



## **Agenda 2030 y los Objetivos de Desarrollo Sostenible (ODS)**

La Agenda 2030 para el Desarrollo Sostenible es un plan de acción a favor de las personas, el planeta y la prosperidad, que también tiene la intención de fortalecer la paz universal y el acceso a la justicia. La Agenda plantea 17 Objetivos (ODS - Objetivos de Desarrollo Sostenible) que abarcan las esferas económica, social y ambiental. Al adoptarla, los Estados se comprometieron a movilizar los medios necesarios para su implementación mediante alianzas centradas especialmente en las necesidades de los más pobres y vulnerables.

Los Objetivos de Desarrollo Sostenible (ODS) constituyen un llamamiento universal a la acción para poner fin a la pobreza, proteger el planeta y mejorar las vidas y las perspectivas de las personas en todo el mundo.

El ODS 6 se refiere al agua potable y al saneamiento, y tiene como objetivo garantizar la disponibilidad de agua y su gestión sostenible para todas las personas en el mundo. 884 millones de personas en todo el mundo todavía no tienen acceso a agua potable. El agua libre de impurezas y accesible para todos es un sueño. Hay suficiente agua dulce en el planeta; sin embargo, el reparto del agua no es el adecuado y para el año 2050 se espera que al menos un 25 % de la población mundial viva en un país afectado por escasez crónica de agua dulce. La sequía afecta a algunos de los países más pobres del mundo, recrudece el hambre y la desnutrición.

El ejercicio que se propone para esta prueba de evaluación continua está relacionado con este ODS 6

## **Estudio del consumo de agua en comunidad**

Con el objetivo de tomar medidas para un consumo de agua sostenible, eficiente y responsable, un administrador de fincas que trabaja con varias comunidades de vecinos nos solicita realizar un programa que permita confeccionar estudios de consumo de agua dentro de diferentes comunidades de vecinos, con distintos números de viviendas.

Para ello, recogerá en un vector de registros los metros cúbicos de agua suministrados a cada vivienda a lo largo de un año. Cada registro representa una vivienda del bloque o comunidad en estudio y contendrá un campo con los metros cúbicos de agua suministrados a la respectiva vivienda a lo largo de un año.

Así, una parte del software a implementar será:

1. Crear, en lenguaje C, el siguiente tipo estructurado, `struct tipoVivienda`, que contendrá los siguientes campos:
  - Campo `idCatastro`, cadena de caracteres de longitud 21 caracteres, incluyendo carácter nulo.
  - Campo `propietario`, cadena de caracteres de longitud 80 caracteres, incluyendo el carácter nulo.
  - Campo `direccion`, cadena de caracteres de longitud 128 caracteres, incluyendo el carácter nulo.



- Campo superficie, valor de tipo float con la superficie en metros cuadrados de la vivienda.
- Campo consumos vector de tipo float, de tamaño constante y definido por la constante NUM\_MESES y cuyas celdas contendrán el consumo de agua, expresado en metros cúbicos, realizado por la vivienda en cada uno de los NUM\_MESES.

2. Se debe implementar la siguiente función:

```
int procesarConsumos(struct tipoVivienda* viviendas, int numV,  
                    float * conMedioViviendas, float * conMaxViviendas,  
                    float * conMedioMeses, float * conMaxMeses);
```

que tiene la siguiente lista de parámetros formales:

- Primer parámetro, viviendas, un vector de registros de tipo struct tipoVivienda y en el que cada registro representa una vivienda del bloque o comunidad de vecinos en estudio. Este vector con datos de entrada contiene todos los valores ya cargados: identificador de catastro, propietario, dirección, superficie y los valores de consumos de agua de la vivienda, expresados en metros cúbicos, en el campo consumos.
- Segundo parámetro, numV, un entero indicando el tamaño efectivo del vector viviendas.
- Tercer parámetro, conMedioViviendas, vector de celdas de tipo float en el que, en cada celda, se devolverá el promedio de consumo de agua de cada vivienda. El tamaño de este vector será el mismo que el del parámetro viviendas.
- Cuarto parámetro, conMaxViviendas, vector de celdas de tipo float en el que, en cada celda, se devolverá el valor máximo de consumo de agua de cada vivienda. El tamaño de este vector será el mismo que el del parámetro viviendas.
- Quinto parámetro, conMedioMeses, vector de celdas de tipo float, en el que, en cada celda, se devolverá el promedio de consumos de agua de las numV viviendas por cada mes. El tamaño del vector que se pasará en la invocación está definido por la constante NUM\_MESES.
- Sexto parámetro, conMaxMeses, vector de celdas de tipo float en el que, en cada celda, se devolverá el valor máximo de consumo de agua del conjunto de viviendas por cada mes. El tamaño del vector que se pasará en la invocación está definido por la constante NUM\_MESES.

Se establecen las siguientes precondiciones, a evaluar al comienzo de la ejecución de la función:

- si el parámetro viviendas, es un puntero NULL, la función finalizará devolviendo un -1



- si el tamaño del vector, numV es 0 o negativo, o superior al tamaño físico del vector de viviendas, representado por la macro NUM\_VIVIENDAS, la función debe finalizar devolviendo un -2
- si uno cualquiera de los parámetros conMedioViviendas, conMaxViviendas, conMedioMeses, conMaxMeses, es un puntero NULL, la función finalizará devolviendo un -3

La función debe realizar:

- Un primer recorrido del vector de registros calculando, para cada vivienda, el promedio y el máximo de consumo de agua a partir de los datos de consumos almacenados en el campo consumos. El resultado del cálculo de consumo medio se almacenará en la correspondiente celda del vector recibido a través del parámetro conMedioViviendas. Adicionalmente, se debe identificar el consumo de agua máximo realizado por cada vivienda en los NUM\_MESES, almacenando dicho máximo en la respectiva celda del vector conMaxViviendas.
- En un segundo recorrido, se debe calcular, para cada mes, el promedio de consumos de agua realizados por todas las viviendas y se identificará el consumo máximo realizado por una de las viviendas en ese mes. Por cada celda del campo consumos se realizará un recorrido del vector de registros viviendas para calcular el promedio de consumos cada mes. El resultado se almacenará en la celda correspondiente del vector conMedioMeses. Adicionalmente, se debe identificar el consumo de agua máximo realizado por una vivienda en cada mes, almacenando dicho máximo en la respectiva celda del vector conMaxMeses.

Si la función tiene éxito en su tarea devolverá un valor int con un 0.

3. Finalmente se escribirá un programa completo donde poder probar el funcionamiento de esta función. Para facilitar las pruebas, se recomienda escribir las siguientes funciones, muy similares a las ya implementadas en “Enunciados ejercicios adicionales” del Tema 1:

```
void cargarRegistrosAleatorios(struct tipoVivienda *v,  
                             int num);
```

Esta función recibe un vector de registros de tipo struct tipoVivienda y cargará todos los campos de dichos registros con valores aleatorios. Para implementar esta función se puede tomar como base o ejemplo la equivalente del ejercicio de struct empleados ya realizado en sesiones anteriores.

```
void mostrarRegistros(struct tipoVivienda *v, int num);
```

Esta función recibe un vector de registros de tipo struct tipoVivienda y mostrará por pantalla el contenido de todos los campos de dichos registros.