

Arreglos, listas y vectores. ejercitación adicional programación orientada a objetos.

a. Llevar a cabo un programa que incluya una *lista* que almacene los nombres: Frodo, Sam, Meriadoc, Peregrin, Gimli, Legolas, Gandalf, Aragorn, Boromir. Realizar un recorrido de *lista* para cargar los datos(que los mete el usuario), otro recorrido para mostrar los mismos por pantalla, luego elimine los datos “Gandalf” y “Boromir” y presente la lista resultante. Por último, repita este ejercicio con un *arreglo*.

b. Realizar un programa que permita la carga de 10 números en una *lista* o *vector*. Una vez cargados, se necesita que el programa determine cuál es el mayor y cuál es el menor de ellos.

c. Realizar un programa que permita la carga de n números (es decir, cuantos el usuario quiera) en un *vector*, un *arreglo* ó una *lista*. Una vez cargados, se necesita que el programa cuente e informe por pantalla cuántas veces se cargó el número 3 y por otro lado cuántos de esos números son primos.

d. Realiza un programa que permita la carga de los puntajes obtenidos en 12 niveles de un juego. Calcule y muestre:

- La suma total de los puntajes
- El promedio de puntajes obtenidos
- El puntaje más alto y el más bajo.
- Compare el promedio de puntajes de los primeros seis niveles niveles 1-6) con los últimos 6 (niveles 7-12)
- Calcule las variaciones de puntaje entre niveles consecutivos.
- Los niveles en los que el puntaje fue superior al promedio general.

e. Usted está desarrollando un juego de deportes, el “pro evolution futbol”, para almacenar las estadísticas de los jugadores en un torneo define 3 variables:

- Nombres de los jugadores
- Desempeño en número de goles
- Distancia recorrida en la cancha durante un torneo.

Se necesita un programa que permita la carga de la información para las variables y evalúe e informe por pantalla las estadísticas de los jugadores con mayor número de goles y que hayan recorrido mayor distancia (No necesariamente será el mismo jugador).

f. Desarrolle un sistema para gestionar recursos en un juego de estrategia y gestión de recursos. Use una *lista* para almacenar los recursos, cada uno representado por una clase **Recurso** con atributos como nombre, cantidad y tipo, puede implementar una clase

TablaRecursos que use una lista para almacenar, modificar, mostrar y eliminar varios recursos y que puedan accederse a los mismos.

g. Cree un sistema para gestionar misiones en un juego RPG. Use una *lista, arreglo o vector* para almacenar misiones (representada por una clase **Mision**) con atributos como nombre, descripción y estado (pendiente, completado) puede implementar una clase **DiarioDeMision** que use una lista de **Mision** para gestionar los eventos.