



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Ingeniero en Computación

Asignatura:

Programación Estructurada

Actividad 9/4:

CURP

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 14 de Abril del

2024

Introducción:

Para la actividad de elaboración para el programa de la CURP, estuve abarcando diversos temas vistos previamente en clases, entre ellos: declaración y ejecución de funciones, arreglos, cadenas, entre otros.

Competencia:

Usar la lógica para la resolución de este ejercicio, hacer las validaciones correspondientes y de los conocimientos previos obtenidos en clase y prácticas. Permitirá crear programas avanzados sin tener que depender constantemente de buscar en Internet o consultar libros. En otras palabras, estaremos programando de manera más eficiente y lógica.

Fundamentos:

```
int main()
{
    int op;
    char curp[20];
    srand(time(NULL));
    system("CLS");
    printf("ACTIVIDAD 9 3/4\n");
    do {
        printf("Generador de CURP: \n");
        imprimir_curp(curp);
        reemplazar_palabras_tabu(curp);
        printf("CURP generada: %s\n", curp);
        op = ValidarCadena("\nDeseas buscar otra curp(SI = 1, NO = 2): \n", 1, 2);
    } while (op != 2);
    return 0;
}
```

```
void imprimir_curp(char curp[19])
{
    char pri_ape[20], seg_ape[20];
    pedir_apellidos(pri_ape, seg_ape);
    char pri_nom[20], seg_nom[20], ter_nom[20];
    pedir_nombres(pri_nom, seg_nom, ter_nom);
    int anio[4], mes[2], dia[2];
    pedir_fecha_nacimiento(anio, mes, dia);
    char estado[30], estado_abreviado[3];
    pedir_estado(estado, estado_abreviado);
    char genero[2];
    pedir_genero(genero); eliminar_palabras_compuestas(pri_ape);
    eliminar_palabras_compuestas(seg_ape);
    eliminar_palabras_compuestas(pri_nom);
    eliminar_palabras_compuestas(seg_nom);
    eliminar_palabras_compuestas(ter_nom);
    char primer_consonante_ape = buscar_primera_consonante_interna(pri_ape);
    char primer_consonante_seg_ape = buscar_primera_consonante_interna(seg_ape);
    char primer_consonante_pri_nom = buscar_primera_consonante_interna(pri_nom);
}
```

Procedimiento:

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

Instructivo:

<http://www.ordenjuridico.gob.mx/Federal/PE/APF/APC/SEGOB/Instructivos/InstructivoNormativo.pdf>

Conclusiones:

1. Manejo de cadenas de texto: La generación de la CURP implica una gran cantidad de manipulación de cadenas de texto, lo que puede ayudarte a familiarizarte con las funciones de manejo de cadenas en C++.
2. Validación de datos: Para generar una CURP válida, necesitas validar los datos de entrada, como el nombre, la fecha de nacimiento y el sexo. Esto puede ayudarte a entender la importancia de la validación de datos en la programación.
3. Uso de estructuras de control: La generación de la CURP también requiere el uso de varias estructuras de control, como los bucles y las declaraciones condicionales.
4. Formato y presentación de datos: Finalmente, generar una CURP te permite practicar el formato y la presentación de datos en la salida del programa.

Anexo: