

```
In [ ]: import csv
import os
import pandas as pd
```

Importaciones csv: Utilizado para trabajar con archivos CSV (valores separados por comas).
os: Utilizado para verificar si un archivo ya existe. pandas: Utilizado para cargar archivos CSV en un DataFrame y realizar manipulaciones de datos.

```
In [ ]: class Cedula:
    def __init__(self, numero, nombre, apellido, fecha_nacimiento, lugar_nacimiento):
```

Clase Cedula:

El constructor (**init**) inicializa una nueva instancia de la clase Cedula, con los siguientes atributos:

numero: El número de cédula que queremos que tenga exactamente 11 dígitos. nombre: Nombre de la persona. apellido: Apellido de la persona. fecha_nacimiento: Fecha de nacimiento de la persona en formato DD/MM/YYYY. lugar_nacimiento: (Opcional) Lugar de nacimiento de la persona. genero: (Opcional) Género de la persona.

```
In [ ]:     numero_str = str(numero)
    if len(numero_str) != 11 or not numero_str.isdigit():
        raise ValueError("El número de cédula debe tener exactamente 11 dígitos")
```

Esto convierte el número de cédula a cadena para verificar que tenga exactamente 11 dígitos y que sea un número válido. Si no cumple con estos criterios, lanza un ValueError.

```
In [ ]:     def formatear_cedula(self):
        """
        Devuelve el número de cédula en formato XXX-XXXXXXX-X.
        """
        numero_str = str(self.numero)
        return f"{numero_str[:3]}-{numero_str[3:10]}-{numero_str[10:]}"
```

Método formatear_cedula() Este método formatea el número de cédula para que se muestre en el formato XXX-XXXXXXX-X, dividiendo la cadena en bloques apropiados.

```
In [ ]:     def __str__(self):
        """
        Devuelve una representación legible de la cédula.
        """
        representacion = f"Cedula N°: {self.formatear_cedula()}\nNombre: {self.nombre}"
        if self.lugar_nacimiento:
            representacion += f"\nLugar de Nacimiento: {self.lugar_nacimiento}"
        if self.genero:
            representacion += f"\nGenero: {self.genero}"
        return representacion
```

Método **str()** este método especial devuelve una representación legible de la instancia de la clase Cedula, formateando los atributos en una salida estructurada. Incluye información opcional si está disponible, como el lugar de nacimiento y el género.

```
In [ ]: @classmethod
def from_input(cls):
    """
    Método de clase para capturar la información del usuario y crear una instancia de Cedula.
    """
```

Este método de clase captura los datos del usuario directamente desde la consola usando la función `input()`. Recoge los datos para todos los atributos de la clase y devuelve una nueva instancia de Cedula.

```
In [ ]: try:
    numero = int(input("Ingrese el numero de cedula (11 dígitos): "))
    nombre = input("Ingrese el nombre: ")
    apellido = input("Ingrese el apellido: ")
    fecha_nacimiento = input("Ingrese la fecha de nacimiento (DD/MM/YYYY): ")
    lugar_nacimiento = input("Ingrese el lugar de nacimiento (opcional): ")
    genero = input("Ingrese el genero (opcional): ")
```

Captura datos como el número de cédula, nombre, apellido, fecha de nacimiento, lugar de nacimiento y género. Si se proporciona un valor vacío para el lugar de nacimiento o género, se establece como `None`.

```
In [ ]: return cls(numero, nombre, apellido, fecha_nacimiento, lugar_nacimiento)

except ValueError as e:
    print(f"Error: {e}")
    return None
```

Si el número de cédula no es válido (por ejemplo, si no es un entero), captura la excepción y muestra un mensaje de error.

```
In [ ]: def guardar_en_csv(self, filename="cedulas.csv"):
    """
    Guarda los datos de la instancia en un archivo CSV.
    Si el archivo no existe, lo crea con la cabecera adecuada.
    """
```

Método `guardar_en_csv()` Este método guarda los datos de la instancia de Cedula en un archivo CSV.

```
In [ ]: archivo_existe = os.path.isfile(filename)
```

Verifica si el archivo CSV ya existe usando la función `os.path.isfile()`. Esto asegura que la cabecera solo se escriba si el archivo es nuevo.

```
In [ ]: with open(filename, mode='a', newline='') as file:
    escritor_csv = csv.writer(file)

    if not archivo_existe:
```

```
escritor_csv.writerow(['numero', 'nombre', 'apellido', 'fecha_nacim  
escritor_csv.writerow([self.numero, self.nombre, self.apellido, self.fe
```

Abre el archivo en modo append ('a') para añadir información sin sobrescribir. Si el archivo no existe, escribe la cabecera con las columnas numero, nombre, apellido, fecha_nacimiento. Luego, escribe los datos de la cédula en una nueva fila del archivo CSV.

```
In [ ]: def leer_cedulas_csv(filename="cedulas.csv"):
        """
        Lee el archivo cedulas.csv y lo carga en un DataFrame de pandas.
        """
        try:
            df = pd.read_csv(filename)
            print(df)
            return df
        except FileNotFoundError:
            print(f"El archivo {filename} no existe.")
            return None
```

Función leer_cedulas_csv() Esta función utiliza la librería Pandas para leer el archivo cedulas.csv y cargarlo en un DataFrame. Si el archivo no existe, captura la excepción y muestra un mensaje.

```
In [ ]:
```