

PORTAFOLIO EJERCICIO DE CÁTEDRA N° 2

FECHA DE ENTREGA AL ESTUDIANTE: 10 de mayo

FECHA DE ENTREGA AL DOCENTE: 22 de mayo

CONDICIONES DEL PORTAFOLIO:

El ejercicio #2 del Portafolio de Aprendizaje del curso EIF200 Fundamentos de Informática comprende el tema de la resolución de problemas usando arreglos unidimensionales. Se presenta un problema que el estudiante debe implementar en Zinjal, y posteriormente compartirlo en el Google Drive (GD) que ha creado con el objetivo de compartir sus portafolios con su profesor y el asistente del curso. El archivo debe tener el nombre del estudiante y número de grupo además del número de portafolio, por ejemplo: Port2G10RosaRojas. El ejercicio se trabaja de manera individual, cualquier plagio se calificará con 0, como lo establece el artículo 24 del Reglamento General sobre los procesos de Enseñanza y Aprendizaje de la Universidad Nacional.

ENUNCIADO DEL PROBLEMA:

En el país “ENSUEÑO” se llevó a cabo este año un proceso electoral en el cual participaron 20 candidatos a la Presidencia. Dicho país cuenta con un millón de habitantes (1.000.000). Los resultados obtenidos en cada elección se encuentran almacenados en un arreglo unidimensional. Cada 4 años el número de participantes (candidatos presidenciales) puede cambiar. Cada posición del arreglo corresponde a un candidato. El Tribunal Supremo de Elecciones de dicho país desea obtener algunas estadísticas con el fin de mostrar los resultados finales.

Se le ha solicitado a usted que haga un programa en C++ que tenga las siguientes funcionalidades:

1. **(Valor 5 pts)** Crear una clase **Coleccion** que permita almacenar los datos de las votaciones que se realizan en el país, de tal manera que pueda adaptarse a los distintos procesos de elecciones por los que pasa el país cada 4 años.
2. **(Valor 5 pts)** Crear los métodos básicos de la clase **Coleccion** (get, set, constructores, destructor).
3. **(Valor 10 pts)** Como son un **millón** de posibles votantes, la suma de los valores del vector no puede exceder esa cantidad. Se deberá llenar el vector de la clase utilizando la función RAND, de la siguiente manera:
 - a. En la primera posición se almacenará cualquier valor entre 0 y 1000000. Ejemplo 67000.
 - b. En la segunda posición se almacenará cualquier valor entre 0 y 933000. (1000000-67000=933000). Ejemplo 125000
 - c. En la tercera posición se almacenará cualquier valor entre 0 y 808000 (1000000 - 67000 - 125000 = 808000).

Así se almacenará para cada candidato un valor entre 0 y una cantidad de votos resultante de restarle a un millón las cantidades asignadas a las posiciones anteriores, para garantizar que los resultados no excedan a la población electora.

4. (**Valor 5 pts**). Un método **toString()** que permita desplegar el valor de todos los elementos de la colección.
5. (**Valor 5 pts**) Un método que devuelva ¿cuál fue la posición del vector que obtuvo mayor número de votos? (suponga que no hubo empate).
6. (**Valor 10 pts**) Un método **toStringOrdenado()** que permita desplegar la cantidad de votos obtenidos en orden ascendente, de menor a mayor.
7. (**Valor 5 pts**) Un método **toStringMayor()** que permita desplegar cuáles posiciones obtuvieron 30% o más del total de los votos emitidos.
8. (**Valor 5 pts**) Un método **hayGanador()** que devuelva true si al menos un candidato obtuvo el 40% o más de los votos emitidos y false si ninguno lo obtuvo.
9. (**Valor 5 pts**) Un método **abstencionismo()** que devuelva el abstencionismo que hubo en el proceso electoral (cantidad de votantes que no votaron en el proceso).
10. (**Valor 10 pts**) Un método **declaracionGanador()** que declare al ganador de la elección, o determine si hay un segundo proceso. Esto ocurre si ningún participante llegó al 40% de los votos emitidos, y se deben escoger los dos que obtuvieron la mayor cantidad de votos para que participen en una segunda ronda.
11. (**Valor 10 pts**) Un método **toStringCompara()** que reciba como parámetro un objeto de tipo **Coleccion** y muestre datos comparativos de ambas colecciones. Debe mostrar la cantidad de candidatos participantes, el total de votos emitidos, el candidato ganador y el nivel de abstencionismo, como mínimo.

En el main():

12. (**Valor 3 pts**) Crear, con el constructor sin parámetros, un objeto **ELECCION-PROCESO1-2018** de la clase **Coleccion**.
13. (**Valor 5 pts**) Llamar los métodos creados en los puntos 3 al 10 y desplegar un mensaje con el resultado de cada uno.
14. En caso que el resultado del método del punto 8 sea false, se deberá:
 - a- (**Valor 3 pts**) Crear un segundo objeto de la clase **Coleccion** con el constructor sin parámetros llamado objeto **ELECCION-PROCESO2-2018**.
 - b- (**Valor 3 pts**) Llenar el vector de este objeto llamando el método 3, suponga que participan de nuevo todos los 20 candidatos que participaron en la primera ronda.
 - c- (**Valor 3 pts**) Llamar el método 5. Desplegar un mensaje indicando cuál fue el número del candidato ganador.
 - d- (**Valor 3 pts**) Llamar el método 9. Desplegar un mensaje indicando la cantidad de abstencionismo que hubo en el segundo proceso.
15. (**Valor 5 pts**) Cree otro objeto de tipo **Coleccion** con los datos de una elección pasada y muestre los resultados comparativos, como se especifica en el punto 11.