

ENUNCIADOListaOrdenada.pdf



Gengar



Programación Orientada a Objetos



2º Grado en Ingeniería Informática



**Escuela Superior de Ingeniería
Universidad de Cádiz**

Estudiar **sin publi** es posible.

Compra Wuolah Coins y que nada te distraiga durante el estudio.



Lista Ordenada

Se quiere implementar la clase Lista Ordenada de número en coma flotante y doble precisión (posibles repetidos) con las operaciones que se declaran:

```
class ListaOrdenada
{
public:
    typedef list<double>::const_iterator; // bidireccional
    listaOrdenada();                    // Construir una lista vacía
    void insertar(double e);             // Insertar elemento
    iterator buscar(doublé e) const;    // Iterador al primer elemento e
    iterator begin() const;              // Primera posición de la lista
    iterator end() const;                // Posición siguiente a la del último
private:
    //...
};
```

Para ello se usarán la clase genérico estándar list y se propone 2 opciones:

- Definir ListaOrdenada como una especialización de list.
- Definir ListaOrdenada como una composición de list y usar la técnica de delegación de operaciones.

a) Implemente ListaOrdenada seleccionando la opción que considere más adecuada y explique por qué.

b) Añada un nuevo método `size_t contar(double e) const`, que cuenta el número de ocurrencias de un elemento dado. Utiliza para ello el algoritmo de la STL `count_if` () el cual recibe dos iteradores y un predicado (clase objeto función cuyo operador () devuelve un bool). Defina el predicado como una función anónima (lambda).

*Nota: Se recuerda que el contenido estándar para listas posee funciones miembros `begin` (), `end` (), `cbegin` (), `cend` (), `insert` (), `erase` (), etc.

*Nota2: No contiene `find` ()



WUOLAH