Práctica Final: El cocinero integral

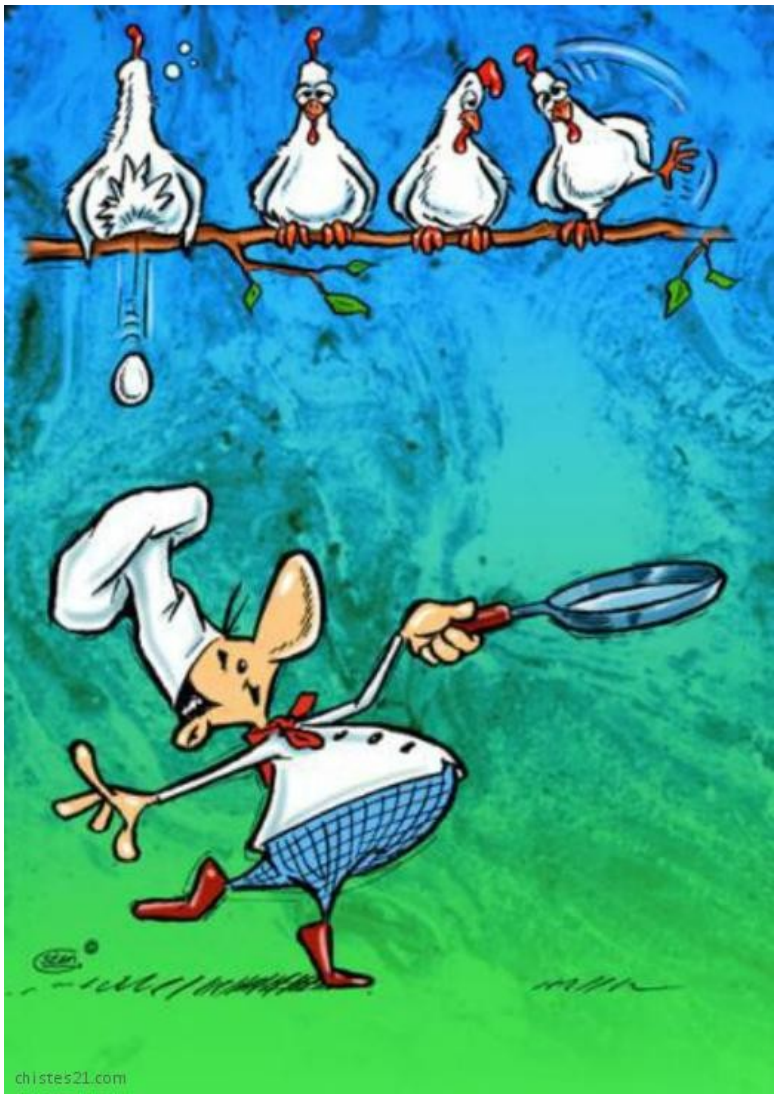
Dpto. Ciencias de la Computación e Inteligencia Artificial
E.T.S. de Ingenierías Informática y de Telecomunicación
Universidad de Granada

DECSAI

Estructuras de Datos

Grado en Ingeniería Informática
Doble Grado en Informática y ADE
Doble Grado en Informática y Matemáticas

Índice de contenido



1. Introducción

Los objetivos de este guión de prácticas son los siguientes:

- Resolver un problema eligiendo la mejor estructura de datos para las operaciones que se solicitan

Los requisitos para poder realizar esta práctica son:

1. Haber estudiado el Tema 1: Introducción a la eficiencia de los algoritmos
2. Haber estudiado el Tema 2: Abstracción de datos. Templates.
3. Haber estudiado el Tema 3: T.D.A. Lineales.
4. Haber estudiado el Tema 4: STL e Iteradores.
5. Haber estudiado estructuras de datos jerárquicas: Árboles

2. Objetivo.

El objetivo de esta práctica es llevar a cabo el análisis, diseño e implementación de un proyecto. Con tal fin el alumno abordará un problema donde se requiere estructuras de datos que permiten almacenar grandes volúmenes de datos y poder acceder a ellos de la forma más eficiente.

3. Ejercicio

Dado un conjunto de recetas, un usuario sobre una receta podrá obtener:

1. Los ingredientes para elaborarla con sus cantidades
2. El valor nutricional de la receta
3. Los pasos para elaborar la receta. Un ejemplo de salida que cumpla con los objetivos 1-3 se muestra en el recuadro a la derecha.
4. Adicionalmente el usuario dado un conjunto de recetas podrá seleccionar dos recetas para fusionar, y sobre esta nueva receta fusión deberá obtener igualmente los ingredientes, información nutricional y pasos a a seguir.

```
CODE:R9 RECETA:Macarrones con Tomate PLATO: 1
Ingredientes:

       Macarrones 250
       Tomate Frito 400
       Cebolla 25
       Queso Mozzarella 25
       Sal 1
       Aceite Oliva 10

Información Nutricional:

       Calorias:1355.65
       Hidratos de Carbono:210.75
       Grasas:43
       Proteina:44.25
       Fibra:19.75

Pasos a seguir:

       Cocer   Macarrones
       Reservar
       Calentar        Aceite Oliva
       Pelar   Cebolla
       Cortar
       Mezclar
       Reservar
       Mezclar
       Add     Tomate Frito
       Add     Sal
       Add     Queso Mozzarella
       Hornear
```

4. Programa cocinero_integral

Este programa tiene un doble objetivo:

1. Obtiene toda la información de una receta: ingredientes, información nutricional y pasos a seguir para realizar la receta.
2. Fusiona dos recetas como un nueva receta.

Un ejemplo de llamada a este programa desde la línea de órdenes sería:

```
prompt% bin/cocinero_integral datos/Acciones.txt datos/recetas.txt datos/ingredientes.txt
datos/instrucciones/
```

Los parámetros de entrada son los siguientes:

1. El nombre del fichero con las acciones posibles a realizar en la cocina.
2. El nombre del fichero con todas las recetas
3. El nombre del fichero con todos los ingredientes alimenticios
4. La ruta donde se encuentra los ficheros con las instrucciones

En primer lugar la ejecución de este comando muestra los códigos, nombres y número de plato de las recetas leídas. Un ejemplo de salida sería el siguiente:

```
CODE:R1  NOMBRE: Ensalada Mixta       PLATO:1
CODE:R10 NOMBRE: Cocido               PLATO:1
CODE:R11 NOMBRE: Merluza al Horno     PLATO:2
CODE:R12 NOMBRE: Tofu a la Pimienta   PLATO:2
CODE:R13 NOMBRE: Bacalao con Tomate   PLATO:2
CODE:R14 NOMBRE: Guisante con Huevo   PLATO:2
CODE:R15 NOMBRE: Hamburguesas de Tofu PLATO:2
CODE:R16 NOMBRE: Cuscus con Pimiento  PLATO:2
….
```

A continuación el programa pedirá al usuario un código de receta. En nuestro ejemplo le hemos insertado la receta con código R1 y el programa ha mostrado los ingredientes, valor nutricional y pasos a seguir de la receta.

```
CODE:R1 RECETA: Ensalada Mixta PLATO: 1

Ingredientes:

       Lechuga 200
       Tomate 50
       Pepino 50
       Cebolla 50
       Aceite Oliva 5
       Vinagre De Vino 10
       Sal 1

Información Nutricional:

       Calorias:92.85
       Hidratos de Carbono:7.1
       Grasas:5
       Proteina:3.5
       Fibra:5.5

Pasos a seguir:

       Cortar  Lechuga
       Cortar  Tomate
       Mezclar
       Pelar   Pepino
       Cortar
       Mezclar
       Pelar   Cebolla
       Cortar
       Mezclar
       Add     Aceite Oliva
       Add     Vinagre De Vino
       Add     Sal
```

A continuación se le pide al usuario que introduzca el código de dos recetas para fusionar. En nuestro ejemplo se ha insertado los códigos R1 y R9 obteniendo la siguiente receta fusión:

```
CODE:F_R1_R9 RECETA:Fusion Ensalada Mixta y Macarrones con Tomate PLATO: 1

Ingredientes:

        Lechuga 200
        Tomate 50
        Pepino 50
        Cebolla 75
        Aceite Oliva 15
        Vinagre De Vino 10
        Sal 2
        Macarrones 250
        Tomate Frito 400
        Queso Mozzarella 25

Información Nutricional:

        Calorias:1448.5
        Hidratos de Carbono:217.85
        Grasas:48
        Proteina:47.75
        Fibra:25.25

Pasos a seguir:

        Cortar  Lechuga
        Cortar  Tomate
        Mezclar
        Pelar   Pepino
        Cortar
        Mezclar
        Pelar   Cebolla
        Cortar
        Mezclar
        Add     Aceite Oliva
        Add     Vinagre De Vino
        Add     Sal
        Cocer   Macarrones
        Reservar
        Calentar        Aceite Oliva
        Pelar   Cebolla
        Cortar
        Mezclar
        Reservar
        Mezclar
        Add     Tomate Frito
        Add     Sal
        Add     Queso Mozzarella
        Hornear
        Acompañar
```

5. Tareas a realizar

Antes de abordar el programa descrito en la sección 4 el alumno/a deberá desarrollar diferentes tipos de datos y realizar los test correspondientes. Así las tareas a realizar son:

1. Desarrollar y probar el TDA Acciones
2. Desarrollar y probar el TDA Instrucciones.
3. Modificar el TDA Receta para que incluya el conjunto de instrucciones para poder implementar la receta. Añadir a receta un método que dada otra receta construya un nueva receta como la fusión de la original y la que le pasamos como parámetro.
4. Recuperar los TDA Recetas, Ingredientes e Ingrediente desarrollado en la práctica 3. Adaptar los TDAs.
5. Construir el programa cocinero_integral descrito en la sección 4

5.1. TDA Acciones

Un objeto de tipo **acciones** es una colección de pares formado por una acción (p.e: cocer, mezclar, añadir, pelar, cortar, etc) junto con la ariedad (número de operandos necesarios) de la acción. Es decir la acción puede ser unaria o binaria si necesita 1 o 2 ingredientes básicos o ingredientes procesados por otras acciones. Por ejemplo la acción ***"pelar"*** es unaria, así podemos decir "Pelar Patatas" (se aplica sobre un solo ingrediente). O la acción puede ser binaria se aplica sobre dos ingredientes por ejemplo:

```
Mezclar Arroz  Tomate
```

En este ejemplo se esta aplicando la acción ***"Mezclar"*** sobre dos ingredientes básicos. Pero podríamos tener la siguiente secuencia:

```
Cortar  Lechuga
Cortar  Tomate
Mezclar
```

En esta secuencia primero se corta la lechuga en segundo lugar se corta el tomate y a continuación, la acción "***Mezclar***" se aplica sobre los dos ingredientes previamente cortados.

Una posible representación eficiente de acciones podría se la siguiente:

```
//FICHERO acciones.h
class acciones{
  private:
    map<string,unsigned  char > datos;
…
}
//FIN DE acciones.h
```

5.2. TDA Instrucciones

Un objeto de tipo **instrucciones** contiene la secuencia de pasos que deben darse para cocinar una receta. Los pasos se almacenarán en un árbol binario. Los nodos de este árbol binario se forman por un conjunto de nodos hojas que serán los ingredientes base de la receta y el resto de nodos serán acciones del conjunto de acciones permitidas (ver sección 5.1). Por lo tanto una posible representación sería:

```
//FICHERO instrucciones.h
class instrucciones{
….
  private:
     ArbolBinario<string>datos;
     acciones acc[1];

…
}
//FIN DE instrucciones.h
```

La complejidad de este TDA reside en como a partir de la secuencia de instrucciones obtener

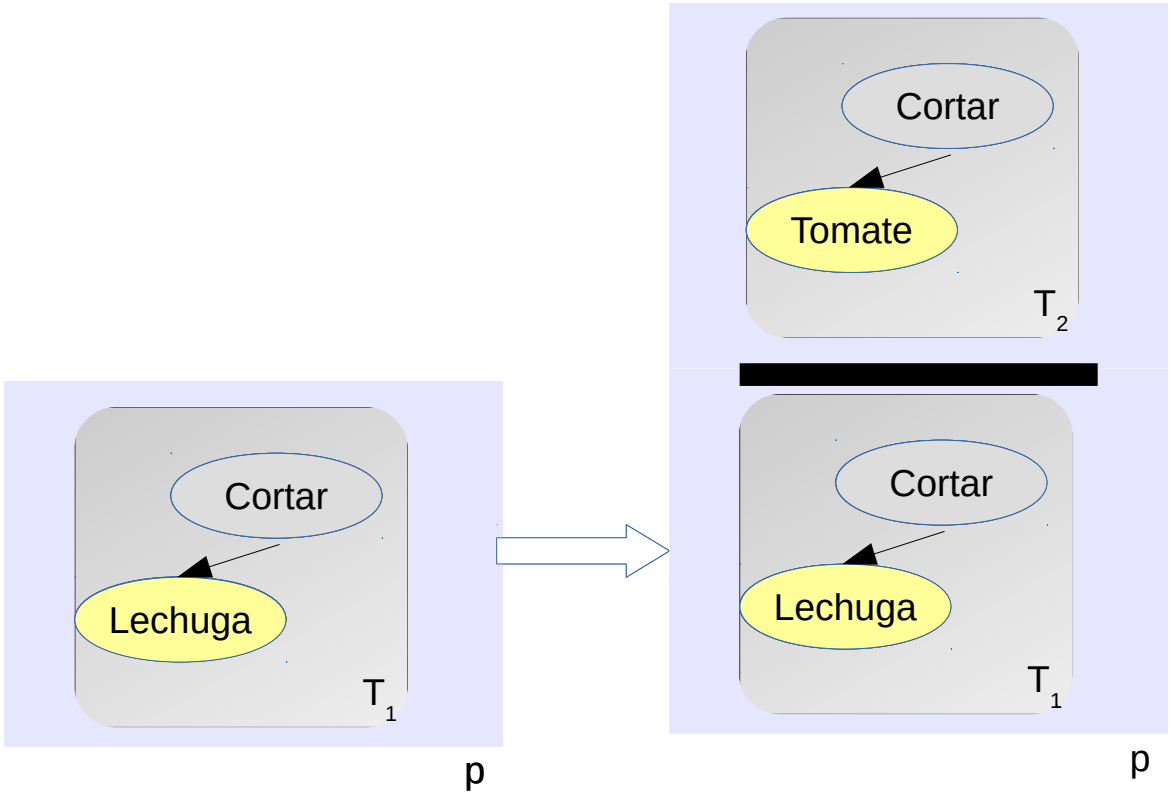
1 *Pensad que las acciones son comunes a todas las recetas y por lo tanto a todas las instrucciones. De forma que con una sola ocurrencia de acciones para todas las instrucciones sería suficiente. Por lo tanto se podría definir como* ***static acciones &acc;***

el objeto ArbolBinario asociado (datos). Las instrucciones se almacenan en un fichero como una secuencia de pares (acción, ingredientes). El número de ingredientes pueden ser 1 o 2 según la ariedad de la acción. Pero podría no aparecer ningún ingrediente a continuación de la acción, en este caso la acción se aplica sobre ingredientes previamente procesados. En nuestro programa las instrucciones las cargaremos en memoria sobre un objeto ***instrucciones*** (en particular se almacenarán sobre un objeto de tipo árbol binario).

Veamos con un ejemplo como se logra pasar de la descripción en el fichero de las instrucciones al árbol binario correspondiente. Suponer la siguiente secuencia de instrucciones:

```
Cortar   Lechuga
Cortar   Tomate
Mezclar
```

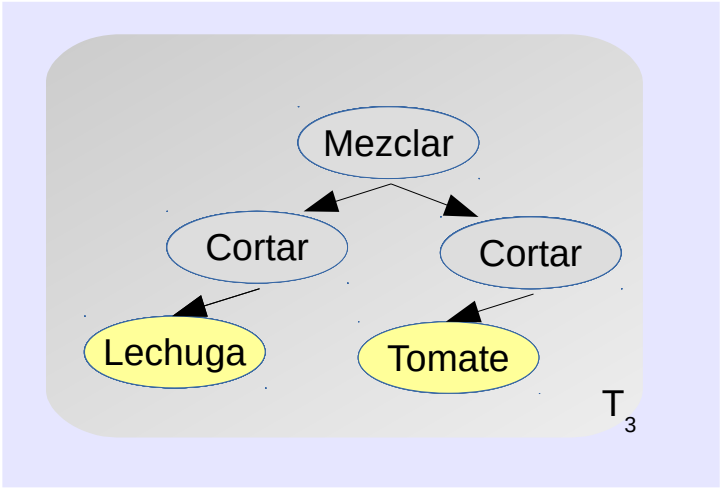
Nuestro árbol binario ***datos*** esta inicialmente vacío. Al ver la primera línea ***"Cortar Lechuga"*** se reconoce la acción ***"Cortar"*** y como ingrediente ***"Lechuga"*** por lo tanto crea el árbol correspondiente y lo mete en una pila. Sea la pila **p**, que vamos a necesitar en nuestro proceso de construcción.



Con la primera sentencia ***"Cortar Lechuga"*** se crea el árbol binario T_1 y se pone en la pila **p**[2]. A continuación con la sentencia ***"Cortar Tomate"*** se crea el árbol binario T_2 y se pone en la pila **p.** Cuando aparece la sentencia ***"Mezclar",*** consultamos al conjunto de acciones su ariedad, siendo en este caso binaria. Por lo tanto necesitamos para poder aplicarla dos ingredientes. Estos dos ingredientes los vamos a tomar de la pila creando el árbol T_3 y poniéndolo en el tope de la pila. Para crear el árbol T_3, creamos un árbol con un solo nodo cuya etiqueta es ***"Mezclar".*** A continuación tomamos el árbol T_2 del tope de la pila que lo

2 *Para construir el árbol binario la acción sería la etiqueta de la raíz del árbol. Si solamente después de la acción hay un ingrediente, este será la etiqueta del hijo izquierda. En el caso que hubiese dos ingredientes uno se coloca como la etiqueta del hijo a la izquierda y el segundo ingrediente sería la etiqueta del hijo a la derecha.*

insertamos como hijo derecho de T_3 y de nuevo del tope de la pila sacamos T_1 que lo insertamos como hijo izquierdo de T_3.

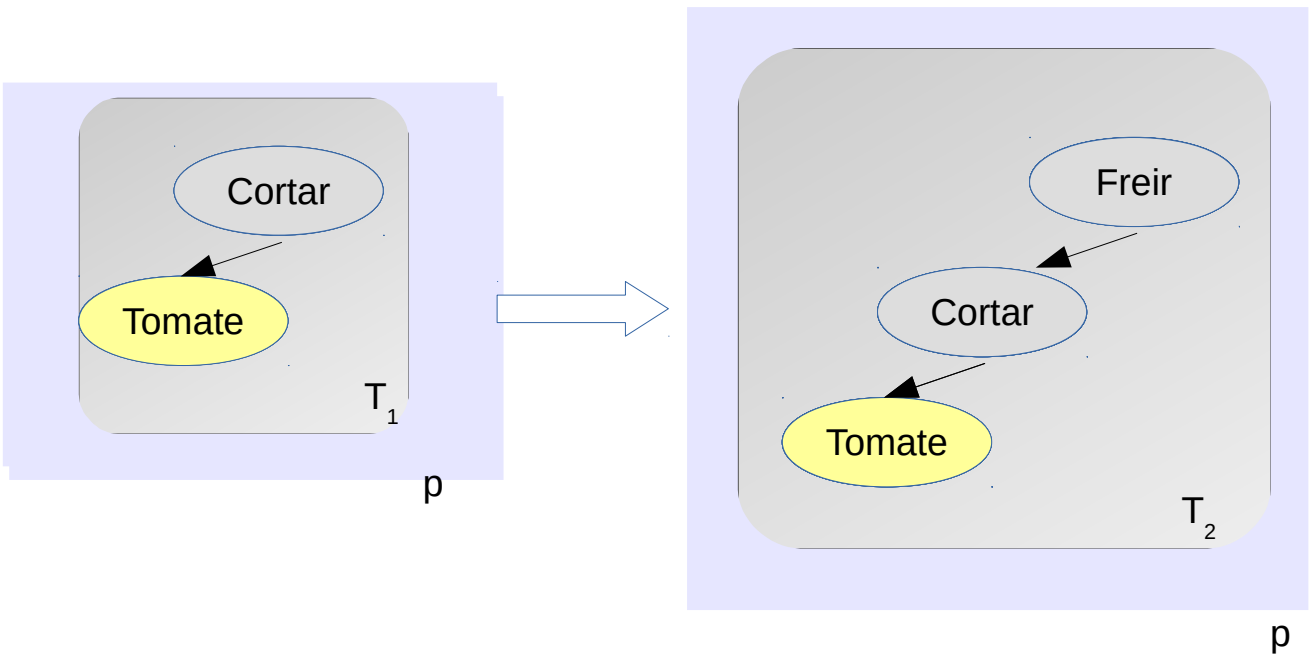


Si ya no tenemos más sentencias que procesar, en el tope de la pila tenemos el árbol binario asociado a las instrucciones.

En el caso que la operación en vez de binaria fuese unaria se cogería el tope de la pila y se insertaría como hijo a la izquierda. Un caso de esta situación ocurren con las siguientes instrucciones:

```
Cortar  Tomate
Freir
```

La evolución de la pila sería la que se muestran a continuación:



Otro operador de especial interés en el TDA Instrucciones es el operador de salida (<<). Este operador tiene que obtener la secuencia de instrucciones (como se muestran en los ficheros de instrucciones) a partir de los datos almacenados en el árbol binario. Una posible solución para llevar a cabo la implementación de este operador es recorrer el árbol binario en postorden. Cuando se encuentra en un nodo hoja (que contendrá como etiqueta el nombre de un ingrediente) simplemente la almacenamos en una variable. Cuando nos encontramos en un nodo interior (que hará referencia a un nombre de una acción) se escribirá en el flujo de salida, en la misma línea, la acción junto con los ingredientes leídos previamente (como máximo dos).

5.3. TDA Receta

Un objeto de tipo receta se compone de un nombre, código, plato y una lista de ingredientes con las cantidades en gramos necesarias. Además contiene un conjunto de instrucciones necesarias para su elaboración.

Una posible representación de receta sería la siguiente:

```
//FICHERO receta.h
class receta{
….
 private:
     string code;
     unsigned int plato;
     string nombre;
     list<pair<string,unsigned int> > ings;
     float calorias,hc,grasas,proteinas,fibra;
     instrucciones inst;
...
  public:
  ...
      class iterator {
      private:
       list<pair<string,unsigned int> >::iterator  it;
      public:

      };
      class const_iterator {
      private:
        list<pair<string,unsigned int> >::const_iterator  it;
      public:
        const_iterator(){}
       ...
      };

  ...
      iterator begin();

      iterator end();

      const_iterator begin()const;

      const_iterator end()const ;
```

```
    friend istream& operator>>(istream& in, receta & r );

    friend ostream& operator<< (ostream& out, const receta & r);

...
};
//end receta.h
```

6. Ficheros

6.1. Fichero con los ingredientes

Para poder probar nuestro programa usaremos un fichero compuesto de una serie de líneas. Cada línea se corresponde con la información de un ingrediente

```
Alimento (100 gramos);Calorias (Kcal.);Hidratos de Carb.;Proteinas;Grasas;Fibra;Tipo
Ketchup;98;24;2;0;0;Procesado
Salmon;182;0;18;12;0;Pescado
Cereales Cornflakes;368;85;9;2;11;Cereal
Gallo;81;0;17;1;0;Carne
Queso Emmental;377;0;29;29;0;Leche-Derivados
Cerezas;47;12;0;2;0;Fruta
Gatorade;39;10;0;0;0;Bebida
Salsa Ketchup;98;24;2;0;0;Procesado
Esparragos Enlatados;14;1;2;0;0;Verdura
Endibias;11;1;2;0;0;Verdura
Harina Trigo, Panificada;337;75;11;1;3;Cereal
Nuez Brasil;617;4;12;62;9;Frutos Secos
Gofio Millo;377;83;6;5;0;Cereal
Pavo;107;0;22;2;0;Carne
Mantequilla;740;0;0;82;0;Aceite
Yogur Liquido;78;11;3;2;0;Leche-Derivados
Cacao Polvo;357;11;20;24;38;Semillas
Arenque Ahumado;205;0;26;11;0;Pescado
….
```

El fichero contiene:

1. En la primera línea es un comentario. Indicando que atributos tendrá cada uno de los alimentos.
2. A continuación en cada línea viene la información de cada ingrediente. Cada atributo de un ingrediente se encuentra separada por “;”. Los atributos son
    - Nombre de ingrediente (puede tener más de una palabra)
    - Calorías por cada 100 gramos del ingrediente
    - Porcentaje de hidratos de carbono
    - Porcentaje de proteínas
    - Porcentaje de grasas
    - Porcentaje de fibra

- Tipo de ingrediente

En el directorio datos el alumno encontrará el fichero “ingredientes.txt” con un conjunto de ingredientes.

6.2. Ficheros con las recetas

El fichero con las recetas contiene un línea por receta. Cada receta se describe por:

1. Un código de receta. A continuación un carácter ';'
2. Un número indicando si es un primer, segundo o tercer plato.
3. El nombre de la receta
4. Y un conjunto de pares separados ';'. Cada par se compone del nombre de un ingrediente y un número que indica la cantidad del ingrediente en gramos.

Un ejemplo de fichero de recetas es el siguiente:

```
R1;1;Ensalada Mixta;Lechuga 200;Tomate 50;Pepino 50;Cebolla 50;Aceite Oliva 5;Vinagre De Vino 10;Sal 1
R2;1;Ensaladilla Rusa;Patata cocida 100;Zanahoria 50;Judia Verde 50;Huevo Duro 100;Mayonesa 125;Sal 1
R3;1;Revuelto de Acelgas;Acelgas 250;Huevo 50;Ajo 4;Pimiento 15;Aceite Oliva 25;Sal 1
R4;1;Salteado de Brocoli con Jamon;Jamon Serrano 100;Brocoli 500;Ajo 20;Aceite Oliva 20;Sal 1
R5;1;Crema de Calabacin;Patata 250;Pimiento 100;Cebolla 200;Calabacin 250;Aceite Oliva 20;Queso Quark
50;Sal 1
R6;1;Salmorejo;Huevo Cocido 50;Tomate 125;Vinagre De Vino 5;Aceite Oliva 5;Ajo 5;Sal 1;Pan Blanco 75
R7;1;Ajo Blanco;Almendras 100;Ajo 10;Pan Blanco 50;Agua 1000;Aceite Oliva 15;Vinagre De Vino 20;S
```

En el directorio datos del material tenéis un ejemplo de fichero de recetas en recetas.txt.

6.3. Fichero con las acciones

El fichero con las acciones posibles para crear una receta se compone de una serie de líneas. Cada línea especifica una acción formada por el nombre y la ariedad de la acción. La ariedad se refiere al número de ingredientes (básicos o procesados) que necesita la acción para ejecutarse. Un ejemplo de fichero podría ser el siguiente:

```
Pelar 1
Add 2
Calentar 1
Reservar 1
Triturar 1
Cocer 1
```

En este ejemplo todas las acciones son unarias excepto *Add* (añadir) que sería binaria.

En el directorio datos del material podéis encontrar un ejemplo de fichero de acciones en Acciones.txt.

6.4. Fichero con las instrucciones

Un fichero con las instrucciones contiene una serie de líneas compuesta por una acción y a continuación puede venir 0 o más ingredientes. Un ejemplo de fichero de instrucciones sería el que se muestra en el siguiente recuadro:

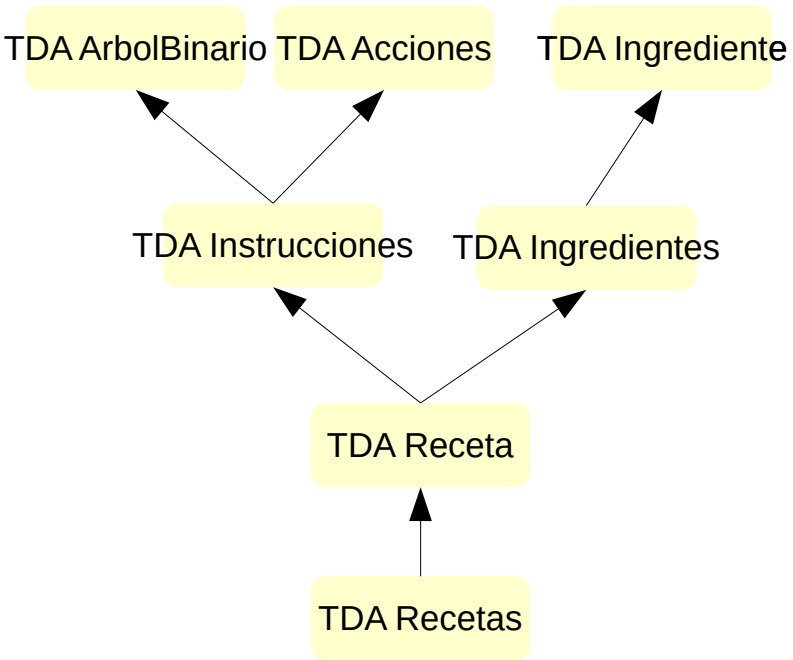
```
Cortar Lechuga
Cortar Tomate
Mezclar
Pelar Pepino
Cortar
Mezclar
Pelar Cebolla
Cortar
Mezclar
Add Aceite Oliva
Add Vinagre De Vino
Add Sal
```

El número de ingredientes necesarios para la acción de la instrucción se indicó en el fichero acciones cargado.

El alumno en el directorio datos/instrucciones puede ver las instrucciones para las distintas recetas que se muestran en el fichero recetas.txt. Para relacionar la receta con el fichero de instrucciones se ha usado el codigo de la receta y se le ha añadido m. Así para la receta con codigo R1 sus instrucciones están en el fichero ***R1m.txt.***

7. Esquema del Proyecto

En el siguiente esquema se ve los TDA que como mínimo debería tener el proyecto:



En este esquema se indica los TDA existentes en el proyecto y las dependencias de un TDA con otro. Por ejemplo el TDA Receta depende de TDA Instrucciones y TDA Ingredientes, ya que una receta debe contener un conjuntos de instrucciones. Por otro lado una receta necesita conocer la información de los ingredientes para formalizar el valor nutricional de la receta. A su vez el TDA Instrucciones depende del conjunto de acciones y del TDA

ArbolBinario que es donde se almacenará en memoria las instrucciones.

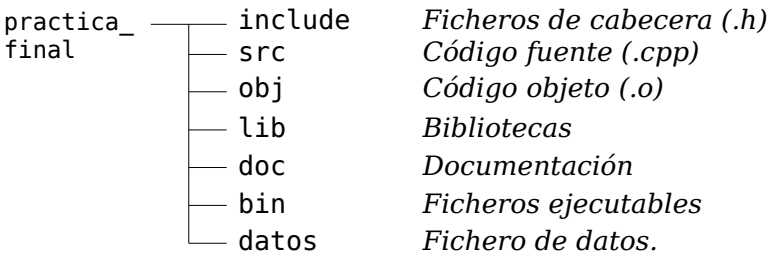
Por otro lado los pasos fundamentales de nuestro programa cocinero_integral serán los siguientes:

1. Cargar en memoria las acciones
2. Cargar en memoria los ingredientes
3. Cargar en memoria las recetas
4. Obtener el valor nutricional de todas las recetas usando los ingredientes
5. Escribir el código, nombre y plato de todas las recetas
6. Preguntar al usuario sobre que código de receta quiere obtener la información
7. Mostrar la información completa de la receta. Esto implica mostrar los nombres de los ingredientes con sus cantidades, valor nutricional de la receta y pasos a seguir para crear la receta
8. Pedir al usuario dos códigos de recetas a fusionar
9. Obtener una nueva receta fusión creada a partir de las dos recetas dadas en el paso 8.
10. Mostrar toda la información completa de la receta fusión como se hizo en el paso 7.

8. Práctica a entregar

El alumno deberá empaquetar todos los archivos relacionados en el proyecto en un archivo con nombre "practica_final.*tgz*" y entregarlo antes de la fecha que se publicará en la página web de la asignatura. Tenga en cuenta que no se incluirán ficheros objeto, ni ejecutables, ni la carpeta datos. Es recomendable que haga una "limpieza" para eliminar los archivos temporales o que se puedan generar a partir de los fuentes.

El alumno debe incluir el archivo *Makefile* para realizar la compilación. Tenga en cuenta que los archivos deben estar distribuidos en directorios*:*



Para realizar la entrega, en primer lugar, realice la limpieza de archivos que no se incluirán en ella, y sitúese en la carpeta superior (en el mismo nivel de la carpeta "*practica_final*") para ejecutar:

```
prompt% tar zcv practica_final.tgz practica_final
```

tras lo cual, dispondrá de un nuevo archivo practica_final.tgz que contiene la carpeta letras así como todas las carpetas y archivos que cuelgan de ella.