

Ejercicio Ayudantía

INF2223 – Estructura de datos
I Semestre 2025

Información evaluación		Nota
Nombre y apellido:		
Signatura:	INF2223 – Estructura de datos	
Fecha:		

Enunciado

La Federación de Estudiantes de la Universidad VCUP agrupa a múltiples Centros de Estudiantes (correspondientes a cada carrera). Cada centro posee una lista simplemente enlazada de Integrantes CEE, quienes representan a las distintas generaciones de estudiantes.

Actualmente, la federación enfrenta desafíos relacionados con la actualización de los datos: algunas carreras tienen registros incompletos o desactualizados, dificultando así un catastro real de la conformación de sus centros.

Para considerar un Centro de Estudiantes como correctamente conformado, se establecen las siguientes condiciones:

- Debe tener exactamente 5 integrantes, que cumplan los roles de: Presidente, Vicepresidente, Secretario, Tesorero y Gestor de Comunicaciones.
- Todos los integrantes deben pertenecer a una generación igual o posterior a 2020.

Con el fin de llevar un control actualizado, se solicita desarrollar funciones utilizando las siguientes estructuras:

```
struct Estudiante {
    char *nombre;
    int anioGeneracion;
    int tam;
};

struct IntegranteCEE {
    struct Estudiante *datoCoordinador;
    struct IntegranteCEE *sig;
}; /* Simplemente enlazada */

struct CentroEstudiantes {
    char *nombreCarrera;
    struct IntegranteCEE *head;
};

struct NodoCentro {
    struct CentroEstudiantes *datoCentro;
    struct NodoCentro *sig;
}; /* Simplemente enlazada */

struct Federacion {
    char *nombreFederacion;
    struct NodoCentro *head;
};
```

Se solicita que implemente la siguiente función:

```
void limpiarCentrosIncompletos(struct Federacion *federacion);
```

Esta función debe recorrer todos los centros de estudiantes de una federación. Para cada centro, debe verificar que tenga exactamente 5 integrantes y que ningún estudiante pertenezca a una generación anterior a 2020. En caso de que un centro no cumpla ambas condiciones (por exceso/falta de integrantes o por generación inválida), debe convertir en NULL a todos sus integrantes, dejando sin integrantes el Centro de Estudiantes, pero sin eliminar la estructura.

Consideraciones obligatorias

- Debe modularizar las soluciones e implementar lo solicitado.
- Comente los supuestos en caso de ser necesario
- Respete las reglas y buenas prácticas de programación y desarrollo.

```

void limpiarIntegrantes(struct IntegranteCEE *head) {
    struct IntegranteCEE *actual = head;
    while (actual != NULL) {
        if (actual->datoCoordinador != NULL) {
            actual->datoCoordinador = NULL;
        }
        actual = actual->sig;
    }
}

```

```

int contarIntegrantes(struct IntegranteCEE *head, int *hayAntiguo) {
    int contador = 0;
    *hayAntiguo = 0;
    struct IntegranteCEE *actual = head;

    while (actual != NULL) {
        contador++;
        if (actual->datoCoordinador != NULL && actual->datoCoordinador->anioGeneracion < 2020) {
            *hayAntiguo = 1;
        }
        actual = actual->sig;
    }

    return contador;
}

```

```

void limpiarCentrosIncompletos(struct Federacion *federacion) {
    if (federacion == NULL) return;

    struct NodoCentro *centroActual = federacion->head;
    while (centroActual != NULL) {
        int hayAntiguo = 0;
        int cantidad = contarIntegrantes(centroActual->datoCentro->head, &hayAntiguo);

        if (cantidad != 5 || hayAntiguo) {
            limpiarIntegrantes(centroActual->datoCentro->head);
        }
        centroActual = centroActual->sig;
    }
}

```

```
}
```

```
centroActual = centroActual->sig;
```

```
}
```

```
}
```