



# **Universidad Autónoma de Baja California**

## **Facultad de Ingeniería, Arquitectura y Diseño**

Ingeniero en Computación

### **Asignatura:**

Programación Estructurada

### **Actividad Curp:**

Librerías

**Brayan Arturo Rocha Meneses**

### **Matricula:**

371049

**Ensenada Baja California 23 de Octubre del 2023**

## **Introducción:**

Realiza operaciones relacionadas con la generación de la Clave Única de Registro de Población (CURP) en México. La CURP es un identificador único asignado a cada ciudadano mexicano y residente permanente. El programa solicita información personal como nombres, apellidos, fecha de nacimiento, sexo, estado de nacimiento, y matrícula, y utiliza esta información para generar y guardar la CURP

## **Competencia:**

El programa demuestra competencia en el manejo de entrada de usuario, validación de datos, manipulación de cadenas, y escritura en archivos. Además, incluye funciones específicas para validar la fecha de nacimiento, el estado de nacimiento y la matrícula. La inclusión de abreviaturas de estados y la validación de la matrícula sugieren un enfoque hacia la generación de la CURP de manera robusta.

## **Fundamentos:**

### **Estructuras de Datos:**

Curp para organizar los datos personales.

### **Entrada y Validación:**

Se emplean diversas funciones para solicitar información al usuario y validarla, como la fecha de nacimiento, el estado y la matrícula.

### **Manipulación de Cadenas:**

Se implementan funciones para manipular y procesar cadenas, convirtiendo letras a mayúsculas y extrayendo consonantes de apellidos y nombres.

### **Algoritmo de Generación de CURP:**

La función constructor implementa el algoritmo para generar la CURP utilizando las reglas establecidas, incluyendo el manejo de palabras inconvenientes.

### **Escritura en Archivos:**

Se utiliza la función guardarDatosEnArchivo para almacenar los datos ingresados y la CURP generada en un archivo de texto llamado 'datos.txt'.

### **Validación de Sexo:**

La función validar\_sexo garantiza que el usuario ingrese un valor válido para el sexo (1 para hombre, 2 para mujer). Se podría mejorar incluyendo un mensaje de error si se ingresa un valor incorrecto.

### **Abreviaturas de Estados:**

Se utiliza un array bidimensional abreviaturas para almacenar las abreviaturas de los estados. Esto facilita la asignación del estado de nacimiento en la CURP, proporcionando una interfaz más amigable al usuario.

### **Función de Validación de Cadenas:**

La función `vali_cadena` se encarga de verificar que una cadena contenga solo letras, ignorando espacios. Esto es esencial para asegurar que los nombres y apellidos ingresados sean válidos.

### **Uso de Estructuras de Control:**

Se emplean bucles `while` para validar la entrada del usuario, garantizando que solo se acepten valores válidos para la fecha de nacimiento, el estado, la matrícula y el sexo.

### **Separación de Responsabilidades:**

El código sigue un enfoque modular, dividiendo las tareas en funciones específicas como `ingresa_nombre`, `ingresa_apellido_pat`, `ingresa_apellido_mat`, `mayusculador`, `consonantes`, y `constructor`. Esto facilita la comprensión y mantenimiento del código.

### **Manejo de Archivos:**

La función `guardarDatosEnArchivo` utiliza operaciones de archivo para abrir, escribir y cerrar el archivo `'datos.txt'`. Sería beneficioso agregar manejo de errores para abordar posibles problemas durante estas operaciones.

### **Mensajes Interactivos:**

Se utilizan mensajes descriptivos para guiar al usuario durante la entrada de datos. Estos mensajes ayudan a mejorar la experiencia del usuario y aclaran las expectativas.

### **Manejo de Errores de Entrada:**

Se han implementado mecanismos para manejar errores de entrada, como la verificación de que la matrícula sea un valor entero y la validación de la fecha de nacimiento, proporcionando mensajes informativos en caso de errores.

### **Documentación de Código:**

El código podría beneficiarse de comentarios adicionales para explicar el propósito y la funcionalidad de secciones específicas, facilitando la comprensión y colaboración futura.

### **Usabilidad y Experiencia del Usuario:**

El programa podría mejorarse incluyendo una interfaz más interactiva y amigable para el usuario, proporcionando mensajes claros y detallados sobre el progreso y cualquier error que pueda ocurrir durante la ejecución. Esto contribuiría a una mejor experiencia del usuario.

**Procedimiento:**

# **CURP**

## **CADENAS.**

### **INSTRUCCIONES:**

- 1.- APLICA TODAS LAS FUNCIONES QUE REALIZASTE EN LA ACTIVIDAD 7 PARTE 2.
- 2.- REALIZA REPORTE DE PRACTICA
- 3.- SUBE TODOS LOS ARCHIVOS (REPORTE PRACTICA Y PROGRAMA)

## **CURP**

Realiza un programa que sirva para generar el **CURP** de una persona.  
El programa debe pedir los datos al usuario, generar, almacenar en una cadena y desplegar el CURP.  
**El programa deberá repetirse cuantas veces desee el usuario**

**Nota:** el programa deberá estar **100% validado**, de datos entrada así como las reglas que e deben cumplir al generar el curp

### **DATOS PARA PROBAR FUNCIONAMIENTO DE CURP**

**FRAIRE**

**LOPEZ**

**JOSE**

**OMAR**

**1997**

**31**

**ENERO**

**HOMBRE**

**GUERRERO**

**CURP: FXLO970131HGRRPM02**

**CHAN**

**CWEN**

**2002**

**MARZO**

**28**

**HOMBRE**

**NACIDO EN EL EXTRANJERO**

**CURP: CAXC020328HNEHXWA2**

**DE LA LOMA**

**XIU**

**JOSE**

ANDRES

2007

DICIEMBRE

28

HOMBRE

YUCATAN

**CURP: LOXA071228HYNMXNA3**



## **Conclusiones:**

El código presenta una implementación funcional para generar la CURP a partir de la información personal proporcionada por el usuario.

**Anexo:**

<https://github.com/BrayanArM98/Estructurada>