo', Cedula='4567'  
Especialidad='Calculo'  
Horas='48'  
Sueldo a pagar (20$/h): $960
```

3. Prueba de Robustez (Try-Catch)

Igual que en matriculación, probamos la seguridad del sistema ante entradas inválidas.

- **Acción:** En el menú principal, en lugar de un número, escribe una palabra (ej. "hola").
- **Resultado esperado:** El bloque catch debe atrapar el error, mostrar "Error: Ingrese un dato válido" y reiniciar el menú sin que el programa se cuelgue.

```
----- OPCIONES -----
1.- Mostrar listado general (Todos)
2.- Ingresar Alumno pregrado
3.- Ingresar Alumno magister
4.- Ingresar Profesor hora
5.- Listar Alumnos Pregrado
6.- Listar Alumnos Magister
7.- Listar Profesores
8.- SALIR
INGRESE UNA OPCION: hola
Error: Ingrese un dato válido.
```

Sistema de Registro de Métodos de Autenticación:

1. Prueba de "Unicidad Biométrica" (Regla de Negocio)

Tu código tiene una regla especial: un empleado solo puede tener **un** rostro registrado, pero muchas huellas o tokens. Esta prueba es vital.

- **Acción:**
 1. Registra un empleado.
 2. Agrega un Reconocimiento Facial (Opción 3).
 3. Intenta agregar otro Reconocimiento Facial al mismo empleado.
- **Resultado esperado:** El sistema debe impedirlo y mostrar: "El empleado ya tiene un rostro registrado.".

```
1
Ingrese la cedula del empleado que desea agregar:
1234
Ingrese el nombre:
Matias
```

```
3
Ingrese la cedula del empleado al que le quiere añadir el rostro
1234
Ingrese el patron del rostro:
gaswefaa
Ingrese el nivel de seguridad de su metodo:
3

3
Ingrese la cedula del empleado al que le quiere añadir el rostro
1234
Ingrese el patron del rostro:
qrqew
Ingrese el nivel de seguridad de su metodo:
3
El empleado ya tiene un rostro registrado.
```

2. Prueba de "Autenticación Real" (Acceso Concedido vs Denegado)

Aquí pruebas el núcleo del sistema: verificar si la "contraseña" (token/patrón) coincide.

- **Acción:**

1. Registra un Token para un empleado (ej. Token: "ABC-123").
2. Ve a la Opción 12 (Autenticar empleado).
3. **Intento Fallido:** Escribe el token incorrecto ("XYZ-999"). -> Debe decir: "Acceso denegado".
4. **Intento Exitoso:** Repite y escribe el token correcto ("ABC-123"). -> Debe decir: "Acceso concedido".

```
2
Ingrese la cedula del empleado al que le quiere añadir el token
1234
Ingrese el token:
ABC-123
Ingrese el nivel de seguridad de su metodo:
2
- - - - -
```

```
--- SISTEMA DE AUTENTICACIÓN ---  
Ingrese la cédula del usuario:  
1234  
¿Qué método usará? (Escriba: 'Huella Digital', 'Reconocimiento Facial' o 'Token')  
Token  
Ingrese el dato (Patrón, Huella o Token):  
XYZ-999  
Resultado: Acceso denegado
```

3. Prueba de "Filtro de Seguridad" (Umbral de Riesgo)

Esta prueba demuestra que tu sistema puede filtrar métodos débiles vs. fuertes.

- **Acción:**
 1. Al mismo empleado, agrégale una Huella con nivel de seguridad 1 y un Token con nivel de seguridad 10.
 2. Usa la Opción 11 (Métodos mayor a umbral).
 3. Ingresa un umbral de 5.
- **Resultado esperado:** El sistema solo debe listar el Token (porque $10 > 5$). La huella no debe aparecer.

```
4  
Ingrese la cedula del empleado al que le quiere añadir una huella  
1234  
Ingrese el patron de la huella:  
qmcil  
Ingrese el nivel de seguridad de su metodo:  
1  
4  
Ingrese la cedula del empleado al que le quiere añadir una huella  
1234  
Ingrese el patron de la huella:  
qmcil  
Ingrese el nivel de seguridad de su metodo:  
1  
  
11  
Ingrese la cédula del empleado:  
1234  
Ingrese el nivel de seguridad minimo (Umbral):  
5  
Métodos con seguridad superior a 5:  
Token
```

4. Prueba de "Polimorfismo Completo" (Reporte General)

Demuestra que un empleado es una entidad compleja que "contiene" varios métodos distintos.

- **Acción:**

1. Asegúrate de que el empleado tenga los 3 tipos: Huella, Rostro y Token.
 2. Usa la Opción 6 (Buscar empleado y mostrar datos) o la Opción 5.
- **Resultado esperado:** El `toString()` del empleado debe desplegar la lista completa formateada, mostrando: Tipo: Huella Digital, Tipo: Token, etc., todo bajo el mismo nombre.

```
4
Ingrese la cedula del empleado que quiere buscar:
1234
Nombre: Matias - Cédula: 1234
Métodos de Autenticación Asociados:
    -> Nivel de seguridad: 1
    Tipo: Huella Digital
    Patron Huella: gmcil
-----
    -> Nivel de seguridad: 10
    Tipo: Token
-----
    -> Nivel de seguridad: 6
    Tipo: Reconocimiento FacialPatron rostro: 1094
-----
```

5. Prueba de "Validación de Usuario" (Empleado Inexistente)

Verifica que no se le pueda asignar seguridad a fantasmas.

- **Acción:** Intenta agregar un Token (Opción 2) a una cédula que **no** has registrado (ej. "999999").
- **Resultado esperado:** El sistema debe responder: "No existe Empleado con esa cedula".

```
2
Ingrese la cedula del empleado al que le quiere añadir el token
99999
No existe Empleado con esa cedula
```

Sistema de Gestión de Proveedores:

1. Prueba de "Integridad Relacional" (Contrato sin Asociación)

Esta es la regla de negocio más importante: No puedes contratar servicios de un proveedor que no "conoces".

- Acción:

1. Crea un Cliente (Opción 2) y un Proveedor (Opción 1).
2. NO los asociés (sáltate la opción 3).
3. Intenta crear un contrato (Opción 4) entre ellos.

- Resultado esperado: El sistema debe buscar en la lista interna del cliente y decir: "Error: El proveedor no está asociado a este cliente.".

```
Ingresar opción: 2
----CREAR NUEVO CLIENTE----
Escriba un nombre para el nuevo cliente: Temu
Cliente Temu creado exitosamente!

Ingresar opción: 1
----CREAR PROVEEDOR----
Ingrese el nombre del Proveedor: ChinaCorp
Ingrese el país del proveedor: China

Escriba el nombre del proveedor con el que quiere generar un contrato: ChinaCor
Error: El proveedor no está asociado a este cliente.
```

2. Prueba de "Verificación de Tipos" (Polimorfismo)

Demuestra que el sistema sabe distinguir entre servicios Cloud, SaaS y Pagos.

- Acción:

1. Asocia un proveedor de tipo Cloud (Opción 1 -> tipo 1) a un cliente.
2. Usa la Opción 5 (Buscar proveedores según su tipo).
3. Pregunta si el cliente tiene proveedores de tipo "Pasarelas de Pago" (Opción 3).

- **Resultado esperado:** El sistema debe responder: "El cliente NO tiene proveedores de tipo Proveedor Pasarelas Pago".

```

Ingresar opción: 1
----CREAR PROVEEDOR----
Ingrese el nombre del Proveedor: Amazon
Ingrese el país del proveedor: US
Ingrese el tipo del proveedor (1.Cloud , 2.SaaS , 3.Pasarelas Pago): 1
Éxito: Se ha creado un Proveedor Cloud

Ingresar opción: 3
----ASOCIAR PROVEEDOR A CLIENTE----
Escriba el nombre del cliente que desea asociar (o 'salir'): JuanCorp
Escribe el nombre del proveedor para asociar (o 'salir'): Amazon
Proveedor agregado al cliente JuanCorp

Ingresar opción: 5
----BUSCAR PROVEEDORES SEGÚN SU TIPO----
Escriba el nombre del cliente: JuanCorp
Seleccione el tipo de proveedor:
1. Cloud
2. SaaS
3. Pasarelas de Pago
Opción: 3
El cliente NO tiene proveedores de tipo Proveedor Pasarelas Pago

```

3. Prueba de "Flujo Completo"

Esta captura demuestra que todo el sistema funciona unido.

- **Acción:**

1. Crea Cliente "Empresa X".
2. Crea Proveedor "Amazon AWS" (Cloud).
3. **Asocia** el proveedor al cliente (Opción 3).
4. Crea un **Contrato** de \$1000 por 12 meses (Opción 4).
5. Muestra el resultado final con la **Opción 6 (Listar contratos activos)**.

- **Resultado esperado:** Debe aparecer el nombre del Cliente, luego el del Proveedor y finalmente los datos del contrato (Precio y Duración) anidados.

```
Ingresar opción: 2
----CREAR NUEVO CLIENTE----
Escriba un nombre para el nuevo cliente: Empresa X
Cliente Empresa X creado exitosamente!

Ingresar opción: 1
----CREAR PROVEEDOR----
Ingrese el nombre del Proveedor: Amazon AWS
Ingrese el país del proveedor: US
Ingrese el tipo del proveedor (1.Cloud , 2.SaaS , 3.Pasarelas Pago): 1
Éxito: Se ha creado un Proveedor Cloud

Ingresar opción: 3
----ASOCIAR PROVEEDOR A CLIENTE----
Escriba el nombre del cliente que desea asociar (o 'salir'): Empresa X
Escribe el nombre del proveedor para asociar (o 'salir'): Amazon AWS
Proveedor agregado al cliente Empresa X

Ingresar opción: 4
----CREAR CONTRATOS----
Escriba el nombre de un cliente para buscar sus proveedores: Empresa X
----PROVEEDORES ASOCIADOS AL CLIENTE----
Nombre: Amazon AWS - País: US

Escriba el nombre del proveedor con el que quiere generar un contrato: Amazon AWS
Escriba la duración del contrato en meses: 12
Ingrese el monto a pagar por el contrato: 1000
Contrato agregado

-----MENU-----
1. Crear Proveedor
2. Crear Cliente
3. Asociar proveedores a los clientes
4. Crear contratos entre proveedores y clientes
5. Verificar si un cliente posee proveedores de un tipo determinado
6. Listar todos los contratos activos del sistema
7. Salir
Ingresar opción: 6
Total de contratos Activos en el Sistema
CLIENTE: Empresa X
Proveedor: Amazon AWS
Precio: 1000.0 - Duracion: 12
```

4. Prueba de "Unicidad de Clientes" (Validación)

- **Acción:**
 1. Crea un cliente llamado "Google".
 2. Intenta crear de nuevo al cliente "Google" (Opción 2).
- **Resultado esperado:** El sistema debe proteger la lista y decir: "Cliente ya existe".

```
----CREAR NUEVO CLIENTE----  
Escriba un nombre para el nuevo cliente: Google  
Cliente Google creado exitosamente!  
-----MENU-----  
1. Crear Proveedor  
2. Crear Cliente  
3. Asociar proveedores a los clientes  
4. Crear contratos entre proveedores y clientes  
5. Verificar si un cliente posee proveedores de un tipo determinado  
6. Listar todos los contratos activos del sistema  
7. Salir  
Ingresar opción: 2  
----CREAR NUEVO CLIENTE----  
Escriba un nombre para el nuevo cliente: Google  
Cliente ya existe
```

5. Prueba de "Robustez Numérica" (Try-Catch)

- **Acción:**
 1. Entra a crear un Proveedor (Opción 1).
 2. Cuando te pida el **tipo** (1, 2 o 3), escribe una letra como "A".
- **Resultado esperado:** Tu bloque try-catch debe interceptar la excepción, mostrar "Error: Debe ingresar un número válido" y regresar al menú sin explotar.

```
Ingresar opción: 1  
----CREAR PROVEEDOR----  
Ingrese el nombre del Proveedor: Temu  
Ingrese el país del proveedor: China  
Ingrese el tipo del proveedor (1.Cloud , 2.SaaS , 3.Pasarelas Pago): A  
Error: Debe ingresar un número válido.
```