



Fundamentos de Programación

Grado en Ingeniería Informática

- Septiembre 2018 -

Dpto. Tecnologías de la Información
Escuela Técnica Superior de Ingeniería - Universidad de Huelva

NOTAS

La puntuación máxima de este examen es de 5 puntos

(Modelo A)

I. Introducción.

Puntuación máxima 5 puntos de los 10 de la asignatura.

La práctica consiste en la programación de un programa para la gestión de una receta de cocina. Para ello contamos con una estructura **Ingrediente**, una clase **Receta** (se especificarán más adelante) y un programa principal que utilizará ambas estructuras.

II. Definición de las Estructuras de Datos

a) Estructura Ingrediente.

Esta estructura es utilizada para almacenar los datos de un ingrediente que será utilizado en una receta. La información que almacena en sus campos, son **nombre**, peso en **gramos**, **cantidad** de producto y **calorías** del ingrediente. La estructura de dicha clase es la siguiente:

```
struct Ingrediente
{
    Cadena Nombre;
    float Gramos;
    int Cantidad;
    int Calorias;
};
```

b) Clase Receta.

Esta clase es utilizada para gestionar los ingredientes de una única receta además de los atributos necesarios que nos dan información sobre su nombre, dificultad, y número de personas que pueden comer con ella.

```
class Receta
{
    Cadena Nombre;
    Ingrediente Elementos[MAX_INGREDIENTES];
    int NIngredientes;
    int Dificultad;
    int NPersonas;

public:
    Receta();
    void ActualizarDatos(Cadena pNombre, int pNDificultad,
                        int pNPersonas);           //0.25 puntos
    int Buscar(Cadena pNombre);                   //1 punto
    Ingrediente get_Ingrediente(int pPos);
    int get_NIngredientes();
    void get_DatosAdicionales(Cadena pNombre, int &pDificultad,
                             int &pNPersonas);    //0.25 puntos
};
```

```
void set_Ingrediente(int pPos, Ingrediente pIng);  
bool eliminarIngrediente(Cadena pNombre);           // 1.5 puntos  
bool anadirIngrediente(Ingrediente pIng);           // 1 punto  
};
```

Los atributos de la clase son los siguientes:

Nombre: contiene el nombre de la receta.

Elementos: Vector cuyos elementos son estructuras **Ingrediente**.

NIngredientes: Atributo que contiene cuantos elementos del vector **Elementos** son correctos y contiene ingredientes de la receta.

Dificultad; Valor entero de 0 a 10 que indica el nivel de dificultad para realizar la receta. 0 muy fácil y 10 muy difícil.

NPersonas: Atributo que contiene cuantas personas pueden comer cuando esta receta esté cocinada.

La descripción de los métodos es la siguiente:

Receta()

Método constructor que inicializa los atributos de la clase a un valor inicial, 0 para los numéricos y "" para las cadenas.

void ActualizarDatos(Cadena pNombre, int pNDificultad, int pNPersonas)

(Implementar 0.25 puntos) Método que actualiza los atributos de la clase con los datos pasados por parámetro e inicializa el atributo número de ingredientes a cero.

int Buscar(Cadena pNombre)

(Implementar 1 punto) Método que busca la posición dentro del vector de ingredientes de aquel cuyo nombre coincide con el nombre pasado por parámetro. Si no se encuentra el ingrediente el método devolverá -1.

Ingrediente get_Ingrediente(int pPos)

Método que devuelve el ingrediente del vector cuya posición es pasada en el parámetro **pPos**.

int get_NIngredientes()

Método que devuelve el número de ingredientes de la receta.

void get_DatosAdicionales(Cadena pNombre, int &pDificultad, int &pNPersonas)

(Implementar 0.25 puntos) Método que devuelve los atributos de la clase nombre de la receta, la dificultad y el número de personas que comerán con la receta.

void set_Ingrediente(int pPos, Ingrediente pIng)

Añade el ingrediente pasado en el por parámetro **pIng** en el vector de ingredientes en la posición indicada por el parámetro **pPos**.

bool eliminarIngrediente(Cadena pNombre)

(Implementar 1.5 puntos) Método que elimina el ingrediente cuyo nombre es pasado por parámetro. Si encuentra el ingrediente, el método deberá desplazar el resto de ingredientes de la receta una posición hacia abajo. Si el método encuentra y elimina el ingrediente devolverá **true**, en caso contrario devolverá **false**.

bool anadirIngrediente(Ingrediente pIng)

(Implementar 1 punto) Método que añade un nuevo ingrediente al final del vector de ingredientes siempre y cuando dicho ingrediente no exista previamente en la receta. El método devuelve **true** si ha podido añadir el nuevo ingrediente y **false** si no lo ha añadido.

III. Programa Principal

El programa principal está compuesto por un menú cuya estructura y descripción es la siguiente:

El Menú consta de 5 opciones incluida la opción de salir. El programa no finalizará hasta que la opción salir sea seleccionada.

```
MENU PRINCIPAL
*****

1.- Actualizar Datos de la Receta
2.- Añadir un Ingrediente
3.- Eliminar un Ingrediente
4.- Mostar la Receta (Datos e Ingredientes)
5.- Salir
Elige Opción:
```

Opción 1: Esta opción actualizará el nombre, dificultad y número de comensales de la receta.

Opción 2: Esta opción se encarga de añadir un ingrediente a la receta. Para ello pedirá todos los datos de un Ingrediente y los añadirá a la receta. Si se ha podido añadir correctamente, se mostrará por pantalla un mensaje indicándolo y si no, mostrará un mensaje que el ingrediente ya existe en la receta.

Opción 3: Esta opción se encarga de eliminar un ingrediente de la receta. Para ello pedirá el nombre de la receta y procederá a eliminarla. Si ha podido eliminarla satisfactoriamente, emitirá un mensaje por pantalla indicándolo, si no mostrará un mensaje indicando que no se ha encontrado el ingrediente.

Opción 4: (Implementar 1 punto) Esta opción mostrará por pantalla los datos de la receta (nombre, dificultad, nº de personas y número de ingredientes). A continuación, mostrará todos los datos de cada ingrediente de la receta y terminará el listado mostrando **el número de calorías de la receta por gramo de comida de ésta**.

Opción 5: Esta opción es la única que hace que el programa termine.