

Ingeniería De Sistemas Basados En Conocimientos

Práctica 1. Sistema CBR



Miguel Alexander Maldonado Lenis



Índice

Estructura del proyecto.....3

Funcionamiento.....8

Restricciones.....11

Estructura del proyecto

En esta sección se describirá la estructura de paquetes del proyecto.

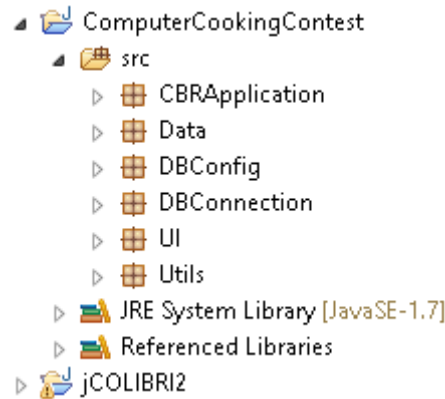


Ilustración 1

Paquete CBRApplication

El paquete **CBRApplication** contiene las clases **SandwichDescription.java**, **SandwichRecommender.java** y **SandwichSolution.java**

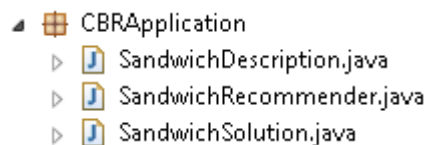


Ilustración 2

SandwichDescription.java contiene la representación de los casos.

```
private String caseID;  
private String ingredient1;  
private String ingredient2;  
private String ingredient3;  
private String ingredient4;  
private String ingredient5;
```

Ilustración 3

SandwichRecommender.java representa la clase principal de la aplicación y contiene los métodos que se encargan de realizar el ciclo CBR (Configure, PreCycle, Cycle y PostCycle).

SandwichSolution.java contiene la representación de la solución (casos recuperados).

```
private String caseID;  
private String ingredient1;  
private String ingredient2;  
private String ingredient3;  
private String ingredient4;  
private String ingredient5;
```

Ilustración 4

Paquete Data

El paquete **Data** contiene un conjunto de clases que representan de manera lógica la siguiente jerarquía de alimentos:



Ilustración 5

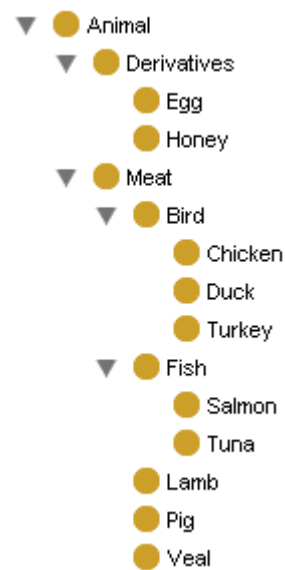


Ilustración 6

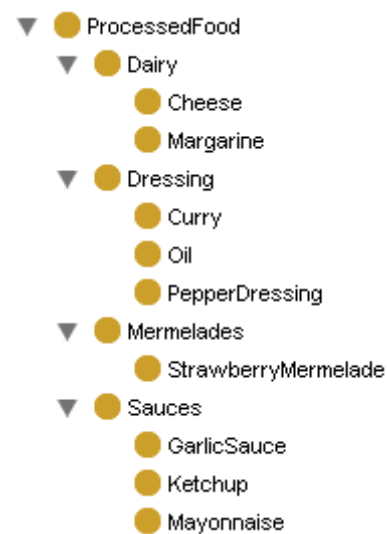


Ilustración 7

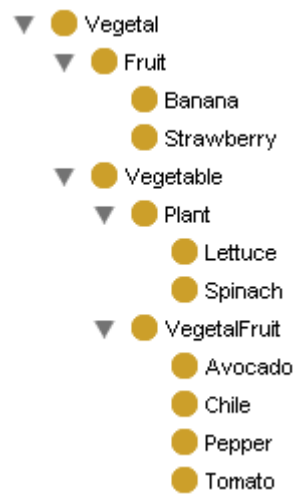


Ilustración 8

Cabe destacar el método ***getLastChildsInHierarchy***, implementado por todas las clases, el cual permite acceder a las hojas de cualquier rama de la jerarquía pudiendo, de esta forma, acceder a los ingredientes “finales”.

Por ejemplo:

Partiendo de la categoría **Vegetable**, el método devuelve las hojas: **Lettuce**, **Spinach**, **Avocado**, **Chile**, **Pepper** y **Tomato**.



Ilustración 9

Paquete DBConfig

El paquete **DBConfig** contiene un conjunto de clases necesarias para la configuración de la BBDD, así como la configuración del mapeo entre los atributos de los casos y las columnas de la tabla en la BBDD, la configuración de **Hibernate** (librería encargada de leer y escribir de la BBDD)

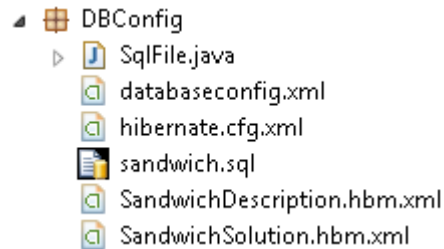


Ilustración 10

Paquete DBConnection

El paquete **DBConnection** contiene la clase **HSQldbServer.java**, la cual se encarga de crear la BBDD, configurar su conexión y lanzar el servidor.

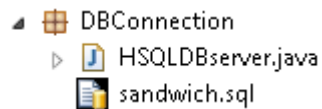


Ilustración 11

Paquete UI

El paquete **UI** contiene las clases es las que se define la interfaz de usuario.



Ilustración 12

Paquete Utils

El paquete **Utils** contiene la clase **LevelChildHierarchy.java**, define una estructura que permite comprobar si cualquier ingrediente es un ingrediente “final” (si es una hoja en la jerarquía de alimentos).

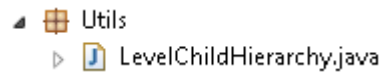


Ilustración 13

Funcionamiento

En esta sección se describirá la interfaz de usuario y el funcionamiento de la aplicación.

Interfaz

En la interfaz nos encontramos con un ingrediente fijo, el pan, imprescindible para hacer cualquier sándwich (existen diferentes tipos de pan para elegir en este **DropDowList**).

Cuatro filas con distintos **DropDowList**, en los cuales se puede elegir una categoría de la jerarquía de alimentos o un ingrediente “final”, para ello, debemos desplazarnos por la jerarquía a través de los distintos **DropDowList** de cada fila, en los cuales se carga de forma dinámica la jerarquía atendiendo al elemento que se ha seleccionado en el **DropDowList** anterior.

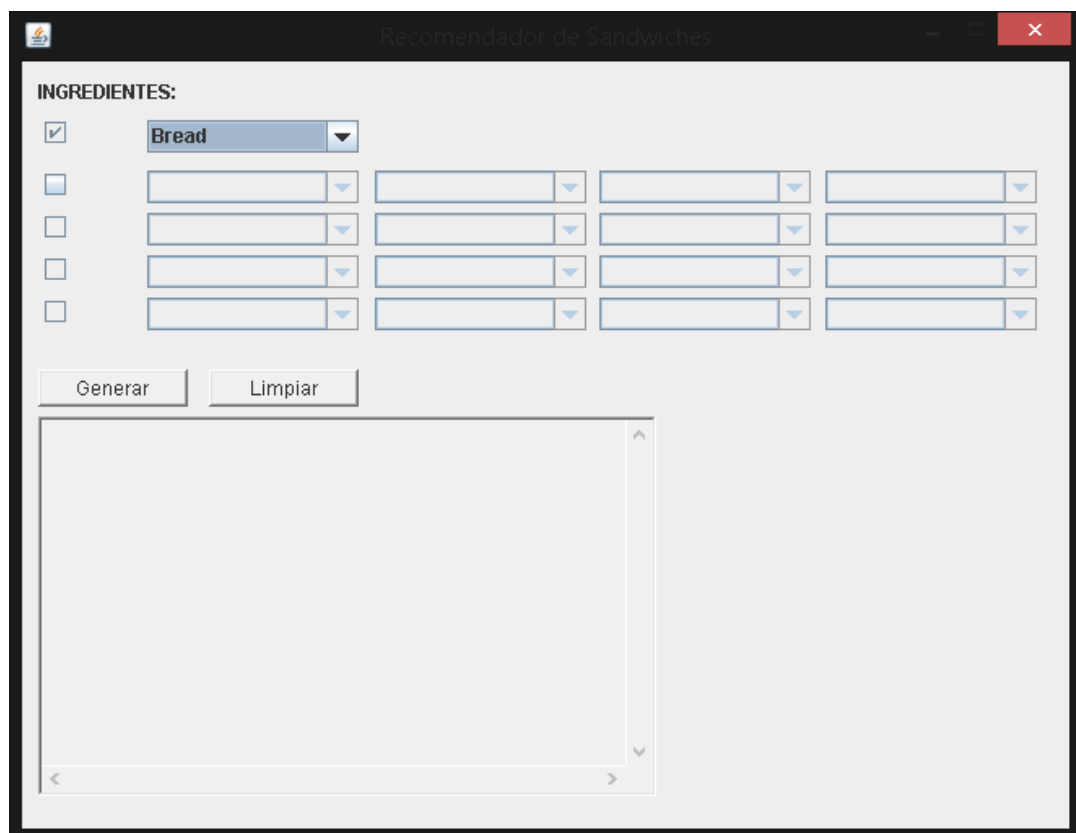


Ilustración 14

1. Carga de la jerarquía de alimentos

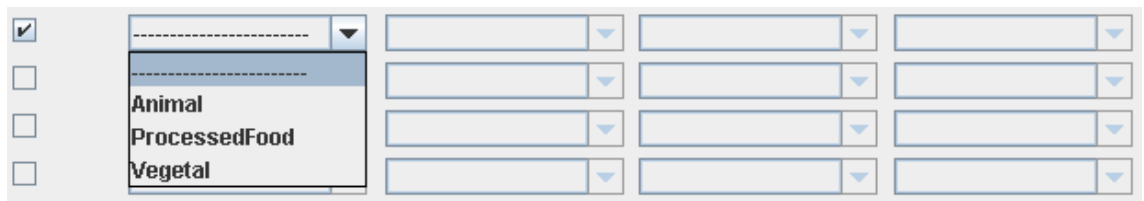


Ilustración 15



Ilustración 16

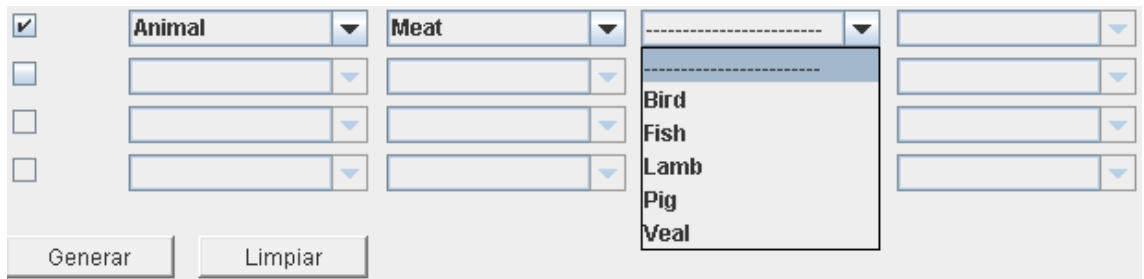


Ilustración 17



Ilustración 18

Por otro lado, tenemos el botón **‘Generar’**, que pone en funcionamiento el ciclo CBR tomando como consulta los ingredientes seleccionados y el botón **‘Limpiar’**, el cual borra el contenido del **textArea** donde se muestran las soluciones.



Ilustración 19

2. Generación de consultas

Atendiendo al siguiente caso (*Ilustración 20*):

<input checked="" type="checkbox"/>	Animal	Meat	Fish	Tuna
<input checked="" type="checkbox"/>	Vegetal	Vegetable		
<input type="checkbox"/>				
<input type="checkbox"/>				

Ilustración 20

Tenemos la siguiente consulta (**Bread, Tuna, Vegetable, X, X**), pero **Vegetable** no es un ingrediente “final”, por lo tanto, la aplicación se encarga de desplazarse por la jerarquía hasta llegar a las hojas de esa rama y así generar las consultas adecuadas, por lo tanto, se generan en este caso las siguientes consultas:

1. (**Bread, Tuna, Lettuce, X, X**)
2. (**Bread, Tuna, Spinach, X, X**)
3. (**Bread, Tuna, Avocado, X, X**)
4. (**Bread, Tuna, Chile, X, X**)
5. (**Bread, Tuna, Pepper, X, X**)
6. (**Bread, Tuna, Tomato, X, X**)

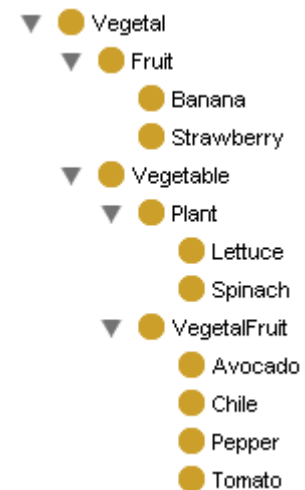


Ilustración 21

Resultado:

```
Results:
(Sandwich1;Bread;Tuna;Oil;Avocado;Lettuce)
(Sandwich6;Bread;Tuna;Lettuce;Avocado;Mayonnaise)
```

Ilustración 22

Con el contenido de la siguiente BBDD

```
insert into sandwich values('Sandwich1','Bread','Tuna','Oil','Avocado','Lettuce');
insert into sandwich values('Sandwich2','Baguette','Egg','Avocado','Turkey','Mayonnaise');
insert into sandwich values('Sandwich3','Bread','Avocado','Oil','Egg','Turkey');
insert into sandwich values('Sandwich4','Wholemeal','Cheese','Pig','Lettuce','Oil');
insert into sandwich values('Sandwich5','Ciabatta','Chile','Oil','Veal','Cheese');
insert into sandwich values('Sandwich6','Bread','Tuna','Lettuce','Avocado','Mayonnaise');
insert into sandwich values('Sandwich7','Bread','Oil','Avocado','Mayonnaise','Lettuce');
insert into sandwich values('Sandwich8','Bread','Lettuce','Avocado','Oil','Mayonnaise');
insert into sandwich values('Sandwich9','Bread','Lettuce','Lettuce','Oil','Mayonnaise');
```

Ilustración 23

Restricciones

1. Máximo 4 ingredientes, sin contar con el pan.
2. El orden de los ingredientes no importa (casos recuperados), aunque en la consulta he fijado el pan como ingrediente número uno, esto no afecta en el momento de la recuperación de casos.
3. Función de similitud: Igualdad.