

Escuela Politécnica Superior (Jaén)

# Ingeniería en Informática



**UNIVERSIDAD DE JAÉN**

*Escuela Politécnica Superior (Jaén)*

## **PROTOTIPO SOFTWARE PARA LA CLASIFICACIÓN DEL ACEITE DE OLIVA VIRGEN CON DIFERENTES DOMINIOS DE EXPRESIÓN**

**Alumna: Ana Arroyo Moral**

Tutora: Dra. Macarena Espinilla  
Estévez

Dpto: Informática

Área: Arquitectura y Tecnología de  
Computadores

**Agosto 2013**





Universidad de Jaén

Escuela Politécnica Superior de Jaén

Departamento de Informática

Dra. Prof. D<sup>a</sup>. Macarena Espinilla Estévez, tutora del Proyecto Fin de Carrera titulado:  
Prototipo Software para la Clasificación del Aceite de Oliva Virgen con Diferentes  
Dominios de Expresión, que presenta D<sup>a</sup>. Ana Arroyo Moral, autoriza su presentación  
para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Agosto 2013

La alumna:

La tutora:

D. Ana Arroyo Moral

D<sup>a</sup>. Macarena Espinilla Estévez

---

## ÍNDICE GENERAL

<b>Índice de contenidos.....</b>	<b>4</b>
<b>Índice de figuras .....</b>	<b>5</b>
<b>Índice de tablas.....</b>	<b>8</b>
<b>Capítulo 1 - Introducción.....</b>	<b>10</b>
1. Introducción al proyecto.....	10
2. Propósito.....	12
3. Objetivos .....	12
<b>Capítulo 2 - Evaluación Sensorial del Aceite de Oliva y Modelado de la Información .....</b>	<b>13</b>
1. Toma de decisiones y Evaluación.....	13
2. Información involucrada en el proceso de evaluación sensorial.....	27
3. Evaluación Sensorial del aceite de oliva virgen. Método COI .....	38
<b>Capítulo 3 - Proyecto.....</b>	<b>58</b>
1. Descripción.....	58
2. Especificación de requerimientos.....	59
3. Análisis del sistema .....	67
4. Diseño del sistema .....	83
5. Implementación .....	132
6. Pruebas .....	155
<b>Capítulo 4 - Conclusiones .....</b>	<b>161</b>
1. Conclusiones finales.....	161
2. Trabajos futuros.....	162
<b>Bibliografía .....</b>	<b>163</b>
<b>Anexo I - Manual de instalación .....</b>	<b>169</b>
<b>Anexo II - Manual de usuario: Administrador .....</b>	<b>176</b>
<b>Anexo III - Manual de usuario: Catador .....</b>	<b>192</b>
<b>Anexo IV - Manual del desarrollador .....</b>	<b>197</b>

## ÍNDICE DE FIGURAS

FIGURA 1: TIPOS DE DOMINIOS DE EXPRESIÓN .....	12
FIGURA 2: RESOLUCIÓN DE UN PROBLEMA DE TOMA DE DECISIÓN.....	25
FIGURA 3: ESQUEMA DE UN PROCESO DE TOMA DE DECISIÓN.....	25
FIGURA 4: EJEMPLO DE FUNCIÓN DE PERTENENCIA .....	30
FIGURA 5: INTERSECCIÓN Y UNIÓN EN CONJUNTOS DIFUSOS.....	32
FIGURA 6: REPRESENTACIÓN GRÁFICA DE LAS FUNCIONES DE PERTENENCIA TRIANGULAR, TRAPEZOIDAL Y GAUSSIANA .....	34
FIGURA 7: HOJA DE PERFIL ORIGINAL.....	40
FIGURA 8: FASES DEL MODELO SENSORIAL DEL ACEITE DE OLIVA.....	41
FIGURA 9: ESCALA NUMÉRICA DE 0 A 10 .....	42
FIGURA 10: HOJA DE PERFIL PARA DOMINIO DE EXPRESIÓN NUMÉRICO 0-10 .....	43
FIGURA 11: ESCALA LINGÜÍSTICA DE 5 ETIQUETAS .....	46
FIGURA 12: HOJA DE PERFIL PARA DOMINIO LINGÜÍSTICO DE 5 ETIQUETAS .....	47
FIGURA 13: ETIQUETA TRAPEZOIDAL.....	50
FIGURA 14: ESCALA DIFUSO NO BALANCEADA DE 4 ETIQUETAS.....	50
FIGURA 15: RESTRICCIONES DE LA SEMÁNTICA DE LAS ETIQUETAS DIFUSAS .....	50
FIGURA 16: HOJA DE PERFIL PARA DOMINIO DIFUSO DE 4 ETIQUETAS .....	51
FIGURA 17: ESCALA 2-TUPLA DE 5 ETIQUETAS .....	54
FIGURA 18: HOJA DE PERFIL PARA DOMINIO LINGÜÍSTICO DE 5 ETIQUETAS .....	55
FIGURA 19: DIAGRAMA FRONTERA.....	70
FIGURA 20: CASO DE USO - GESTIONAR USUARIOS .....	72
FIGURA 21: CASO DE USO - GESTIONAR MUESTRAS .....	75
FIGURA 22: CASO DE USO - GESTIONAR SESIONES DE CATA.....	77
FIGURA 23: CASOS DE USO - GESTIONAR RESULTADOS .....	79
FIGURA 24: CASO DE USO - EVALUAR MUESTRAS.....	80
FIGURA 25: DIAGRAMA ENTIDAD - RELACIÓN.....	103
FIGURA 26: ESQUEMA ENTIDAD RELACIÓN MODIFICADO .....	105
FIGURA 27: EJEMPLO DE PANTALLA I .....	116
FIGURA 28: EJEMPLO DE PANTALLA II .....	117
FIGURA 29: POSIBLES ERRORES EN LA ENTRADA AL SISTEMA .....	124
FIGURA 30: POSIBLES ERRORES DURANTE EL ALTA DE USUARIOS .....	125
FIGURA 31: POSIBLES ERRORES DURANTE LA ELIMINACIÓN DE USUARIOS.....	126
FIGURA 32: POSIBLES ERRORES DURANTE EL ALTA DE MUESTRAS .....	127
FIGURA 33: POSIBLES ERRORES DURANTE LA ELIMINACIÓN DE MUESTRAS.....	127
FIGURA 34: POSIBLES ERRORES DURANTE LA CONSULTA DE LAS SESIONES DE CATA DE LAS MUESTRAS .....	128
FIGURA 35: POSIBLES ERRORES EN LA CONSULTA DE LAS HOJAS DE PERFIL DE UNA MUESTRA .....	128
FIGURA 36: POSIBLES ERRORES CON RANGOS NUMÉRICOS .....	129
FIGURA 37: POSIBLES ERRORES CON RANGOS LINGÜÍSTICOS .....	129
FIGURA 38: POSIBLES ERRORES AL GESTIONAR ETIQUETAS SEMÁNTICAS .....	130
FIGURA 39: POSIBLE ERROR AL CONSULTAR UN JUEGO DE ETIQUETAS .....	131
FIGURA 40: POSIBLE ERROR AL ELIMINAR UN RANGO.....	131
FIGURA 41: POSIBLES ERRORES AL CREAR SESIONES DE CATA.....	131
FIGURA 42: EDICIONES DE JAVA.....	133
FIGURA 43: COMPILACIÓN / EJECUCIÓN DE CÓDIGO EN JAVA.....	134

---

FIGURA 44: PROBLEMÁTICA DE LA PERSISTENCIA EN LAS BBDD .....	138
FIGURA 45: USO DE DATOS RELACIONALES EN LA APLICACIÓN .....	138
FIGURA 46: USO DE DATOS RELACIONALES EN LA APLICACIÓN USANDO DAO .....	139
FIGURA 47: USO DE UNA BD ORIENTADA A OBJETOS.....	139
FIGURA 48: USO DE UN MOTOR DE PERSISTENCIA.....	140
FIGURA 49: EJEMPLO JPA - RELACIÓN ENTRE LAS CLASES .....	142
FIGURA 50: EJEMPLO JPA- IMPLEMENTACIÓN DE LAS CLASES .....	142
FIGURA 51: EJEMPLO JPA - TABLAS EXISTENTES EN LA BD .....	142
FIGURA 52: EJEMPLO JPA - CLASE CUSTOMER CON ANOTACIONES .....	143
FIGURA 53: EJEMPLO JPA - CLASE ORDER CON ANOTACIONES.....	143
FIGURA 54: MAPEO DE RELACIONES DE 1 A N .....	144
FIGURA 55: MAPEO DE RELACIONES DE N A M .....	144
FIGURA 56: EJEMPLO JPA - MANIPULACIÓN DE LAS CLASES .....	145
FIGURA 57: EJEMPLO DE JPA - CONSULTAS CON JPQL .....	145
FIGURA 58: EJEMPLO DE ENTITYMANAGER.....	146
FIGURA 59: LIBRERÍAS NECESARIAS PARA LA CONEXIÓN CON LA BD Y MOTOR DE PERSISTENCIA JPA .....	148
FIGURA 60: CONEXIÓN CON EL SERVIDOR DE BD MYSQL I .....	149
FIGURA 61: CONEXIÓN CON EL SERVIDOR DE BD MYSQL II .....	149
FIGURA 62: CREAR UNIDAD DE PERSISTENCIA .....	150
FIGURA 63: CREAR LAS CLASES DESDE LA BD I .....	150
FIGURA 64: CREAR LAS CLASES DESDE LA BD II .....	151
FIGURA 65: CREAR LAS CLASES DESDE LA BD III .....	151
FIGURA 66: CREAR LAS CLASES DESDE LA BD IV .....	152
FIGURA 67: CREAR CLASES JPA CONTROLLER I .....	152
FIGURA 68: CREAR CLASES JPA CONTROLLER II.....	153
FIGURA 69: HERRAMIENTA MYSQL WORKBENCH .....	154
ILUSTRACIÓN 70: INSTALACIÓN DEL SERVIDOR DE BASES DE DATOS MYSQL.....	174
ILUSTRACIÓN 71: EJECUCIÓN DEL SCRIPT QUE CONTIENE LA BASE DE DATOS .....	175
FIGURA 72: GESTIÓN DE USUARIOS .....	177
FIGURA 73: ALTA DE USUARIOS .....	177
FIGURA 74: BAJA DE USUARIOS.....	178
FIGURA 75: GESTIÓN DE ATRIBUTOS .....	178
FIGURA 76: ALTA DE ATRIBUTOS .....	179
FIGURA 77: BAJA DE ATRIBUTOS .....	179
FIGURA 78: GESTIÓN DE MUESTRAS .....	180
FIGURA 79: ALTA DE MUESTRAS DE ACEITE.....	180
FIGURA 80: BAJA DE MUESTRAS DE ACEITE.....	181
FIGURA 81: CONSULTAR LAS SESIONES DE CATA QUE ESTÁN ASOCIADAS A UNA DETERMINADA MUESTRA DE ACEITE .....	182
FIGURA 82: CONSULTAR HOJA DE PERFIL AGREGADA .....	182
FIGURA 83: GESTIÓN DE DOMINIOS DE EXPRESIÓN .....	183
FIGURA 84: ALTA DE RANGOS NUMÉRICOS.....	183
FIGURA 85: BAJA DE RANGO NUMÉRICO .....	184
FIGURA 86: ALTA DE UN JUEGO CON 5 ETIQUETAS LINGÜÍSTICAS .....	185
FIGURA 87: BAJA DE UN JUEGO DE ETIQUETAS .....	185
FIGURA 88: ALTA DE UN JUEGO CON 3 ETIQUETAS DIFUSAS .....	186
FIGURA 89: CONSULTAR LAS ESCALAS EXISTENTES EN EL SISTEMA .....	186

---

---

FIGURA 90: CONSULTAR LAS ETIQUETAS DE UN DETERMINADO JUEGO Y ESCALA .....	187
FIGURA 91: ALTA DE UNA SESIÓN DE CATA .....	187
FIGURA 92: ELIMINAR UNA SESIÓN DE CATA .....	188
FIGURA 93: GESTIÓN DE RESULTADOS .....	188
FIGURA 94: ESTADO DE LAS SESIONES DE CATA .....	189
FIGURA 95: ELIMINAR CATADOR/A DE UNA SESIÓN DE CATA .....	189
FIGURA 96: CLASIFICACIÓN DE LA MUESTRA DE ACEITE DE OLIVA .....	190
FIGURA 97: RESULTADOS DE LAS SESIONES DE CATA FINALIZADAS .....	190
FIGURA 98: CERRAR SESIÓN .....	191
FIGURA 99: CERRAR LA APLICACIÓN .....	191
FIGURA 100: MODIFICAR CONTRASEÑA DE UN CATADOR .....	193
FIGURA 101: CONSULTAR HOJAS DE PERFIL SIN EVALUAR .....	194
FIGURA 102: CONSULTAR HOJAS DE PERFIL EVALUADAS.....	195
FIGURA 103: CERRAR SESIÓN CATADOR.....	195
FIGURA 104: CERRAR LA APLICACIÓN CATADOR .....	196
FIGURA 105: REPARACIÓN DE PROBLEMAS ASOCIADOS A LAS LIBRERÍAS DE UN PROYECTO EN NETBEANS .....	199

## ÍNDICE DE TABLAS

TABLA 1: ESQUEMA GENERAL DE UN PROBLEMA DE TOMA DE DECISIÓN CON UN ÚNICO CRITERIO.....	15
TABLA 2: ESQUEMA GENERAL DE UN PROBLEMA DE TOMA DE DECISIÓN MULTICRITERIO .....	16
TABLA 3: ESQUEMA GENERAL DE UN PROBLEMA DE TOMA DE DECISIÓN CON UN SOLO EXPERTO .....	18
TABLA 4: ESQUEMA GENERAL DE UN PROBLEMA DE TOMA DE DECISIÓN MULTIEPERTO .....	19
TABLA 5: ESQUEMA GENERAL DE UN PROBLEMA DE TOMA DE DECISIÓN MULTICRITERIO .....	19
TABLA 6: PROPIEDADES DE LAS FUNCIONES .....	32
TABLA 7: VECTORES DE UTILIDAD POR EL PANEL DE CATADORES DOMINIO NUMÉRICO 0-10 .....	44
TABLA 8: MEDIANAS OBTENIDAS - DOMINIO NUMÉRICO 0-10 .....	44
TABLA 9: EVALUACIÓN DE LA CALIDAD DEL ACEITE DE OLIVA - DOMINIO NUMÉRICO 0-10 .....	45
TABLA 10: ESCALA LINGÜÍSTICA DE 5 ETIQUETAS .....	46
TABLA 11: VECTORES DE UTILIDAD POR EL PANEL DE CATADORES DOMINIO LINGÜÍSTICO DE 5 ETIQUETAS .....	48
TABLA 12: MEDIANAS OBTENIDAS - DOMINIO LINGÜÍSTICO DE 5 ETIQUETAS .....	48
TABLA 13: EVALUACIÓN DE LA CALIDAD DEL ACEITE DE OLIVA - DOMINIO LINGÜÍSTICO DE 5 ETIQUETAS .....	49
TABLA 14: ESCALA DIFUSA DE 4 ETIQUETAS.....	51
TABLA 15: VECTORES DE UTILIDAD POR EL PANEL DE CATADORES DOMINIO DIFUSO DE 5 ETIQUETAS.....	52
TABLA 16: MEDIANAS OBTENIDAS - DOMINIO DIFUSO DE 4 ETIQUETAS .....	52
TABLA 17: EVALUACIÓN DE LA CALIDAD DEL ACEITE DE OLIVA - DOMINIO DIFUSO DE 4 ETIQUETAS.....	53
TABLA 14: ESCALA 2-TUPLA DE 5 ETIQUETAS.....	54
TABLA 15: VECTORES DE UTILIDAD POR EL PANEL DE CATADORES DOMINIO 2-TUPLA DE 5 ETIQUETAS .....	56
TABLA 16: EQUIVALENCIA ENTRE 2-TUPLAS Y NUMÉRICO .....	56
TABLA 17: MEDIANAS OBTENIDAS - DOMINIO 2-TUPLA DE 5 ETIQUETAS .....	56
TABLA 18: EVALUACIÓN DE LA CALIDAD DEL ACEITE DE OLIVA - DOMINIO 2-TUPLA DE 5 ETIQUETAS .....	57
TABLA 19: CAMPOS DE LA TABLA ATTRIBUTE .....	106
TABLA 20: CAMPOS DE LAS TABLAS NUMATTRIBUTE, NUMATTRIBUTE_U, NUMATTRIBUTE_I.....	106
TABLA 21: CAMPOS DE LA TABLA RANK .....	106
TABLA 22: CAMPOS DE LA TABLA LINGATTRIBUTE .....	106
TABLA 23: CAMPOS DE LA TABLA SCALE .....	107
TABLA 24: CAMPOS DE LA TABLA SEMANTICS .....	107
TABLA 25: CAMPOS DE LA TABLA LABEL.....	107
TABLA 26: CAMPOS DE LA TABLA PROFILESHEET .....	108
TABLA 27: CAMPOS DE LA TABLA USER .....	108
TABLA 28: CAMPOS DE LA TABLA SAMPLE.....	108
TABLA 29: CAMPOS DE LA TABLA TASTINGSESSION.....	109
TABLA 30: CAMPOS DE LA TABLA LA_Ps.....	109
TABLA 31: CAMPOS DE LA TABLA NAU_Ps.....	109
TABLA 32: CAMPOS DE LA TABLA NAI_Ps .....	110
TABLA 33: CAMPOS DE LA TABLA USER_TASTINGSESSION.....	110
TABLA 34: CAMPOS DE LA TABLA TASTINGSESSION_SAMPLE .....	110
TABLA 35: EJEMPLOS DE METÁFORAS .....	114
TABLA 36: TIPOS DE INSTRUCCIONES DDL .....	136
TABLA 37: TIPOS DE INSTRUCCIONES DCL.....	136
TABLA 38: TIPOS DE INSTRUCCIONES DML .....	137
TABLA 39: TEST 1 .....	155
TABLA 40: TEST 2 .....	156
TABLA 41: TEST 3 .....	157

---

TABLA 42: TEST 4 .....	158
TABLA 43: TEST 5 .....	159
TABLA 44: RESULTADOS OBTENIDOS EN LOS TESTS.....	160

## CAPÍTULO 1 - INTRODUCCIÓN

### 1. Introducción al proyecto

---

La evaluación es uno de los procesos más importantes en la industria, la alimentación, las empresas, etc., con multitud de disciplinas tales como: la evaluación sensorial, la evaluación de calidad, la evaluación de desempeño, etc. En todo proceso de evaluación se realiza un análisis de las características o indicadores del elemento que es objeto de evaluación. Este análisis es similar al que se realiza en los problemas de toma de decisiones [20,29,73], en los que antes de tomar una decisión, se lleva a cabo un proceso de análisis que permite a los decisores tomar decisiones de una forma coherente. Además, dicho análisis es responsable de realizar un estudio metódico y analítico que ayuda a analizar las alternativas, indicadores, etc., del elemento que se está estudiando.

Nuestro interés en este proyecto fin de carrera se centra en la *evaluación sensorial* [3] debido a la gran utilidad que tiene a nivel industrial, siendo también fundamental para realizar controles de calidad de productos, destacando además a nivel comercial y de mercado para conocer la aceptación y la preferencia de los productos sobre los consumidores, entre otras aplicaciones. La evaluación sensorial es una disciplina científica que a través de sus métodos evalúa, mide, califica y describe los atributos sensoriales de un producto (alimentos, tejidos, perfumes, etc.), utilizando como herramienta analítica los sentidos de los seres humanos, la vista, el olfato, el oído, el gusto y el tacto, lo que conlleva que el tipo de información implicada en el proceso de evaluación sensorial presente vaguedad e incertidumbre.

Este proyecto fin de carrera está enfocado en el ámbito de la evaluación sensorial del aceite de oliva. El objetivo de la evaluación sensorial del aceite de oliva es clasificar el aceite en una de las tres categorías existentes: Virgen Extra, Virgen y Lampante. Para ello, un conjunto de catadores evalúa las características sensoriales del aceite en una escala no estructura de 10 cm de longitud, interpretándose la valoración como un número real en el intervalo 0-10 [1].

Actualmente, el dominio de expresión utilizado para valorar las características sensoriales del aceite de oliva está sujeto a debate, ya que se les exige a los catadores un nivel de precisión muy elevado para expresar información que perciben a través de los sentidos, siendo esta información vaga e imprecisa.

Por otro lado, no existen en el mercado aplicaciones informáticas que permitan gestionar la información presente en la evaluación sensorial del aceite de oliva. Aplicaciones informáticas en este ámbito permitirían realizar los procesos de evaluación sensorial de un modo más eficiente y eficaz.

El objetivo de este proyecto fin de carrera es realizar una aplicación de escritorio para la clasificación de las muestras de aceite de oliva que ofrezca una flexibilización a la hora de expresar las valoraciones de los atributos sensoriales del aceite de oliva a través de diferentes dominios de expresión. Para ello, se va a dar la posibilidad de trabajar con los siguientes dominios de expresión: *numérico* e *intervalar*, tanto para valores reales como naturales positivos, y a través de variables lingüísticas, modelando dicha información a través de diferentes metodologías: basadas en el lógica difusa, enfoque lingüístico difuso y lingüístico 2-tuplas.

A continuación, se presenta la estructura de la presenta memoria. En el capítulo 2 vamos a revisar los conceptos básicos sobre la toma de decisiones y el tipo de información involucrada en los procesos de evaluación sensorial. Así como las nociones de lógica difusa y teoría de conjuntos difusos, el enfoque lingüístico difuso y 2-tuplas. Además se indicará cómo se lleva a cabo la evaluación sensorial del aceite de oliva virgen según el método oficial que establece el Consejo Oleícola Internacional (COI).

En el capítulo 3 se describen las diferentes fases que forman el ciclo de vida del software como son las especificaciones de requerimientos tanto funcionales como no funcionales, análisis del sistema, diseño del sistema, implementación y pruebas.

En el capítulo 4 se indican unas conclusiones finales junto con posibles mejoras en trabajos futuros.

Por último unos anexos que incluyen los manuales de instalación, desarrollador y de usuario tanto catador como administrador.

---

## 2. Propósito

El propósito de este proyecto es la implementación de una aplicación de escritorio con tecnología JAVA que permita realizar de forma automática la clasificación del aceite de oliva virgen utilizando para ello diferentes dominios de expresión:

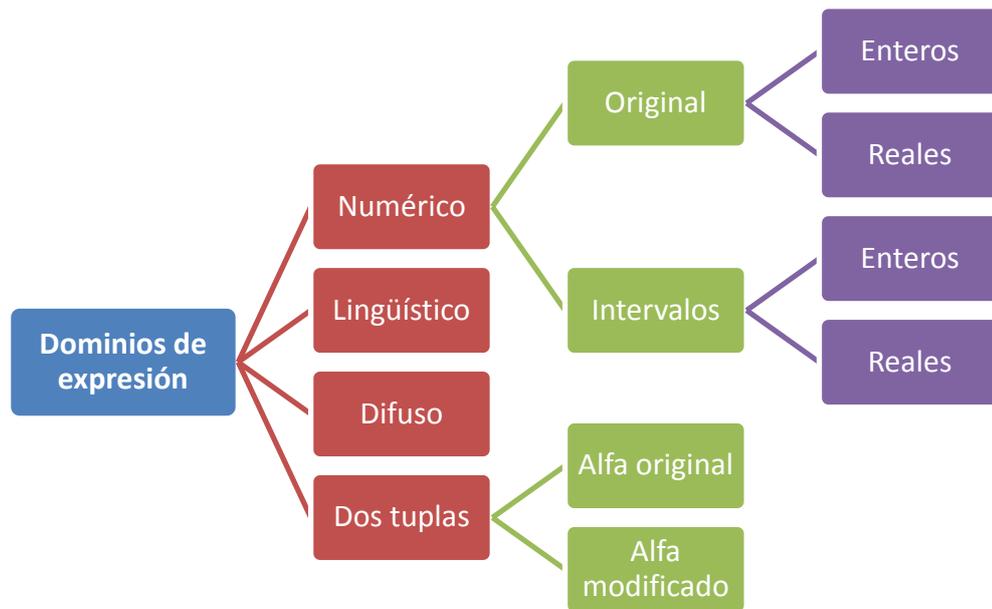


Figura 1: Tipos de dominios de expresión

## 3. Objetivos

Los objetivos que se pretenden alcanzar son los siguientes:

1. Buscar y revisar fuentes bibliográficas.
2. Analizar el modelo existente sobre información heterogénea.
3. Implementar los diferentes dominios de expresión propuestos y el algoritmo que permite agregar toda la información de la sesión de cata.
4. Diseñar e implementar una base de datos que recoja la información generada en las sesiones de cata.
5. Diseñar e implementar de una interfaz de usuario amigable y usable.
6. Realizar manuales para el uso correcto de la aplicación.

## **CAPÍTULO 2 - EVALUACIÓN SENSORIAL DEL ACEITE DE OLIVA Y MODELADO DE LA INFORMACIÓN**

### **1. Toma de decisiones y Evaluación**

---

En este apartado se indicarán distintos conceptos y características de los problemas de toma de decisión.

#### **1.1. Toma de decisión**

En un sentido amplio, tomar una decisión consiste en elegir la mejor opción o alternativa entre un conjunto de opciones o alternativas posibles. Los problemas de toma de decisión presentan los siguientes elementos básicos [20]:

1. Uno o varios objetivos por alcanzar.
2. Un conjunto de alternativas o decisiones posibles para alcanzar dichos objetivos.
3. Un conjunto de factores o estados de la naturaleza que definen el contexto en el que se plantea el problema de decisión.
4. Un conjunto de valores de utilidad o consecuencias asociados a los pares formados por cada alternativa y estado de la naturaleza.

Dependiendo de las características de los elementos del problema de decisión, éstos podrán clasificarse atendiendo a distintos puntos de vista. Por otro lado, el tipo de información que define el marco del problema y su modelado influirían también en el modelo de resolución del problema de decisión.

En las siguientes secciones, describiremos la clasificación de los problemas de toma de decisión según la teoría de decisión. Revisaremos el modelado de preferencias,

---

debido a su papel principal en los procesos de toma de decisión, ya que definirá cómo se expresa el conocimiento, los gustos, las afinidades, etc., que forman parte del problema de toma de decisión. Por último, veremos de forma breve un esquema básico de resolución de un problema de decisión.

### **1.1.1. Clasificación de los Problemas de Toma de Decisiones**

Ante la gran variedad de situaciones o problemas de decisión que se pueden presentar en la vida real, la teoría de la decisión [29, 73] ha establecido una serie de criterios que permiten clasificar los problemas de decisión atendiendo a diferentes puntos de vista:

1. Según el número de criterios o atributos que se han de valorar en la toma de decisión.
2. Según el ambiente de decisión en el que se han de tomar las decisiones.
3. Según el número de expertos que participan en el proceso de decisión.

La descripción de cada punto de vista de dicha clasificación se muestra a continuación.

#### ***A) Número de Criterios***

El número de criterios o atributos que se tienen en cuenta en los procesos de decisión para obtener la solución permite clasificar a los problemas de decisión en dos tipos [22, 23, 30, 31, 32, 46, 51, 52, 59, 62]:

1. *Problemas con un sólo criterio o atributo.* Problemas de decisión en los que, para evaluar las alternativas, se tiene en cuenta un único criterio o atributo que representa la valoración dada a esa alternativa. La solución se obtiene como la alternativa que mejor resuelve el problema teniendo en cuenta este único criterio.

Supongamos un problema de decisión en el que nos planteamos cambiar de coche y nos ofrecen tres posibles alternativas, siendo el criterio de optimización de la

---

decisión el criterio *precio*. Este problema de decisión sería muy simple de resolver puesto que escogeríamos la alternativa con el menor precio.

En los problemas de decisión de un único criterio, cada alternativa es caracterizada por un único valor. Sea  $X = \{x_1, x_2, \dots, x_n\}$  el conjunto de alternativas del problema. Una forma de representación de la información del problema se muestra en la siguiente tabla:

Alternativas	Valoración
$x_1$	$y_1$
...	...
$x_n$	$y_n$

Tabla 1: Esquema general de un problema de toma de decisión con un único criterio

Cada entrada,  $y_i$ , de la tabla indica la valoración de la alternativa,  $x_i$ . Según el marco de definición del problema, cada,  $y_i$ , estaría valorada en un dominio de expresión determinado (numérico, lingüístico, etc.).

2. *Problemas multicriterio o multiatributo*. Problemas de decisión en los que, para evaluar las alternativas, se tienen en cuenta dos o más criterios o atributos que definen cada alternativa. La alternativa solución sería aquella que mejor resolviera el problema considerando todos estos criterios o atributos.

Supongamos el problema presentado anteriormente. Éste se complicaría y el proceso para resolverlo sería diferente si, además de considerar el precio, también tuviésemos en cuenta otros criterios o atributos como el diseño y el consumo del coche. En este caso, nos enfrentaríamos a un problema en el que se consideran varios criterios o atributos para tomar una decisión y, por lo tanto, hablamos de un problema de decisión multicriterio o multiatributo.

El número de criterios en un problema de decisión multicriterio se asume que es finito. Sean  $X = \{x_1, x_2, \dots, x_n\}$  y  $C = \{c_1, c_2, \dots, c_h\}$  el conjunto de alternativas y el conjunto de criterios respectivamente, que caracterizan una situación de decisión determinada, entonces una representación de la información del problema puede expresarse mediante la siguiente tabla:

Alternativas	<i>Criterios</i>			
	$c_1$	$c_2$	...	$c_h$
$(x_i)$				
$x_1$	$y_{11}$	$y_{12}$	...	$y_{1h}$
...	...	...	...	...
$x_n$	$y_{n1}$	$y_{n2}$	...	$y_{nh}$

Tabla 2: Esquema general de un problema de toma de decisión multicriterio

Cada entrada,  $y_{ij}$ , indica la preferencia de la alternativa,  $x_i$ , respecto del criterio,  $c_j$ . Los problemas de toma de decisión multicriterio son más complejos de resolver que los problemas en los que hay un solo criterio. Cada criterio puede establecer un orden de preferencia particular y diferente sobre el conjunto de alternativas. A partir del conjunto de órdenes de preferencia particulares, será necesario establecer algún mecanismo que permita construir un orden global de preferencia. En la literatura, podemos encontrar varios ejemplos [67, 73].

### B) Ambiente de Decisión

El ambiente de decisión viene definido por las características y el marco en el que se va a llevar a cabo la toma de decisión. La teoría clásica de la decisión distingue tres situaciones o ambientes de decisión [29, 73]:

1. *Ambiente de certidumbre*. Un problema de decisión está definido en un ambiente de certidumbre cuando son conocidos con exactitud todos los elementos o factores que intervienen en el problema. Esta situación permite asignar valores cuantitativos de utilidad a cada una de las alternativas presentes en el problema.

Como ejemplo, supongamos que una directora debe mandar la impresión de unos programas para un festival musical. La directora conoce el objetivo impresión de los programas al mínimo coste y no tiene dificultad para comparar los presupuestos de las distintas imprentas locales para la impresión de los programas.

En este caso, la directora conoce todos los factores que se han de tener en cuenta para tomar la decisión y puede elegir una imprenta y saber con certeza cuánto costará la impresión.

2. *Ambiente de riesgo*. Un problema de decisión está definido en un ambiente de riesgo cuando alguno de los elementos o factores que intervienen están sujetos a las

leyes del azar. En estos casos, los problemas pueden ser resueltos utilizando la teoría de la probabilidad. Supongamos el problema presentado anteriormente. La información presente no le servirá a la directora para saber ¿Cuántos programas debe imprimir? Para tomar esta decisión debe considerar el hecho de que, si bien no quiere quedarse sin programas, ordenar demasiados programas es un gasto innecesario. Si el festival es el décimo evento anual que tiene en su población en esa época del año, la directora puede analizar los datos existentes para determinar, aunque con cierto riesgo, la cantidad de programas que probablemente se necesitarán y podrá basarse en estimaciones estadísticas de eventos anteriores.

3. *Ambiente de Incertidumbre*. Un problema de decisión está definido en un ambiente de incertidumbre cuando la información disponible sobre las distintas alternativas puede ser incompleta, vaga o imprecisa, lo que implica que la utilidad asignada a cada alternativa tenga que ser valorada de forma cualitativa. Esta incertidumbre surge a raíz del intento de modelar la imprecisión propia del comportamiento humano o la inherente a ciertos fenómenos que por su naturaleza son inciertos (imprecisos).

Como ejemplo, supongamos la situación anterior. La directora quiere escoger el *diseño de la portada* del programa que más agrade a la población. En este caso, la directora se encuentra en un problema de decisión bajo un ambiente de incertidumbre.

Problemas relacionados con la toma de decisión han sido tradicionalmente tratados, o bien con enfoques determinísticos, o probabilísticos. El primero ignora completamente la incertidumbre, mientras que el segundo asume que cualquier incertidumbre puede ser representada como una distribución de probabilidad.

Sin embargo, en problemas reales (ingeniería, planificación, etc.) las decisiones han de tomarse bajo situaciones con información vaga, imprecisa e incierta. Habitualmente, la incertidumbre es de naturaleza no probabilística como, por ejemplo, la que puede surgir al intentar valorar fenómenos relacionados con las apreciaciones sensoriales. Esto ha generado la necesidad de recurrir a la definición de herramientas para tratar este tipo de incertidumbre, entre ellos se encuentran la teoría de los conjuntos difusos [27], rough sets [28, 34, 45], conjuntos difusos intuicionistas [6, 14, 15, 17, 69, 79], etc. La lógica difusa y el enfoque lingüístico difuso serán las herramientas que utilizaremos en esta memoria para el tratamiento de este tipo de incertidumbre, ya

---

que el uso de la información lingüística hace más flexibles y fiables los modelos de decisión bajo incertidumbre.

### C) Número de Expertos

Finalmente, otro punto de vista, a la hora de clasificar los problemas de decisión, hace referencia al número de expertos o fuentes de información que toman parte en el problema de decisión. Un problema de toma de decisión, en el que participan varios expertos, es más complejo que otro en el que la toma de decisión se realiza de forma individual. Sin embargo, el hecho de que intervengan varios expertos con puntos de vista diferentes puede ofrecer una solución más satisfactoria al problema [66, 73].

Atendiendo al número de expertos que toman parte en el proceso de Toma de Decisión, los problemas de decisión se pueden clasificar en dos tipos [38, 39, 40, 41, 42, 48, 60, 74, 76, 80]:

1. *Unipersonales o individuales*. Las decisiones son tomadas por un único experto.

En los problemas de decisión unipersonales o individuales, cada alternativa es valorada por un único experto. Sea  $X = \{x_1, x_2, \dots, x_n\}$  el conjunto de alternativas que son valoradas por el experto,  $e$ , entonces una forma de representación de la información del problema es la mostrada en la siguiente tabla:

Alternativas	Experto
$(x_i)$	$e$
$x_1$	$y_1$
...	...
$x_n$	$y_n$

Tabla 3: Esquema general de un problema de toma de decisión con un solo experto

Cada entrada,  $y_i$ , de la tabla indica la valoración dada por el experto,  $e$ , sobre la alternativa,  $x_i$ . Según el marco de definición del problema, cada  $y_i$  estará valorada en un dominio de expresión determinado (numérico, lingüístico, etc.).

2. *En Grupo o Multiexperto*. Las decisiones son tomadas en conjunto por un grupo de expertos que intentan alcanzar una solución, en común, al problema. El número de expertos en problemas de decisión multiexperto se asume que es finito. Sean  $X = \{x_1, x_2,$

...,  $x_n$  y  $E = \{e_1, e_2, \dots, e_m\}$  el conjunto de alternativas y el conjunto de expertos respectivamente, que valoran cada alternativa que caracteriza una situación de decisión determinada, entonces una forma de representación de la información del problema puede verse en la siguiente tabla:

Alternativas ( $x_i$ )	Expertos			
	$e_1$	$e_2$	...	$e_m$
$x_1$	$y_{11}$	$y_{12}$	...	$y_{1m}$
...	...	...	...	...
$x_n$	$y_{n1}$	$y_{n2}$	...	$y_{nm}$

Tabla 4: Esquema general de un problema de toma de decisión multiexperto

Cada entrada,  $y_{ij}$ , de la tabla indica la preferencia del experto,  $e_j$ , sobre la alternativa,  $x_i$ .

Una situación de decisión habitual son los problemas de decisión multiexperto multicriterio, en los que cada experto expresa las preferencias sobre distintos criterios que definen cada alternativa [73]. Estas situaciones están relacionadas con los procesos de evaluación sensorial, donde un conjunto de jueces expresan sus valoraciones sobre las características sensoriales de un producto y es necesario fusionar todas ellas para obtener la valoración global del producto.

Experto ( $e_1$ )	Criterios				...	Experto ( $e_m$ )	Criterios			
Alternativas	$c_1$	$c_2$	...	$c_h$		Alternativas	$c_1$	$c_2$	...	$c_h$
$x_1$	$y_{11}^1$	$y_{12}^1$	...	$y_{1h}^1$		$x_1$	$y_{11}^m$	$y_{12}^m$	...	$y_{1h}^m$
...	...	...	...	...		...	...	...	...	...
$x_n$	$y_{n1}^1$	$y_{n2}^1$	...	$y_{nh}^1$		$x_n$	$y_{n1}^m$	$y_{n2}^m$	...	$y_{nh}^m$

Tabla 5: Esquema general de un problema de toma de decisión multicriterio

Cada experto,  $e_i$ , proporciona la entrada de la tabla,  $y_{jk}^i$ , que indica la preferencia de la alternativa,  $x_j$ , respecto del criterio,  $c_k$ .

Luego en este proyecto nos encontramos con un problema de toma de decisión caracterizado por ser: multicriterio, en un ambiente de incertidumbre y multiexperto.

### 1.1.2. Modelado de Preferencias en Toma de Decisión

El modelado de preferencias juega un papel fundamental en los procesos de toma de decisión ya que se utiliza para que los expertos, en base a su conocimiento, experiencias y creencias, expresen sus valoraciones sobre el conjunto de alternativas y establezcan una preferencia sobre la idoneidad de cada una ellas como solución al problema. Un modelado de la información adecuado permite que los expertos expresen mejor sus preferencias. Factores como la naturaleza de la alternativa y los criterios que la definen, junto con la formación y el conocimiento del experto, influirán en cómo modelar la información en cada problema.

Para hacer una revisión breve sobre el modelado de preferencias, vamos a considerar dos puntos de vista diferenciados pero igualmente importantes:

a) *El dominio de expresión*. Conjunto de valores utilizados por los expertos para expresar sus preferencias.

b) *La estructura de representación*. Estructura usada en el problema para almacenar las preferencias de los expertos.

Ambos puntos de vista se revisan a continuación.

#### ***A) Dominios de Expresión de Preferencias***

En problemas de decisión entendemos por dominio de expresión de preferencias al conjunto de valores utilizado por los expertos para emitir sus preferencias. En los problemas de decisión, la elección de un dominio de información para expresar las preferencias puede deberse a varios motivos:

1. ***Naturaleza cuantitativa o cualitativa de la información***. La naturaleza del atributo puede condicionar el dominio utilizado para su valoración. Atributos de naturaleza cuantitativa se adecúan mejor a valoraciones de tipo numérico que aquellos de naturaleza cualitativa en los que al tratarse, por ejemplo, de sensaciones o percepciones subjetivas, el uso de otro tipo de valoraciones como palabras o términos lingüísticos (*bueno, malo, mejor*, etc.) suele ser más apropiado.

---

2. ***Pertenencia de los expertos a diferentes áreas de conocimiento.*** Siempre que sea posible, cada experto tenderá a utilizar un dominio de información que le resulte cercano al tipo de información con el que esté acostumbrado a trabajar en su respectiva área de trabajo. Así, expertos pertenecientes a áreas técnicas se sentirán cómodos utilizando valoraciones numéricas. Sin embargo, aquellos pertenecientes a áreas sociales pueden preferir utilizar otro tipo de valoraciones no numéricas como las lingüísticas.

3. ***Expertos con diferente grado de conocimiento sobre el problema.*** La experiencia de los expertos en la resolución de problemas similares puede implicar que unos expertos opten por elegir dominios de expresión precisos como, por ejemplo, escalas con un elevado número de valoraciones, frente a otros expertos con menos experiencia y que se sientan más habituales utilizando escalas con un menor número de valoraciones.

Adaptar el modelado de preferencias al marco en el que se desarrolla el problema de decisión logra que los expertos se sientan más cómodos y seguros a la hora de valorar sus preferencias y, por lo tanto, que la solución final tenga mayor fiabilidad. En la literatura, encontramos que en los problemas de toma de decisión la información puede ser expresada en distintos dominios, siendo los más comunes los siguientes:

1. *Dominio Numérico* [29, 61, 62].
2. *Dominio Intervalar* [12, 16, 38, 50, 77].
3. *Dominio Lingüístico* [19, 36, 37, 70, 76, 80].

En proyecto, nos centramos en procesos de evaluación sensorial, donde las preferencias de los expertos son percibidas a través de sus órganos sensoriales.

Este hecho hace que el conocimiento sobre los atributos sensoriales evaluados sea vago e impreciso. Dicha incertidumbre no es probabilística y el uso de información lingüística para modelar esta información nos da mayor flexibilidad y fiabilidad, entre otras razones, porque los expertos se sienten más cómodos al valorar lingüísticamente atributos de naturaleza cualitativa relacionados con percepciones.

---

## B) Estructuras de Representación de Preferencias

Una vez conocidos los dominios de expresión más comunes en problemas de toma de decisión, vamos a revisar qué tipos de estructuras son más comunes en los problemas de decisión para representar las preferencias:

1. *Órdenes de Preferencia*. Este formato de representación de preferencias establece un ranking u orden de alternativas que representa la idoneidad de cada alternativa como solución al problema de decisión según el punto de vista de cada experto. Las preferencias de un experto,  $e_i \in E$ , sobre un conjunto de alternativas,  $X$ , se describen mediante un orden de preferencias individual,  $O^i = \{\sigma(1), \dots, \sigma(n)\}$ , donde  $\sigma(\cdot)$  es una función de permutación sobre el conjunto de índices,  $\{1, \dots, n\}$ , para dicho experto [57, 68, 71, 72].

De esta forma, un experto, de acuerdo a su punto de vista, proporciona un vector de alternativas ordenado de mejor a peor. Para todo orden de preferencia,  $O^i$ , supondremos, sin pérdida de generalidad, que cuanto menor es la posición de una alternativa en dicho orden, mejor satisface dicha alternativa el objetivo del experto que proporciona dicho orden y viceversa. Por ejemplo, sea  $E = \{e_1, \dots, e_m\}$  ( $m \geq 2$ ) un conjunto finito de expertos que han de expresar sus preferencias sobre un conjunto finito de alternativas  $X = \{x_1, \dots, x_n\}$  ( $n = 4$ ). Si el experto  $e_2$  proporciona el siguiente orden de preferencia  $\{x_2; x_4; x_1; x_3\}$ , entonces  $\sigma^2(1) = 2$ ;  $\sigma^2(2) = 4$ ;  $\sigma^2(3) = 1$ ;  $\sigma^2(4) = 3$ , lo que significa que la alternativa  $x_2$  es la mejor para dicho experto, mientras que la alternativa  $x_3$  es la peor.

2. *Vectores de Utilidad*. Los valores de utilidad han sido un formato de representación de preferencias muy utilizado en la literatura clásica [21, 26, 75]. En este caso, las preferencias de un experto,  $e_i \in E$ , sobre un conjunto de posibles alternativas,  $X$ , se describen mediante un vector de  $n$  valores de utilidad en un dominio  $D$ ,  $U^i = \{u^i_1, \dots, u^i_n\}$ ,  $u^i_j \in D$ .

Este formato de representación de preferencias es más fino que los anteriores, por lo que un experto puede utilizarlo para representar correctamente sus preferencias sobre las alternativas, siendo necesario que el experto sea capaz de evaluar cada alternativa de manera global con respecto a las demás. Por ejemplo, sea  $E = \{e_1, \dots, e_m\}$

( $m \geq 2$ ) un conjunto finito de expertos que han de expresar sus preferencias sobre un conjunto finito de alternativas  $X = \{x_1, \dots, x_n\}$  ( $n = 4$ ). Si el experto  $e_2$  proporciona el siguiente vector de utilidad:  $U^2 = \{0.3, 0.7, 0.9, 0.4\}$ , significará que, desde su punto de vista, la alternativa  $x_1$  es la peor de todas y que  $x_3$  es la mejor.

Este proyecto, el cual está centrado en procesos de evaluación sensorial, los catadores utilizarán vectores de utilidad para expresar sus preferencias, debido a que la forma habitual de evaluar un producto en evaluación sensorial es valorar cada uno de los atributos sensoriales que lo definen. Por ello, en los modelos de evaluación sensorial que proponemos en esta memoria, las preferencias de los catadores serán recogidas a través de vectores de utilidad y en un dominio de expresión lingüístico.

3. *Relaciones de Preferencia.* En la Teoría Clásica de Preferencias [65], las preferencias sobre un conjunto de alternativas,  $X = \{x_1, \dots, x_n\}$ , se pueden modelar a través de una relación binaria  $R$  definida como sigue:

$$x_i R x_k \Leftrightarrow "x_i \text{ no es peor que } x_k"$$

Esta definición considera una relación binaria como una relación de preferencia débil, e implica que dicha relación,  $R$ , es reflexiva. Con esta definición, es natural asociar un número real, llamado valoración y denotado  $R(x_i; x_j) \in \mathbb{R}$ , el cual representa el grado de verdad de la afirmación " $x_i$  no es peor que  $x_k$ ", o grado de preferencia de la alternativa  $x_i$  sobre la alternativa  $x_j$ . Cuando el conjunto de alternativas es finito, podemos asociar una matriz  $PR$  a la relación  $R$ , tomando como elemento  $ij$ ésimo el valor  $R(x_i; x_j)$  [47, 58, 69, 72].

Sea  $E = \{e_1, \dots, e_m\}$  ( $m \geq 2$ ) un conjunto finito de expertos que han de expresar sus preferencias sobre un conjunto finito de alternativas  $X = \{x_1, \dots, x_n\}$  ( $n = 4$ ). Una relación de preferencia individual se representa como una matriz  $P_{e_i} \subset X \times X$ , donde el valor  $\mu_{P_{e_i}}(x_i, x_k) = p_{ik}^i$  representa el grado de preferencia de la alternativa  $x_i$  sobre la alternativa  $x_k$ .

$$P_{e_i} = \begin{pmatrix} p_i^{11} & \dots & p_i^{1n} \\ \dots & \dots & \dots \\ p_i^{m1} & \dots & p_i^{mn} \end{pmatrix}$$

### 1.1.3. Procesos de Resolución de Problemas de Toma de Decisiones

A pesar de la existencia de los distintos tipos de problemas de decisión, se propone un esquema básico de resolución [24, 30, 64] para los mismos, basado en dos fases: (a) agregación y (b) explotación, que son descritas a continuación.

a) *Fase de Agregación*. Se lleva a cabo a partir de las preferencias individuales de los expertos sobre el conjunto de alternativas para obtener una preferencia global, que agrega sus propiedades. El problema de la agregación de información ha sido estudiado en profundidad y existe gran cantidad de publicaciones al respecto [8, 9, 10, 18, 25, 33, 56, 81, 82, 83].

b) *Fase de Explotación*. A partir de las preferencias colectivas, se aplica un grado de selección para generar el conjunto solución de alternativas del problema [5, 35, 58, 63].

Podemos ver gráficamente este proceso en la Figura 2

## 1.2. Análisis de Decisión y Procesos de Evaluación

A continuación, vamos a mostrar con mayor detalle dicho proceso de resolución y notaremos cuáles son las fases que están directamente relacionadas con un proceso de evaluación.

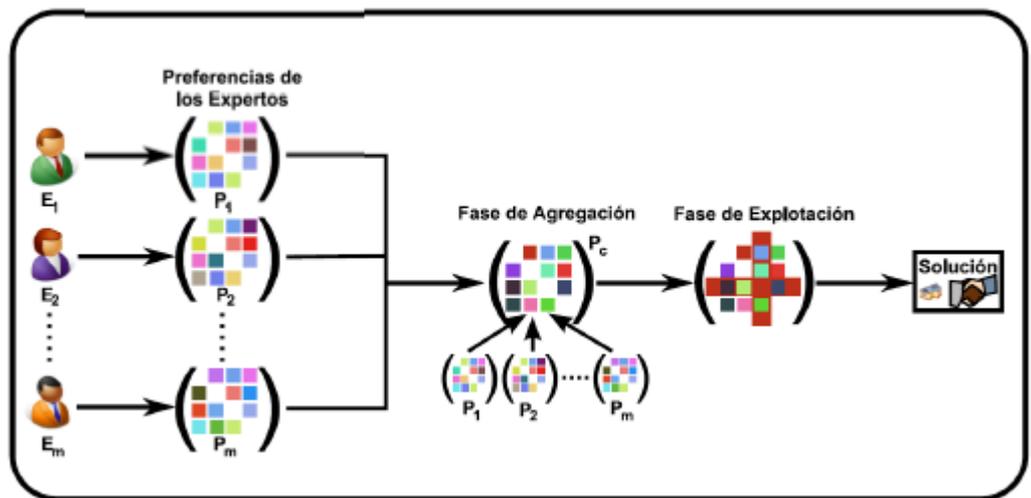


Figura 2: Resolución de un problema de toma de decisión

La toma de decisión, como señalan Keeney y Raiffa [49], intenta ayudar a los individuos a tomar decisiones difíciles y complejas de una forma racional. Esta racionalidad implica el desarrollo de métodos y modelos que permitan representar fielmente cada problema y analizar las distintas alternativas con criterios objetivos. Sin embargo, no todo problema de decisión se resuelve por medio de un proceso completamente racional [7,44]. De hecho, muchos factores externos y subjetivos afectan a un problema de decisión. Por lo tanto, la solución puede variar si las condiciones en las que se presenta el problema cambian. El esquema de resolución de un problema de toma de decisión puede descomponerse en 8 fases [24], que se muestran gráficamente en la Figura 3, siendo denominado el conjunto formado por las 7 primeras fases Proceso de Análisis de Decisión.

Proceso de Toma de Decisión

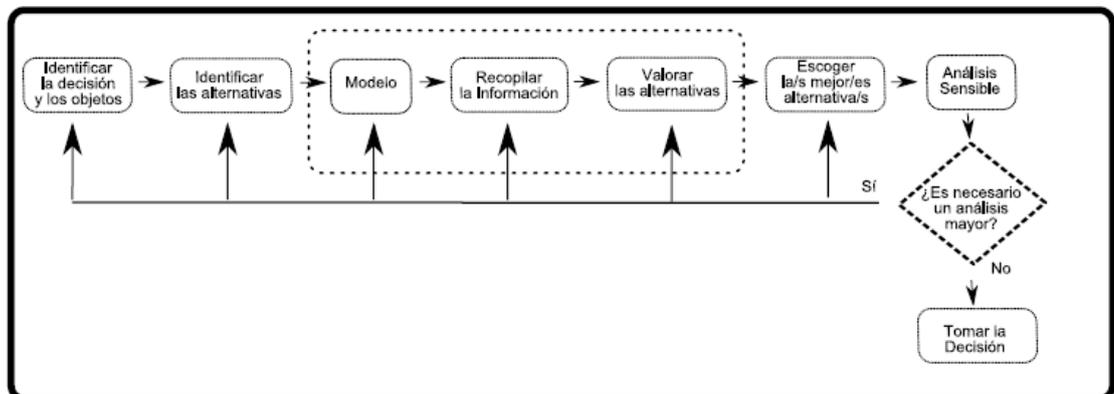


Figura 3: Esquema de un proceso de toma de decisión

Por ejemplo, supongamos que un varón (cercano a los cuarenta años, de ciudad, padre de familia de clase media) se dispone a cambiar de coche. Un coche es muy costoso y hay que analizar detenidamente las alternativas para realizar la mejor adquisición posible, así que el sujeto intenta recabar toda la información posible, busca en Internet, lee folletos de vehículos de distintas categorías, compra revistas de automovilismo, estudia análisis comparativos, etc. Dado que la mayor parte de sus desplazamientos son urbanos y en compañía de sus hijos, el análisis de las distintas alternativas indica que un coche familiar de bajo consumo es el más adecuado, pero finalmente el sujeto compra un coche deportivo, potente, de espacio reducido y mayor consumo. El motivo de esta decisión es que el sujeto, aunque haya hecho un análisis racional, ha tomado una decisión emocional, basada, por lo general, en factores ajenos a la razón.

Por tanto, podemos ver que aunque el análisis de decisión (parte racional del proceso de toma de decisión) no sea seguido siempre por los decisores a la hora de tomar sus decisiones (parte emocional del proceso de toma de decisión), sí es responsable de realizar un estudio metódico y analítico que ayuda a los expertos a analizar las alternativas, indicadores, etc., de los elementos que se están estudiando, siendo éste el objetivo de los procesos de evaluación.

Dado que el campo de interés de este proyecto es la evaluación sensorial, debemos señalar que la evaluación es un proceso cognitivo complejo que implica diversos mecanismos en los cuales es necesario identificar los elementos que van a ser evaluados, fijar el marco en el que se va a realizar la evaluación, recopilar la información necesaria y, finalmente, obtener una valoración de los elementos evaluados. El objetivo de cualquier proceso de evaluación es obtener una valoración de un artículo (producto, servicio, material, etc.) para mejorarlo o compararlo con otros artículos y, así, saber cuál es mejor. En la literatura científica, podemos encontrar cómo el uso de técnicas de análisis de decisión [2, 13, 46, 53] ha producido buenos resultados en la resolución de procesos de evaluación.

## 2. Información involucrada en el proceso de evaluación sensorial

---

### 2.1. Percepciones

La evaluación sensorial es una disciplina de la evaluación donde la información proporcionada por un panel de personas (panel de expertos) es percibida por los órganos sensoriales de la vista, el olfato, el oído, el gusto y el tacto. Este tipo de información implica siempre subjetividad, imprecisión, vaguedad,... Por ello, se propone el uso de la aproximación lingüística difusa para modelar este tipo de información en lugar del uso de información precisa, ayudando a los expertos que participan en el proceso de evaluación sensorial a expresar sus percepciones.

Sin embargo, se considera que el aparato sensorial humano muestra grados de variación sensorial de persona a persona y que la sensibilidad que puede ser influenciada fácilmente por circunstancias externas o del medio. Esto implica que uno de los grandes problemas asociados al análisis sensorial sea conseguir que la respuesta humana sea precisa. Por lo tanto, pueden existir situaciones en las que los expertos necesiten distintas escalas lingüísticas para expresar sus percepciones hacia un mismo indicador sensorial. En estos casos, nos encontramos ante un proceso de evaluación sensorial con múltiples escalas lingüísticas. De la misma manera, pueden ser necesarias este tipo de escalas cuando evaluamos un objeto o producto donde estén involucrados varios índices sensoriales.

Los atributos sensoriales son, en general, todo lo que se percibe a través de los sentidos. Se puede hacer una división de los atributos de acuerdo con los sentidos por los que son percibidos:

- Apariencia, color y forma mediante la vista.
- Consistencia y las características relacionadas (fluidez, viscosidad, dureza, fibrosidad, crujiente, flexibilidad) mediante el tacto y el oído.
- Aroma mediante el olfato.
- Sensaciones gustativas mediante el gusto.
- Sabor mediante una combinación del olfato, gusto y tacto.

Algunas de las aplicaciones que tiene el análisis sensorial en la industria se detallan a continuación:

- Control del Proceso de Fabricación: influencia en el cambio de materia prima, ingredientes o cambios en las condiciones del proceso.
- Desarrollo del producto.
- Estudios de vida útil.
- Establecimiento de los límites y grados de calidad.
- Caracterización del producto (establecimiento perfil sensorial).
- Estudio comparativo de muestras, estudio de aceptación con consumidores.

En este proyecto, aplicaremos uno de los modelos de evaluación sensorial a la caracterización del aceite de oliva en una de las categorías existentes (extra, virgen y lampante).

## 2.2. Modelado de información que presenta incertidumbre

Nuestro interés se centra en los procesos de evaluación sensorial donde la información proporcionada por los jueces ha sido percibida a través de los sentidos, lo que implica incertidumbre e imprecisión. Dicha incertidumbre no tiene un carácter probabilístico, ya que está asociada a la imprecisión de las valoraciones de los jueces [55]. El uso de la Lógica Difusa y el Enfoque Lingüístico Difuso facilita el tratamiento de este tipo de información. Por tanto, consideramos que el uso de un modelado adecuado para el tratamiento de la incertidumbre e imprecisión mejorará los resultados en los procesos de evaluación sensorial. El uso del enfoque lingüístico difuso ha producido buenos resultados a la hora de modelar la información vaga e imprecisa en diversos campos de aplicación [4, 11, 42, 43, 54, 78]. Dicha aproximación está basada en la teoría de conjuntos difusos.

### 2.2.1. Conjuntos difusos

#### Conjuntos difusos y funciones de pertenencia

La noción de conjunto refleja la tendencia a organizar, generalizar y clasificar el conocimiento sobre los objetos del mundo real. El encapsulamiento de los objetos es una colección cuyos miembros comparten una serie de características o propiedades

---

---

que implican la noción de conjunto. Los conjuntos introducen una noción de dicotomía, que en esencia es una clasificación binaria: o se acepta o se rechaza la pertenencia de un objeto a una categoría determinada. Habitualmente la decisión de aceptar se denota como 1 y la de rechazar como 0. Esta decisión de aceptar o rechazar se expresa mediante una función característica, según las propiedades que posean los objetos del conjunto.

La Lógica Difusa se fundamenta en el concepto de conjunto difuso [25] que suaviza el requerimiento anterior y admite valores intermedios en la función característica, que se denomina función de pertenencia. Esto permite una interpretación más realista de la información, puesto que la mayoría de las categorías que describen los objetos del mundo real, no tienen unos límites claros y bien definidos.

Un conjunto difuso puede definirse como una colección de objetos con valores de pertenencia entre 0 (exclusión total) y 1 (pertenencia total). Los valores de pertenencia expresan los grados con los que cada objeto es compatible con las propiedades o características distintivas de la colección. Formalmente podemos definir un conjunto difuso como sigue.

**Definición 2.1:** *Un conjunto difuso A sobre un dominio o universo de discurso U está caracterizado por una función de pertenencia que asocia a cada elemento del conjunto el grado con que pertenece a dicho conjunto, asignándole un valor en el intervalo [0,1]:*

$$\mu_A : U \rightarrow [0, 1]$$

Así, un conjunto difuso A sobre U puede representarse como un conjunto de pares ordenados de un elemento perteneciente a U y su grado de pertenencia,  $A = \{(x, \mu_A(x)) / x \in U, \mu_A(x) \in [0, 1]\}$ . Por ejemplo, consideremos el concepto persona alta, en un contexto donde la estatura oscila entre 1 y 2 m. Como es de suponer, alguien que mida 1,30 m. no se puede considerar como persona alta por lo que su grado de pertenencia al conjunto de personas altas será de 0. Por el contrario, una persona que

mida 1,90m. si la consideramos alta por lo que su grado de pertenencia al conjunto es de 1.

La representación de una función de pertenencia puede adoptar distintas formas, cumpliendo propiedades específicas, pero es el contexto de la aplicación lo que determina la representación más adecuada en cada caso. Puesto que las valoraciones lingüísticas dadas por los usuarios son únicamente aproximaciones, algunos autores consideran que las funciones de pertenencia paramétricas son suficientemente buenas para capturar la imprecisión de tales valoraciones lingüísticas. La representación paramétrica es obtenida a partir de una 4-tupla  $(a, b, \alpha, \beta)$ , ver figura 4, donde  $a$  y  $b$  indican el intervalo en que el valor de pertenencia es 1, con  $\alpha$  y  $\beta$  indicando los límites izquierdo y derecho del dominio de definición de la función de pertenencia trapezoidal. Un caso particular de este tipo de representación son las valoraciones lingüísticas cuyas funciones de pertenencia son triangulares, es decir,  $a = b$ , por lo que se representan por medio de una 3-tupla  $(\alpha, a, \beta)$ . La siguiente figura muestra la descripción y la representación gráfica de un ejemplo de función de pertenencia trapezoidal.

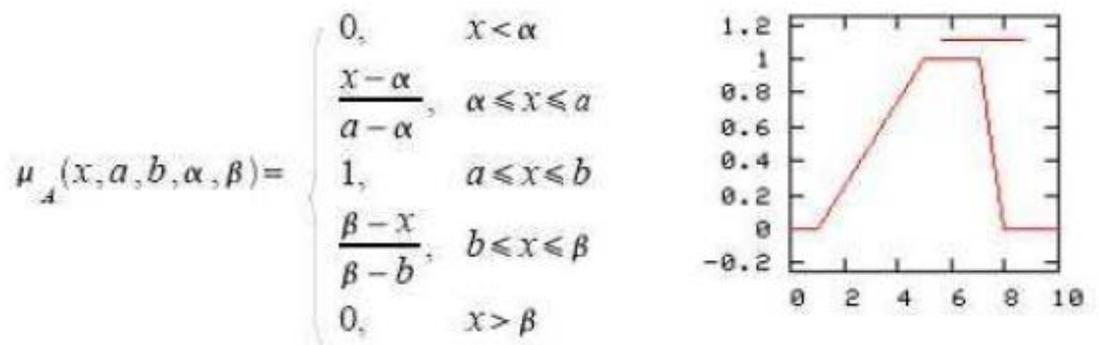


Figura 4: Ejemplo de función de pertenencia

### Definiciones básicas

**Definición 2.2:** Se define el soporte de un conjunto difuso  $A$  en el universo  $U$ , como el conjunto formado por todos los elementos de  $U$  cuyo grado de pertenencia a  $A$  sea mayor que 0:

$$\text{sup}(A) = \{x \in U / \mu_A(x) > 0\}$$

**Definición 2.3:** La altura de un conjunto difuso  $A$  se define como el mayor grado de pertenencia de todos los elementos de dicho conjunto:

$$h(A) = \max\{\mu_A(x) / x \in U\}$$

**Definición 2.4:** El  $\alpha$ -corte de un conjunto difuso  $A$  es el conjunto formado por todos los elementos del universo  $U$  cuyos grados de pertenencia en  $A$  son mayores o iguales que el valor de corte  $\alpha \in [0, 1]$ :

$$A_\alpha = \{x \in U / \mu_A(x) \geq \alpha\}$$

**Definición 2.5:** Se denomina conjunto de niveles de un conjunto difuso  $A$ , al conjunto de grados de pertenencia de sus elementos:

$$L(A) = \{a / \mu_A(x) = a, x \in U\}$$

### Operaciones con conjuntos difusos

Al igual que en la lógica tradicional, las operaciones lógicas que se pueden establecer entre conjuntos difusos son la intersección, la unión y el complemento. Al igual que el resultado de operar dos conjuntos clásicos es un nuevo conjunto clásico, las mismas operaciones con conjuntos difusos nos darán como resultado otros conjuntos también difusos.

Hay muchas formas de definir estas operaciones. Cualquier operación que cumpla las propiedades de una t-norma puede ser usada para hacer la intersección, de igual manera que cualquier operación que cumpla las propiedades de una t-conorma puede ser empleada para la unión. La tabla siguiente muestra las propiedades que deben cumplir las dos familias de funciones y algunos ejemplos.

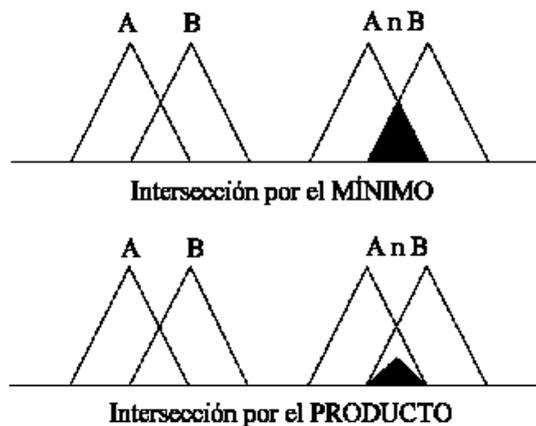
	Propiedades	Ejemplos
<b>T-Normas</b> $T: [0,1] \times [0,1] \rightarrow [0,1]$ $\mu_{A \cap B}(x) = T[\mu_A(x), \mu_B(x)]$	<b>Conmutativa:</b> $T(a,b) = T(b,a)$ <b>Asociativa:</b> $T(a, T(b,c)) = T(T(a,b), c)$ <b>Monotonía:</b> $T(a,b) \geq T(c,d)$ si $a \geq c$ , y $b \geq d$ <b>Condiciones frontera:</b> $T(a,1) = a$	<b>Intersección estándar</b> $T(a,b) = \min(a,b)$ <b>Producto algebraico</b> $T(a,b) = a \cdot b$ <b>Intersección drástica</b> $T(a,b) = \begin{cases} a, & \text{si } b=1 \\ b, & \text{si } a=1 \\ 0, & \text{en otro caso} \end{cases}$
<b>S-Normas</b> $S: [0,1] \times [0,1] \rightarrow [0,1]$ $\mu_{A \cup B}(x) = S[\mu_A(x), \mu_B(x)]$	<b>Conmutativa:</b> $S(a,b) = S(b,a)$ <b>Asociativa:</b> $S(a, S(b,c)) = S(S(a,b), c)$ <b>Monotonía:</b> $S(a,b) \geq S(c,d)$ si $a \geq c$ , y $b \geq d$ <b>Condiciones frontera:</b> $S(a,0) = a$	<b>Unión estándar</b> $S(a,b) = \max(a,b)$ <b>Suma algebraica</b> $S(a,b) = a + b - a \cdot b$ <b>Unión drástica</b> $S(a,b) = \begin{cases} a, & \text{si } b=0 \\ b, & \text{si } a=0 \\ 1, & \text{en otro caso} \end{cases}$

Tabla 6: Propiedades de las funciones

Las operaciones se definen de la siguiente manera:

- Intersección:  $A \cap B = \{(x, \mu_{A \cap B}) / \mu_{A \cap B}(x) = T[\mu_A(x), \mu_B(x)]\}$ .
- Unión:  $A \cup B = \{(x, \mu_{A \cup B}) / \mu_{A \cup B}(x) = S[\mu_A(x), \mu_B(x)]\}$ .
- Complemento:  $\mu_{\neg A}(x) = 1 - \mu_A(x)$ .

En la figura siguiente podemos ver una representación grafica de dichas operaciones.



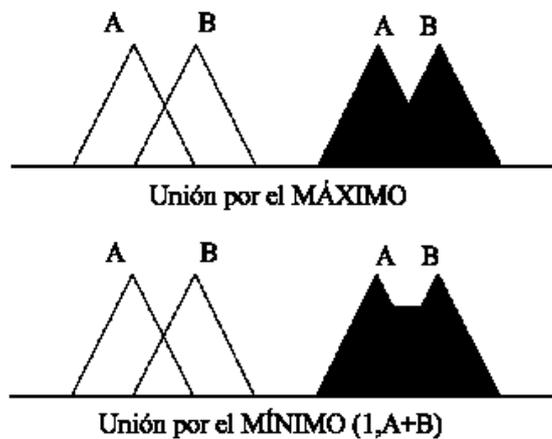


Figura 5: Intersección y Unión en conjuntos difusos

### 2.2.2. Enfoque lingüístico difuso

El enfoque lingüístico difuso representa información lingüística utilizando valores lingüísticos mediante variables lingüísticas [16].

Cuando trabajamos con información lingüística hay que seleccionar los descriptores lingüísticos adecuados y su semántica. Para llevar a cabo esta tarea, un aspecto importante a considerar es la "granularidad de la incertidumbre", es decir, la capacidad de discriminación entre distintos valores de información. Valores típicos de cardinalidad en los modelos lingüísticos son valores impares, tales como 7 ó 9, donde el término medio representa una valoración de aproximadamente 0,5e el resto de términos están distribuidos simétricamente a su alrededor. Una posibilidad es generar directamente el conjunto de términos considerando todos los términos distribuidos en una escala sobre la cuál se define un orden total. Un ejemplo de un conjunto de 7 términos es:

$$S = \{s_0 : N, s_1 : V L, s_2 : L, s_3 : M, s_4 : H, s_5 : V H, s_6 : P\}$$

La semántica de los términos lingüísticos viene dada por números difusos definidos en el intervalo [0,1]. Estos son valoraciones aproximadas, por lo que diferentes autores consideran que una función de pertenencia trapezoidal es lo suficientemente buena para capturar la vaguedad de los términos lingüísticos. Una representación

trapezoidal se consigue con una 4-tupla  $(a, b, d, c)$ , donde  $b$  y  $d$  indican el intervalo donde la función de pertenencia vale 1, con  $a$  y  $c$  indicando los límites izquierdo y derecho de la función [1]. Un caso particular de este tipo de representación son las valoraciones lingüísticas cuya función sea triangular, es decir,  $b = d$ , entonces esta función de pertenencia triangular se representa como  $(a, b, c)$ . Otros autores usan representaciones no trapezoidales, por ejemplo, funciones gaussianas [2].

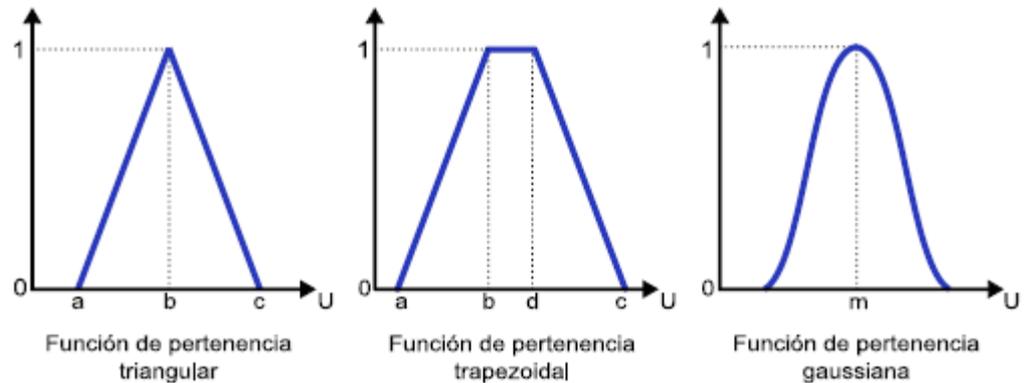


Figura 6: Representación gráfica de las funciones de pertenencia triangular, trapezoidal y gaussiana

### 2.2.3. Modelo lingüístico basado en 2-tupla

El modelo de representación lingüístico basado en 2-tuplas fue presentado en [7] para mejorar los problemas de pérdida de información en los procesos de computación con palabras de otros modelos: (i) Modelo basado en el Principio de Extensión [3], (ii) Modelo simbólico [4]. Además se ha demostrado como un modelo útil en el tratamiento de contextos de información no homogéneos [10, 11, 12].

Este modelo se basa en el concepto de traslación simbólica. Sea  $S = \{s_0, \dots, s_g\}$  un conjunto de términos lingüísticos, y  $f \in U [0, g]$  un valor en el intervalo de granularidad de  $S$ .

**Definición 1.** La *Traslación Simbólica* de un término lingüístico  $s_i$  es un número valorado en el intervalo  $[-5, 5)$  que expresa la “diferencia de información” entre una cantidad de información expresada por el valor  $\beta \in U [0, g]$  obtenido en una operación simbólica y el valor entero más próximo,  $i \in \{0, \dots, g\}$ , que indica el índice de la etiqueta lingüística ( $s_i$ ) más cercana en  $S$ .

A partir de este concepto desarrollaremos un nuevo modelo de representación para la información lingüística, el cuál usa como base de representación un par de valores o 2-tupla,  $(r_i, \alpha_i)$ , donde  $r_i \in S$  y  $\alpha_i \in [-5, 5)$ . Este modelo de representación define un conjunto de funciones que facilitan las operaciones sobre 2-tuplas.

**Definición 2.** Sea  $S = \{s_0, \dots, s_g\}$  un conjunto de términos lingüísticos y  $\beta \in [0, g]$  un valor que representa el resultado de una operación simbólica, entonces la 2-tupla lingüística que expresa la información equivalente a  $\beta$  se obtiene usando la siguiente función:

$$\Delta : [0, g] \longrightarrow S \times [-5, 5)$$

$$\Delta(\beta) = (s_i, \alpha), \text{ con } \begin{cases} s_i, & i = \text{round}(\beta) \\ \alpha = \beta - i, & \alpha \in [-5, 5), \end{cases}$$

donde round es el operador usual de redondeo,  $s_i$  es la etiqueta con índice más cercano a  $\beta$  y  $\alpha$  es el valor de la traslación simbólica.

**Proposición 1.** Sea  $S = \{s_0, \dots, s_g\}$  un conjunto de términos lingüísticos y  $(s_i, \alpha)$  una 2-tupla lingüística. Existe la función  $\Delta^{-1}$ , tal que, dada una 2-tupla  $(s_i, \alpha)$  esta función devuelve su valor numérico equivalente  $\beta \in [0, g]$ .

#### Demostración.

Es trivial si consideramos la siguiente función.

$$\Delta^{-1} : S \times [-5, 5) \longrightarrow [0, g]$$

$$\Delta^{-1}(s_i, \alpha) = i + \alpha = \beta.$$

**Comentario:** A partir de las definiciones 1 y 2 y de la proposición 1, la conversión de un término lingüístico en una 2-tupla consiste en añadir el valor cero como traslación simbólica:

$$s_i \in S \longrightarrow (s_i, 0)$$

Este modelo de representación tiene un modelo computacional asociado presentado en [7]:

**1. Agregación de 2-tuplas:** La agregación de 2-tuplas lingüísticas consiste en la obtención de un valor que resuma un conjunto de valores, por lo tanto, el resultado de la agregación del conjunto de 2- tuplas debe ser una 2-tupla lingüística. En la literatura, podemos encontrar numerosos operadores de agregación que nos permiten combinar la información de acuerdo a distintos criterios. Cualquiera de estos operadores puede ser fácilmente extendido para trabajar con 2-tuplas usando funciones  $\Delta$  y  $\Delta^{-1}$ , que transforman valores numéricos en 2-tupla y viceversa sin pérdida de información.

Algunos ejemplos de estos operadores son los siguientes:

**Definición 9.** Siendo  $X = \{(s_1, \alpha_1), \dots, (s_n, \alpha_n)\}$  un conjunto de varias 2-tuplas lingüísticas, la 2-tupla que simboliza la media aritmética,  $\bar{x}^e$ , se calcula de la siguiente forma:

$$\bar{x}^e[(r_1, \alpha_1), \dots, (r_n, \alpha_n)] = \Delta \left( \sum_{i=1}^n \frac{1}{n} \Delta^{-1}(r_i, \alpha_i) \right) = \Delta \left( \frac{1}{n} \sum_{i=1}^n \beta_i \right)$$

**Definición 10.** Siendo  $X = \{(s_1, \alpha_1)_1, \dots, (s_n, \alpha_n)_n\}$ ,  $s_j \in S = \{s_0, \dots, s_g\}$  un conjunto ordenado de 2-tuplas, siendo  $(s_j, \alpha_j)_k$  el k-ésimo elemento mayor del conjunto X, la 2-tupla que simboliza la mediana,  $Me(X)$ , se define de la siguiente forma:

$$Me(X) = \begin{cases} Me(X) = (s_j, \alpha_j)_{\frac{n+1}{2}} & \text{si } n \text{ es impar} \\ Me(X) = (s_j, \alpha_j)_{\frac{n}{2}} & \text{si } n \text{ es par} \end{cases}$$

Cuando n es par, el valor de la Mediana no es único,  $Me(X) \in [(s_j, \alpha_j)_{\frac{n}{2}}, (s_j, \alpha_j)_{\frac{n+1}{2}}]$

Generalizando:

$$Me(X) = \Delta \left( \frac{\Delta^{-1}(s_j, \alpha_j)_{\frac{n}{2}} + \Delta^{-1}(s_j, \alpha_j)_{\frac{n+1}{2}}}{2} \right)$$

**2. Comparación de 2-tuplas:** La información de comparación representada por las 2-tuplas la realizamos de acuerdo a un orden lexicográfico. Sea  $(s_k, \alpha_1)$  y  $(s_l, \alpha_2)$  dos 2-tuplas que representan dos valoraciones:

- Si  $k < l$  entonces  $(s_k, \alpha_1)$  es más pequeño que  $(s_l, \alpha_2)$
- Si  $k = l$  entonces:
  - a) Si  $\alpha_1 = \alpha_2$  entonces  $(s_k, \alpha_1)$  y  $(s_l, \alpha_2)$  representan el mismo valor.
  - b) Si  $\alpha_1 < \alpha_2$  entonces  $(s_k, \alpha_1)$  es más pequeño que  $(s_l, \alpha_2)$ .
  - c) Si  $\alpha_1 > \alpha_2$  entonces  $(s_k, \alpha_1)$  es mayor que  $(s_l, \alpha_2)$ .

3. **Operador de negación de una 2-tupla:** El operador de negación de una 2-tupla se

define como:

$$Neg(s_i, \alpha) = \Delta(g - (\Delta^{-1}(s_i, \alpha))),$$

donde  $g + 1$  es la cardinalidad de  $S$ ,  $s_i \in S = \{s_0, \dots, s_g\}$ .

### 3. Evaluación Sensorial del aceite de oliva virgen. Método COI

---

#### 3.1. Evaluación sensorial para el aceite de oliva.

El objetivo de este proyecto es realizar de forma automática la clasificación del aceite de oliva, para ello se realiza la evaluación sensorial del aceite de oliva permitiendo evaluar diversas características sensoriales del aceite con diferentes dominios de expresión. La utilización de variables lingüísticas ha dado buenos resultados a la hora de modelar y trabajar con la información involucrada en los procesos de evaluación sensorial del aceite de oliva [1, 54].

En esta sección, se va a revisar el proceso de actual de aceite de oliva virgen que marca el COI.

El aceite virgen de oliva es el zumo de la aceituna, obtenido exclusivamente por procesos físicos como la presión, la decantación o la centrifugación. Como en todos los alimentos, la calidad del aceite virgen depende de la calidad de la aceituna, así como de la calidad de los procesos de cosecha, postcosecha, extracción, conservación y distribución del producto final hasta ser consumido. Se trata de una cadena de producción donde una pérdida de calidad en un punto intermedio es irreparable en etapas posteriores.

A diferencia de otros productos alimenticios, el aceite virgen de oliva debe ser analizado, como define la normativa, tanto desde el punto de vista químico como sensorial. Los resultados de ambas analíticas tiene igual importancia a la hora de clasificar el aceite virgen en sus diferentes categorías comerciales.

El aceite se clasifica en una de las 3 categorías existentes: extra, virgen o lampante. Esta clasificación responde a unos parámetros físico-químicos y sensoriales fijados a nivel mundial por el Consejo Oleícola Internacional. El análisis químico mide el grado de acidez, el índice de peróxidos y la transmisión ultravioleta (K270). El sensorial valora y puntúa las cualidades organolépticas.

El análisis sensorial de aceites debe realizarse utilizando un panel de catadores, seleccionados y entrenados para dicha evaluación. La norma COI T20/Doc.14(17 de noviembre de 1994) define el método de selección de catadores, para constituir un grupo cuyas evaluaciones sean suficientemente fiables para constituir un panel de cata.

A modo de resumen, la formación de un panel de catadores se realiza en cuatro etapas:

- ✗ Entrenamiento previo en el análisis de aceites y selección preliminar por aspectos psicológicos.
- ✗ Determinación de los umbrales medios de sensibilidad del grupo de aspirantes, para los defectos del aceite que determinan su categoría comercial.
- ✗ Determinación del umbral de sensibilidad de cada catador, para dichos atributos y eliminación de los catadores menos sensibles.
- ✗ Determinación de la repetibilidad, reproducibilidad y fiabilidad de las evaluaciones y entrenamiento continuado del grupo para mantenerlas.

El panel mínimo es de ocho catadores y un jefe de panel, ya que un catador aislado está sometido a una posibilidad de error excesiva, especialmente para aquellos aceites que están entre dos categorías comerciales.

En el proceso de evaluación sensorial del aceite, cada catador que forme parte del panel deberá oler y probar el aceite sometido a examen, con el fin de analizar las percepciones olfativas, gustativas y táctiles. A continuación, deberá señalar en la hoja de perfil puesta a su disposición la intensidad a la que percibe cada uno de los atributos negativos y positivos.

Las normas CE Nº.796/2002 y COI T20/Doc.nº5 Rev.1 de 1996 fijan un modelo oficial de hoja de perfil para realizar un análisis de clasificación comercial común a todos los paneles oficiales del mundo (ver Figura 5.1). En dicha hoja, se analiza la intensidad de cada atributo sobre una escala no estructurada de 10 cm de longitud, donde la valoración se extrae al medir la distancia entre el inicio de la escala y la marca, tratándose de un valor numérico en el intervalo 0-10 con un decimal. En concreto, se

---

analizan siempre seis atributos negativos del aceite relacionados con una mala calidad del fruto (atrojado, avinado, moho) y del proceso de extracción y conservación (borras, rancio, metálico). Se pueden analizar otros defectos si los hubiera, siempre que estén definidos en el vocabulario recogido en la norma vigente. También se analizan tres atributos positivos (frutado, amargo y picante), aunque para la evaluación de la calidad del aceite de oliva virgen solamente intervenga el atributo positivo frutado.

### Hoja de Perfil Intensidad

Código del Catador: \_\_\_\_\_  
Código de la Muestra: \_\_\_\_\_  
Fecha: \_\_\_\_\_

Percepción de los atributos negativos

Atrojado	_____→
Moho	_____→
Avinado	_____→
Borras	_____→
Metálico	_____→
Rancio	_____→
Otros	_____→

Percepción del atributo positivo

Frutado	_____→
---------	--------

Figura 7: Hoja de perfil original

El jefe del panel analiza los resultados, utilizando los estimadores estadísticos siguientes: la mediana, la desviación típica sólida, el coeficiente de variación sólido y los intervalos de confianza al 95% sobre la mediana. Los métodos del cálculo se detallan en el Reglamento (CE) Nº 796/2002. Desde el punto de vista de la clasificación comercial del aceite, las diferentes categorías se establecen según los siguientes límites expuestos a continuación:

- ⊕ *Extra* cuando la mediana de los defectos sea igual a 0 y la mediana del frutado sea superior a 0.
- ⊕ *Virgen* cuando:
  - La mediana de los defectos sea  $0 \leq x \leq 3,5$ .
  - La mediana del frutado sea 0.
  - La mediana de los defectos sea inferior o igual a 3.5 y la mediana del frutado igual a 0.
- ⊕ *Lampante* cuando la mediana de los defectos sea superior a 3,5.

**Nota 1:** Por mediana de los defectos se entiende la mediana del defecto percibido con la intensidad más fuerte. El valor del coeficiente de variación sólido para este defecto será inferior o igual a 20%.

**Nota 2:** Cuando la mediana del amargo y/o picante sea superior a 5,0, el jefe de panel lo señalará en el certificado de análisis del aceite.

#### Proceso de Evaluación Sensorial del Aceite de Oliva

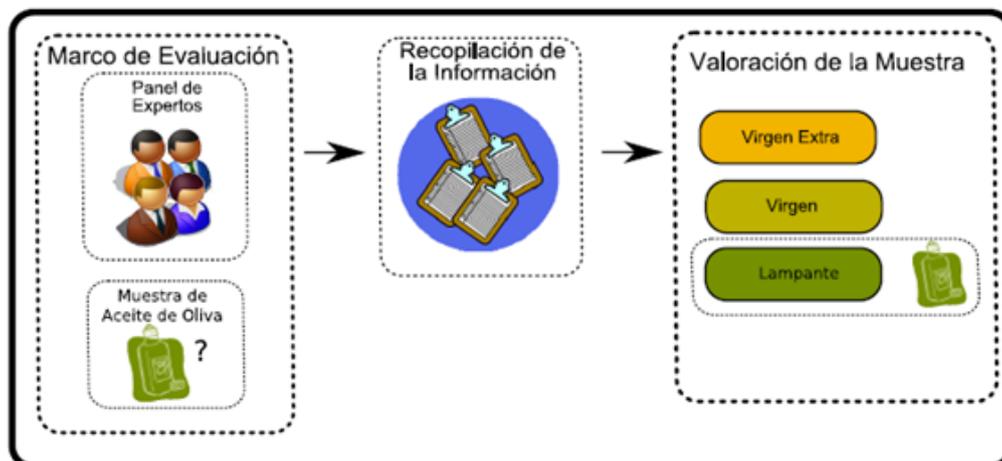


Figura 8: Fases del modelo sensorial del aceite de oliva

### 3.2. Flexibilización del marco de evaluación.

El objetivo del proyecto es aumentar la flexibilidad en el marco de evaluación del proceso de evaluación sensorial del aceite de oliva, proporcionando diferentes alternativas como dominios de expresión para medir las intensidades en la hoja de perfil.

#### 3.2.1. Marco de evaluación: Dominios de expresión

A continuación se van a exponer un ejemplo de obtención de la clasificación del aceite para cada uno de los dominios de expresión, cada uno de éstos utiliza las herramientas que han sido revisadas en el capítulo 2.

#### A) Dominio de expresión numérico

##### i. Marco de evaluación

El marco de evaluación en el que se realizarán las valoraciones será una escala de 0 a 10.

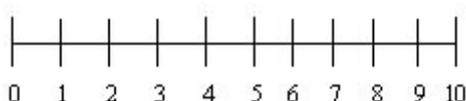


Figura 9: Escala numérica de 0 a 10

Supongamos un panel de cata formado por 9 individuos,  $E = \{e_1, \dots, e_9\}$ , que someterán a evaluación una muestra de aceite de oliva,  $X = \{x_1\}$ , de la que se evaluarán 8 atributos sensoriales,  $F = \{f_1, \dots, f_8\}$ . En concreto, se analizan los siguientes atributos negativos del aceite de oliva: Atrojado, Avinado, Moho, Borrás, Rancio, Metálico, Otros y un único atributo positivo, Frutado, que es el que interviene en la evaluación de la calidad del aceite de oliva.

**Hoja de Perfil  
Intensidad**

Código del Catador: \_\_\_\_\_  
Código de la Muestra: \_\_\_\_\_  
Fecha: \_\_\_\_\_

**Percepción de los atributos negativos**

Atrojado	
	0 1 2 3 4 5 6 7 8 9 10
Moho	
	0 1 2 3 4 5 6 7 8 9 10
Avinado	
	0 1 2 3 4 5 6 7 8 9 10
Borras	
	0 1 2 3 4 5 6 7 8 9 10
Metálico	
	0 1 2 3 4 5 6 7 8 9 10
Rancio	
	0 1 2 3 4 5 6 7 8 9 10
Otros	
	0 1 2 3 4 5 6 7 8 9 10

**Percepción del atributo positivo**

Frutado	
	0 1 2 3 4 5 6 7 8 9 10

Figura 10: Hoja de perfil para dominio de expresión numérico 0-10

## ii. Recopilación de información

En esta fase, se recogerán las intensidades de los atributos percibidos por el panel de catadores utilizando para ello la hoja de perfil indicada en el apartado anterior.

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
$e_1$	0	1	0	1	2	3	0	3
$e_2$	0	0	1	0	3	2	0	4
$e_3$	0	1	0	0	2	4	0	3
$e_4$	1	1	0	1	2	4	0	5
$e_5$	0	1	0	0	1	2	0	3
$e_6$	2	2	0	0	2	4	0	3
$e_7$	1	1	1	0	2	3	0	2
$e_8$	0	0	0	0	4	4	0	3
$e_9$	0	1	2	0	1	2	0	3

Tabla 7: Vectores de utilidad por el panel de catadores dominio numérico 0-10

### iii. Valoración de la muestra

En esta fase, los vectores de utilidad serán agregados para obtener la valoración global de la muestra de aceite de oliva evaluada.

La agregación de las valoraciones se realizará mediante el operador mediana, en caso de que el número de valoraciones sea par se realizará la media de las dos valoraciones centrales.

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
<b>Mediana</b>	0	1	0	0	2	3	0	3

Tabla 8: Medianas obtenidas - dominio numérico 0-10

✘ **Mediana de los defectos:**  $\text{Max}(0,1,0,0,2,3,0) = 3$ .

✘ **Mediana del Frutado:** 3.

Para la evaluación, una vez que para cada uno de los atributos evaluados realizamos la mediana entre las valoraciones de los diferentes expertos y nos quedamos con la mediana máxima de los atributos negativos y con la mediana del atributo positivo frutado.

Categoría	Mediana (Me) del atributo negativo de mayor intensidad	Mediana (Me) del atributo Frutado
Extra	Me = 0	Me > 0
Virgen	$0 \leq \text{Me} \leq 3,5$	
		Me = 0
	$0 \leq \text{Me} \leq 3,5$	Me = 0
Lampante	Me > 3,5	

Tabla 9: Evaluación de la calidad del aceite de oliva - dominio numérico 0-10

En nuestro ejemplo, según la tabla de evaluación la muestra de aceite sería de tipo **Virgen**.

La aplicación desarrollada también da la posibilidad de trabajar con:

- ✘ Valoraciones reales, el proceso sería idéntico al descrito.
- ✘ Valoraciones intervalares tanto enteras como reales. La única diferencia sería que antes de pasar a calcular la mediana de los atributos, tenemos que calcular el valor intermedio del intervalo asignado.
- ✘ Diferentes escalas, para ello, la tabla de clasificación que está enfocada para un rango de 0-10, hay que transformarla a la escala utilizada de la siguiente forma:  **$(3,5 * \text{granularidad})/10$** .

## B) Dominio de expresión lingüístico

### i. Marco de evaluación

El marco de evaluación en el que se realizarán las valoraciones será una escala lingüística de 5 etiquetas.

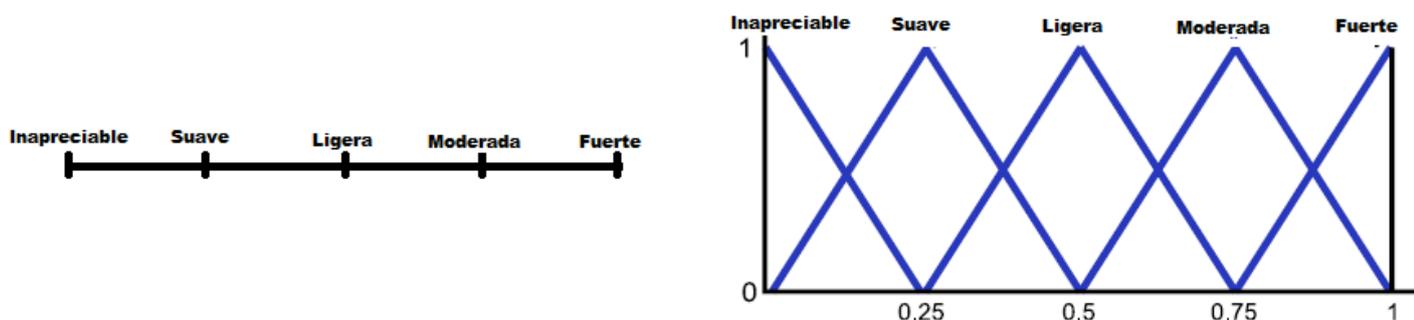


Figura 11: Escala lingüística de 5 etiquetas

A cada una de las etiquetas la identificaremos de la siguiente forma:

Notación	Sintaxis	Semántica
$S_0^5$	Inapreciable	(0,0,0.25)
$S_1^5$	Suave	(0,0.25,0.5)
$S_2^5$	Ligera	(0.25,0.5,0.75)
$S_3^5$	Moderada	(0.5,0.75,1)
$S_4^5$	Fuerte	(0.75,1,1)

Tabla 10: Escala lingüística de 5 etiquetas

Supongamos un panel de cata formado por 9 individuos,  $E = \{e_1, \dots, e_9\}$ , que someterán a evaluación una muestra de aceite de oliva,  $X = \{x_1\}$ , de la que se evaluarán 8 atributos sensoriales,  $F = \{f_1, \dots, f_8\}$ . En concreto, se analizan los siguientes atributos negativos del aceite de oliva: Atrojado, Avinado, Moho, Borrás, Rancio, Metálico, Otros y un único atributo positivo, Frutado, que es el que interviene en la evaluación de la calidad del aceite de oliva.

## Hoja de Perfil Intensidad

Código del Catador: \_\_\_\_\_  
 Código de la Muestra: \_\_\_\_\_  
 Fecha: \_\_\_\_\_

**Percepción de los atributos negativos**

	Inapreciable	Suave	Ligera	Moderada	Fuerte
Atrojado	----- ----- ----- ----- -----				
Moho	----- ----- ----- ----- -----				
Avinado	----- ----- ----- ----- -----				
Borras	----- ----- ----- ----- -----				
Metálico	----- ----- ----- ----- -----				
Rancio	----- ----- ----- ----- -----				
Otros	----- ----- ----- ----- -----				
<b>Percepción del atributo positivo</b>					
Frutado	----- ----- ----- ----- -----				

Figura 12: Hoja de perfil para dominio lingüístico de 5 etiquetas

## ii. Recopilación de información

En esta fase, se recogerán las intensidades de los atributos percibidos por el panel de catadores utilizando para ello la hoja de perfil indicada en el apartado anterior.

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
$e_1$	Inapreciable	Fuerte						
$e_2$	Inapreciable	Moderada						
$e_3$	Inapreciable	Ligera						
$e_4$	Inapreciable	Fuerte						
$e_5$	Inapreciable	Fuerte						
$e_6$	Inapreciable	Moderada						
$e_7$	Inapreciable	Fuerte						
$e_8$	Inapreciable	Ligera						
$e_9$	Inapreciable	Fuerte						

Tabla 11: Vectores de utilidad por el panel de catadores dominio lingüístico de 5 etiquetas

### iii. Valoración de la muestra

En esta fase, los vectores de utilidad serán agregados para obtener la valoración global de la muestra de aceite de oliva evaluada.

La agregación de las valoraciones se realizará mediante el operador mediana, en caso de que el número de valoraciones sea par se realizará la media de las dos valoraciones centrales.

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
<b>Mediana</b>	$S_0^5$	$S_4^5$						

Tabla 12: Medianas obtenidas - dominio lingüístico de 5 etiquetas

✘ **Mediana de los defectos:**

$$\text{Max}(S_0^5, S_0^5, S_0^5, S_0^5, S_0^5, S_0^5, S_0^5) = S_0^5 \rightarrow \text{Inapreciable.}$$

✘ **Mediana del Frutado:**  $S_4^5 \rightarrow$  Extremo.

Para la evaluación, una vez que para cada uno de los atributos evaluados realizamos la mediana entre las valoraciones de los diferentes expertos y nos quedamos

con la mediana máxima de los atributos negativos y con la mediana del atributo positivo frutado.

Como en este ejemplo hemos trabajado con un rango de 0-4 y la tabla de clasificación está enfocada para un rango de 0-10, tenemos que establecer la correspondencia entre ambos de la siguiente forma:  $(3,5 * \text{granularidad})/10$ .

Categoría	Mediana (Me) del atributo negativo de mayor intensidad	Mediana (Me) del atributo Frutado
<b>Extra</b>	$Me = S_0^5$ $Me = 0$	$Me > S_0^5$ $Me > 0$
<b>Virgen</b>	$S_0^5 \leq Me \leq S_2^5$ $0 \leq Me \leq 1,75$	
		$Me = S_0^5$ $Me = 0$
	$S_0^5 \leq Me \leq S_2^5$ $0 \leq Me \leq 1,75$	$Me = S_0^5$ $Me = 0$
<b>Lampante</b>	$Me > S_2^5$ $Me > 1,75$	

Tabla 13: Evaluación de la calidad del aceite de oliva - dominio lingüístico de 5 etiquetas

En nuestro ejemplo, según la tabla de evaluación la muestra de aceite sería de tipo **Virgen Extra**.

**Nota:** En el caso de que la mediana obtenida sea por ejemplo 2.3 la etiqueta correspondiente será la nº 2. Si por el contrario se obtienen valores como 2.5 o 2.8 la etiqueta será la nº 3.

C) Dominio de expresión difuso

i. Marco de evaluación

El marco de evaluación en el que se realizarán las valoraciones será una escala lingüística difusa no balanceada de 4 etiquetas. En este caso pasamos a trabajar con etiquetas trapezoidales cuya semántica está formada por 4 valores (a,b,c,d), es definida por jefe de cata y que deben cumplir las siguientes restricciones:

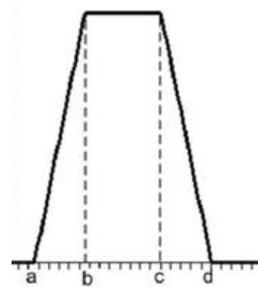


Figura 13: Etiqueta trapezoidal

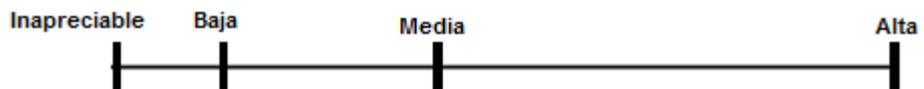
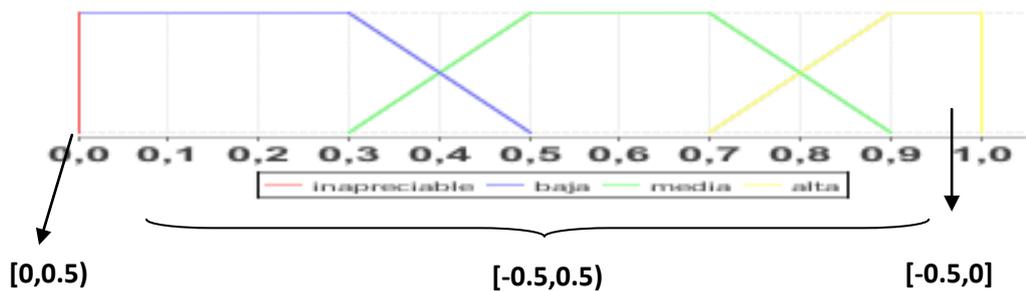


Figura 14: Escala difuso no balanceada de 4 etiquetas



$$a \leq b \leq c \leq d$$

Figura 15: Restricciones de la semántica de las etiquetas difusas

A cada una de las etiquetas la identificaremos de la siguiente forma:

Notación	Sintaxis	Semántica
$S_0^4$	Inapreciable	(0,0,0,0)
$S_1^4$	Baja	(0,0,0.3,0.5)
$S_2^4$	Media	(0.3,0.5,0.7,0.9)
$S_3^4$	Alta	(0.7,0.9,1,1)

Tabla 14: Escala difusa de 4 etiquetas

Supongamos un panel de cata formado por 9 individuos,  $E = \{e_1, \dots, e_9\}$ , que someterán a evaluación una muestra de aceite de oliva,  $X = \{x_1\}$ , de la que se evaluarán 8 atributos sensoriales,  $F = \{f_1, \dots, f_8\}$ . En concreto, se analizan los siguientes atributos negativos del aceite de oliva: Atrojado, Avinado, Moho, Borrás, Rancio, Metálico, Otros y un único atributo positivo, Frutado, que es el que interviene en la evaluación de la calidad del aceite de oliva.

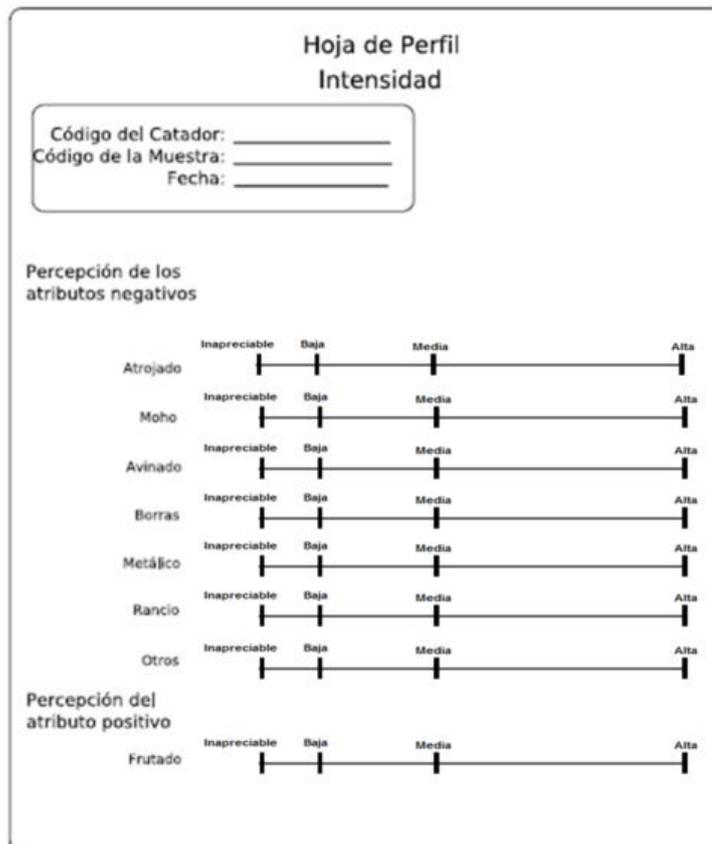


Figura 16: Hoja de perfil para dominio difuso de 4 etiquetas

## ii. Recopilación de información

En esta fase, se recogerán las intensidades de los atributos percibidos por el panel de catadores utilizando para ello la hoja de perfil indicada en el apartado anterior.

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
$e_1$	Inapreciable	Baja						
$e_2$	Inapreciable	Baja						
$e_3$	Inapreciable	Media						
$e_4$	Inapreciable	Media						
$e_5$	Inapreciable	Media						
$e_6$	Inapreciable	Baja						
$e_7$	Inapreciable	Baja						
$e_8$	Inapreciable	Media						
$e_9$	Inapreciable	Media						

Tabla 15: Vectores de utilidad por el panel de catadores dominio difuso de 5 etiquetas

## iii. Valoración de la muestra

En esta fase, los vectores de utilidad serán agregados para obtener la valoración global de la muestra de aceite de oliva evaluada.

La agregación de las valoraciones se realizará mediante el operador mediana, en caso de que el número de valoraciones sea par se realizará la media de las dos valoraciones centrales.

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
<b>Mediana</b>	$S_0^4$	$S_2^4$						

Tabla 16: Medianas obtenidas - dominio difuso de 4 etiquetas

✗ **Mediana de los defectos:**  $\text{Max}(S_0^4, S_0^4, S_0^4, S_0^4, S_0^4, S_0^4, S_0^4) = S_0^4 \rightarrow$  Inapreciable.

✗ **Mediana del Frutado:**  $S_2^4 \rightarrow$  Media.

Ahora pasamos a calcular el centroide de las etiquetas  $S_0^4$  y  $S_2^4$ :

Centroide =  $[h (2l_1 + l_2)/3(l_1 + l_2)]$ , siendo h la altura,  $l_1 = (c-b)$  y  $l_2 = (d-a)$

- ✗ Centroide de  $S_0^4 \rightarrow 0$ .
- ✗ Centroide de  $S_2^4 \rightarrow 0.42$ .

Para la evaluación, una vez que para cada uno de los atributos evaluados realizamos la mediana entre las valoraciones de los diferentes expertos y nos quedamos con la mediana máxima de los atributos negativos y con la mediana del atributo positivo frutado.

Categoría	Centroide de la Mediana (Me) del atributo negativo de mayor intensidad	Centroide de la Mediana (Me) del atributo Frutado
Extra	Me = 0	Me > 0
Virgen	$0 \leq Me \leq 0.35$	
		Me = 0
	$0 \leq Me \leq 0.35$	Me = 0
Lampante	Me > 0.35	

Tabla 17: Evaluación de la calidad del aceite de oliva - dominio difuso de 4 etiquetas

En nuestro ejemplo, según la tabla de evaluación la muestra de aceite sería de tipo **Virgen Extra**.

### D) Dominio de expresión 2-tupla

#### i. Marco de evaluación

El marco de evaluación en el que se realizarán las valoraciones será una escala lingüística de 5 etiquetas.

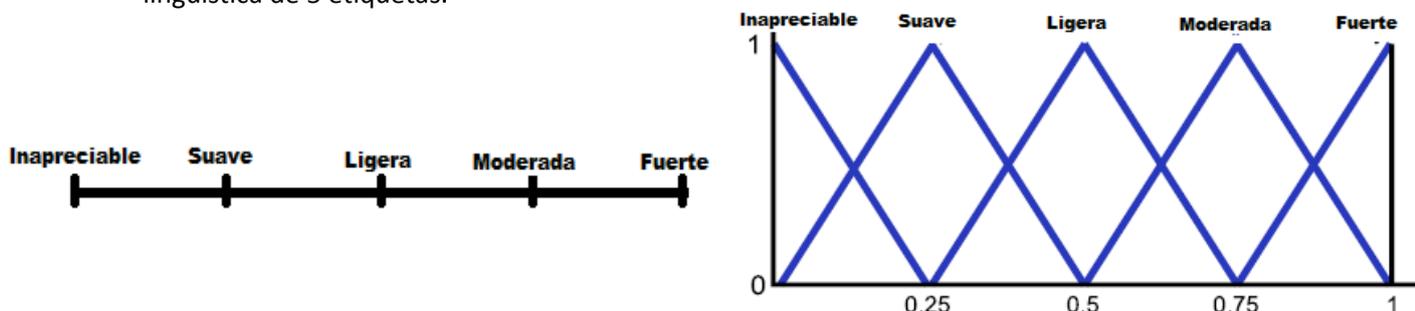


Figura 17: Escala 2-tupla de 5 etiquetas

A cada una de las etiquetas la identificaremos de la siguiente forma:

Notación	Sintaxis	Semántica
$S_0^5$	Inapreciable	(0,0,0.25)
$S_1^5$	Suave	(0,0.25,0.5)
$S_2^5$	Ligera	(0.25,0.5,0.75)
$S_3^5$	Moderada	(0.5,0.75,1)
$S_4^5$	Fuerte	(0.75,1,1)

Tabla 18: Escala 2-tupla de 5 etiquetas

Supongamos un panel de cata formado por 9 individuos,  $E = \{e_1, \dots, e_9\}$ , que someterán a evaluación una muestra de aceite de oliva,  $X = \{x_1\}$ , de la que se evaluarán 8 atributos sensoriales,  $F = \{f_1, \dots, f_8\}$ . En concreto, se analizan los siguientes atributos negativos del aceite de oliva: Atrojado, Avinado, Moho, Borrás, Rancio, Metálico, Otros y un único atributo positivo, Frutado, que es el que interviene en la evaluación de la calidad del aceite de oliva.

En la aplicación se da la posibilidad de trabajar con alfa original (0) o alfa modificado que será determinado por cada catador ( $-0.5 > \alpha \leq 0.5$ ). En este ejemplo utilizaremos alfa modificado.

### Hoja de Perfil Intensidad

Código del Catador: \_\_\_\_\_  
Código de la Muestra: \_\_\_\_\_  
Fecha: \_\_\_\_\_

**Percepción de los atributos negativos**

	Inapreciable	Suave	Ligera	Moderada	Fuerte
Atrojado	----- ----- ----- ----- -----				
Moho	----- ----- ----- ----- -----				
Avinado	----- ----- ----- ----- -----				
Borras	----- ----- ----- ----- -----				
Metálico	----- ----- ----- ----- -----				
Rancio	----- ----- ----- ----- -----				
Otros	----- ----- ----- ----- -----				

**Percepción del atributo positivo**

Frutado	----- ----- ----- ----- -----				
---------	-------------------------------	--	--	--	--

Figura 18: Hoja de perfil para dominio lingüístico de 5 etiquetas

## ii. Recopilación de información

En esta fase, se recogerán las intensidades de los atributos percibidos por el panel de catadores utilizando para ello la hoja de perfil indicada en el apartado anterior.

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
$e_1$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_2^5, 0.4)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_3^5, 0)$
$e_2$	$(S_1^5, 0.4)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_2^5, 0.4)$
$e_3$	$(S_0^5, 0.2)$	$(S_1^5, 0)$	$(S_0^5, 0)$	$(S_4^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_4^5, -0.2)$
$e_4$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_3^5, 0)$
$e_5$	$(S_2^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_3^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_3^5, 0)$
$e_6$	$(S_2^5, -0.3)$	$(S_1^5, -0.4)$	$(S_0^5, 0)$	$(S_3^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_4^5, 0)$
$e_7$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_2^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_4^5, 0.2)$
$e_8$	$(S_0^5, 0)$	$(S_2^5, 0.1)$	$(S_0^5, 0)$	$(S_3^5, 0.2)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_3^5, 0)$
$e_9$	$(S_1^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_4^5, -0.4)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_4^5, 0.4)$

Tabla 19: Vectores de utilidad por el panel de catadores dominio 2-tupla de 5 etiquetas

### iii. Valoración de la muestra

En esta fase, los vectores de utilidad serán agregados para obtener la valoración global de la muestra de aceite de oliva evaluada.

La agregación de las valoraciones se realizará mediante el operador mediana, en caso de que el número de valoraciones sea par se realizará la media de las dos valoraciones centrales. Para realización de la pasaremos las valoraciones 2-tuplas a cantidades numéricas mediante la función inversaDelta().

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
$e_1$	0	0	0	2,4	0	0	0	3
$e_2$	1,4	0	0	0	0	0	0	2,4
$e_3$	0,2	1	0	4	0	0	0	3,8
$e_4$	0	0	0	0	0	0	0	3
$e_5$	2	0	0	3	2	3	0	3
$e_6$	1,7	0,6	0	3	0	0	0	4
$e_7$	0	0	0	2	0	0	0	3
$e_8$	0	2,1	0	3,2	0	0	0	3
$e_9$	1	0	0	3,6	0	0	0	4,4

Tabla 20: Equivalencia entre 2-tuplas y numérico

	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
<b>Mediana</b>	0.2	0	0	3	0	0	0	3
	$(S_0^5, 0.2)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_3^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_0^5, 0)$	$(S_3^5, 0)$

Tabla 21: Medianas obtenidas - dominio 2-tupla de 5 etiquetas

✗ **Mediana de los defectos:**  $\text{Max}((S_0^5, 0.2), (S_0^5, 0), (S_0^5, 0), (S_3^5, 0), (S_0^5, 0), (S_0^5, 0), (S_0^5, 0)) = (S_3^5, 0) \rightarrow \text{Moderado}$ .

✗ **Mediana del Frutado:**  $(S_3^5, 0) \rightarrow \text{Moderado}$ .

Para la evaluación, una vez que para cada uno de los atributos evaluados realizamos la mediana entre las valoraciones de los diferentes expertos y nos quedamos con la mediana máxima de los atributos negativos y con la mediana del atributo positivo frutado.

Como en este ejemplo hemos trabajado con un rango de 0-4 y la tabla de clasificación está enfocada para un rango de 0-10, tenemos que establecer la correspondencia entre ambos de la siguiente forma:  **$(3,5 * \text{granularidad})/10$** .

Categoría	Mediana (Me) del atributo negativo de mayor intensidad	Mediana (Me) del atributo Frutado
Extra	$\text{Me} = (S_0^5, 0)$	$\text{Me} > (S_0^5, 0)$
	$\text{Me} = 0$	$\text{Me} > 0$
Virgen	$(S_0^5, 0) \leq \text{Me} \leq (S_2^5, -0.25)$	
	$0 \leq \text{Me} \leq 1,75$	
		$\text{Me} = (S_0^5, 0)$
		$\text{Me} = 0$
Lampante	$(S_0^5, 0) \leq \text{Me} \leq (S_2^5, -0.25)$	$\text{Me} = (S_0^5, 0)$
	$0 \leq \text{Me} \leq 1,75$	$\text{Me} = 0$
	$\text{Me} > (S_2^5, -0.25)$	
	$\text{Me} > 1,75$	

Tabla 22: Evaluación de la calidad del aceite de oliva - dominio 2-tupla de 5 etiquetas

En nuestro ejemplo, según la tabla de evaluación la muestra de aceite sería de tipo **Lampante**.

**Nota:** En el caso de que la mediana obtenida sea por ejemplo 2.3 la etiqueta correspondiente será la nº 2. Si por el contrario se obtienen valores como 2.5 o 2.8 la etiqueta será la nº 3, con el alfa correspondiente

## CAPÍTULO 3 - PROYECTO

### 1. Descripción

---

Una vez presentado el proyecto a grandes rasgos con su propósito, objetivos y realizada una introducción al modelo teórico que vamos a utilizar para la evaluación llega el momento de detallar el desarrollo del proyecto realizado.

El proyecto aborda el desarrollo de una aplicación de escritorio con interfaz gráfica desarrollado con tecnología JAVA que realiza de forma automática el proceso de evaluación descrito anteriormente.

Esta aplicación se ha diseñado para satisfacer las necesidades de dos perfiles:

- **Administrador:** podrá realizar tareas necesarias para la configuración de las sesiones de cata.
- **Catador:** se limitará a evaluar las hojas de perfil pertenecientes a las sesiones de cata en las que participa.

En este capítulo se detallan las fases de Ingeniería del Software para el desarrollo de un prototipo software.

No existe una definición única y estandarizada para la Ingeniería del Software pero las dos que se presentan a continuación pueden resultar válidas:

– *Ingeniería del Software es la construcción de software de calidad con un presupuesto limitado y un plazo de entrega en contextos de cambio continuo.*

– *Ingeniería del Software es el establecimiento y uso de principios y métodos firmes de ingeniería para obtener software económico que sea fiable y funcione de manera eficiente en máquinas reales.*

Las actividades que conforman la **Ingeniería del Software** son las siguientes:

- ✦ *Especificación de Requerimientos:* se obtienen el propósito del sistema y las propiedades y restricciones del mismo.

- ⊕ *Análisis del Sistema:* se obtiene un modelo del sistema correcto, completo, consistente, claro y verificable.
- ⊕ *Diseño del Sistema:* se definen los objetivos del proyecto y las estrategias a seguir para conseguirlos.
- ⊕ *Implementación:* se traduce el modelo a código fuente.

En los siguientes puntos se profundizará en cada una de estas actividades y en cómo se han llevado a cabo en el ámbito nuestro proyecto.

## 2. Especificación de requerimientos

---

El primer paso en la Ingeniería del Software debe ser determinar el propósito final del proyecto, las propiedades que debe satisfacer y las restricciones a las que está sometido.

Este es, sin duda, una paso de vital importancia dentro de su desarrollo ya que, sin conocer el propósito del proyecto y todas las limitaciones de diversa índole a las que debe hacer frente, difícilmente se podrá realizar una aplicación software que cumpla dicho propósito.

En un proyecto de ámbito comercial para una empresa real, para determinar el propósito del mismo se recurre a una serie de estudios como pueden ser entrevistas con los clientes, encuestas con posibles usuarios, estudios de la situación actual del sistema o estudios de viabilidad. En nuestro caso no nos encontramos ante un proyecto comercial sino ante uno académico por lo que el propósito es conocido desde el mismo momento del planteamiento del mismo:

*La principal tarea de este proyecto es desarrollar una aplicación que permita obtener de forma automática la clasificación de una muestra de aceite de oliva pudiendo utilizar para ello diferentes dominios de expresión como: numérico, lingüístico, difuso y 2-tuplas.*

---

Una vez determinado el propósito final del proyecto, el siguiente paso consiste en especificar los requerimientos del mismo. Los requerimientos de un proyecto software son el conjunto de propiedades o restricciones definidas con total precisión, que dicho proyecto software debe satisfacer. Existen dos tipos bien diferenciados de tales requerimientos:

- **Requerimientos funcionales:** aquellos que se refieren específicamente al funcionamiento de la aplicación o sistema.
- **Requerimientos no funcionales:** aquellos no referidos al funcionamiento estricto sino a otros factores externos.

En los dos siguientes subapartados se pasarán a definir cuáles son estos requerimientos para el proyecto del que se ocupa esta memoria.

Sin embargo, estas definiciones sólo serán previas ya que en la actividad de análisis del sistema, donde se crearán los casos de uso y sus escenarios, se descubrirán nuevas necesidades que no son observables en esta primera actividad y que permitirán refinar completamente estos requerimientos.

### 2.1. Requerimientos funcionales

Los requerimientos funcionales de un sistema software son aquellos que se encargan de describir las funcionalidades que el sistema debe proporcionar a los usuarios del mismo para cumplir sus expectativas.

Normalmente, estos requerimientos se obtendrían de la interacción con el cliente mediante diversas entrevistas y/o encuestas. En nuestro caso, al tratarse de un proyecto académico, nos encontramos ante la situación de la no existencia de cliente alguno por lo que la información sobre las funcionalidades que debe disponer el sistema se han obtenido de las necesidades de profesores.

Una vez detectadas las funcionalidades que los distintos perfiles de usuario pueden reclamar a nuestro sistema, se va a proceder a asignar a cada una, un identificador con el siguiente formato: **RF-Número del Requisito:**

---

✚ Para el usuario **Administrador**:

- **RF-01: Crear y eliminar usuarios**

El sistema debe proporcionar la posibilidad de crear y eliminar tanto usuarios de tipo administrador como catador.

- **RF-02: Crear y eliminar atributos**

El sistema debe proporcionar la posibilidad de crear y eliminar atributos de tipo negativo y positivo que serán posteriormente asociados a las hojas de perfil.

No se podrá borrar un atributo que esté vinculado a una hoja de perfil existente en una sesión de cata.

- **RF-03: Crear y eliminar dominios de expresión**

El sistema debe proporcionar la posibilidad de crear y eliminar los diferentes dominios de expresión:

- **Numérico** --> Dar de alta y eliminar rangos
- **Lingüístico, 2-tuplas y difuso** --> Dar de alta y eliminar escalas y juegos de etiquetas.

No se podrá eliminar un dominio de expresión (rango o juego de etiquetas) mientras que esté asociado a alguna sesión de cata.

- **RF-04: Crear y eliminar muestras de aceite**

El sistema debe proporcionar la posibilidad de crear y eliminar muestras de aceite.

Una muestra puede ser asignada a más de una sesión de cata y no podrá ser eliminada mientras esté asociada a alguna de ellas.

- **RF-05: Consultar las valoraciones realizadas por cada catador en las hojas de perfil**

El sistema debe proporcionar la posibilidad de consultar las valoraciones de la hojas de perfil hechas por los catadores una vez que han evaluado la muestra.

---

- **RF-06: Consultar la hoja de perfil agregada**

El sistema debe proporcionar la posibilidad de consultar la hoja de perfil agregada que se obtendría después de agregar las valoraciones de todos los catadores/as que han participado en la sesión.

- **RF-07: Consultar en qué estado se encuentra en cada momento una sesión de cata**

El sistema debe proporcionar la posibilidad de consultar el estado en el que se encuentra las sesiones de cata. Los estados posibles son:

- **Pendiente de catadores** --> Todavía quedan catadores pertenecientes a la sesión que no han valorado la muestra.
- **Pendiente de agregar la información** --> Todos los catadores participantes en la sesión han evaluado la muestra y está a la espera de que el administrador obtenga el resultado.
- **Finalizada** --> Se ha agregado la información de todos los catadores y ya se ha obtenido la clasificación de la muestra.

- **RF-08: Eliminar catadores de las sesiones de cata si aún no han evaluado la hoja de perfil correspondiente.**

El sistema debe proporcionar la posibilidad de eliminar a aquellos catadores pertenecientes a la sesión de cata que todavía no hayan valorado la muestra.

- **RF-09: Crear y eliminar sesiones de cata**

El sistema debe proporcionar la posibilidad de crear y eliminar sesiones de cata.

Para la creación de una sesión debe de existir previamente en el sistema al menos: un usuario de tipo catador, un atributo negativo, un atributo positivo (frutado), una muestra y en el caso de los dominios de expresión, un rango para numérico y un juego de etiquetas para lingüístico, difuso y 2-tupas.

- **RF-10: Clasificar una muestra de aceite**

El sistema debe de dar la posibilidad al usuario administrador de obtener la clasificación de la muestra de aceite valorada.

Para ello previamente todos los catadores que intervengan en la sesión de cata deben de haber completado sus hojas de perfil.

Para el usuario *catador*:

- **RF-11: Evaluar las hojas de perfil pendientes**

El sistema debe proporcionar la posibilidad de que los catadores introduzcan las valoraciones para cada atributo de las hojas de perfil que tiene asignadas.

- **RF-12: Consultar las hojas de perfil ya que han sido evaluadas**

El sistema debe proporcionar la posibilidad de consultar las valoraciones de las hojas de perfil que ha evaluado con anterioridad, dando la posibilidad de modificar los valores si aún no se ha agregado la información con el resto de catadores que participan en la sesión de cata.

## 2.2. Requerimientos no funcionales

Los requerimientos no funcionales son aquellos que restringen los requerimientos funcionales. Son tan importantes como los propios requerimientos funcionales y pueden incluso llegar a ser críticos para la aceptación del sistema. Estos requerimientos normalmente especifican propiedades del sistema o del producto en si (plataforma, velocidad, rendimiento...) y del diseño de la interfaz gráfica con el usuario además de todas las restricciones impuestas por la organización (políticas de empresa, estándares, legalidad vigente...).

Al no ser este un proyecto comercial para ninguna organización o empresa real, no debemos someternos a restricciones de la organización. Por lo tanto, los requerimientos no funcionales que se deben obtener y analizar son los referentes a las necesidades hardware y software de los equipos informáticos para que estos

---

proporcionen al usuario las funcionalidades requeridas de forma eficiente y los referentes a la interfaz gráfica entre la aplicación y el usuario.

### *A) Requerimientos del equipo informático*

Al hablar de los requerimientos del equipo informático y debido a que el marco del desarrollo de la aplicación es local, indicaremos a continuación los requerimientos del equipo en el que se instalará la aplicación.

Los requerimientos del equipo se dividen en dos tipos: los requerimientos de hardware y los requerimientos software.

#### **1) Hardware**

- ✓ *Velocidad*: el equipo debe ser lo suficientemente rápido como para ejecutar la aplicación en el menor tiempo posible y con la mayor fiabilidad. Cualquier microprocesador actual es capaz de cumplir con esta labor.
  
- ✓ *Memoria*: el equipo debe disponer de la suficiente memoria RAM libre para realizar las operaciones que se soliciten entre la aplicación y la base de datos.  
Para un funcionamiento fluido es aconsejable que el equipo tenga al menos 1GB.
  
- ✓ *Almacenamiento*: el equipo debe tener una capacidad de almacenamiento suficiente para almacenar la base de datos. Cualquier disco duro actual es capaz de cumplir con esta labor, si es de tecnología SATAIII mejorará considerablemente el rendimiento.
  
- ✓ *Tarjeta gráfica*: las tarjetas gráficas de las que disponen los equipos informáticos actuales son de gran potencia por lo que cualquiera de ellas puede ser utilizable.
  
- ✓ *Monitor*: debe soportar una resolución de 800x600 y superiores.

## 2) Software

- ✓ *Sistema Operativo:* la aplicación puede trabajar sobre un sistema operativo Windows XP o Linux.
- ✓ Versión de la máquina virtual de Java JRE 1.7 o superiores.
- ✓ Servidor de base de datos MySQL v5.6.
- ✓ .jar que contiene la aplicación y el script para montar la base de datos. Estos archivos serán proporcionados al administrador de la aplicación.

### *B) Requerimientos de la interfaz*

Los requerimientos de la interfaz gráfica entre la aplicación y el usuario están íntimamente ligados a la usabilidad y sus principios. La usabilidad se puede definir de varias formas:

– Usabilidad se define coloquialmente como facilidad de uso, ya sea de una página web, una aplicación informática o cualquier otro sistema que interactúe con un usuario.

– La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso.

– Usabilidad es la efectividad, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico.

A partir de estas tres definiciones se pueden obtener los principios básicos de la usabilidad, los cuales se asociarán a los requerimientos no funcionales que deberá cumplir la interfaz gráfica:

- ✓ **Facilidad de aprendizaje:** se refiere a la facilidad con la que nuevos usuarios pueden tener una interacción efectiva. Depende de los siguientes factores:
  - *Predecible:* una vez conocida la aplicación, se debe saber en cada momento a qué estado se pasará en función de la tarea que se realice.
  - *Sintetizable:* los cambios de estado tras una acción deben ser fácilmente captados.
  - *Generalizable:* las tareas semejantes se resuelven de modo parecido.
  - *Familiarizable:* el aspecto de la interfaz tiene que resultar conocido y familiar para el usuario.
  
- ✓ **Consistente:** siempre se han de seguir una misma serie de pasos para realizar una tarea determinada.
  
- ✓ **Flexible:** relativa a la variedad de posibilidades con las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar la tarea, similitud con tareas anteriores y la optimización entre el usuario y el sistema.
  
- ✓ **Robustez:** es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos o, también, la capacidad del sistema para tolerar fallos. Está relacionada con los siguiente factores:
  - *Navegable:* el usuario debe poder observar el estado del sistema sin que esta observación repercuta de forma negativa en él.

- *Recuperación de información*: la aplicación debe poder deshacer alguna operación y permitir volver a un estado anterior.
- *Valores por defecto*: ayudan al usuario introduciendo posibles valores de uso común permitiendo el recuerdo pasivo.
- *Tiempo de respuesta*: es el tiempo necesario para que el sistema pueda mostrar los cambios realizados por el usuario.

### 3. Análisis del sistema

---

Una vez conocido el propósito del proyecto software, las propiedades que debe cumplir y las restricciones a las que debe someterse, llega el momento de analizar el sistema y crear un modelo del mismo que sea correcto, completo, consistente, claro y verificable. Para conseguir esto se estudiarán los perfiles de usuario y se crearán y definirán casos de uso en base a los requerimientos previamente obtenidos. Por último se describirán ciertos escenarios de acción de dichos casos de uso.

#### 3.1. Perfil de usuario

En esta fase el primer paso es determinar quiénes son los usuarios potenciales de la aplicación, para a partir de esto, obtener las características generales que nos permitan caracterizar los requisitos de usabilidad que posteriormente habrá que tener en cuenta en el diseño de la aplicación y de su interfaz gráfica.

Nuestro sistema, por lo general va a contar con dos tipos de usuarios principales: administrador y catador.

Pasamos a comentar las características de cada uno de los usuarios de nuestro sistema:

**✚ Administrador:**

- *Conocimientos del dominio del problema:* debe tener alto conocimiento del dominio puesto que va a ser el encargado de gestionar toda la información necesario para las sesiones de cata.
- *Sobre uso de equipos/programas informáticos:* no es necesario que tenga elevados conocimientos informáticos puesto que solo se va a utilizar la aplicación.
- *Entorno de trabajo:* estará situado en el lugar de trabajo, es decir, en el lugar donde se lleven a cabo las sesiones de cata.
- *Nivel cultural:* sería recomendable que el administrador tenga un nivel cultural medio, pero no es necesario que tenga un nivel cultural mínimo.
- *Habilidades sociales:* puesto que no es una ocupación de cara al público, el administrador no tiene porque tener habilidades específicas de trato hacia el público, pero si será importante que sepa trabajar en grupo y tener habilidades de interrelación personal con sus compañeros de trabajo.

**✚ Catador:**

- *Conocimientos del dominio del problema:* debe tener experiencia en la cata de muestras de aceite de oliva.
- *Sobre uso de equipos/programas informáticos:* no es necesario que tenga elevados conocimientos informáticos puesto que solo se va a utilizar a utilizar unas funcionalidades muy concretas de la aplicación.
- *Entorno de trabajo:* normalmente será el lugar donde se ha realizado la sesión de cata.

- *Nivel cultural:* No es necesario que tenga un nivel cultural mínimo.
- *Habilidades sociales:* puesto que no es una ocupación de cara al público, el catador no tiene porqué tener habilidades específicas de trato hacia el público, pero si será importante que sepa trabajar en grupo y tener habilidades de interrelación personal con sus compañeros de la sesión de cata.

### 3.2. Casos de Uso

Para estudiar su funcionalidad haremos uso de sus diagramas de casos de uso. Un caso de uso representa una clase de funcionalidad dada por el sistema como un flujo de eventos. También se puede definir como la representación de una situación o tarea de interacción de un usuario con la aplicación. Los casos de uso describen como se realiza una tarea de manera exacta y constan de los siguientes elementos:

- Nombre único e unívoco.
- Actores participantes.
- Condiciones de entrada.
- Flujo de eventos.
- Condiciones de salida.
- Requerimientos especiales.

Por lo tanto, con el estudio de los casos de uso de la aplicación podemos determinar cuáles son los actores participantes. Un actor modela una entidad externa que se comunica con el sistema, es decir, es un tipo de usuario del sistema. Un actor, al igual que un caso de uso, debe tener un nombre único y puede tener una descripción asociada.

En nuestro sistema contamos con los actores descritos en el punto anterior, por lo que es el momento de crear los distintos casos de uso. A la hora de realizar esta acción es importante que cada uno de los requerimientos funcionales ya definidos aparezca en al menos uno de los casos de uso aunque, por otra parte, puede haber casos de uso nuevos, en los que no aparezca ninguno de los requerimientos, ya que

---

estamos en una fase de refinamiento del sistema donde queremos construir un modelo detallado del mismo.

Primer caso de uso del sistema es el diagrama frontera el cual describe completamente la funcionalidad de un sistema.

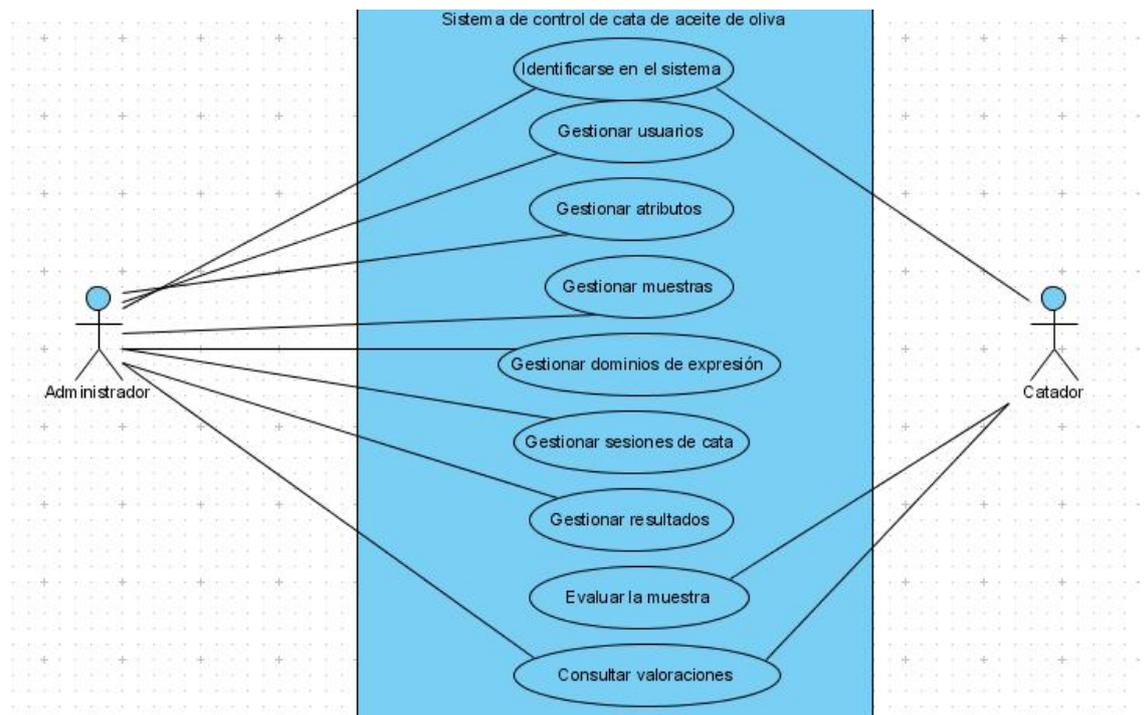


Figura 19: Diagrama frontera

Los casos de uso están representados por elipses y los actores por Figuras humanas. Los actores podrán realizar aquellos casos de uso que señalen con una línea horizontal. Nos podemos encontrar con 2 tipos de actores que interactúan con el sistema así como con 9 casos de uso diferentes.

Los casos de uso mostrados en un diagrama frontera pueden ser lo suficientemente exactos o, por el contrario, pueden ser concretados con un mayor detalle. A la hora de detallar un caso de uso se pueden emplear dos tipos de relaciones:

- **<<extend>>**: es una relación que representa comportamientos excepcionales del caso de uso. La dirección de este tipo de relación es hace el caso de uso extendido.
- **<<include>>**: es una relación cuya dirección es contraria a la de la relación <<extend>> que representa un comportamiento común del caso de uso.

En nuestro caso nos encontramos con algunos casos de uso que requieren ser detallados con mayor profundidad:

### *Caso de uso 1: Gestionar usuarios*

**Actores participantes:** Administrador.

**Condiciones de entrada:** que el Administrador esté identificado en el sistema.

#### **Flujo de eventos:**

1. El sistema muestra el menú principal.
2. Administrador elige la opción " Gestionar usuarios".
3. El sistema ofrece diferentes opciones:
4. Si el Administrador elige "Añadir usuario", se realiza S-1.
5. Si el Administrador elige "Eliminar usuario" se realiza S-2.

#### **Subflujo de eventos:**

##### **S-1: Añadir usuario**

- 1.1 El sistema muestra al Administrador un formulario de entrada de datos.
- 1.2 El Administrador introduce nombre, contraseña por duplicado y tipo de usuario (catador o administrador).
- 1.3 El sistema actualiza la base de datos (E-1) y actualiza el listado de usuarios existentes.
- 1.4 El caso de uso de se inicia de nuevo.

## S-2: Eliminar usuario

2.1 El Administrador selecciona el nombre del usuario que desea eliminar y pulsa el botón "Eliminar usuario".

2.2 El sistema actualiza la base de datos (E-2) y actualiza el listado de usuarios existentes.

2.3 El caso de uso de se inicia de nuevo.

**Condiciones de salida:** Se actualiza la BD con la información introducida y el Administrador visualiza los cambios realizados en los usuarios.

### Excepciones:

E-1: El usuario introducido por el Administrador no es válido porque está repetido el nombre, la contraseña es inferior a 6 caracteres o la repetición de la contraseña es incorrecta. El sistema informa al usuario de dicha situación. El Administrador debe introducir de forma correcta todos los datos del usuario.

E-2: El Administrador va a eliminar el único usuario de tipo administrador que hay en el sistema y esta acción no es posible. El sistema informa al usuario de dicha situación.

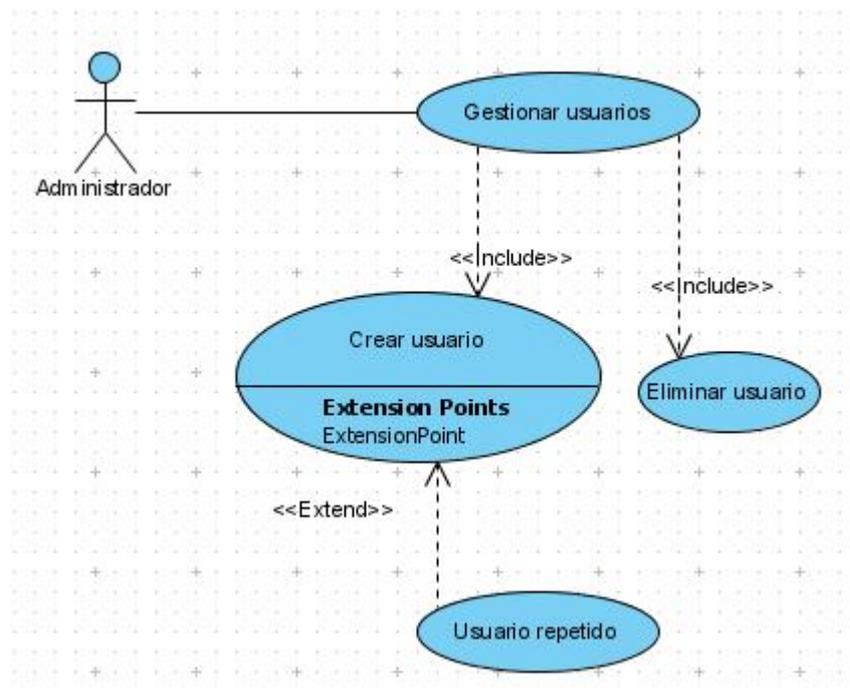


Figura 20: Caso de uso - Gestionar usuarios

### Caso de uso 3: Gestionar muestras

**Actores participantes:** Administrador.

**Condiciones de entrada:** que el Administrador esté identificado en el sistema.

**Flujo de eventos:**

1. El sistema muestra el menú principal.
2. Administrador elige la opción " Gestionar muestras".
3. El sistema ofrece diferentes opciones:
4. Si el Administrador elige "Añadir muestra", se realiza S-1.
5. Si el Administrador elige "Eliminar muestra" se realiza S-2.
6. Si el Administrador elige "Consultar sesiones de cata" S-3.

**Subflujo de eventos**

**S-1: Añadir muestra**

- 1.1 El sistema muestra al Administrador un formulario de entrada de datos.
- 1.2 El Administrador introduce nombre e identificador de la muestra y nombre del cliente propietario de la muestra.
- 1.3 El sistema actualiza la base de datos (E-1) y actualiza el listado de muestras existentes.
- 1.4 El caso de uso de se inicia de nuevo.

**S-2: Eliminar muestra**

- 2.1 El Administrador selecciona la muestra que desea eliminar y pulsa el botón "Eliminar muestra".
- 2.2 El sistema actualiza la base de datos (E-2) y actualiza el listado de muestras existentes.
- 2.3 El caso de uso de se inicia de nuevo.

**S-3: Consultar sesiones de cata**

- 3.1 El Administrador selecciona la muestra sobre la que desea información más detallada y pulsa el botón "Ver sesiones de cata".
-

3.2 El sistema muestra una nueva tabla donde se desglosan las sesiones de cata asociadas a la muestra elegida y los catadores participantes.

3.3 El Administrador selecciona la hoja de perfil que desea consultar y pulsa en "Ver hoja de perfil" (E-3).

3.4 El sistema muestra la hoja de perfil solicitada.

3.5 El caso de uso de se inicia de nuevo.

**Condiciones de salida:** Se actualiza la BD con la información introducida y el Administrador visualiza los cambios realizados en las muestras.

**Excepciones:**

E-1: El identificador o nombre introducidos por el Administrador no es válido porque están repetidos. El sistema informa al usuario de dicha situación. El Administrador debe introducir de forma correcta todos los datos de la muestra.

E-2: El Administrador intenta eliminar una muestra que está asociada a alguna sesión de cata vigente. El sistema informa al usuario de dicha situación. El Administrador debe elegir otra muestra o esperar a que no esté ligada a ninguna sesión de cata para eliminarla.

E-3: El Administrador intenta consultar una hoja de perfil que no ha sido evaluada por el catador todavía o en el caso de las hojas de perfil agregada no se han agregado todavía las valoraciones de todos los catadores. El sistema informa al usuario de dicha situación. El Administrador debe elegir otra hoja de perfil o esperar a que se den las condiciones necesarias para poder consultarla.

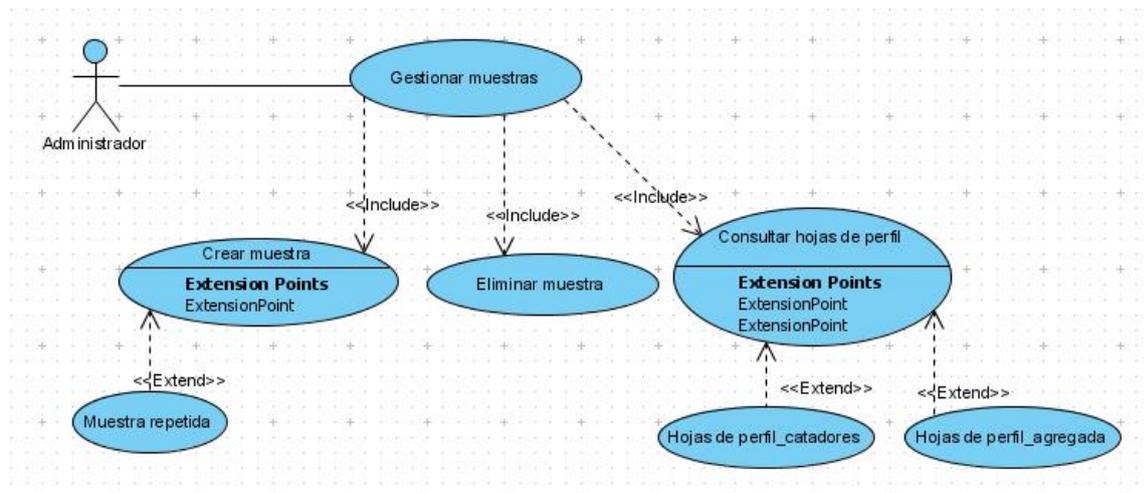


Figura 21: Caso de uso - Gestionar muestras

### Caso de uso 5: Gestionar sesiones de cata

**Actores participantes:** Administrador.

**Condiciones de entrada:** que el Administrador esté identificado en el sistema.

**Flujo de eventos:**

1. El sistema muestra el menú principal.
2. Administrador elige la opción " Gestionar sesiones de cata".
3. El sistema ofrece diferentes opciones:
4. Si el Administrador elige "Añadir sesión de cata", se realiza S-1.
5. Si el Administrador elige "Eliminar sesión de cata" se realiza S-2.

**Subflujo de eventos**

#### **S-1: Añadir sesión de cata**

1.1 El sistema muestra al Administrador un formulario de entrada de datos.

1.2 El Administrador introduce:

- Nombre.
- Identificador.
- Catadores.
- Atributos negativos y positivos.

- Muestra de aceite.
- Dominio de expresión:
  - Numérico --> rango, valores o intervalos, enteros o reales.
  - Lingüístico, difuso --> Escala y juego de etiquetas.
  - 2-tupas --> Escala, juego de etiquetas, alfa original o modificado.

1.3 El sistema actualiza la base de batos (E-1) y actualiza el listado de usuarios existentes.

1.4 El caso de uso de se inicia de nuevo.

## **S-2: Eliminar sesión de cata**

2.1 El Administrador seleccionar el nombre de la sesión de cata que desea eliminar y pulsa el botón "Eliminar sesión".

2.2 El sistema actualiza la base de batos y actualiza el listado de sesiones existentes.

2.3 El caso de uso de se inicia de nuevo.

**Condiciones de salida:** Se actualiza la BD con la información introducida y el Administrador visualiza los cambios realizados en las sesiones de cata.

### **Excepciones**

E-1: El identificador o nombre introducidos por el Administrador no es válido porque están repetidos o se han dejado campos sin rellenar. El sistema informa al usuario de dicha situación. El Administrador debe introducir de forma correcta todos los datos de la sesión.

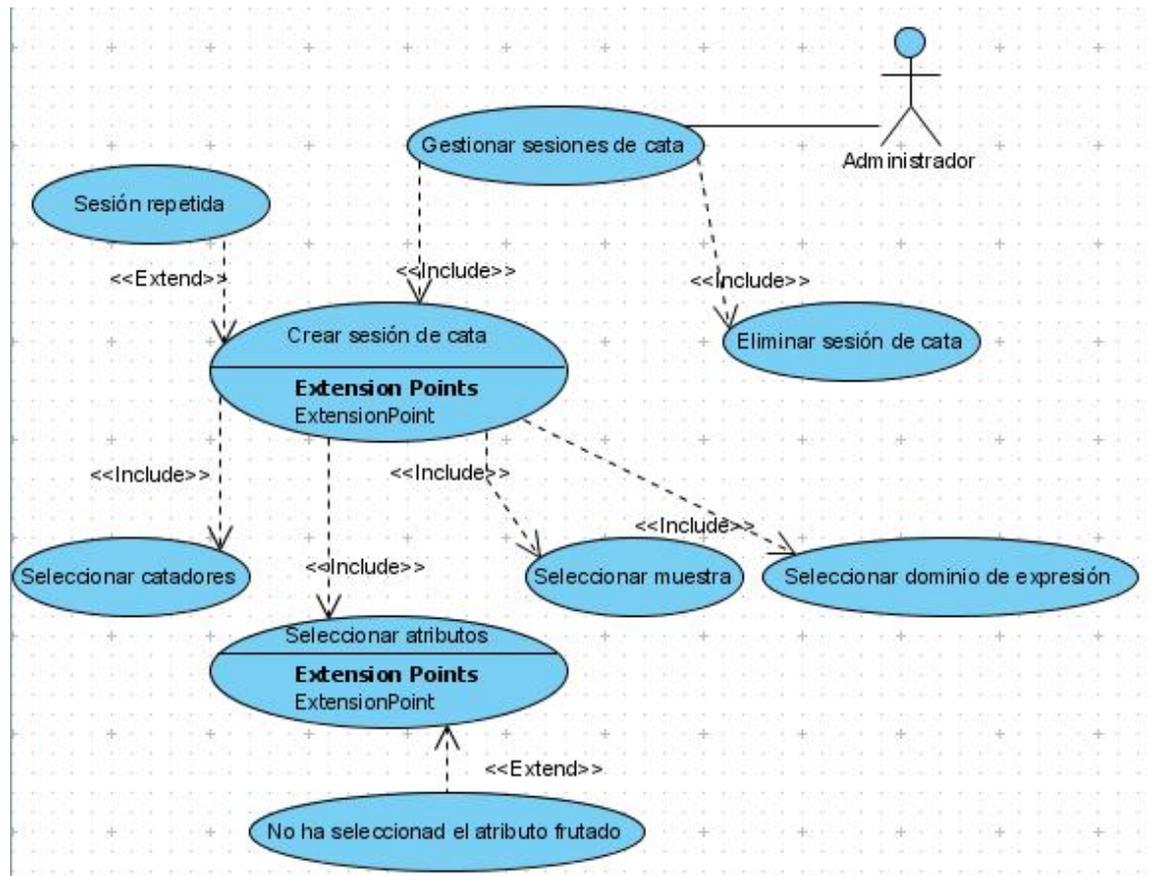


Figura 22: Caso de uso - Gestionar sesiones de cata

### Caso de uso 6: Gestionar resultados

**Actores participantes:** Administrador.

**Condiciones de entrada:** que el Administrador esté identificado en el sistema.

#### **Flujo de eventos:**

1. El sistema muestra el menú principal.
2. Administrador elige la opción " Gestionar resultados".
3. El sistema ofrece diferentes opciones:
4. Si el Administrador elige "Eliminar hoja de perfil", se realiza S-1.
5. Si el Administrador elige "Obtener resultados " se realiza S-2.
6. Si el Administrador elige "Ver resultados" se realiza S-3.

#### **Subflujo de eventos**

**S-1: Eliminar hoja de perfil**

- 1.1 El sistema muestra al Administrador un listado de las sesiones de cata que están aún abiertas porque algunos de los catadores todavía no han evaluado la muestra.
- 1.2 El Administrador puede ver quiénes lo han hecho y cuáles no.
- 1.3 Si el Administrador desea borrar a algún catador que todavía no lo ha hecho, selecciona su nombre y pulsa en "Eliminar hoja de perfil" (E-1).
- 1.4 El sistema actualiza la base de datos y actualiza el listado de hojas de perfil. Si tras esta acción todas las hojas de perfil que pertenecen a la sesión están evaluadas, la sesión de cata sale de la categoría en la que se encuentra para pasar a "Sesiones de cata pendientes de agregar valoraciones".
- 1.5 El caso de uso de se inicia de nuevo.

**S-2: Obtener resultados**

- 2.1 El Administrador selecciona el nombre de la sesión de cata que desea y pulsa el botón "Obtener resultados".
- 2.2 El sistema muestra un mensaje con la clasificación que se ha obtenido para la muestra de aceite implicada.
- 2.3 El sistema actualiza la base de datos y actualiza el listado de sesiones ya que esta sesión pasa a la categoría de "Sesiones de cata finalizadas".
- 2.4 El caso de uso de se inicia de nuevo.

**S-3: Ver resultados**

- 3.1 El Administrador selecciona el nombre de la sesión de cata que desea consultar y pulsa el botón "Ver resultados".
- 3.2 El sistema le muestra un mensaje donde se indica el nombre de la sesión de cata, el nombre de la muestra, fecha de creación y evaluación de la sesión de cata, mediana de los atributos negativos, mediana de los atributos positivo frutado y la clasificación obtenida.
- 3.3 El caso de uso de se inicia de nuevo.

**Condiciones de salida:** Se actualiza la BD con la información introducida y el Administrador visualiza los cambios de estado de la sesión de cata.

---

### Excepciones

E-1: El Administrador ha seleccionado una hoja de perfil que ya está evaluado o si no lo estaba es la única que existe en la sesión de cata. El sistema informa al usuario de dicha situación. El Administrador debe elegir una que esté sin evaluar o si es la única no la podrá eliminar.

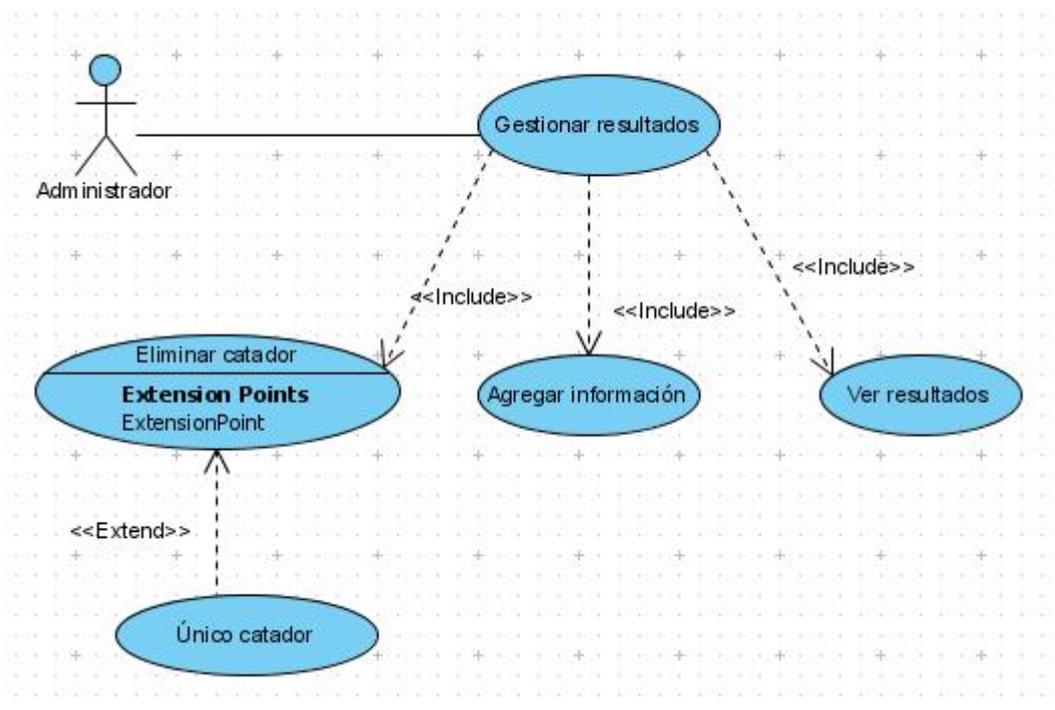


Figura 23: Casos de uso - Gestionar resultados

### Caso de uso 7: Evaluar muestras

**Actores participantes:** Catador.

**Condiciones de entrada:** que el usuario catador esté identificado en el sistema.

**Flujo de eventos:**

1. El sistema muestra el menú.
2. El usuario catador elige la opción " Listar hojas de perfil pendientes".
3. El catador selecciona la hoja de perfil que desea evaluar.
4. Asigna las valoraciones correspondientes a cada atributo.

**Condiciones de salida:** Se actualiza la BD con la información introducida y el catador vuelve a visualizar la pantalla donde podrá elegir otra hoja de perfil pendiente de evaluar.

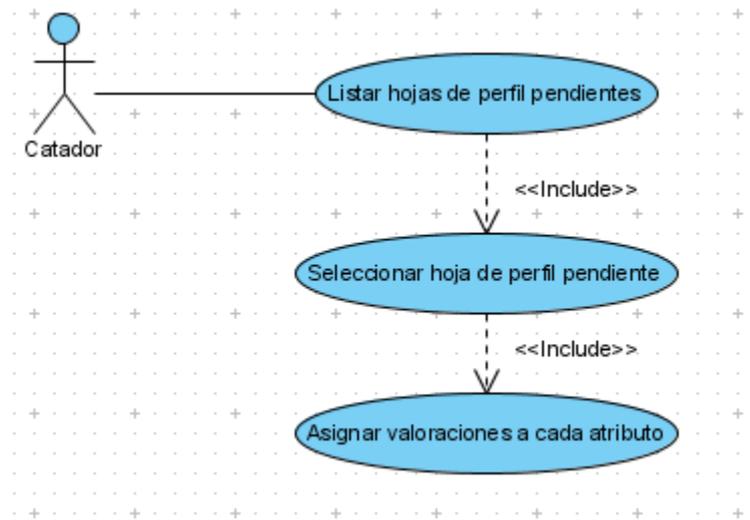


Figura 24: Caso de uso - Evaluar muestras

### 3.3. Escenarios

Un caso de uso es una representación abstracta, una abstracción, de una funcionalidad del sistema a realizar. La representación concreta de un caso de uso se realiza mediante la creación de uno o más escenarios que muestren todas las interacciones posibles entre el sistema y sus usuarios.

Un escenario está formado por los siguientes elementos:

- Un nombre único y unívoco.
- Una descripción.
- Los actores participantes.
- El flujo de eventos.

Como se ha indicado, para cada caso de uso puede haber varios escenarios. Para nuestro proyecto se han creado y descrito una cantidad importante de casos de uso por lo que no vamos a definir todos los escenarios de cada uno de ellos sino que vamos a definir unos pocos que puedan servir como ejemplo de las principales funcionalidades

que el sistema va a desarrollar: Gestionar usuarios, Gestionar muestras, *Gestionar sesiones de cata* y *Obtener resultados*.

### Escenario 1

- **Nombre** : CrearUsuarioCatador.
- **Descripción**: El usuario administrador "admin" desea crear un usuario de tipo catador.
- **Actor**: Administrador.
- **Flujo de eventos**:
  1. El usuario entra al sistema.
  2. El sistema muestra un formulario de entrada.
  3. El usuario introduce sus credenciales: Admin, 123456.
  4. El usuario administrador elige la opción "Gestionar usuarios".
  5. El usuario pulsa en "Añadir usuario".
  6. El sistema muestra una serie de campos en los que el usuario introduce:
    - El nombre del usuario : Manuel.
    - La contraseña : 123abc.
    - Repetición de la contraseña: 123abc.
    - Tipo de usuario: catador.
  7. El usuario guarda los datos.
  8. El sistema se comunica con la BD y almacena dicha información.
  9. El sistema muestra en el listado de usuarios catadores existentes el usuario Manuel que se acaba de crear.

### Escenario 2

- **Nombre** : VerHojaDePerfilAgregada.
  - **Descripción**: El usuario administrador "admin" desea consultar la hoja de perfil agregada que se ha obtenido para una muestra en una determinada sesión de cata.
  - **Actor**: Administrador.
  - **Flujo de eventos**:
    1. El usuario entra al sistema.
    2. El sistema muestra un formulario de entrada.
-

3. El usuario introduce sus credenciales: Admin, 123456.
4. El usuario administrador elige la opción "Gestionar muestras".
5. El usuario pincha sobre la muestra que le interesa y pulsa en el botón "Ver sesiones de cata".
6. El sistema muestra una nueva tabla debajo con la información de las diferentes sesiones de cata y hojas de perfil relacionadas con la muestra.
7. El usuario elige la hoja de perfil agregada de la sesión de cata seleccionada y pulsa en "Ver hoja de perfil".
8. El sistema muestra al usuario la hoja de perfil correspondiente donde podrá ver las valoraciones agregadas para cada atributo evaluado.

### Escenario 3

- **Nombre** : CrearSesiónDeCata.
  - **Descripción**: El usuario administrador "admin" desea crear una sesión de cata.
  - **Actor**: Administrador.
  - **Flujo de eventos**:
    1. El usuario entra al sistema.
    2. El sistema muestra un formulario de entrada.
    3. El usuario introduce sus credenciales: Admin, 123456.
    4. El usuario administrador elige la opción "Gestionar sesiones de cata".
    5. El usuario pulsa en "Añadir sesión de cata".
    6. El sistema muestra una serie de campos en los que el usuario introduce:
      - El nombre de la sesión de cata : 1-NVO-1.
      - El identificador de la sesión de cata : 1.
      - Los catadores que van a participar en ella: Manuel, María, José, Miguel, Ana, Gema, Juan y Pablo.
      - Los atributos que van a formar la hoja de perfil: Atrojado, Avinado, Moho, Metálico, Rancio, Otros, Frutado, Amargo y Picante.
      - La muestra de aceite: AT5.
      - El tipo de dominio de expresión: numérico --> Valor original entero [0-10].
-

7. El usuario guarda la sesión.
8. El sistema se comunica con la BD y almacena dicha información.
9. El sistema muestra en el listado de sesiones de cata existentes la sesión 1-NVO-1 que se acaba de crear.

#### Escenario 4

- **Nombre** : ObtenerResultados.
- **Descripción**: El usuario administrador "admin" desea obtener la clasificación de la muestra de aceite que ha sido evaluada por los catadores que forman parte de la sesión de cata.
- **Actor**: Administrador.
- **Flujo de eventos**:
  1. El usuario entra al sistema.
  2. El sistema muestra un formulario de entrada.
  3. El usuario introduce sus credenciales: Admin, 123456.
  4. El usuario administrador elige la opción "Gestionar resultados".
  5. El usuario observa que la sesión TS001 se encuentra en el apartado de "Sesiones de cata pendientes de agregar valoraciones" y pulsa en el botón "Obtener resultados".
  6. El sistema agrega la información de todos los catadores y muestra un mensaje al usuario administrador donde se indica la clasificación que se ha obtenido para la muestra.
  7. El sistema cambia de categoría la sesión a la de "Sesiones de cata finalizadas" y graba la información obtenida en la BD.

## **4. Diseño del sistema**

---

Sin duda, realizar de manera adecuada cada una de las actividades que conlleva la Ingeniería del Software es indispensable para la realización de un proyecto software de calidad. Por lo tanto, no se puede decir que ninguna de estas actividades sea más importante que otra. Sin embargo, si podemos decir que la actividad de diseño es la más delicada y la más laboriosa de llevar a cabo.

---

Es delicada porque si no se lleva a cabo correctamente se hace imposible el codificar, de manera correcta, en la actividad de implementación el modelo obtenido en el análisis del sistema, lo que puede repercutir en el desperdicio de todo el esfuerzo realizado durante las primeras actividades de la Ingeniería del Software.

Y es laboriosa porque las estrategias a seguir para conseguir que esta traducción entre modelo y código se lleve a cabo correctamente son muy diversas y bastante complejas. Se puede decir, por tanto, que el diseño del sistema es la actividad de la Ingeniería del Software en la que se identifican los objetivos finales del sistema y se plantean las diversas estrategias para alcanzarlos en la actividad de implementación.

Sin embargo, el sistema no se suele diseñar de una sola vez sino que hay que diferenciar entre el diseño y el diagrama de clases que se van a manejar y el diseño de la interfaz entre la aplicación y el usuario. Estas dos fases del diseño no se realizan de forma consecutiva una detrás de la otra sino que lo normal es realizarlas de manera concurrente y finalizarlas a la vez.

Las distintas fases de diseño que veremos a continuación son:

- **Diagrama de clases:** muestran la estructura del sistema.
- **Diseño de los datos:** estructura de los datos.
- **Diseño de la interfaz:** se define la apariencia visual de la aplicación.
- **Implementación:** el modelo obtenido se transforma en código.

#### 4.1. Diagrama de clases

Los diagramas de clases se utilizan para mostrar la estructura estática del sistema modelado. Pueden estar contenidos por clases, interfaces, paquetes, relaciones e incluso instancias, como objetos o enlaces.

Son una potente herramienta de diseño, ayudando a los desarrolladores a planificar y establecer la estructura del sistema y subsistema antes de escribir ningún código. Esto permite asegurar que el sistema está bien diseñado desde el principio.

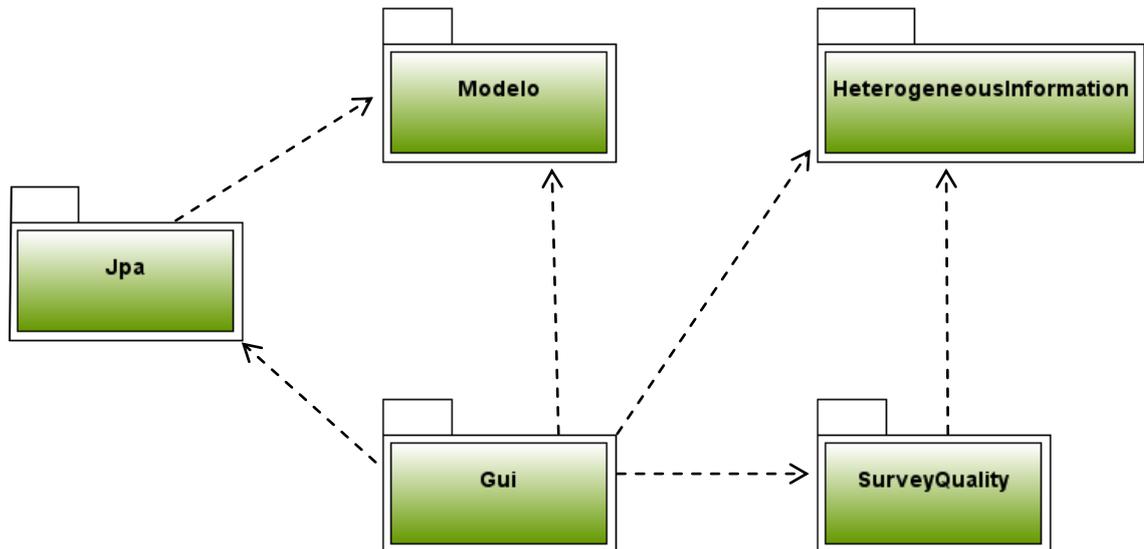
Son utilizados en la fase de diseño prácticamente en la totalidad de los sistemas que utilizan UML para su modelado.

Los diagramas de clases tienen los siguientes componentes:

- **Clases:** son los componentes fundamentales de los diagramas de clase. Su notación general es un rectángulo dividido en tres secciones, mostrando la primera el nombre de la clase, la siguiente los atributos y la última las operaciones.
- **Relaciones:** es una conexión semántica entre elementos. Existen cuatro tipos principales de relaciones:
  - *Generalización:* es una relación de especialización.
  - *Asociación:* es una relación estructural. Existen dos subtipos, la agregación y la composición.
  - *Realización:* es una relación contractual, en la cual una clase especifica un contrato que otra clase garantiza que cumplirá (por ejemplo, una interface).
  - *Dependencia:* Es una relación de uso.

Una vez descritos brevemente los componentes de un diagrama de clases, vamos a proceder a analizar los propios del sistema.

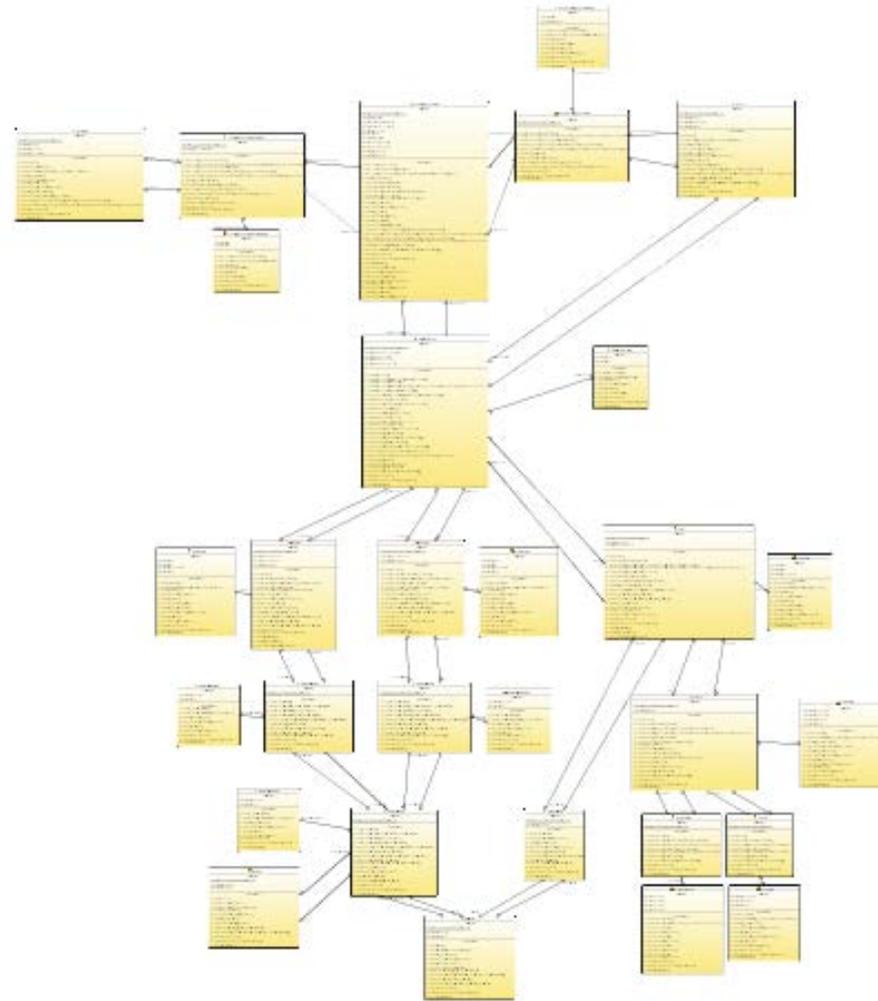
#### 4.1.1. Diagrama completo de clases



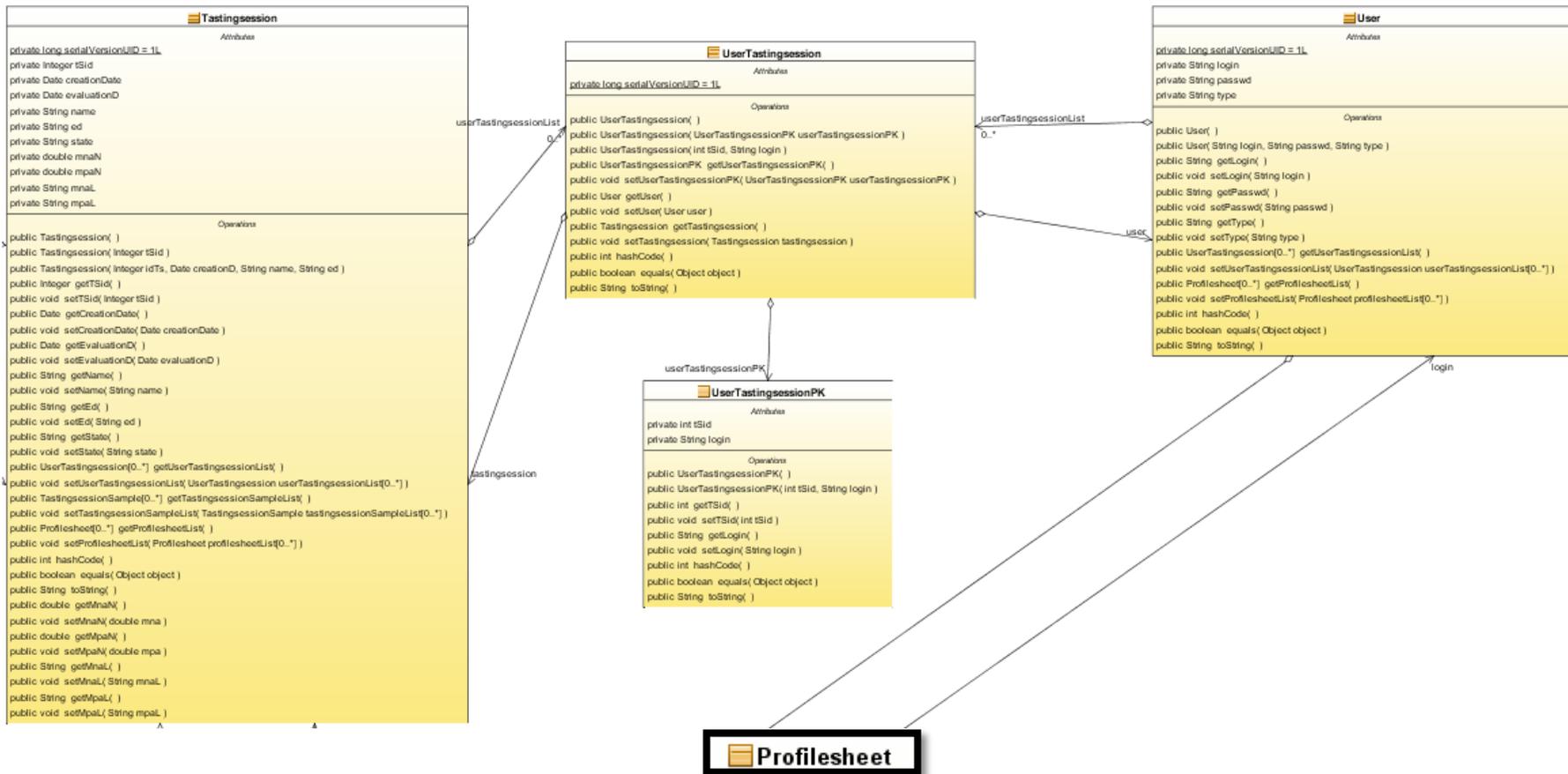
- *Paquete JPA* : es el paquete que contiene todos los controladores jpa que se encargan de la conexión con la BD y de trabajar con los registros.
- *Paquete Modelo*: representa la capa de modelo de la información de la aplicación que será gestionada por otros paquetes.
- *Paquete GUI*: contiene todas las clases relacionadas con la interfaz de la aplicación.
- *Paquete HeterogeneousInformation*: contiene todo lo necesario para el manejo de la información heterogénea (numérica, lingüística, difusa y 2-tupla).
- *Paquete SurveyQuality*: será utilizado para evaluar las diferentes hojas de perfil.

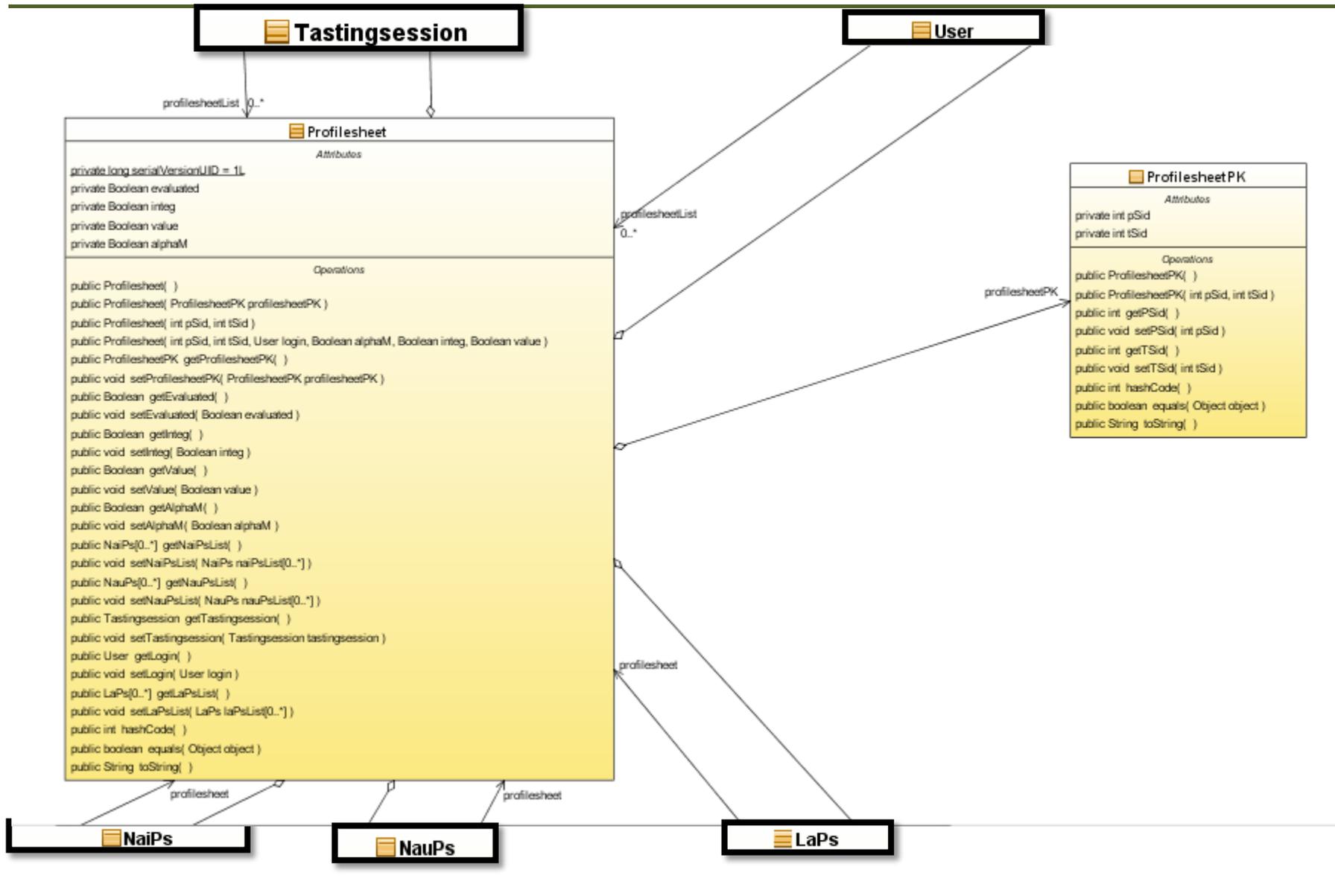
#### 4.1.2. Diagrama por paquete

4.1.2.1. Paquete Model

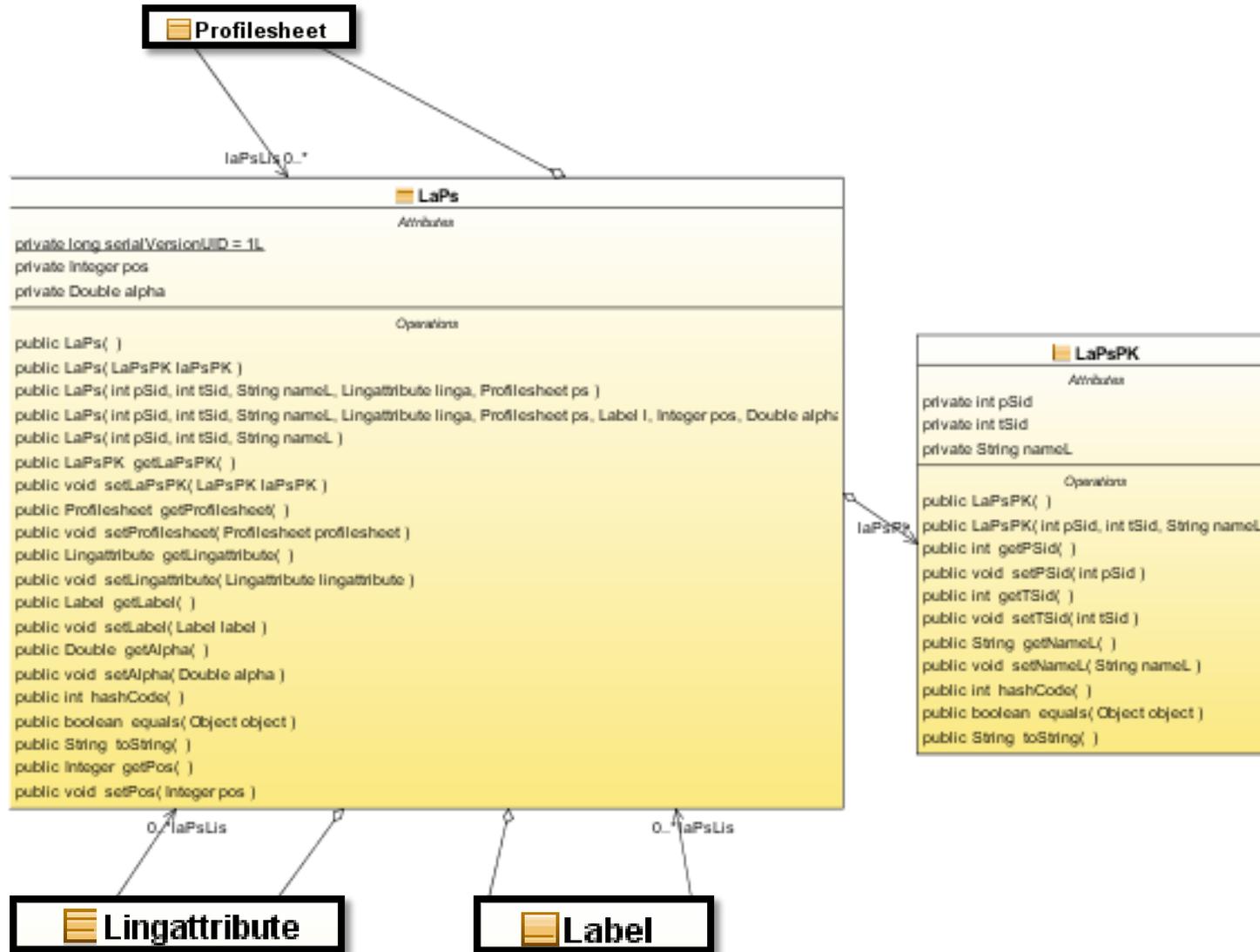


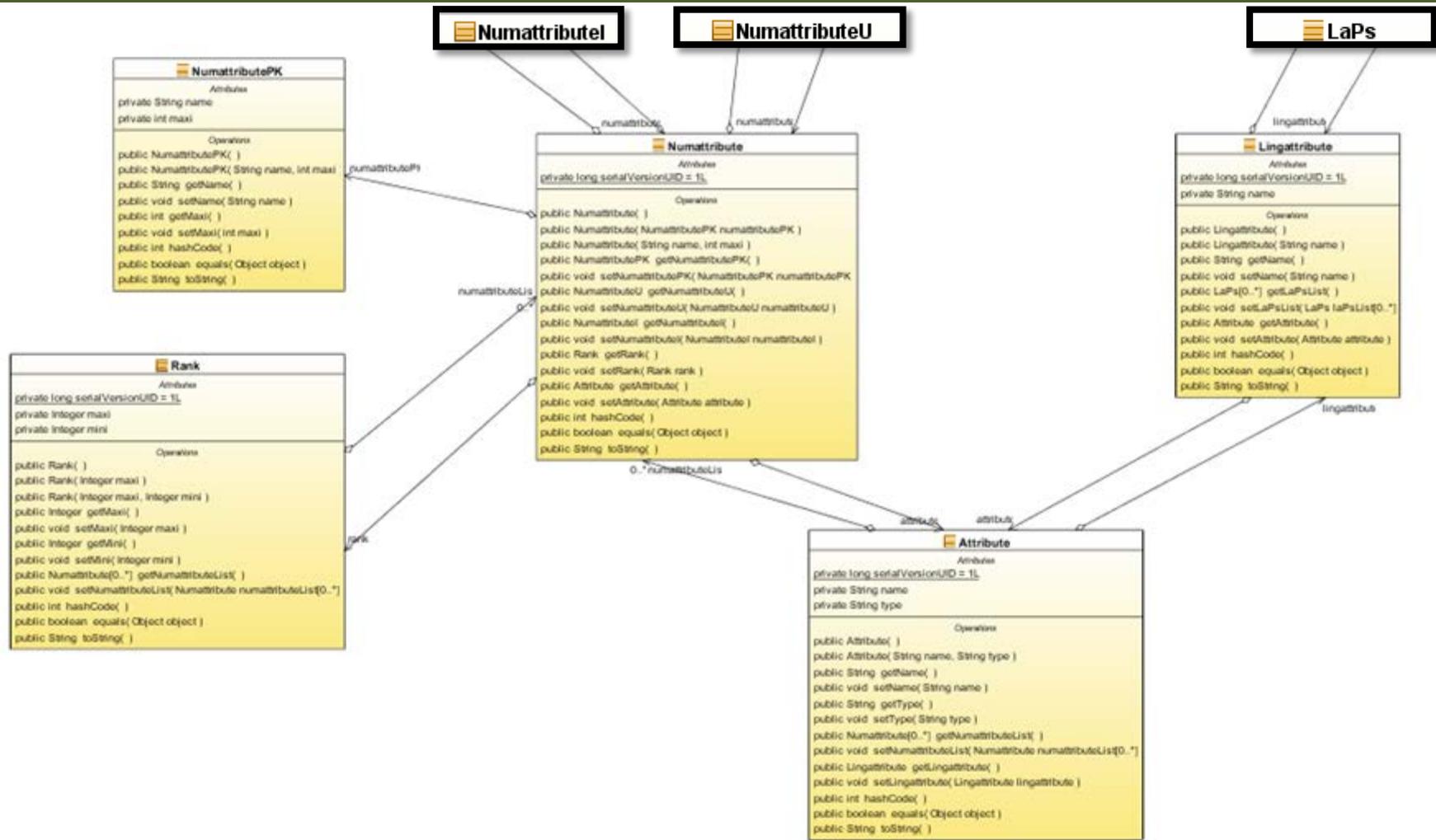


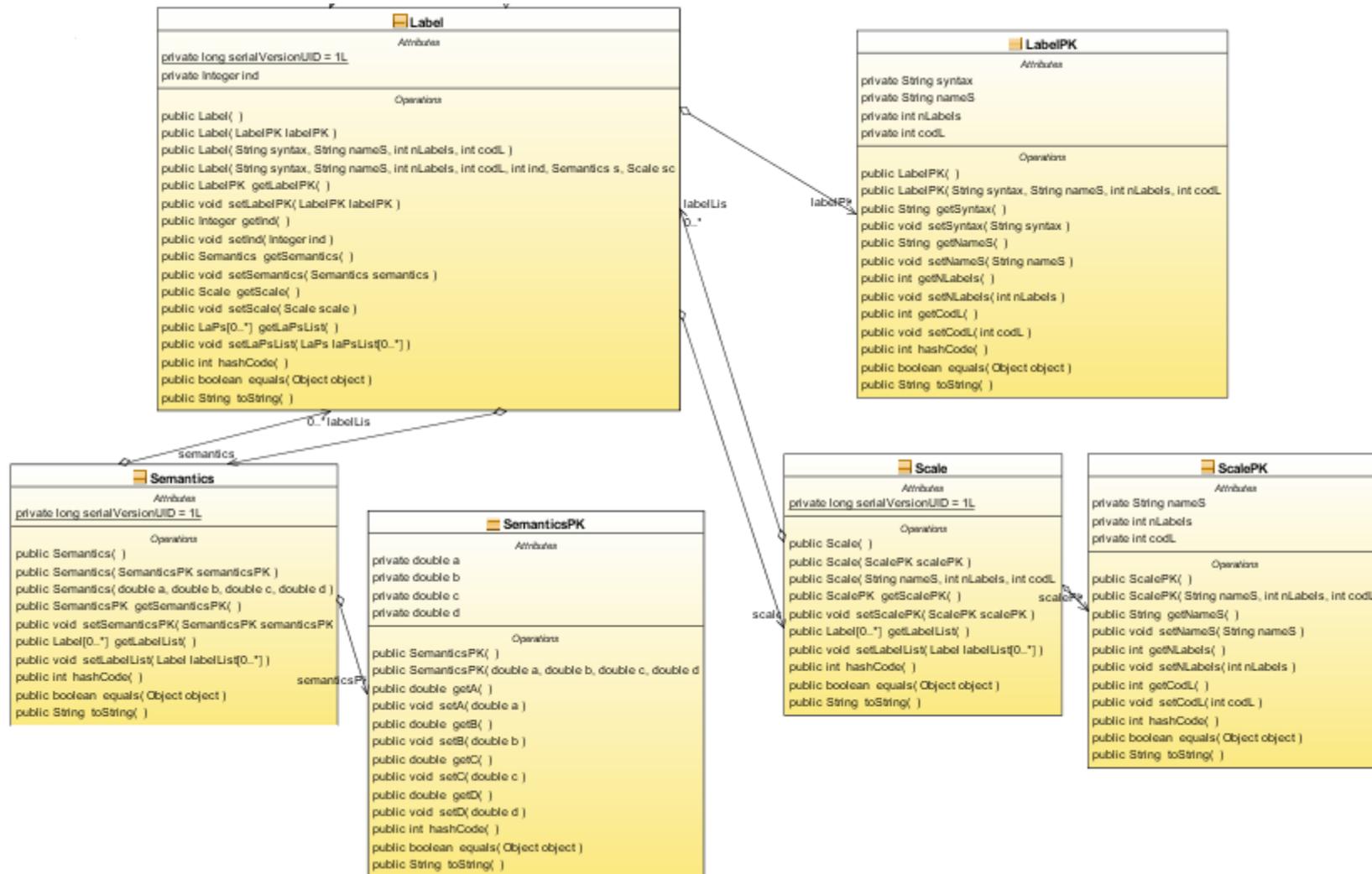












### 4.1.2.2. Paquete GUI

```

Model
  Atributos
  private LinkedList_datos = new LinkedList()
  private LinkedList_listener = new LinkedList()

  Operations
  public int getColumnCount( )
  public int getRowCount( )
  public String getColumnName( int indice_columna )
  public Object getValueAt( int indice_fila, int indice_columna )
  public void insertaFila( Info nuevo )
  public void eliminaFila( int fila )
  public void setValueAt( Object Evaluar, int indice_fila, int indice_columna )
  public void addTableModelListener( TableModelListener tml )
  public void removeTableModelListener( TableModelListener tml )
  private void avisar( TableModelEvent evento )
  public boolean isCellEditable( int row, int col )
    
```

```

AdminFrame
  Atributos
  private JFrame _jBegin
  package JMenuBar _mb
  package JMenu _m1
  package JMenu _m2
  package JMenuItem _j1
  package JMenuItem _j2
  package JMenuItem _j3
  package JMenuItem _j4
  package JMenuItem _j5
  package JMenuItem _j6
  package JMenuItem _j7
  package JMenuItem _j8
  private JLabel jLabel1
  private JLabel jLabel2
  private JLabel jLabel3
  private JLabel jLabel4
  private JLabel jLabel5
  private JLabel jLabel6
  private JLabel jLabel7
  private JLabel jLabel8
  private JLabel jLabel9
  private JPanel jPanel1
  private JButton jbAttribute
  private JButton jbEvaluateSample
  private JButton jbExpressionDomains
  private JButton jbSamples
  private JButton jbTastingSessions
  private JButton jbUsers

  Operations
  public AdminFrame( JFrame f )
  private void initComponents( )
  private void crearMenu( )
  private void jbUsersActionPerformed( ActionEvent evt )
  private void jbSamplesActionPerformed( ActionEvent evt )
  private void cerrarVentana( WindowEvent evt )
  private void JButtonAttributeActionPerformed( ActionEvent evt )
  private void jbExpressionDomainsActionPerformed( ActionEvent evt )
  private void jbEvaluateSampleActionPerformed( ActionEvent evt )
  private void jbTastingSessionsActionPerformed( ActionEvent evt )
  public Image getIconImage( )
  public void main( String args[0..*] )
  public void actionPerformed( ActionEvent e )
    
```

```

AlterUser
  Atributos
  private JButton jButtonOk
  private JPanel jPanel2
  private JLabel jLabel1
  private JLabel jLabel2
  private JLabel jLabel3
  private JPasswordField jfPassword
  private JPasswordField jfPassword2
  private JPasswordField jfPassword2

  Operations
  public AlterUser( User u )
  private void initComponents( )
  public Image getIconImage( )
  private void jButtonOkActionPerformed( ActionEvent evt )
  private void cerrarVentana( WindowEvent evt )
  public void main( String args[0..*] )
    
```

```

ComboEditor
  Atributos
  Operations
    
```

```

RenderLista
  Atributos
  Operations
    
```

```

EditUsers
  Atributos
  Operations
    
```

```

LinguisticTTAMSliderPanel
  Atributos
  Operations
    
```

```

NumericProfileSheet
  Atributos
  Operations
    
```

```

NumericVISliderPanel
  Atributos
  Operations
    
```

```

UnevaluatedProfileSheets
  Atributos
  Operations
    
```

```

ToggleSelectionModel
  Atributos
  private boolean _gestureStarted = false

  Operations
  public void setSelectionInterval( int index0, int index1 )
  public void setValuesAdjusting( boolean isAdjusting )
    
```

```

EditSamples
  Atributos
  Operations
    
```

```

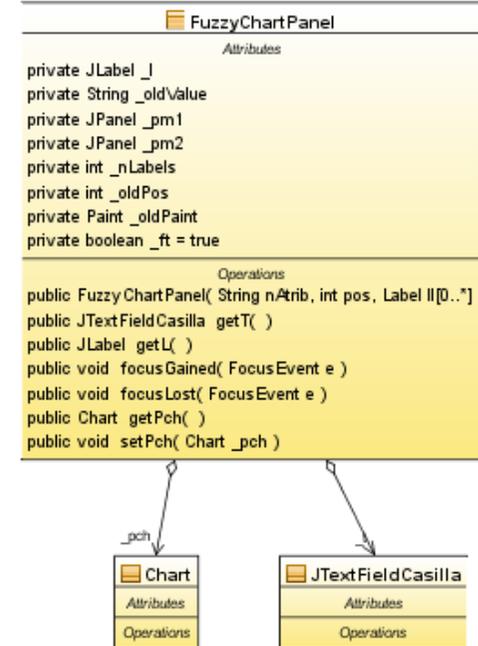
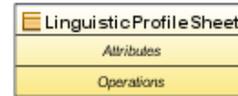
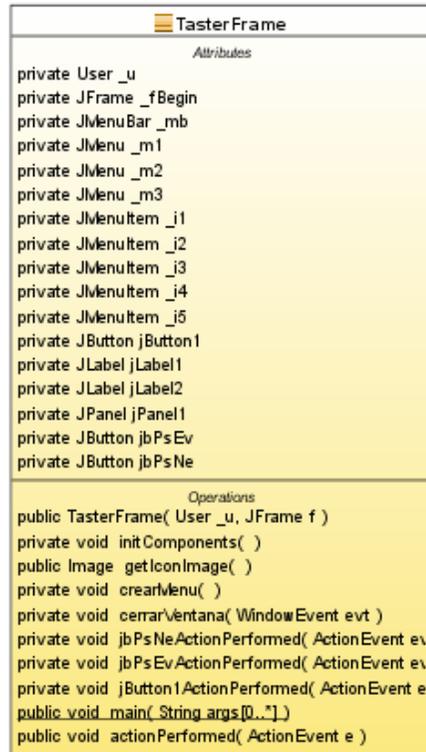
LinguisticSliderPanel
  Atributos
  Operations
    
```

```

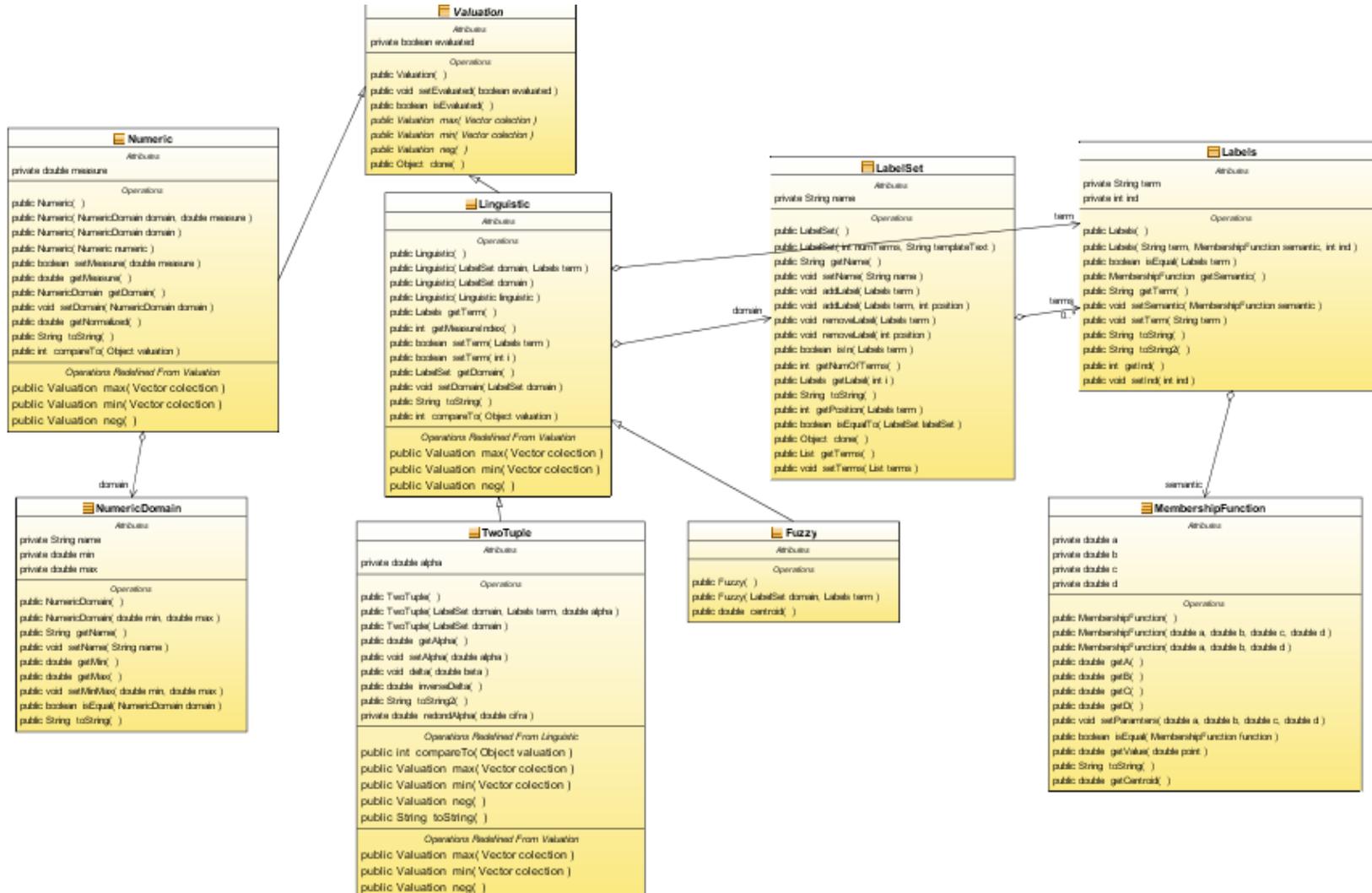
TopPanel
  Atributos
  Operations
    
```

```

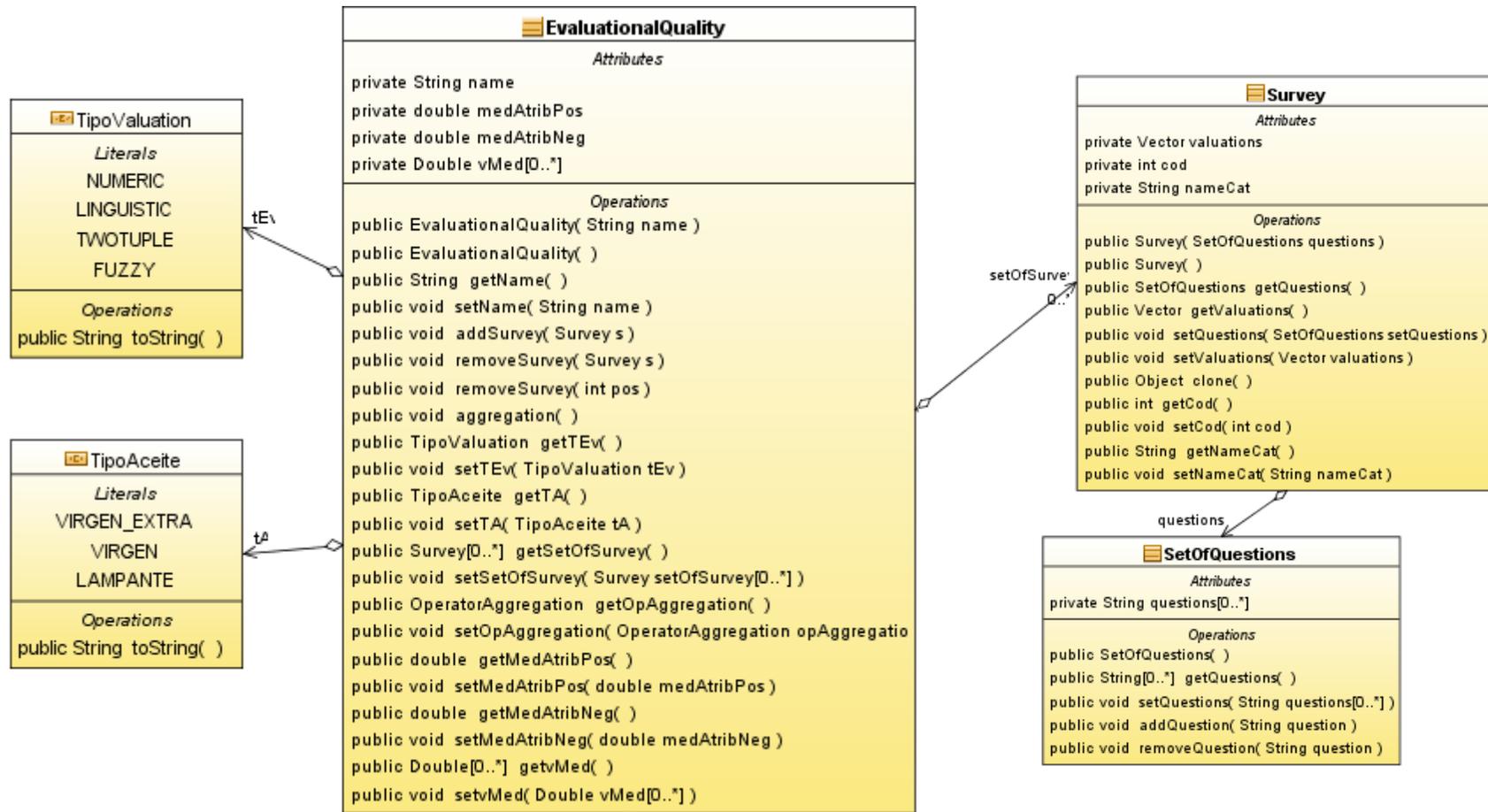
EditAttributes
  Atributos
  Operations
    
```

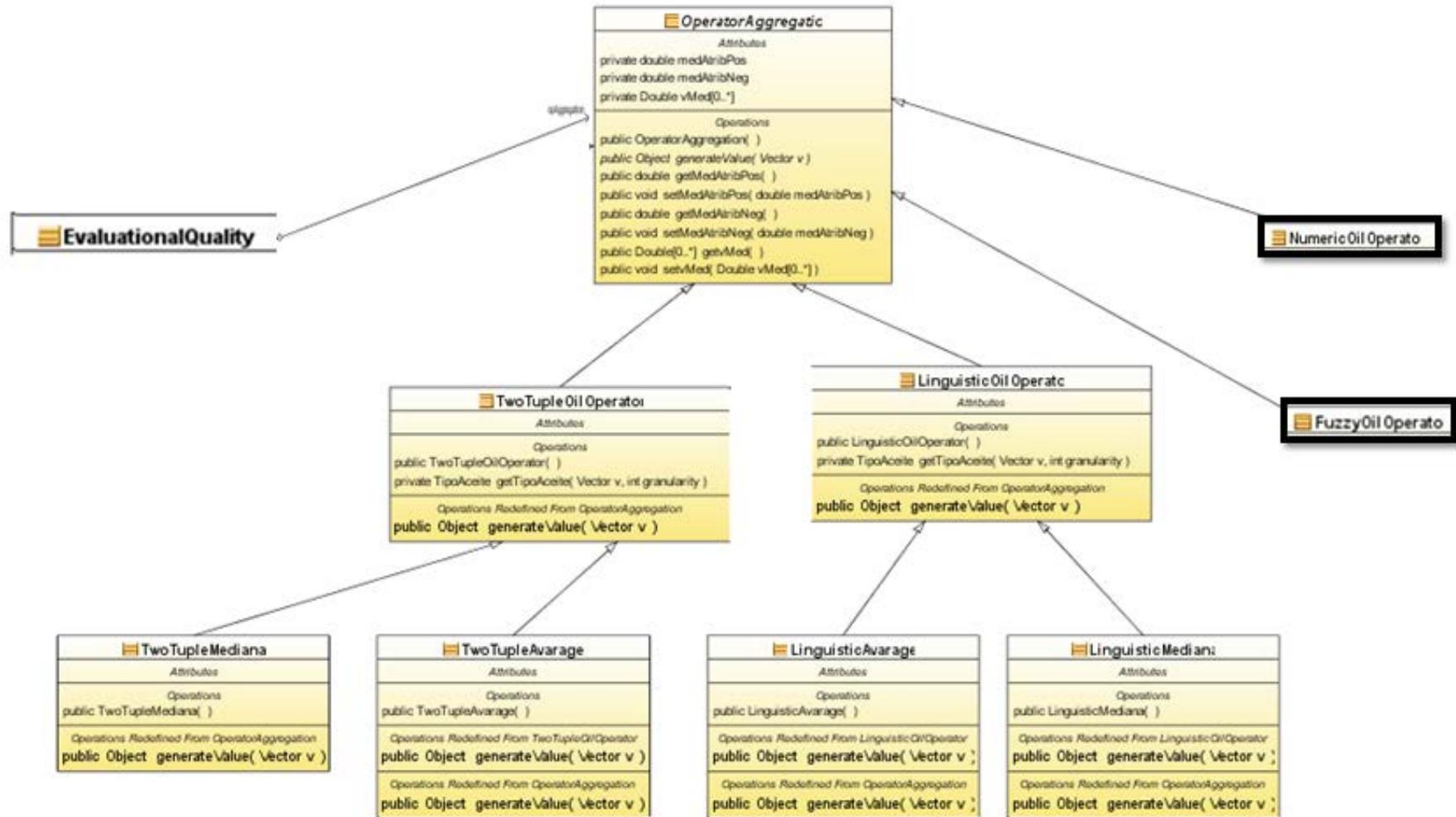


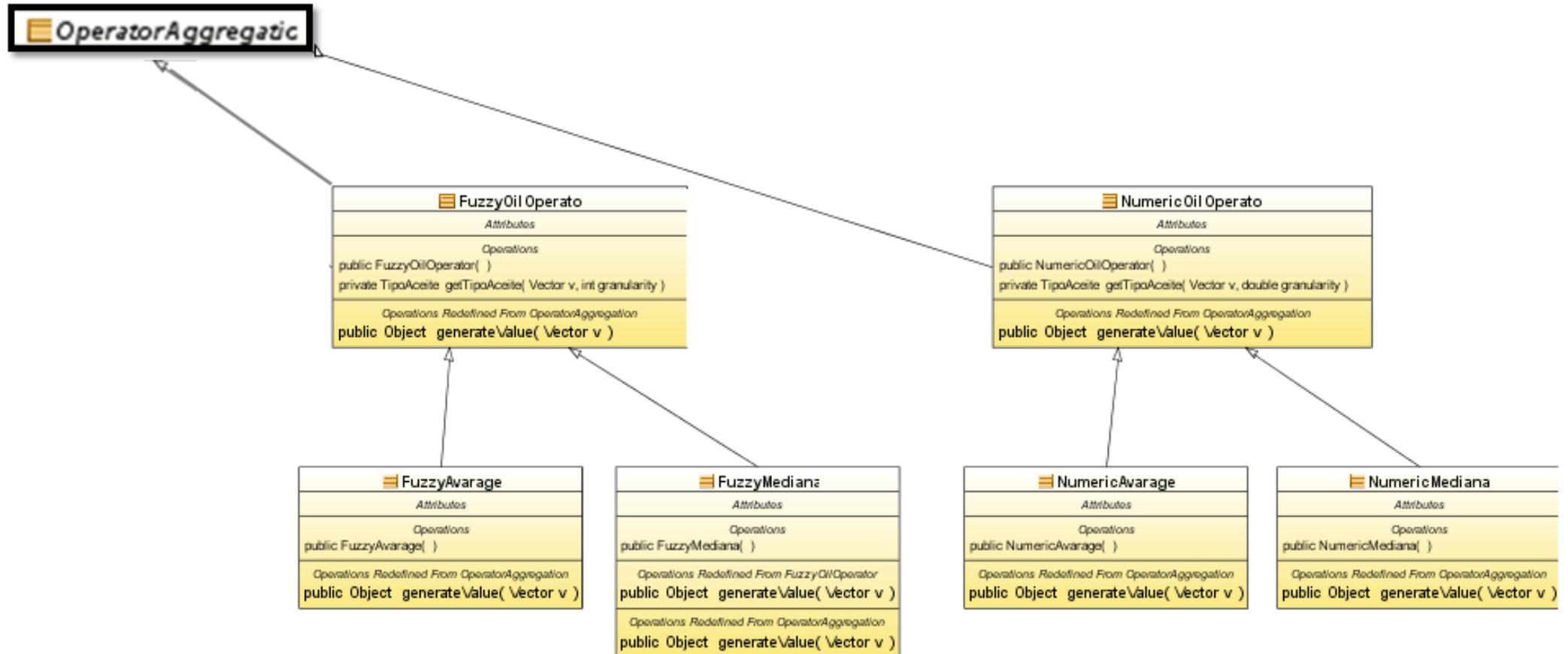
4.1.2.3. Paquete HeterogeneousInformation



4.1.2.4. Paquete SurveyQuality







## 4.2. Diseño de datos

### Modelo Entidad-Relación

El modelo Entidad-Relación (también conocido por sus iniciales: E-R) es una técnica de modelado de datos que utiliza diagramas Entidad-Relación. No es la única técnica de modelado pero si es la más extendida y utilizada.

Un diagrama entidad-relación está compuesto por tres tipos de elementos principales:

- **Entidades:** objetos (cosas, conceptos o personas) sobre los que se tiene información. Se representan mediante rectángulos etiquetados en su interior con un nombre. Una instancia es cualquier ejemplar concreto de una entidad.
- **Relaciones:** interdependencias entre una o más entidades. Se representan mediante rombos etiquetados en su interior con un verbo. Si la relación es entre una entidad consigo mismo se denomina reflexiva, si es entre dos entidades se denomina binaria, ternaria si es entre tres y múltiple si es entre más.
- **Atributos:** características propias de una entidad o relación. Se representan mediante elipses etiquetados en su interior con un nombre.

En los diagramas Entidad-Relación también hay que tener en cuenta otros aspectos como pueden ser:

- **Entidades débiles:** son aquellas que no se pueden identificar unívocamente solo con sus atributos, es decir, necesitan de estar relacionadas con otras entidades para existir. Se representan con dos rectángulos concéntricos de distinto tamaño con un nombre en el interior del más pequeño.
- **Cardinalidad de las relaciones:** existen tres tipos de cardinalidades de una relación según el número de instancias de cada entidad que involucren:
  - *Uno a uno:* una instancia de la entidad A se relaciona solamente con una instancia de la entidad B. (1:1).
  - *Uno a muchos:* cada instancia de la entidad A se relaciona con varias de la entidad B. (1:\*).
  - *Muchos a muchos:* cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B. (\*:\*)

- **Claves:** cada entidad de un diagrama entidad-relación debe tener una clave, debe estar formada por uno o más de sus atributos.

Una vez conocidos los elementos que forman parte de un diagrama Entidad-Relación podemos empezar a desarrollar el modelo Entidad-Relación. Los pasos a seguir son los siguientes:

1. Convertir el enunciado del problema (o, como es nuestro caso, los elementos del sistema software) en un Esquema Conceptual del mismo.
2. Convertir este Esquema Conceptual (o EC) en uno más refinado conocido como Esquema Conceptual Modificado (ECM).
3. Obtener las tablas de la base de datos a partir del Esquema Conceptual Modificado y normalizarlas.

### *Normalización en el modelo Entidad-Relación*

La normalización es un proceso consistente en imponer a las tablas ciertas restricciones mediante una serie de transformaciones consecutivas. Con ello se asegura que las tablas contengan los atributos necesarios y suficientes para describir la realidad de la entidad que representan, separando aquellos que pueden contener información cuya relevancia permite la creación de otra nueva tabla.

Para asegurar la normalización Codd estableció tres formas normales, las cuales hacen que una base de datos (si las cumple) esté normalizada. Estas formas normales son:

- **Primera forma Normal (FN1)**

Una tabla esta en FN1 si todos los atributos no clave, dependen funcionalmente de la clave, o lo que es lo mismo, no existen grupos repetitivos para un valor de clave.

- **Segunda forma Normal (FN2)**

Una tabla está en FN2 si está en FN1 y además todos los atributos que no pertenecen a la clave dependen funcionalmente de forma completa de ella. De esta definición se desprende que una tabla en FN1 y cuya clave está compuesta por un único atributo está en FN2.

- **Tercera forma Normal (FN3)**

Una tabla está en FN3 si está en FN2 y además no existen atributos no clave que dependan transitivamente de la clave.

#### **1. Esquema Conceptual**

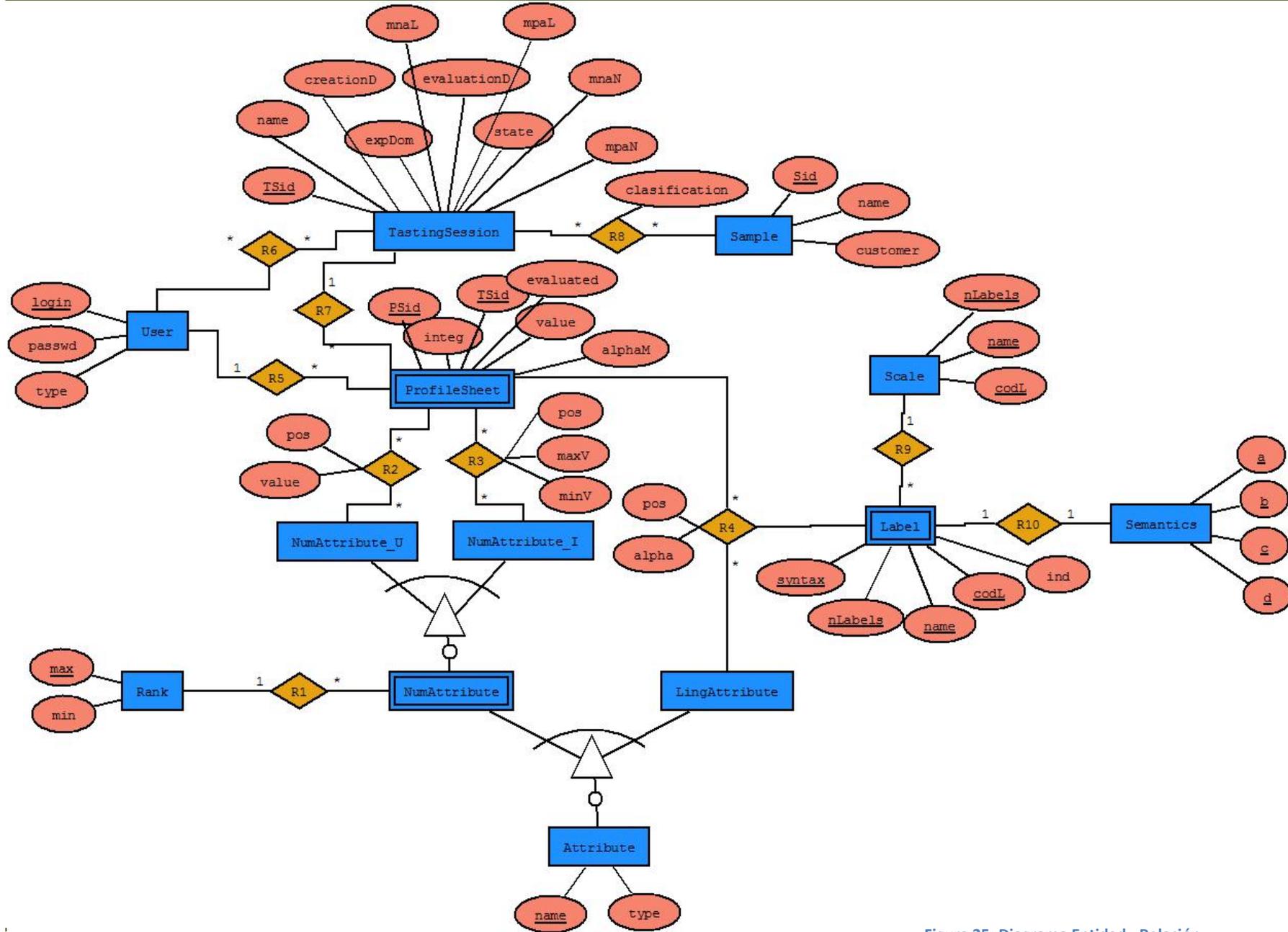


Figura 25: Diagrama Entidad - Relación

---

Necesitamos convertir nuestros elementos del sistema en entidades o relaciones. Es obvio que Usuario (User), Muestra (Sample), HojaPerfil (ProfileSheet), SesiónCata (TastingSession), Escala (Scale), Etiqueta (Label), Semántica (Semantic), Atributo (Attribute), AtributoNumérico (NumAttribute), AtributoLingüístico (LingAttribute), Rango (Rank), AtributoNuméricoÚnico (NumAttribute\_U) y AtributoNuméricoIntervalo (NumAttribute\_I) se convertirán en Entidades en nuestro Esquema Conceptual.

En cuanto a las relaciones:

- **R1:** Un *AtributoNumérico* tiene asignado un *Rango* y un *Rango* puede pertenecer a muchos *AtributoNumérico*.
- **R2:** Un *AtributoNuméricoÚnico* puede pertenecer a muchas *HojasPerfil* y en una *HojaPerfil* hay muchos *AtributoNuméricoÚnico*. Esta relación se transformará en una nueva entidad donde además se incluirá en el campo *Valor* que almacenará la valoración que recibirá el atributo.
- **R3:** Un *AtributoNuméricoIntervalo* puede pertenecer a muchas *HojasPerfil* y en una *HojaPerfil* hay muchos *AtributoNuméricoIntervalo*. Esta relación se transformará en una nueva entidad donde además se incluirán los campos *Max* y *Min* que almacenarán la valoración máxima y mínima que recibirá el atributo.
- **R4:** Un *AtributoLingüístico* puede pertenecer a muchas *HojasPerfil* y en una *HojaPerfil* hay muchos *AtributoLingüístico*. Esta relación se transformará en una nueva entidad donde además se incluirán los campos que forman la clave principal de la entidad *Etiqueta* que almacenarán la información de la etiqueta que ha sido asignada al *atributo*.
- **R5:** Un *Usuario* puede tener asignadas muchas *HojasPerfil* pero una *HojaPerfil* solo corresponde a un *Usuario*.
- **R6:** Un *Usuario* puede formar parte de muchas *SesiónCata* y una *SesiónCata* puede estar formada por muchos *Usuario*. Esta relación se transformará en una nueva entidad.
- **R7:** Una *SesiónCata* puede estar formada por muchas *HojaPerfil* pero una *HojaPerfil* solo puede pertenecer a una *SesiónCata*.
- **R8:** Una *Muestra* puede participar en muchas *SesiónCata* y en una *SesiónCata* pueden participar varias *muestras*.
- **R9:** Una *Etiqueta* pertenece a una *Escala* y una *Escala* puede estar formada por muchas *Etiqueta*.
- **R10:** Una *Etiqueta* está definida por una *Semántica* y una *Semántica* define a una *etiqueta*.

## 2. Esquema Conceptual Modificado

Para obtener el Esquema Conceptual Modificado debemos eliminar todas las entidades débiles, relaciones muchos a muchos y relaciones con atributos que haya en nuestro Esquema Conceptual. Por lo tanto, nuestro ECM queda como sigue:

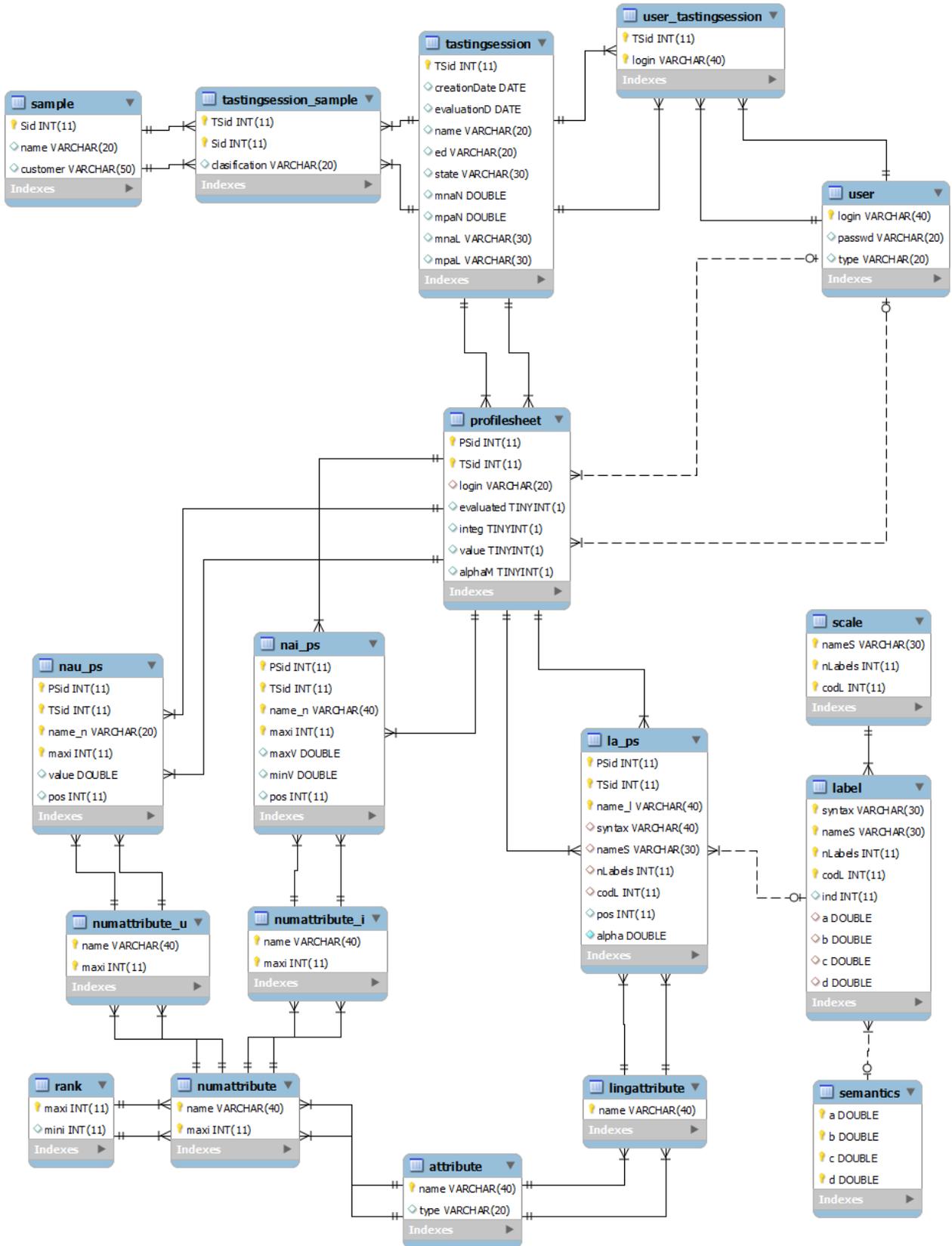


Figura 26: Esquema Entidad Relación Modificado

### 3. Tablas

A partir del ECM obtenido previamente podemos determinar las tablas de la base de datos, teniendo en cuenta que:

- Cada entidad del ECM se transforma en una tabla.
- Los atributos de una entidad se convierten en los campos de las tablas respectivas.

Por lo tanto, obtendremos las siguientes tablas:

#### ATTRIBUTE

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
Name	Varchar (40)	Nombre del atributo.	*	SI
Type	Varchar (20)	Tipo de atributo (positivo o negativo).		SI

Tabla 23: Campos de la tabla Attribute

#### NUMATTRIBUTE, NUMATTRIBUTE\_U, NUMATTRIBUTE\_I

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
name	Varchar (40)	Nombre del atributo.	*	SI
maxi	INT (11)	Identificador del rango al que pertenece, que se corresponde con el valor máximo.	*	SI

Tabla 24: Campos de las tablas NumAttribute, NumAttribute\_U, NumAttribute\_I

#### RANK

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
maxi	INT (11)	Valor mínimo del rango.	*	SI
mini	INT (11)	Valor máximo del rango.	*	SI

Tabla 25: Campos de la tabla Rank

#### LINGATTRIBUTE

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
name	Varchar (40)	Nombre del atributo.	*	SI

Tabla 26: Campos de la tabla LingAttribute

**SCALE**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
nameS	Varchar (30)	Nombre de la escala.	*	SI
nLabels	Int (11)	Número de etiquetas que forma la escala.	*	SI
codL	Int (11)	Nº Juego de etiquetas.	*	SI

Tabla 27: Campos de la tabla Scale

**SEMANTICS**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
a	Double	Primer punto de la etiqueta.	*	SI
b	Double	Segundo punto de la etiqueta.	*	SI
c	Double	Tercer punto de la etiqueta.	*	SI
d	Double	Cuarto punto de la etiqueta.	*	SI

Tabla 28: Campos de la tabla Semantics

**LABEL**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
syntax	Varchar (30)	Nombre de la etiqueta.	*	SI
nameS	Varchar (30)	Nombre de la escala.	*	SI
nLabels	Int (11)	Número de etiquetas que forma la escala.	*	SI
codL	Int (11)	Nº Juego de etiquetas.	*	SI
a	Double	Primer punto de la etiqueta.		
b	Double	Segundo punto de la etiqueta.		
c	Double	Tercer punto de la etiqueta.		
d	Double	Cuarto punto de la etiqueta.		
pos	Int (11)	Indica la posición que ocupa la etiqueta dentro de la escala.		SI

Tabla 29: Campos de la tabla Label

**PROFILESHEET**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
PSid	Int (11)	Identificador de la hoja de perfil.	*	SI
TSid	Int (11)	Identificador de la sesión de cata.	*	SI

<b>login</b>	Varchar (40)	Nombre del usuario catador.	SI
<b>evaluated</b>	Tinyint(1)	Indica si ha sido evaluada (SI/NO).	SI
<b>integ</b>	Tinyint(1)	Indicar si va a trabajar con valores enteros (SI/NO).	
<b>value</b>	Tinyint(1)	Indica si va a trabajar con valores únicos (SI/NO).	
<b>alphaM</b>	Tinyint(1)	Indica si va a trabajar con alfa modificado (SI/NO).	

Tabla 30: Campos de la tabla ProfileSheet

**USER**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
<b>Login</b>	Varchar (40)	Nombre del usuario.	*	SI
<b>Passwd</b>	Varchar (20)	Contraseña del usuario.		SI
<b>Type</b>	Varchar (20)	Tipo de usuario.		SI

Tabla 31: Campos de la tabla User

**SAMPLE**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
<b>Sid</b>	Int (11)	Identificador de la muestra.	*	SI
<b>Name</b>	Varchar (20)	Nombre de la muestra.		SI
<b>Customer</b>	Varchar (50)	Nombre del cliente dueño de la muestra.		SI

Tabla 32: Campos de la tabla Sample

**TASTINGSESSION**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
<b>TSid</b>	Int (11)	Identificador de la sesión de cata.	*	SI
<b>Name</b>	Varchar (20)	Nombre de la sesión de cata.		SI
<b>creationDate</b>	Date	Fecha en la que se crea la sesión de cata.		SI
<b>evaluationDate</b>	Date	Fecha en la que se cierra la sesión de cata.		
<b>Ed</b>	Varchar(20)	Dominio de expresión.		SI
<b>State</b>	Varchar(30)	Estado en el que se encuentra la sesión.		SI

<b>mnaN</b>	Double	Mediana de los atributos negativos numéricos.
<b>mpaN</b>	Double	Mediana del atributo positivo frutado numéricos.
<b>mnaL</b>	Varchar(30)	Mediana de los atributos negativos lingüísticos.
<b>mpaL</b>	Varchar(30)	Mediana del atributo positivo frutado lingüístico.

Tabla 33: Campos de la tabla TastingSession

## LA\_PS

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
<b>TSid</b>	Int (11)	Identificador de la sesión de cata.	*	SI
<b>PSid</b>	Int (11)	Identificador de la hoja de perfil.	*	SI
<b>nameL</b>	Varchar (40)	Nombre del atributo.	*	SI
<b>syntax</b>	Varchar (40)	Nombre de la etiqueta asignada.		
<b>nameS</b>	Varchar (30)	Nombre de la escala.		
<b>nLabels</b>	Int (11)	Número de etiquetas que forma la escala.		
<b>codL</b>	Int (11)	Nº Juego de etiquetas.		
<b>pos</b>	Int (11)	Indica la posición que ocupa el atributo en la hoja de perfil.		SI
<b>alpha</b>	Double	Valor de alfa.		SI

Tabla 34: Campos de la tabla La\_Ps

## NAU\_PS

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
<b>TSid</b>	Int(11)	Identificador de la sesión de cata.	*	SI
<b>PSid</b>	Int(11)	Identificador de la hoja de perfil.	*	SI
<b>name_n</b>	Varchar(20)	Nombre del atributo numérico.	*	SI
<b>maxi</b>	Int(11)	Valor máximo del rango del atributo.	*	SI
<b>value</b>	Double	Valor asignado al atributo.		
<b>Pos</b>	Int(11)	Posición que ocupa el atributo en la hoja de perfil.		SI

Tabla 35: Campos de la tabla Nau\_Ps

**NAI\_PS**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
<b>TSid</b>	Int(11)	Identificador de la sesión de cata.	*	SI
<b>PSid</b>	Int(11)	Identificador de la hoja de perfil.	*	SI
<b>name_n</b>	Varchar(20)	Nombre del atributo numérico.	*	SI
<b>Maxi</b>	Int(11)	Valor máximo del rango del atributo.	*	SI
<b>minV</b>	Double	Valor mínimo asignado atributo.		
<b>maxV</b>	Double	Valor máximo asignado atributo.		
<b>Pos</b>	Int(11)	Posición que ocupa el atributo en la hoja de perfil.		SI

Tabla 36: Campos de la tabla Nai\_Ps

**USERTASTINGSESSION**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
<b>TSid</b>	Int (11)	Identificador de la sesión de cata.	*	SI
<b>Login</b>	Varchar (40)	Nombre del catador.	*	SI

Tabla 37: Campos de la tabla User\_TastingSession

**TASTINGSESSION\_SAMPLE**

CAMPO	TIPO	DESCRIPCIÓN	CLAVE	REQUERIDO
<b>TSid</b>	Int (11)	Identificador de la sesión de cata.	*	SI
<b>Sid</b>	Int (11)	Identificador de la hoja de perfil.	*	SI
<b>Classification</b>	Varchar(20)	Clasificación obtenida por la muestra.		SI

Tabla 38: Campos de la tabla TastingSession\_Sample

### 4.3. Diseño de la interfaz

En esta fase del diseño del sistema software se define cual va a ser la apariencia visual de la aplicación, es decir, se define la interfaz visual entre el usuario y la aplicación. Sin duda, realizar un buen diseño de la interfaz resulta primordial ya que ésta debe presentarse atractiva al usuario de la aplicación pero a la vez le debe de resultar fácil de entender y trabajar sobre ella.

La ingeniería de la usabilidad aporta importantes beneficios referentes a los costes de desarrollo, la calidad del producto y la satisfacción del cliente. Define en esta fase los siguientes aspectos:

- ✓ Definir estilo.
- ✓ Metáforas.
- ✓ Pantallas.
- ✓ Caminos de navegación.
- ✓ Secuencias de diálogo.
- ✓ Mensajes de error.

Cada uno de estos aspectos lo veremos a continuación de forma detallada.

#### 4.3.1. Guía de estilo

Antes de ponerse a diseñar una interfaz de usuario, se debe definir el estilo de la misma.

Esto es de vital importancia cuando el diseño va a ser compartido entre varios diseñadores, ya que ayuda a mantener la coherencia interna de la interfaz.

Sin embargo, en contra de lo que pueda parecer en un principio, también es de mucha utilidad definir una guía de estilo cuando solo hay un diseñador encargado de la interfaz. Esto se debe a varias razones:

A veces es posible que mantener la coherencia y consistencia de una interfaz, si esta es muy grande o muy ambiciosa, sea algo complicado incluso si solo hay un diseñador si no tiene una base.

El diseñador primitivo puede, por las más diversas razones, abandonar el diseño y es de utilidad para sus sustitutos contar con una guía de estilo predefinida para no tener que empezar de cero otra vez. Lo mismo puede aplicarse si no es el diseñador original el que se encarga del mantenimiento o la actualización de la interfaz.

Quedando demostrada la utilidad del uso de guías de estilo podemos pasar a definir las reglas, normas y recomendaciones que contendrá la guía de estilo de nuestra interfaz:

#### *Fuentes:*

- Etiquetas y campos de los formularios: Tahoma 11px Plain.
- Encabezados de los formularios: Tahoma 11px Bold.
- Opciones principales del administrador y catador: Tahoma 14px Bold.

#### *Colores:*

El fondo de la aplicación tendrá un color gris RGB(240,240,240).

Los botones y encabezados de tabla utilizarán en la medida de lo posible los colores verde RGB(95,108,10) y azul RGB(47,69,147) por asociación de ideas con el color del aceite de oliva y de los vasos de cata.

#### *Botones:*

Los botones no llevarán bordes y el fondo del igual color al de la aplicación.

Se empleará una imagen que sirva de metáfora y solo irá acompañado de texto en aquellos casos que se crea necesario aclarar su función.

Todos los botones tendrán configurada la propiedad `toolTipTex` para que cuando el usuario sitúe el cursor encima aparezca una descripción de la funcionalidad de éste.

#### *Diseño de las ventanas:*

Todas llevarán un título que indique la tarea que se llevará a cabo en dicha ventana junto con el logo de la aplicación.



La estructura será similar para un mejor aprendizaje de la herramienta y siempre se utilizarán las mismas imágenes para los botones que tengan la misma funcionalidad.

### ***Metáforas***

Una metáfora es el empleo de un objeto con un significado o dentro de un contexto diferente al habitual. Al diseñar una interfaz gráfica, la utilización de metáforas resulta muy útil ya que permiten al usuario, por comparación con otro objeto o concepto, comprender de una manera más intuitiva las diversas tareas que la interfaz permite desarrollar.

Al igual que pasa en el ámbito de la literatura, para que una metáfora cumpla con su cometido, el desarrollador de la aplicación y el usuario final de ésta deben tener una base cultural similar. Es muy posible que el uso de un icono de manera metafórica sea entendido de una manera por el usuario occidental y de otra bien distinta por un usuario oriental. Hay que intentar, por lo tanto, que las metáforas empleadas sean lo más universales posibles para que así sean comprendidas a la perfección por la mayor parte del público potencial.

Las aplicaciones de escritorio de Windows suelen seguir la Guía de Estilo XP y utilizan una serie de metáforas con las que el usuario está plenamente familiarizado (por ejemplo, una lupa con un signo '+' en su interior establece que la función del icono es, inequívocamente, la de realizar un aumento de zoom). En el mundo de las aplicaciones web también existen una cantidad de metáforas de amplia difusión como puede ser, por ejemplo, el célebre carrito de la compra que emplean casi todos los comercios online.

Pero las metáforas no solo dependen del tipo de aplicación (escritorio o web) sino también del ámbito de la misma. Por ejemplo, el carrito de la compra es una metáfora conocida por todos pero si nuestra aplicación no va a vender nada al usuario no resulta conveniente utilizarla ya que puede confundir.

En nuestra aplicación, hemos usado algunas metáforas como:

	Indica que vamos a añadir un objeto (usuario, muestra, atributo,...) a nuestro sistema.
	Indica que vamos a eliminar un objeto de nuestro sistema.
	Indica que se ha llevado a cabo una determinada tarea de forma correcta.
	Indica que para que entrar en el sistema hay que presentar unos credenciales.
	Indica que vamos a trabajar con las muestras de aceite.
	Indica un listado de tareas que ya se han realizado, en este caso, listado de hojas de perfil que ya se han evaluado.
	Indica un listado de tareas que aún no se han realizado, en este caso, listado de hojas de perfil que no se han evaluado.
	Indica que se va a guardar la información en el sistema.
	<p>Todos los iconos que incluyen una lupa significa que podremos consultar o ver información:</p> <ul style="list-style-type: none"> <li>• Ver hojas de perfil.</li> <li>• Ver escalas.</li> <li>• Ver etiquetas.</li> <li>• Ver resultados.</li> </ul>
	
	
	
	

Tabla 39: Ejemplos de metáforas

### **4.3.2. Pantallas**

Como hemos visto en apartados anteriores, para diseñar una buena interfaz, es necesario un concienzudo trabajo de análisis y diseño.

El diseño gráfico de la interfaz de usuario juega un papel fundamental, ya que una buena distribución, ordenación, y codificación de colores de los elementos de la interfaz hace que ésta sea más clara y por tanto más fácil de usar para el usuario. Por el contrario una colocación incorrecta de los elementos puede hacer muy compleja la utilización de una aplicación por parte de un usuario.

Un principio general es que, los elementos que posee la interfaz tienen que estar colocados de tal modo que, a la hora de realizar una tarea, el usuario deba recorrer la interfaz en la misma dirección que lee un texto, es decir, de izquierda a derecha y de arriba abajo. En países orientales no se realizaría de la misma forma.

Una interfaz de usuario visualmente atractiva hace que sea percibida por el usuario como una mejor interfaz, mientras que una interfaz compleja con sus elementos desordenados, provoca en el usuario insatisfacción y desorientación. No hay que olvidar que una buena interfaz es aquella que encuentra el equilibrio entre la funcionalidad y la estética.

En nuestra aplicación las ventanas los botones relacionados como añadir, eliminar y consultar que encuentran próximos. En la mayoría de las ventanas se muestra de izquierda a derecha, primero la información que ya está en el sistema y luego se da la posibilidad de añadir, eliminar o consultar.

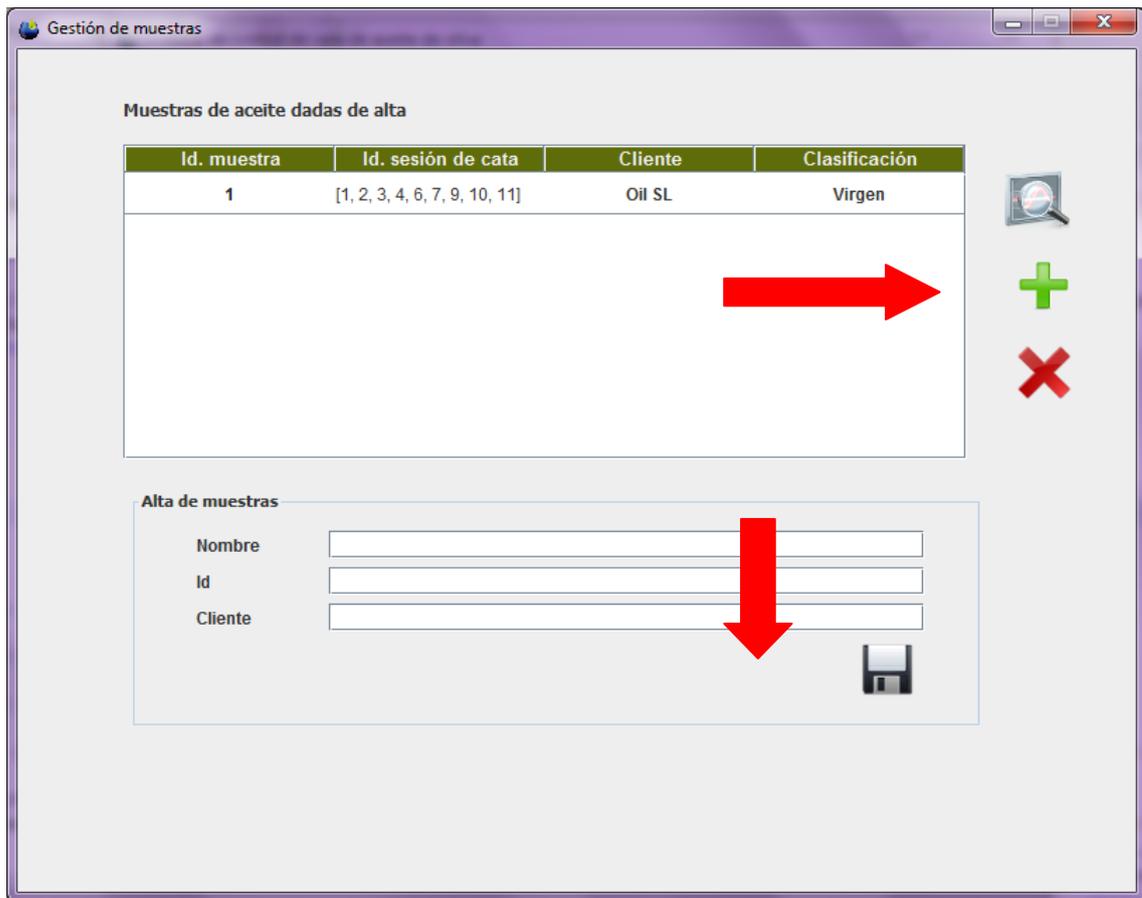


Figura 27: Ejemplo de pantalla I

El botón guardar presente en muchas de las ventanas siempre irá colocada al final, una vez que se han rellenado previamente todos los campos necesarios.

El orden de los botones no es aleatorio. En las diferentes ventanas se han ido colocando por orden decreciente de frecuencia de uso. Por ejemplo, en la ventana de Gestionar Muestras la operación que probablemente se realice más veces sea la de consulta, seguida de añadir y por último eliminar.

En la venta de Gestionar Sesiones de Cata, una de las que más información tiene a configurar por el usuario, se han dispuesto los diferentes elementos de arriba a abajo siguiendo el orden lógico que se seguiría para la configuración de una sesión de cata y una vez finalizado éste, el botón de guardar.

Gestión de las sesiones de cata

Sesiones de cata 1

Alta de una sesión de cata

Nombre

Id

Muestra 1

Agregar catadores

Agregar atributos

Dominio de expresión Numérico

Rango: 1

Valor único
  Intervalo de valores

Valores enteros
  Valores reales

	lun	mar	mié	jue	vie	sáb	dom
22						1	2
23	3	4	5	6	7	8	9
24	10	11	12	13	14	15	16
25	17	18	19	20	21	22	23
26	24	25	26	27	28	29	30

Figura 28: Ejemplo de pantalla II

#### 4.3.3. Caminos de navegación

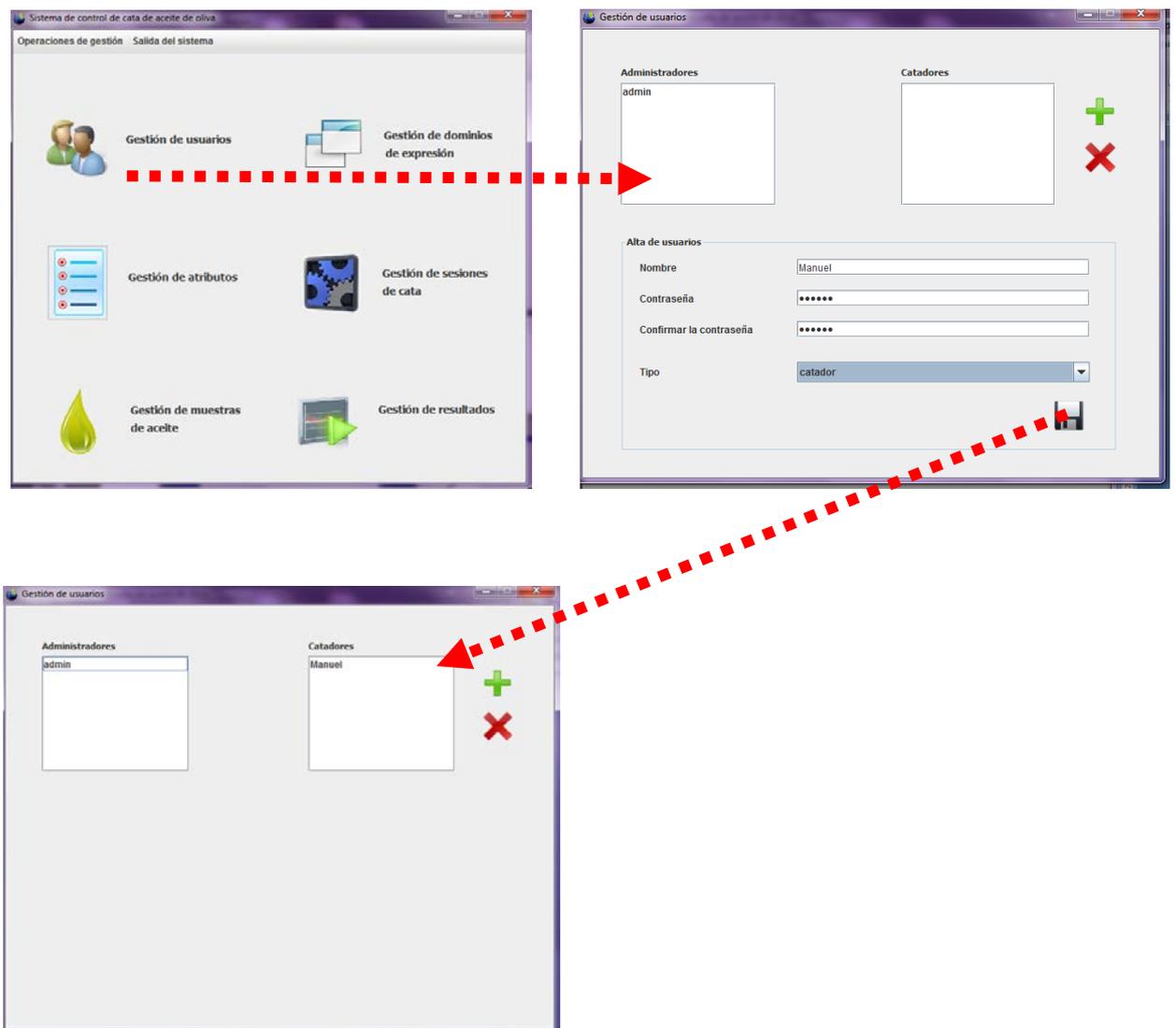
Hasta este momento tenemos un diseño visual de la interfaz estática, es decir, cada pantalla diseñada individualmente, pero no tenemos una idea de si en el conjunto de la interacción, la acción va a transcurrir de forma comprensible para el usuario. Para ello vamos a diseñar la interfaz en movimiento y comprobar que es usable.

Para estudiar los caminos de interacción se empleará una herramienta llamada storyboard, que consiste en mostrar, a modo secuencia, las distintas pantallas por las que se va pasando al realizar el usuario una determinada acción sobre la aplicación. Mediante flechas se ayuda a entender que es lo que ha desencadenado el paso de una pantalla a otra.

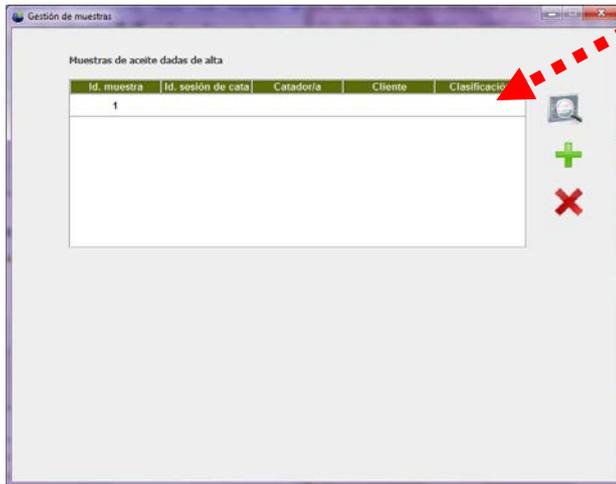
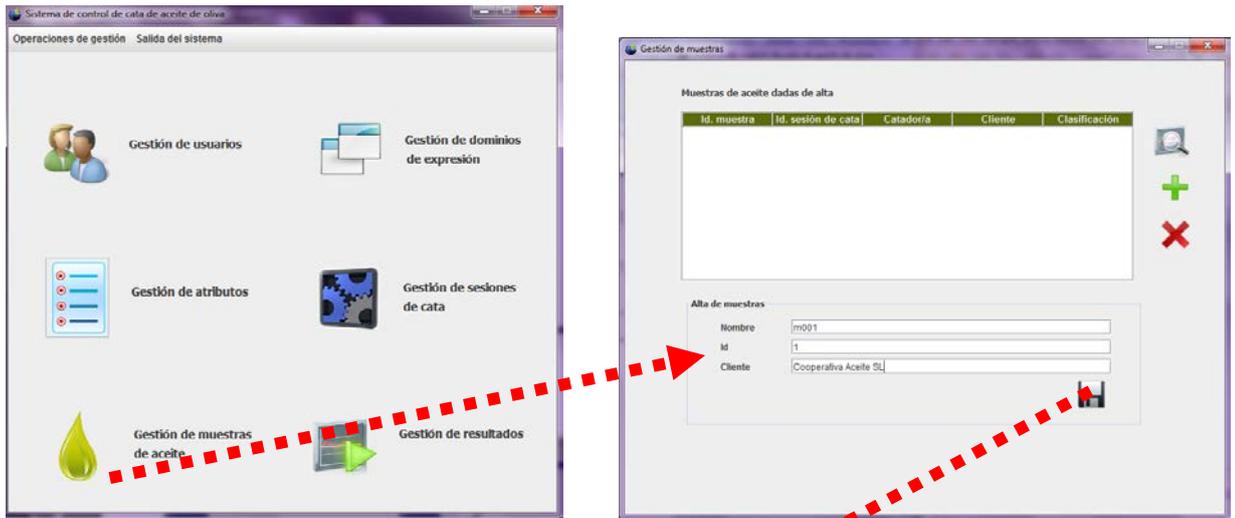
Los storyboards también están muy ligados a los escenarios vistos anteriormente. El storyboard sirve de prototipo para ser evaluado por el usuario y poder introducir correcciones en fases iniciales, ya que cuanto más tiempo se tarde en validar una interfaz, más coste de tiempo y trabajo nos ocasionará.

No se han desarrollado storyboards para todas las acciones de nuestro sistema por lo que los que se muestran a continuación son los que se han considerado más importantes:

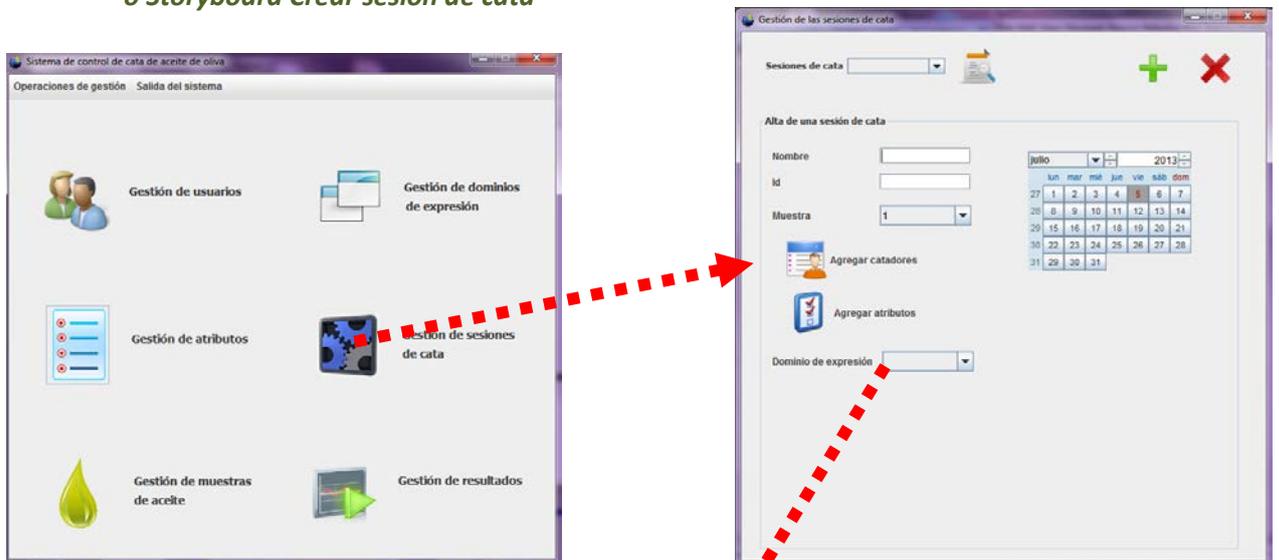
**o Storyboard Crear usuario**

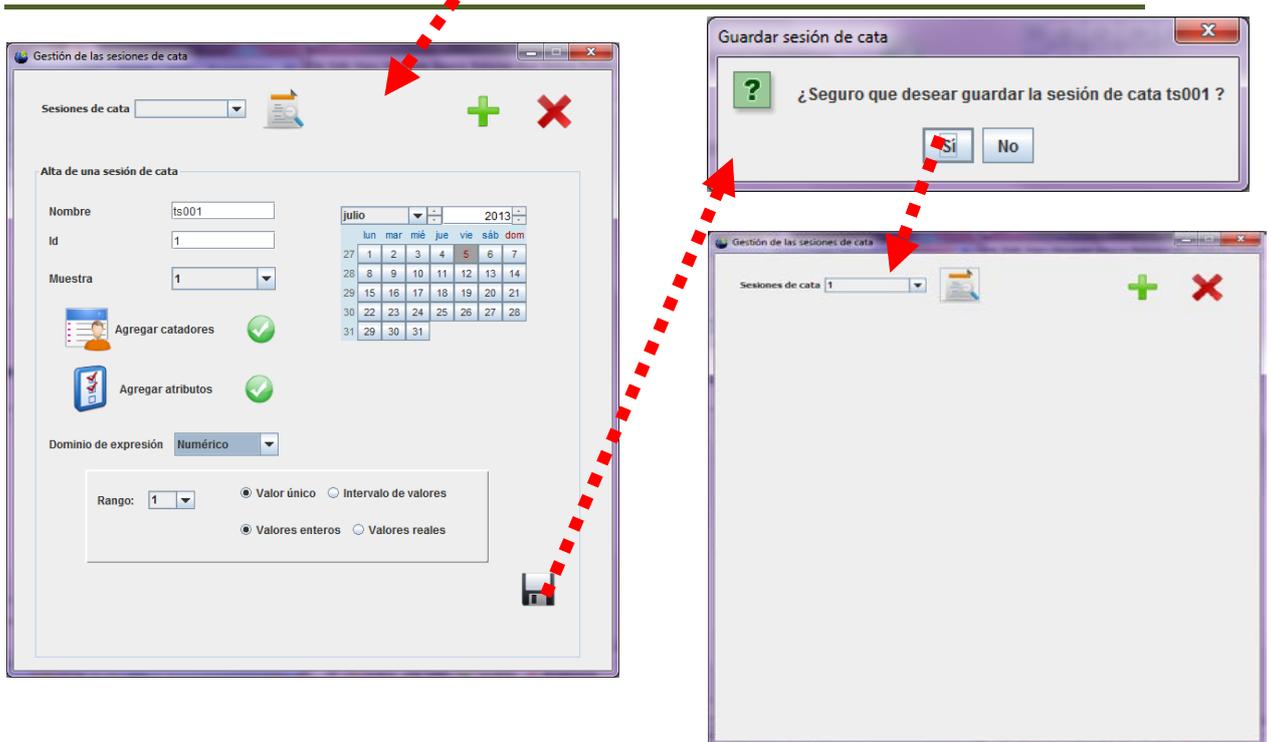


*o Storyboard Dar de alta una muestra de aceite de oliva*

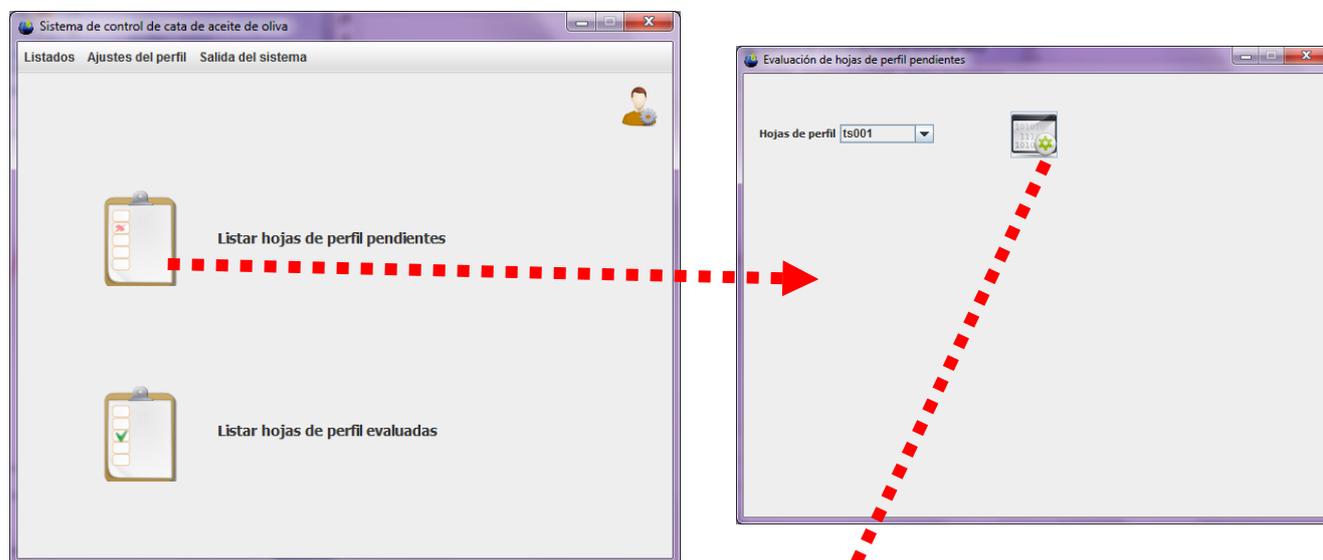


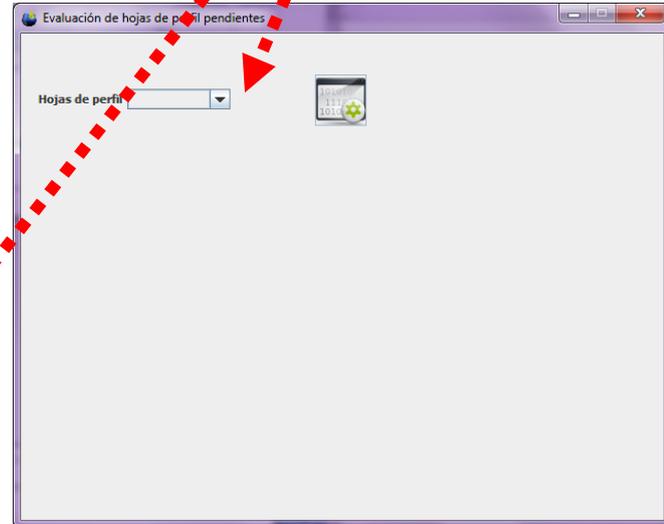
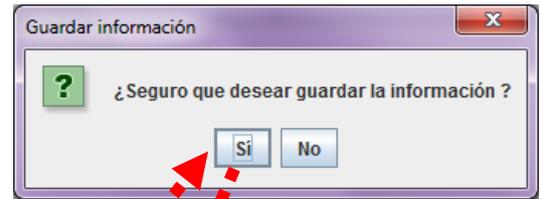
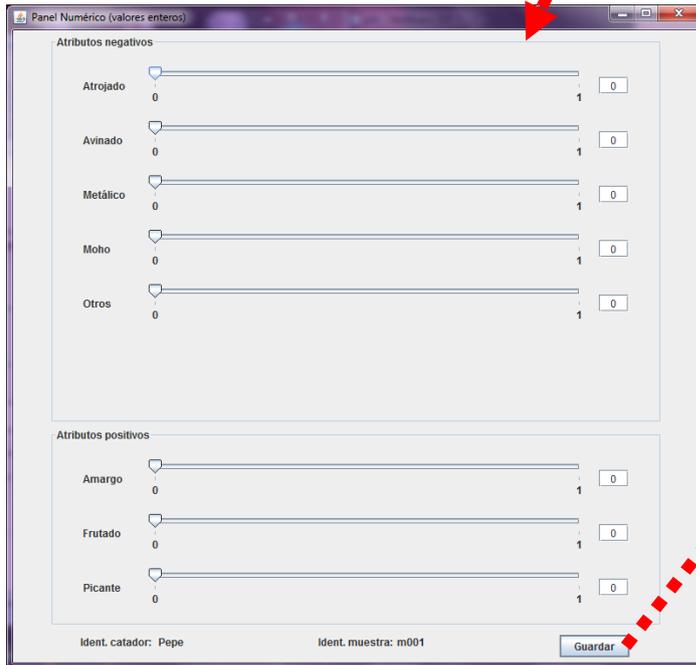
*o Storyboard Crear sesión de cata*



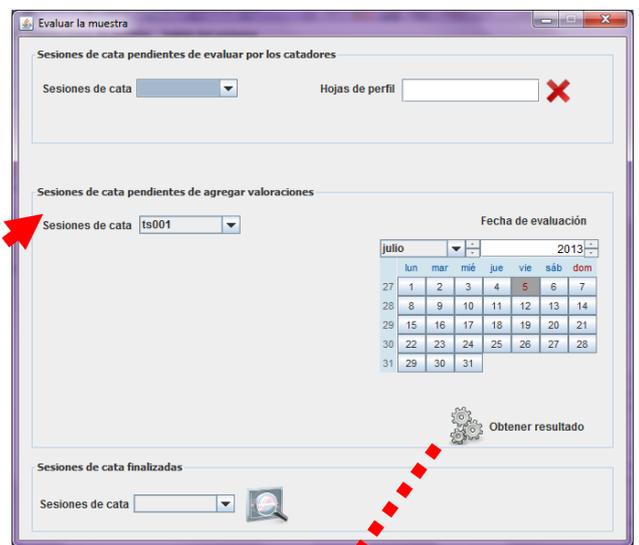
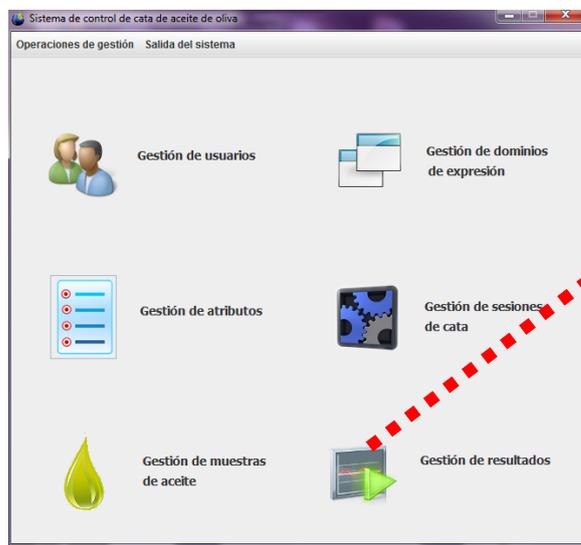


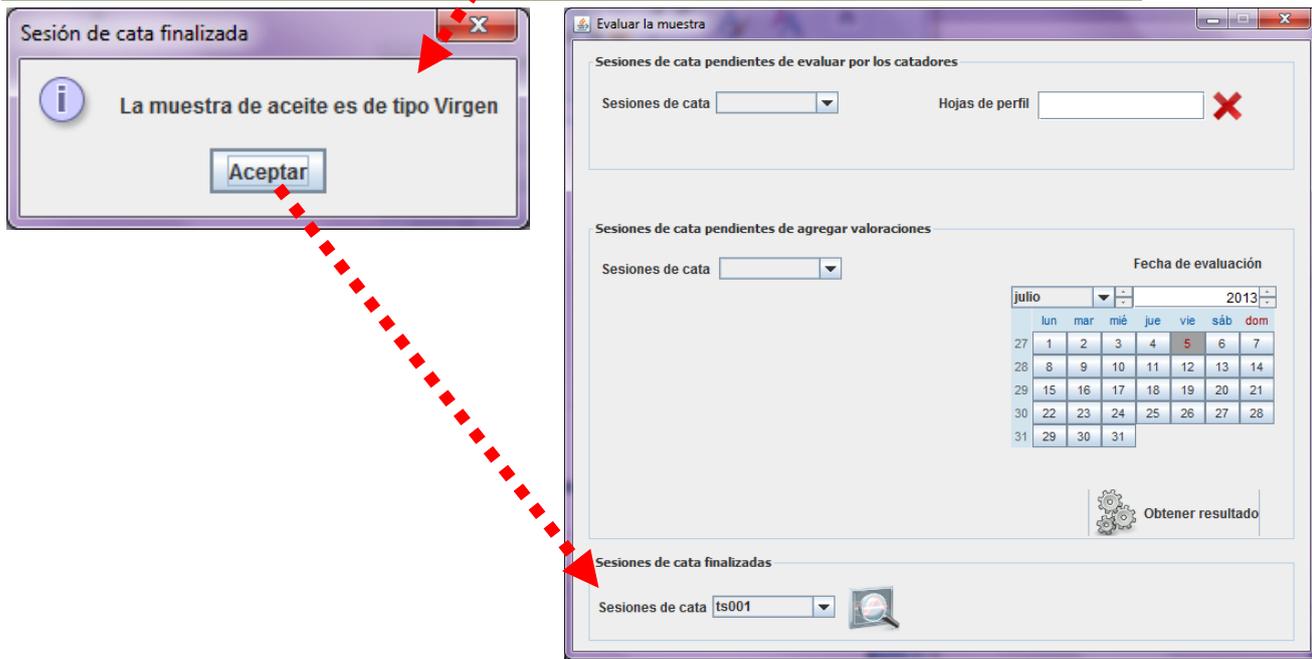
*o Storyboard Evaluar hoja de perfil*



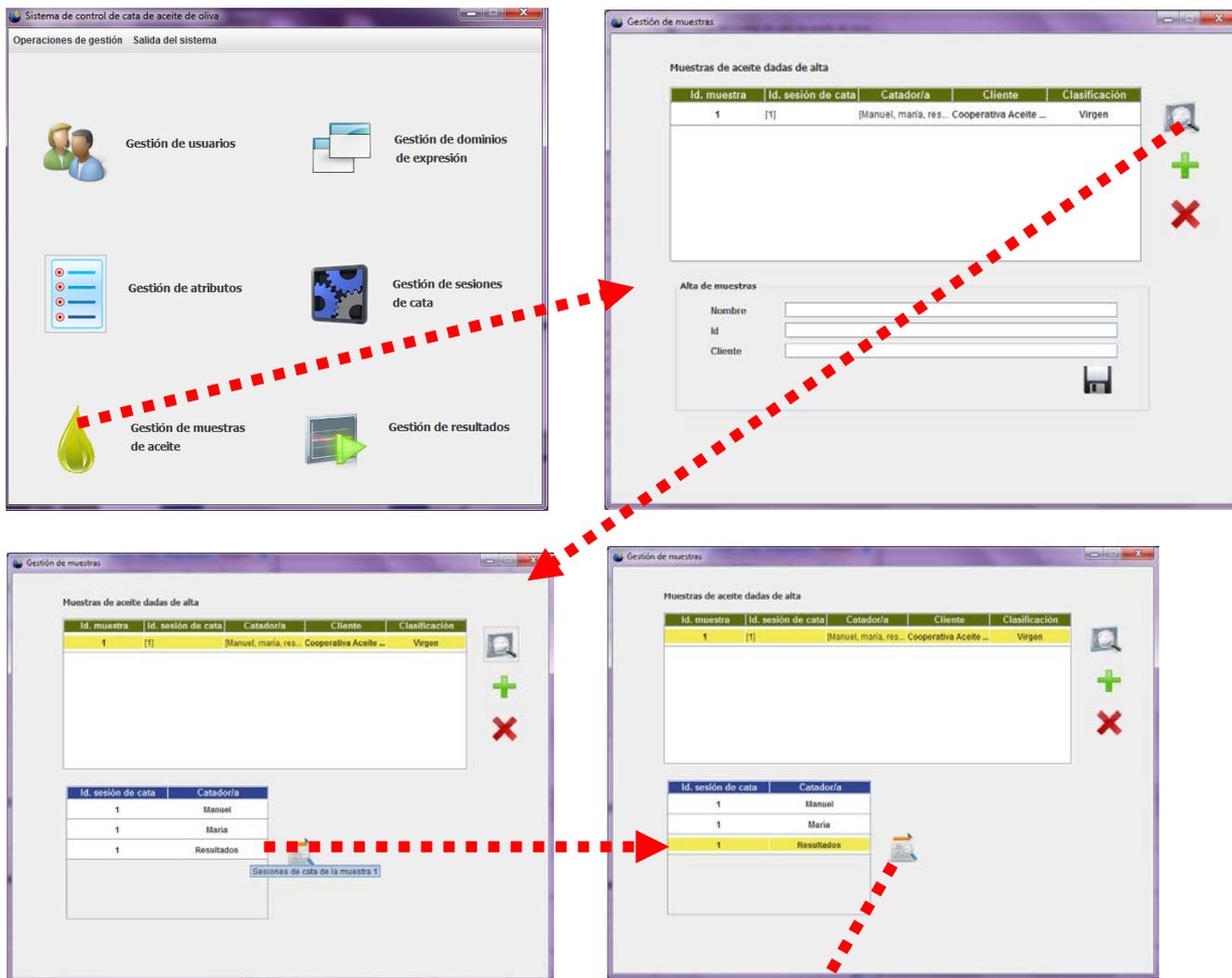


*o Storyboard Obtener resultados de la sesión de cata*





*o Storyboard Consultar información de una sesión de cata*



Panel Numérico (valores enteros)

Atributos negativos

Atributo	Valor
Atrajado	1
Avinado	2
Metálico	1
Moho	1
Otros	0.0
Rancio	0.0

Atributos positivos

Atributo	Valor
Amargo	0.0
Frutado	4
Picante	0.0

Clasificación: Virgen      Ident. muestra: m001

#### 4.3.4. Mensajes de error

Una vez diseñadas las pantallas, los caminos de navegación y las secuencias de diálogo, se tiene una idea muy clara de cómo se va a desarrollar físicamente la interacción entre usuario y ordenador, y por tanto de las posibles situaciones de error que pueden darse en esa interacción. Por ello ahora es el momento de diseñar los mensajes de error.

Los mensajes de error son el medio por el que el sistema comunica al usuario que se ha producido un error en la interacción. Los errores se producen por falta de conocimiento sobre la interfaz, porque se no se ha entendido correctamente el estado del sistema o bien inadvertidamente (por ejemplo se pulsa un botón cuando se intentaba pulsar otro). Ello lleva al usuario a sentirse confuso y aumenta su ansiedad, sobre todo en usuarios noveles en los que su falta de conocimiento y confianza en el uso de la interfaz lleva a amplificar el stress lo que puede llevarles a una experiencia de uso de la interfaz frustrante.

Los mensajes de error son muy importantes de cara a la usabilidad, ya que bien diseñados, permiten aumentar la confianza del usuario en el uso de la interfaz y que pueda seguir con su tarea. Por ello hay que diseñarlos con cuidado. A la hora de diseñar mensajes de error se deberían seguir las siguientes reglas:

- ✓ Ser breves: el usuario es una persona ocupada tratando de llevar a cabo una tarea. Si se le presenta un mensaje de error muy largo, lo normal es que no lo lea ya que no tendrá tiempo para hacerlo. Por ello hay que diseñar mensajes de error claros, pero breves.
- ✓ Ser específico: los mensajes demasiado generales no son adecuados ya que dicen al usuario que algo ha ido mal, pero no le indican claramente qué. Por lo tanto el usuario no sabrá qué hacer para impedir que se produzca el error y su sensación de frustración aumentará Usar un tono positivo y guía constructiva: nunca recriminar al usuario lo que ha hecho mal, en su lugar siempre que sea posible indicarle que debe hacer para eliminar el error.
- ✓ Usar un formato físico apropiado: la mayoría de usuarios encuentran más fácil leer un mensaje donde se mezclan letras mayúsculas y minúsculas de la forma habitual, por tanto este formato es siempre preferible. Los mensajes escritos únicamente en mayúsculas deberían reservarse para avisos breves y graves. Si el mensaje de error debe contener un código numérico, este debería figurar al final del mensaje y entre paréntesis.

Una vez expuestas las reglas a seguir se exponen algunos de los mensajes de error utilizados en nuestro sistema:

- **Pantalla de entrada al sistema:** existe la posibilidad de que un usuario se identifique de forma incorrecta. Si esto ocurre los mensajes de error que se pueden mostrar son los siguientes:

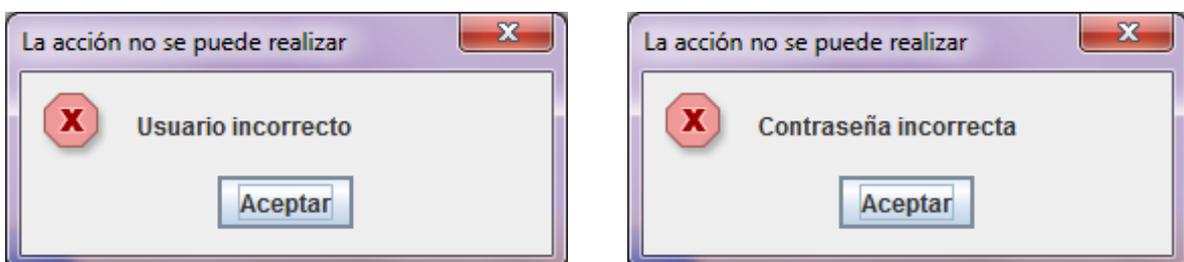


Figura 29: Posibles errores en la entrada al sistema

- **Pantalla de Gestionar usuarios:**

- Existe la posibilidad de que se cree un usuario de forma incorrecta y se pueden encontrar los siguientes mensajes:

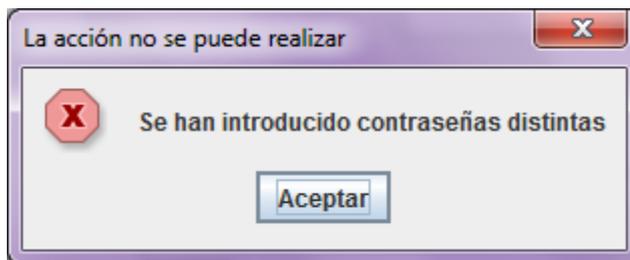
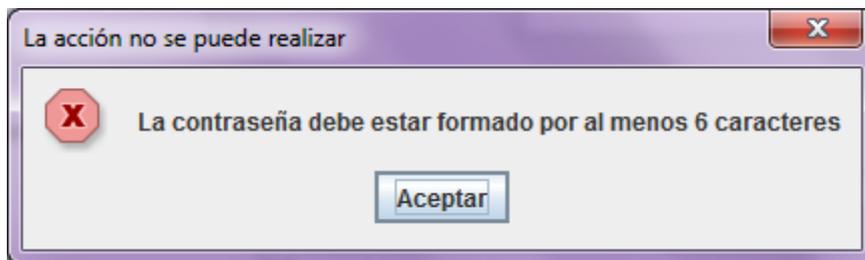
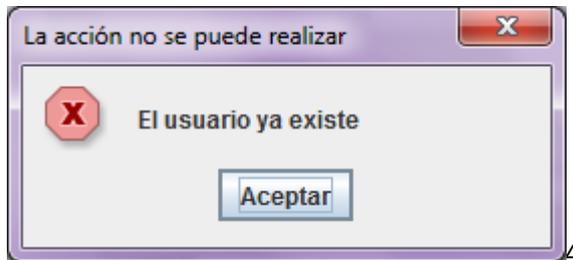
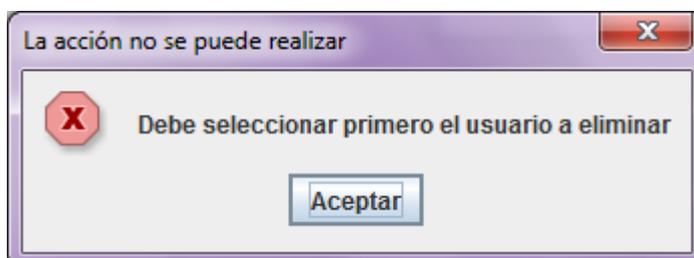


Figura 30: Posibles errores durante el alta de usuarios

- En el caso de querer eliminar un usuario los errores que nos podemos encontrar son:



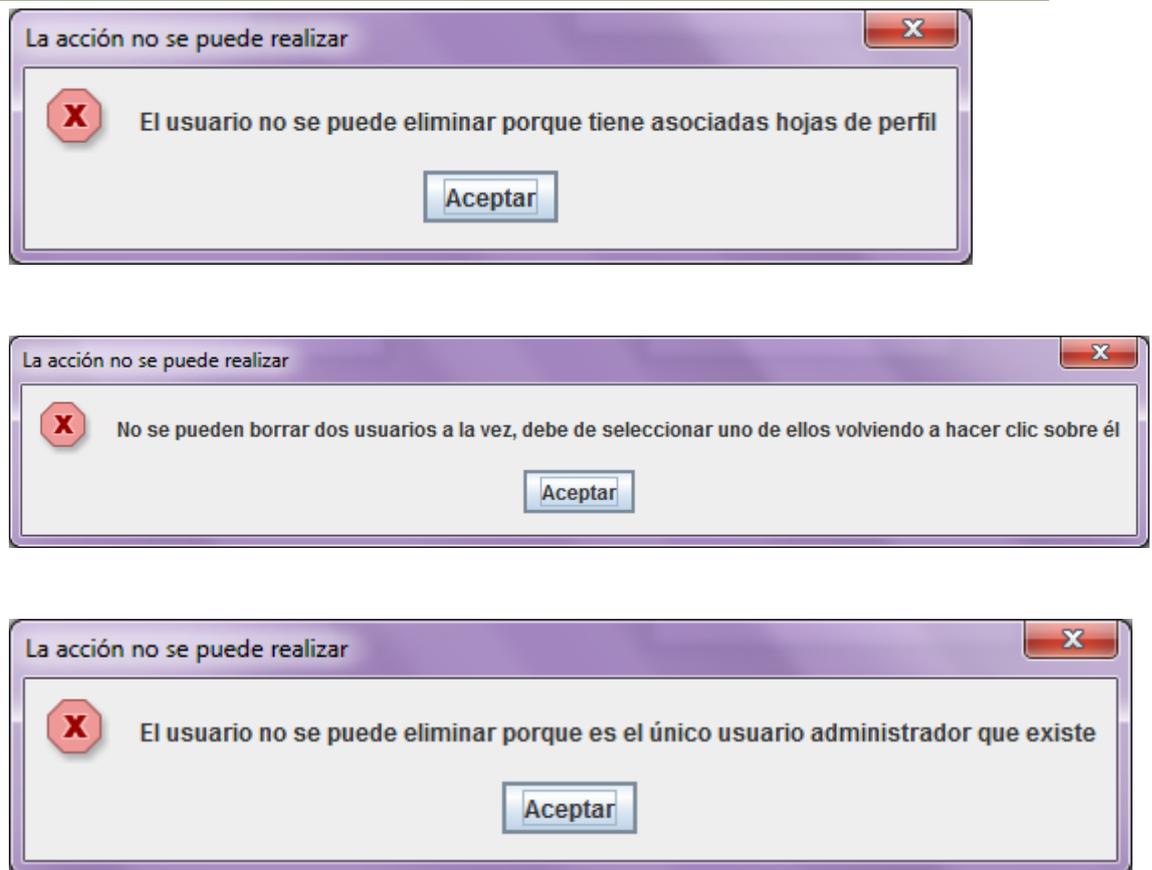


Figura 31: Posibles errores durante la eliminación de usuarios

- **Pantalla de Gestionar muestras:**
  - Existe la posibilidad de que se cree una muestra de forma incorrecta y se pueden encontrar los siguientes mensajes:



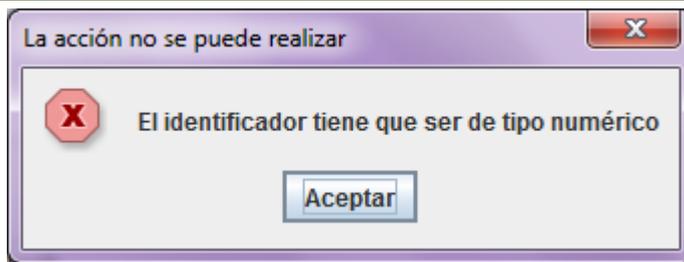


Figura 32: Posibles errores durante el alta de muestras

- o Existe la posibilidad de que se elimine una muestra de forma incorrecta y se pueden encontrar los siguientes mensajes:

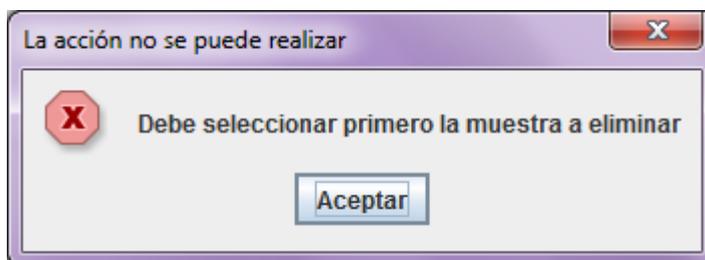
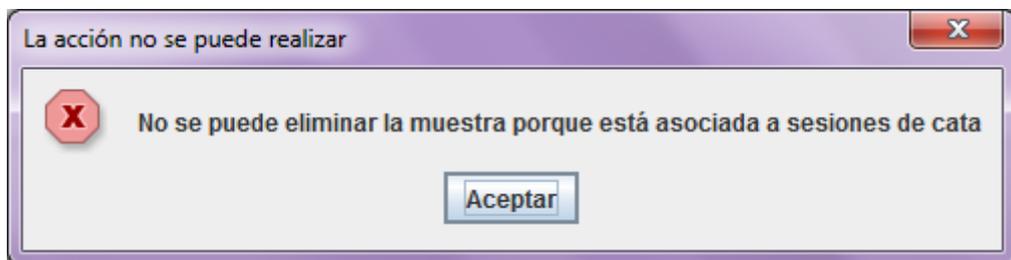
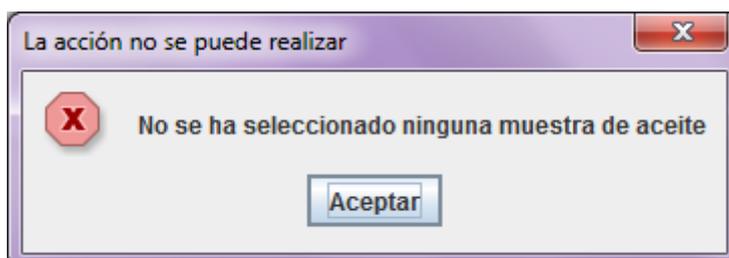


Figura 33: Posibles errores durante la eliminación de muestras

- o En el caso de consulta de información sobre las muestras nos podemos encontrar los siguientes mensajes:



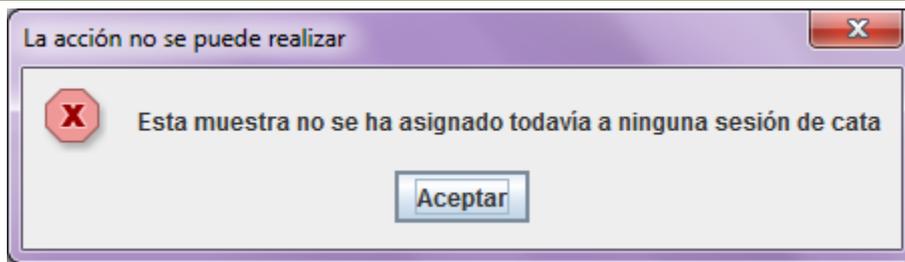


Figura 34: Posibles errores durante la consulta de las sesiones de cata de las muestras

- En el caso de consulta de información sobre las hojas de perfil de la muestra seleccionada nos podemos encontrar los siguientes mensajes:

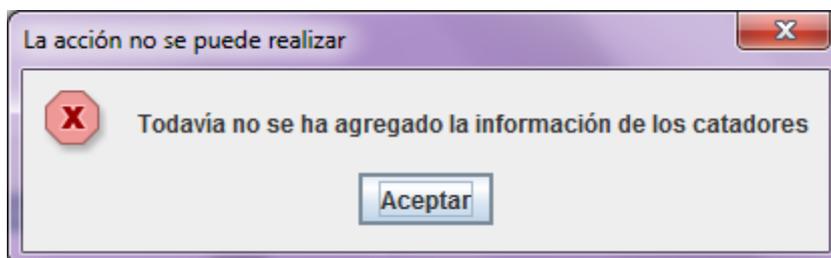
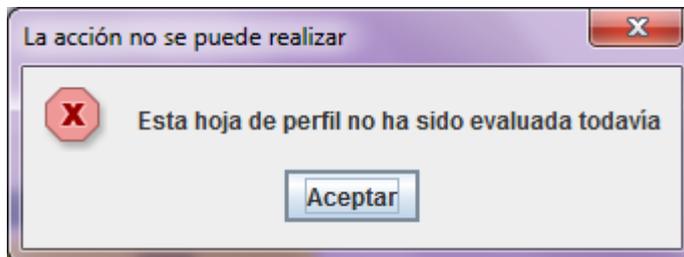


Figura 35: Posibles errores en la consulta de las hojas de perfil de una muestra

- *Pantalla de Gestionar dominios de expresión:*

- Rangos Numéricos

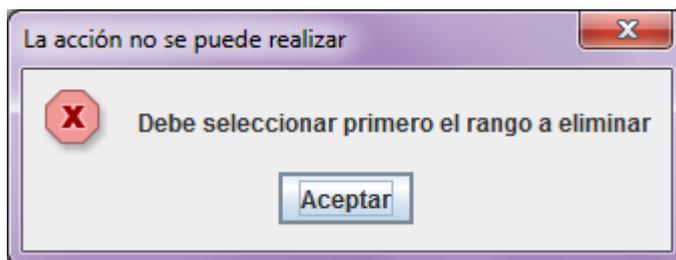
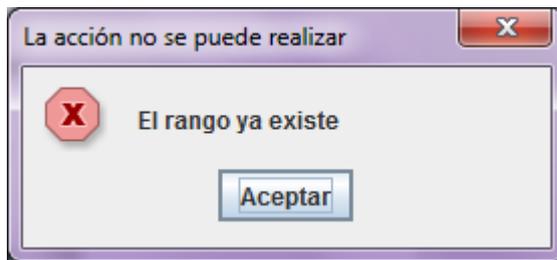


Figura 36: Posibles errores con rangos numéricos

- Rangos Lingüísticos

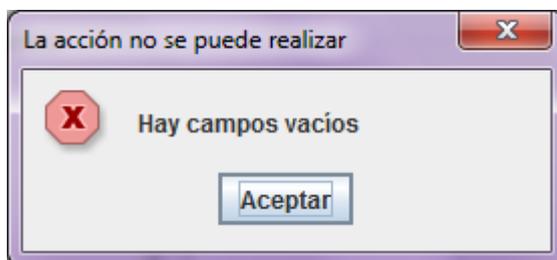


Figura 37: Posibles errores con rangos lingüísticos

- **Mensajes de error relacionados con la semántica de las etiquetas difusas:**

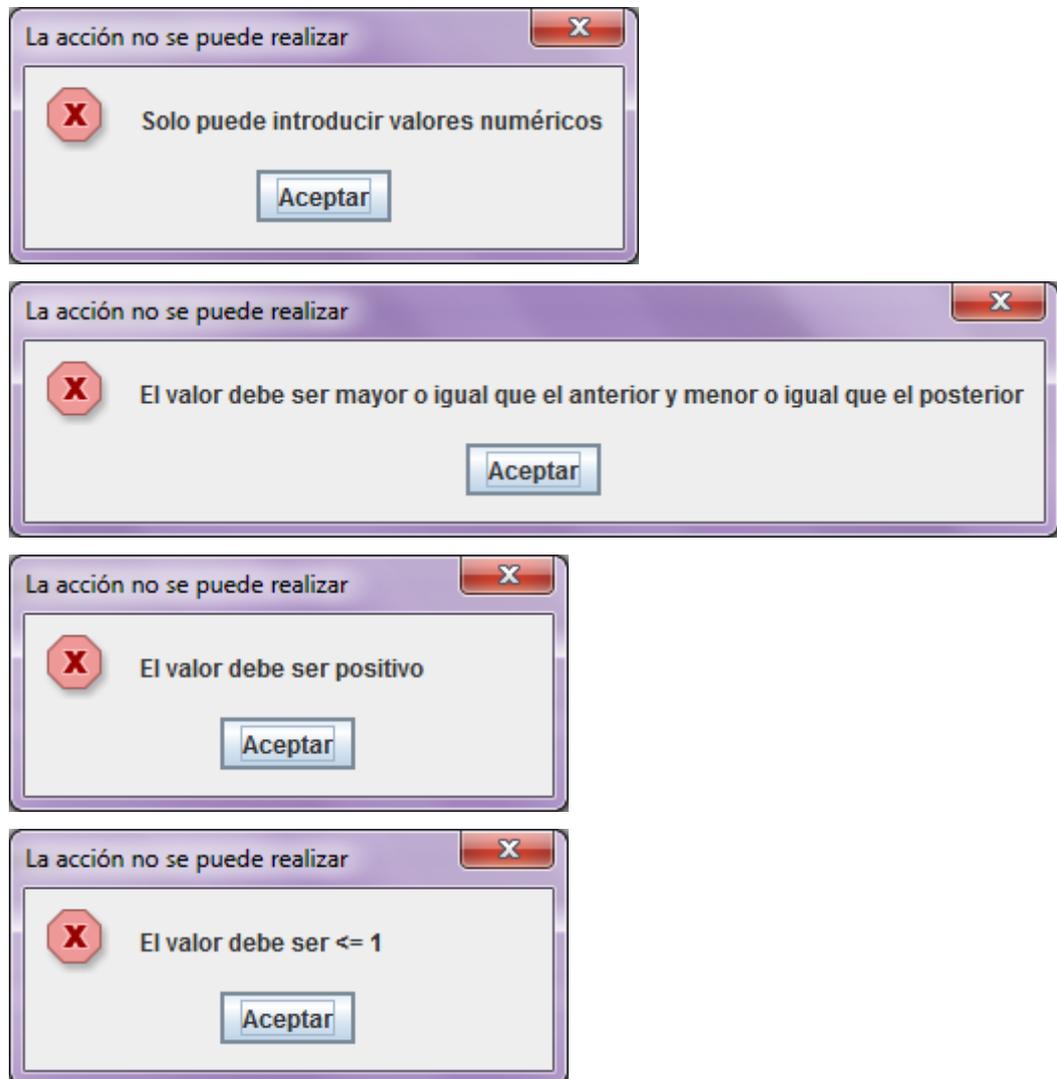


Figura 38: Posibles errores al gestionar etiquetas semánticas

- **Mensajes de error relacionados con la consulta de juego de etiquetas:**

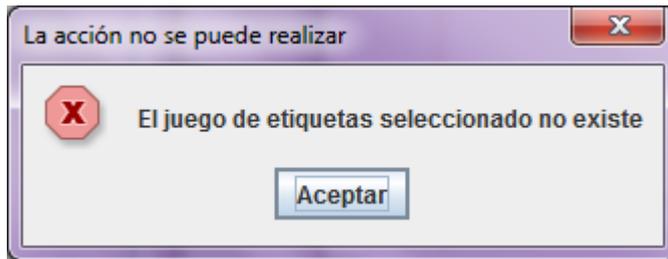


Figura 39: Posible error al consultar un juego de etiquetas

Independientemente del tipo de rango que decidamos borrar, si está siendo utilizado por alguna hoja de perfil no podremos borrarlo y nos aparecerá el siguiente mensaje:

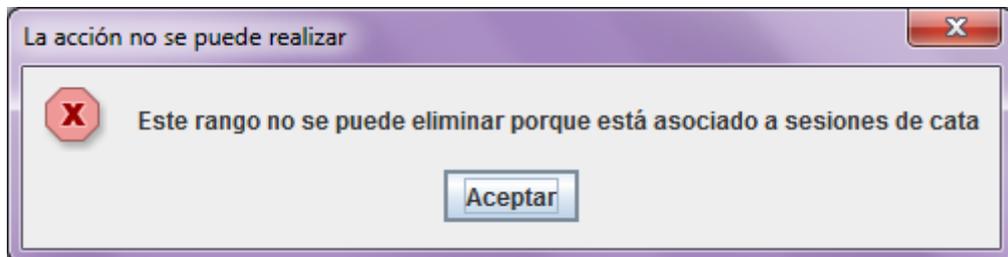


Figura 40: Posible error al eliminar un rango

- **Pantalla de Gestionar sesiones de cata:**
  - A la hora de crear sesiones de cata se pueden producir los siguientes errores:



Figura 41: Posibles errores al crear sesiones de cata

## 5. Implementación

---

La implementación es la actividad final de la Ingeniería del Software, aquella en la que el modelo obtenido en las actividades anteriores se debe transformar en código fuente.

Para ello se debe ser cuidadoso en la elección del lenguaje de programación empleado para la codificación y de la herramienta utilizada para generarla.

### 5.1. Tipo de arquitectura

En nuestro caso, vamos a desarrollar una aplicación de escritorio. El funcionamiento de las arquitecturas de este tipo es sencilla: la aplicación se encuentra instalada en el ordenador a la que los usuarios acceden de forma local, sin necesidad de disponer de una conexión a Internet ni acceder a un servidor local o externo.

### 5.2. Lenguajes y Herramientas usadas

#### 5.2.1. Lenguajes de programación y tecnologías utilizadas

##### 5.2.1.1. *Java*

Desarrollado en los laboratorios de Sun, es uno de los lenguajes de programación orientado a objetos que mayor repercusión ha tenido en los últimos años. Basado en C++ pero simplificado, mucho más fácil de usar, de más alto nivel y menos propenso a errores.

Cuenta con las siguientes características:

- ✓ Una amplísima biblioteca estándar de clases predefinidas.
- ✓ Las aplicaciones Java pueden ser ejecutadas indistintamente en cualquier plataforma sin necesidad de recompilación.
- ✓ Gestión avanzada de memoria mediante un recolector de basura.
- ✓ Gestión avanzada de errores, tanto en tiempo de compilación como de ejecución.
- ✓ Distribuido y multihebra.
- ✓ Lenguaje abierto. Kits de desarrollo y documentación gratuitos en la red.

Existen distintas “ediciones” de Java para el desarrollo de aplicaciones en distintos ámbitos:

- ✦ Aplicaciones de propósito general (J2SE).
- ✦ Aplicaciones de gestión en entornos empresariales (J2EE).
- ✦ Aplicaciones para teléfonos móviles, PDAs y otros dispositivos electrónicos que permitan aplicaciones empotradas (J2ME).

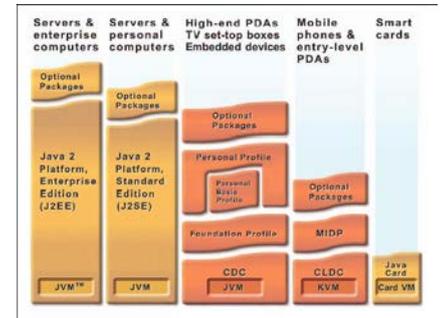


Figura 42: Ediciones de Java

La más utilizada es sin duda la edición estándar J2SE. El código desarrollado en este proyecto corresponden a este tipo y el código enfocado a Internet pertenece a J2EE.

Para la edición J2SE existen dos kits diferentes que pueden ser descargados de la página oficial de Sun:

- ✦ El Java Development Kit (JDK) permite desarrollar y ejecutar aplicaciones Java.
- ✦ El Java Runtime Environment (JRE) permite únicamente la ejecución.

J2SE incluye bibliotecas muy extensas y completas, que permiten la implementación de casi cualquier tipo de aplicación:

- |   |                                     |
|---|-------------------------------------|
| ✗ Seguridad                             | ✗ Interfaz de usuario (AWT, SWING)  |
| ✗ EEDDs                                 | ✗ Gráficos 2D                       |
| ✗ Componentes (JavaBeans)               | ✗ Manipulación, carga y descarga de |
| ✗ Internacionalización                  | ✗ imágenes                          |
| ✗ E/S                                   | ✗ Impresión                         |
| ✗ XML                                   | ✗ Acceso a bases de datos (JDBC)    |
| ✗ Redes y acceso a Internet             | ✗ Gestión de preferencias y         |
| ✗ Programación distribuida (RMI, CORBA) | ✗ configuraciones                   |
| ✗ Matemática de precisión arbitraria    |                                     |
| ✗ Sonido                                |                                     |

En cuanto a la compilación / ejecución Java no sigue un esquema tradicional. La compilación genera un ejecutable en bytecode o código intermedio independiente. Para su ejecución se requiere un JRE específico de la plataforma. El JRE está formado por una máquina virtual java (JVM) y las librerías de clases.

La JVM interpreta el bytecode o realiza su compilación justintime para que su ejecución sea más eficiente.

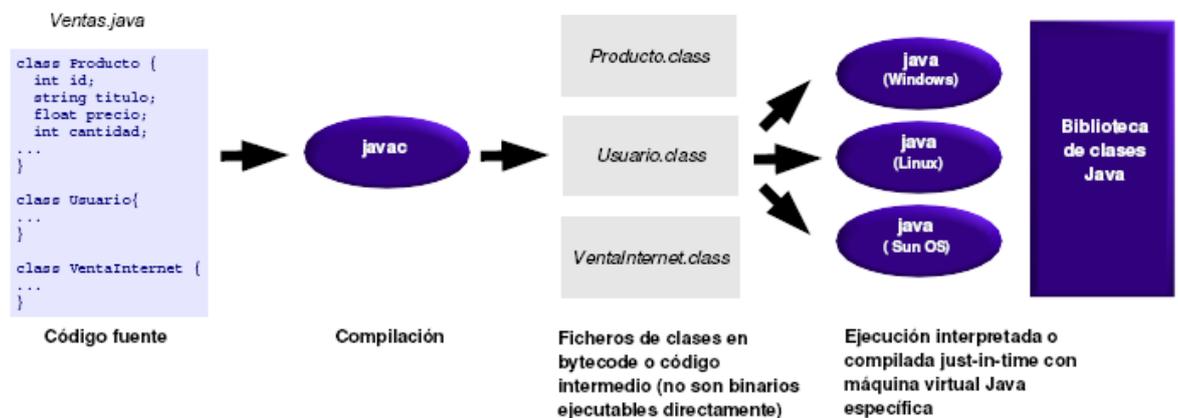


Figura 43: Compilación / ejecución de código en Java

#### Ventajas de este sistema:

- ✘ Se compila la aplicación una única vez y los ejecutables en bytecode obtenidos son válidos para cualquier plataforma. El código fuente queda a salvo.
- ✘ Es muy robusto. La máquina virtual Java es capaz de detectar y notificar gran cantidad de errores durante la ejecución de la aplicación (como accesos a elementos fuera de un vector).
- ✘ El recolector de basura no ocupa espacio en el ejecutable, ya que viene integrado en la JVM.
- ✘ Los ejecutables son pequeños porque las librerías de clases vienen proporcionadas junto a la JVM en el JRE de la plataforma concreta.

**Inconvenientes:**

- ✘ Velocidad. Evidentemente la interpretación o incluso compilación justintime del bytecode produce aplicaciones más lentas que en el caso de la ejecución directa de un binario. El recolector de basura puede suponer una sobrecarga adicional al procesador.
- ✘ La generalidad tiene como inconveniente que no se aprovecha totalmente la potencia de la máquina y del sistema operativo. Por ejemplo, el aspecto de una aplicación Java puede resultar simple y poco atractivo en comparación con las aplicaciones nativas.

**5.2.1.2. SQL**

El SQL (Structured query language), lenguaje de consulta estructurado, es un lenguaje surgido de un proyecto de investigación de IBM para el acceso a bases de datos relacionales. Actualmente se ha convertido en un estándar de lenguaje de bases de datos, y la mayoría de los sistemas de bases de datos lo soportan, desde sistemas para ordenadores personales, hasta grandes ordenadores.

Como su nombre indica, el SQL nos permite realizar consultas a la base de datos. Pero el nombre se queda corto ya que SQL además realiza funciones de definición, control y gestión de la base de datos. Las sentencias SQL se clasifican según su finalidad dando origen a tres sublenguajes:

- **DDL** (Data Description Language), lenguaje de definición de datos, incluye órdenes para definir, modificar o borrar las tablas en las que se almacenan los datos y de las relaciones entre estas.

COMANDO	DESCRIPCIÓN
<b>CREATE</b>	Utilizado para crear nuevas tablas, campos e índices.
<b>DROP</b>	Empleado para eliminar tablas e índices.

**ALTER**

Utilizado para modificar las tablas agregando campos o cambiando la definición de los campos.

Tabla 40: Tipos de instrucciones DDL

- **DCL** (Data Control Language), lenguaje de control de datos, contiene elementos útiles para trabajar en un entorno multiusuario, en el que es importante la protección de los datos, la seguridad de las tablas y el establecimiento de restricciones en el acceso, así como elementos para coordinar la compartición de datos por parte de usuarios concurrentes, asegurando que no interfieren unos con otros.

COMANDO	DESCRIPCIÓN
<b>GRANT</b>	Permite dar permisos a uno o varios usuarios o roles para realizar tareas determinadas.
<b>REVOKE</b>	Permite eliminar permisos que previamente se han concedido con GRANT. Las tareas sobre las que se pueden conceder o denegar permisos son: <ul style="list-style-type: none"> <li>• CONNECT</li> <li>• SELECT</li> <li>• INSERT</li> <li>• UPDATE</li> <li>• DELETE</li> <li>• USAGE</li> </ul>
<b>COMMIT</b>	Finaliza la transacción actual.
<b>ROLLBACK</b>	Aborta la transacción actual.

Tabla 41: Tipos de instrucciones DCL

- **DML** (Data Manipulation Language), lenguaje de manipulación de datos, nos permite recuperar los datos almacenados en la base de datos y también incluye órdenes para permitir al usuario actualizar la base de datos añadiendo

---

nuevos datos, suprimiendo datos antiguos o modificando datos previamente almacenados.

COMANDO	DESCRIPCIÓN
<b>SELECT</b>	Utilizado para consultar registros de la base de datos que satisfagan un criterio determinado.
<b>INSERT</b>	Utilizado para cargar lotes de datos en la base de datos en una única operación.
<b>UPDATE</b>	Utilizado para modificar los valores de los campos y registros especificados.
<b>DELETE</b>	Utilizado para eliminar registros de una tabla de una base de datos.

Tabla 42: Tipos de instrucciones DML

### 5.2.1.3. JPA

La persistencia se define, en computación, como la capacidad de los datos de sobrevivir a la ejecución de la aplicación que los creó. La persistencia se consigue almacenando los datos en ficheros o en bases de datos [164,165].

La forma más sencilla de hacer persistir un objeto Java es a través del mecanismo de serialización o a través de ficheros de texto que almacenen el estado de un objeto para que pueda ser posteriormente recuperado. Para aplicaciones que manejen grandes cantidades de datos interrelacionados entre si, el uso de una base de datos es fundamental.

*Problemática de la persistencia en bases de datos*

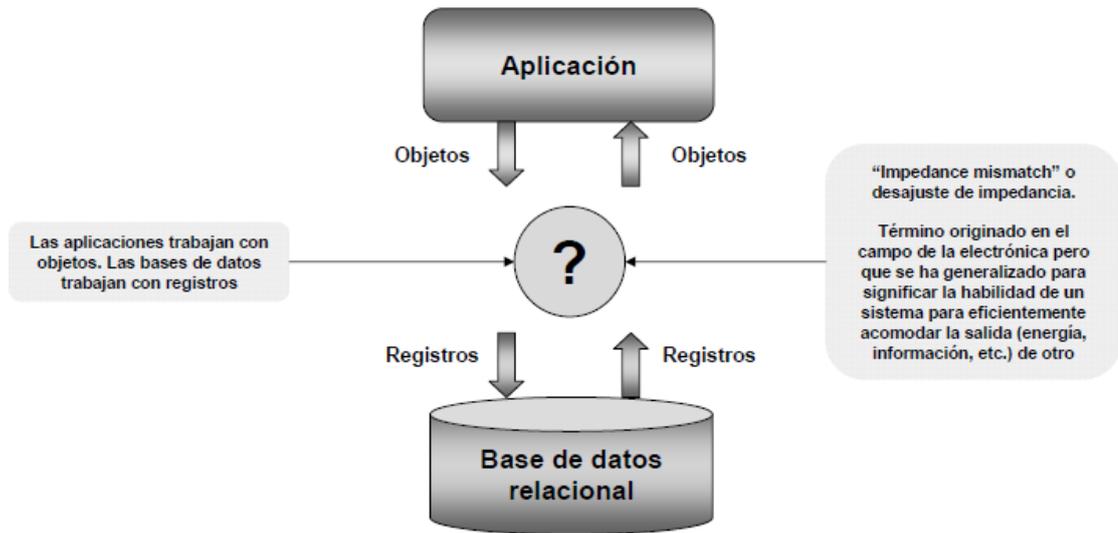


Figura 44: Problemática de la persistencia en las BBDD

*Solución 1: Usar datos relaciones en la aplicación*

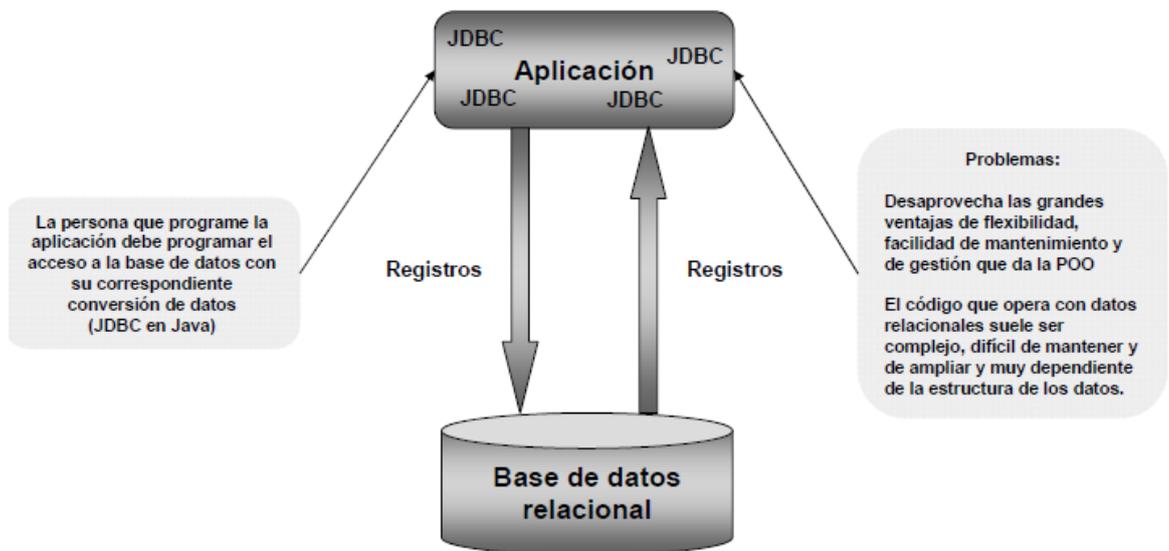


Figura 45: Uso de datos relacionales en la aplicación

**Solución 2: Usar datos relaciones en la aplicación usando DAO (Database Access Object)**

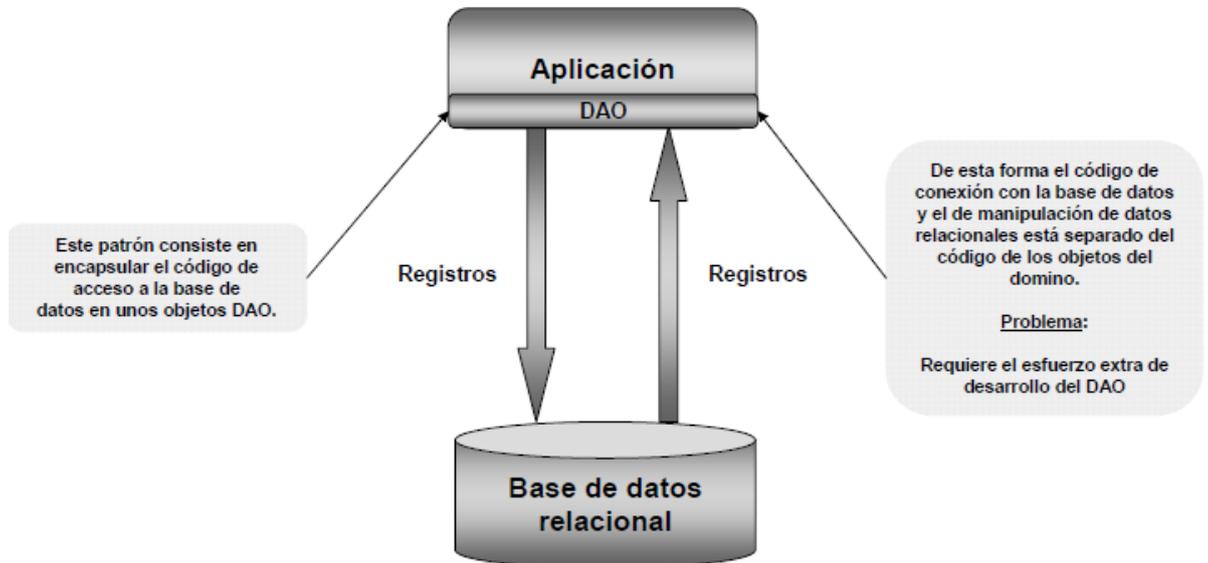


Figura 46: Uso de datos relacionales en la aplicación usando DAO

**Solución 3: Usar una base de datos orientada a objetos**

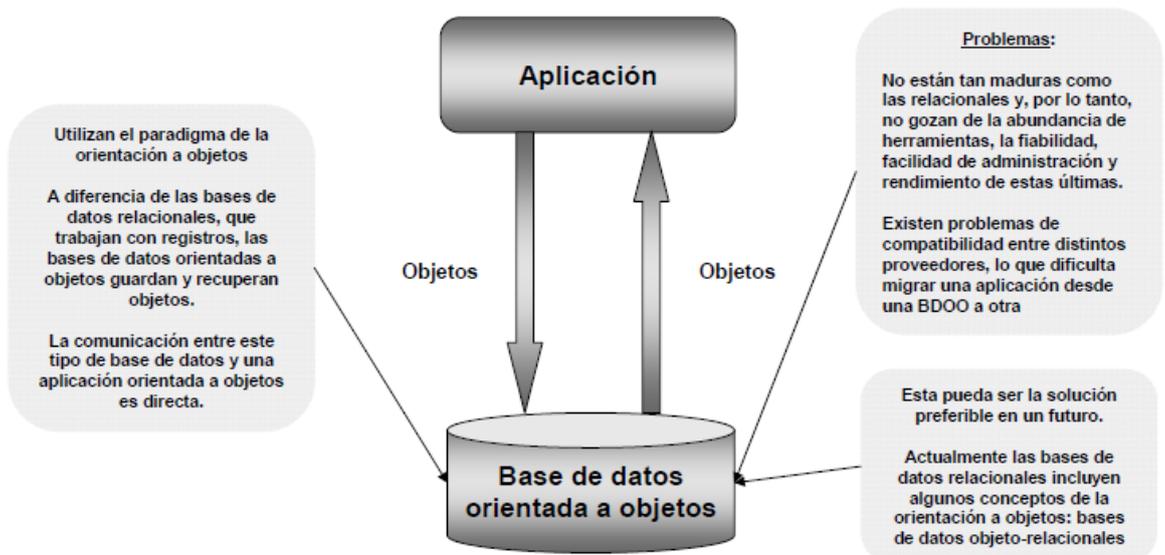


Figura 47: Uso de una BD orientada a objetos

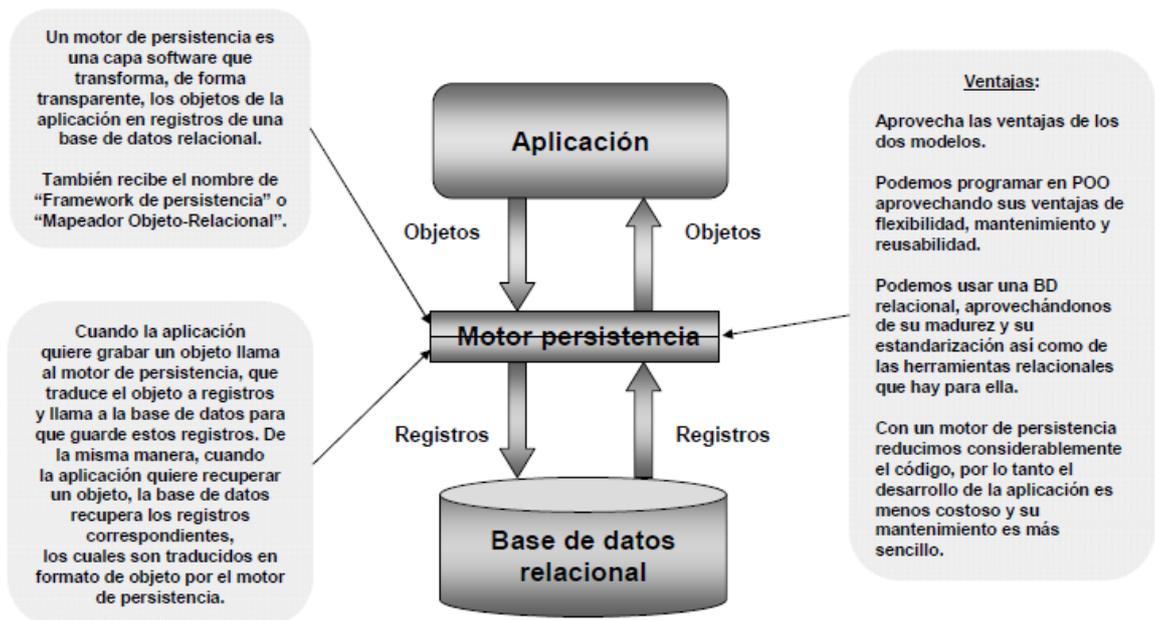
**Solución 4: Usar un motor de persistencia**

Figura 48: Uso de un motor de persistencia

**Características de JPA**

JPA es un nuevo estándar para la persistencia de datos en Java. Está basado en POJOs (Plain Old Java Objects), lo que quiere decir que puede persistir clases Java que no tengan nada de particular (no heredan de una clase en particular, no implementan un determinado interface, etc.).

Forma parte de la especificación EJB 3.0 (Enterprise Java Beans) perteneciente a la plataforma Java EE 5.0. También puede ser usado como *framework* de persistencia para proyectos Java SE. La implementación de referencia se basa en el API TopLink de Oracle.

JPA se basa en las anotaciones introducidas en la versión 5 de Java. Las anotaciones son metadatos, es decir, información sobre los elementos del programa. Vienen a ser algo similar a las etiquetas del javadoc (@author) pero aplicadas al código, no a los comentarios.

Por defecto Java incluye tres anotaciones:

- ✓ **Override:** anota a un método e indica que sobrescribe a otro de la superclase.
- ✓ **Deprecated:** indica que un método o clase está anticuado y no debe usarse.
- ✓ **SuppressWarnings:** el compilador no avisa de warnings ocurridos sobre el elemento anotado.

Las anotaciones no son más que interfaces por lo que pueden declarar métodos o constantes. La potencia de las anotaciones viene de la posibilidad de crear anotaciones personalizadas. Pueden aplicarse a clases, métodos o atributos y se agrupan en dos categorías:

- **Lógicas:** Describen las entidades del modelo desde el punto de vista del modelo de objetos.
- **Físicas:** Describen un modelo de datos concreto en la base de datos (especificando tablas, columnas, restricciones, etc.).

Pueden evitarse las anotaciones usando un fichero de mapeo XML. Generalmente se prefieren las anotaciones por la cercanía entre la propia anotación y el código al que se refieren. Pueden incluso coexistir las anotaciones con un fichero de mapeo XML (en caso de conflicto la información de este último prevalecerá).

### Ejemplo de JPA

El modelo del dominio es el siguiente:

Un cliente (customer) crea pedidos (Order) que deberán ser enviados a determinadas direcciones.



Figura 49: Ejemplo JPA - Relación entre las clases

La implementación del modelo da lugar a las clases Customer y Order (que son POJOs).

```

public class Customer implements Serializable
{
    private int id;
    private String name;
    private Collection<Order> orders = new ArrayList<Order>();

    public int getId()
    { return id; }

    public void setId(int id)
    { this.id = id; }

    public String getName()
    { return name; }

    public void setName(String name)
    { this.name = name; }

    public Collection<Order> getOrders()
    { return orders; }

    public void setOrders(Collection<Order> newValue)
    { this.orders = newValue; }
}

public class Order
{
    private int id;
    private String address;
    private Customer customer;

    public int getId()
    { return id; }

    public void setId(int id)
    { this.id = id; }

    public String getAddress()
    { return address; }

    public void setAddress(String address)
    { this.address = address; }

    public Customer getCustomer()
    { return customer; }

    public void setCustomer(Customer customer)
    { this.customer = customer; }
}
    
```

Figura 50: Ejemplo JPA- Implementación de las clases

Las tablas que existen en la BD son:

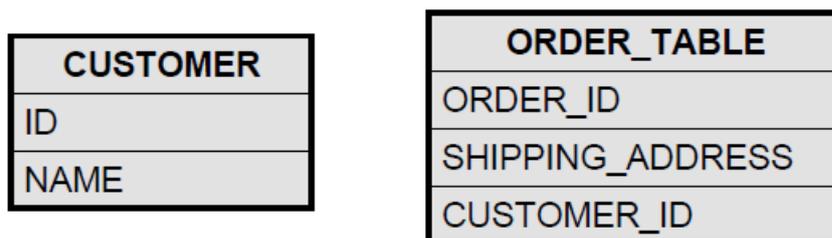


Figura 51: Ejemplo JPA - Tablas existentes en la BD

El mapeado OR se establece a través de anotaciones sobre el código:

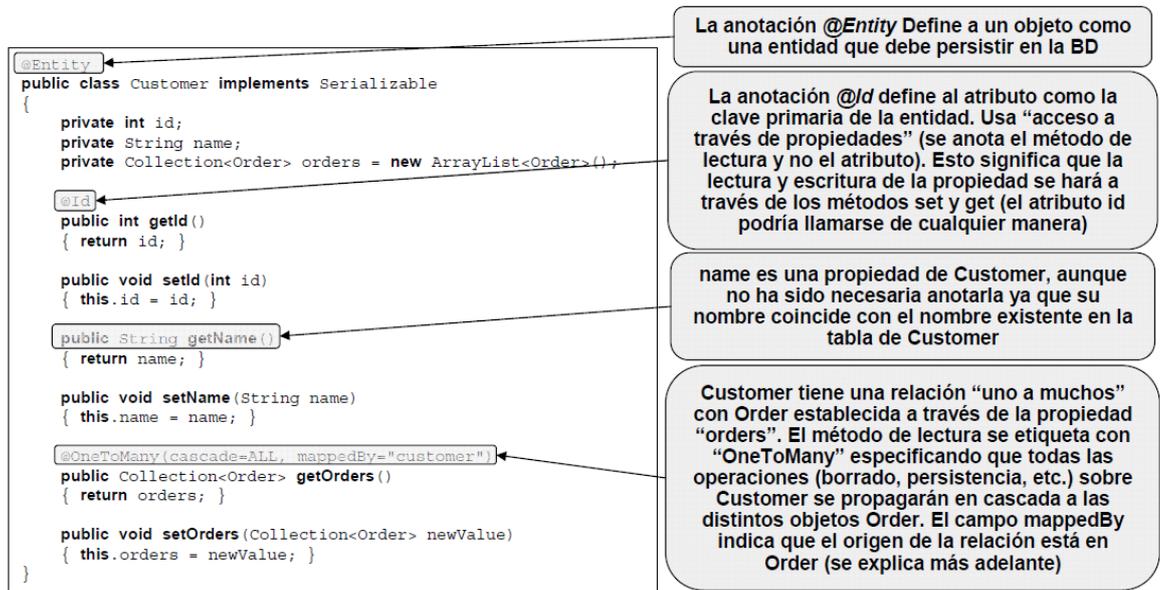


Figura 52: Ejemplo JPA - Clase Customer con anotaciones

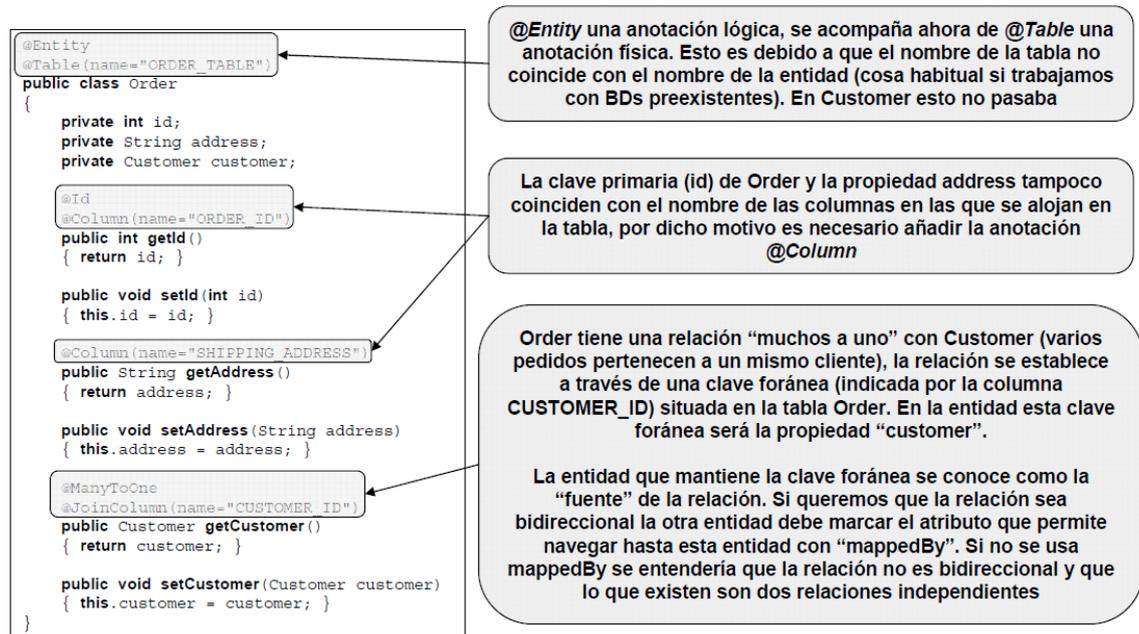


Figura 53: Ejemplo JPA - Clase Order con anotaciones

*Ejemplo de mapeado de relaciones 1 a 1 y de 1 a N*

Se usa la clave de una entidad como clave foránea en la otra entidad.

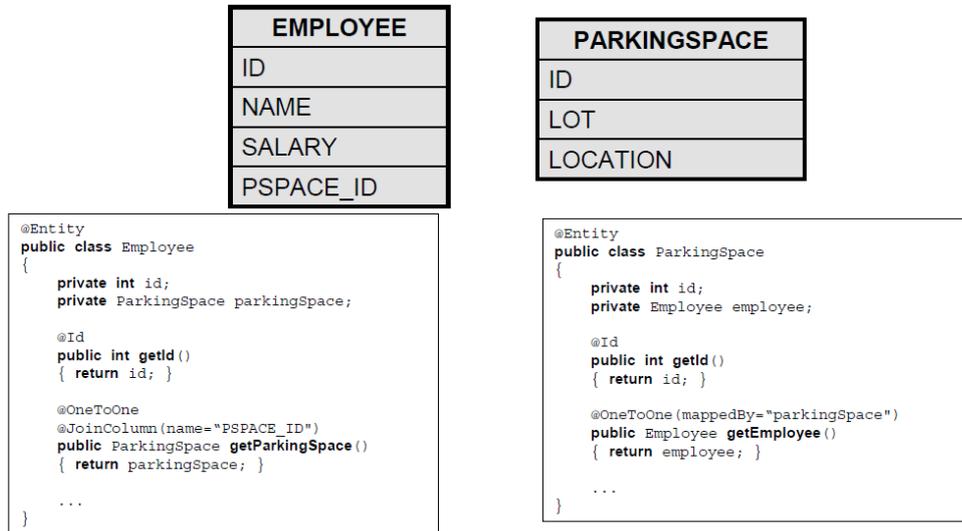


Figura 54: Mapeo de relaciones de 1 a N

*Ejemplo de mapeado de relaciones de N a M*

Para mapear relaciones M a N necesitamos una tabla de unión o relación (join table) que enlace ambas entidades. Esta tabla de unión se forma con las claves primarias de las entidades que se están relacionando.

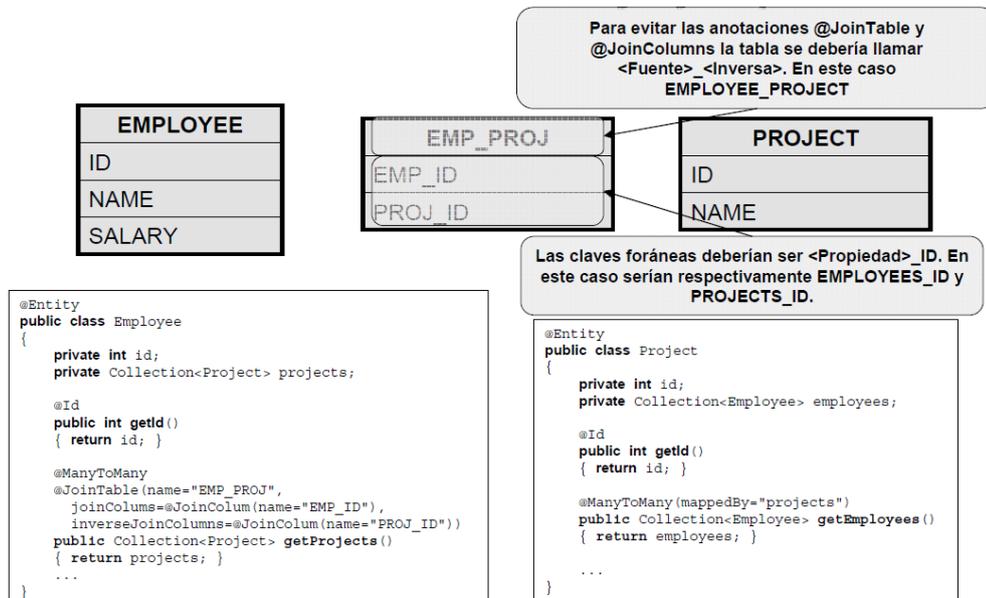


Figura 55: Mapeo de relaciones de N a M

Para crear un nuevo cliente lo haremos de la siguiente manera:

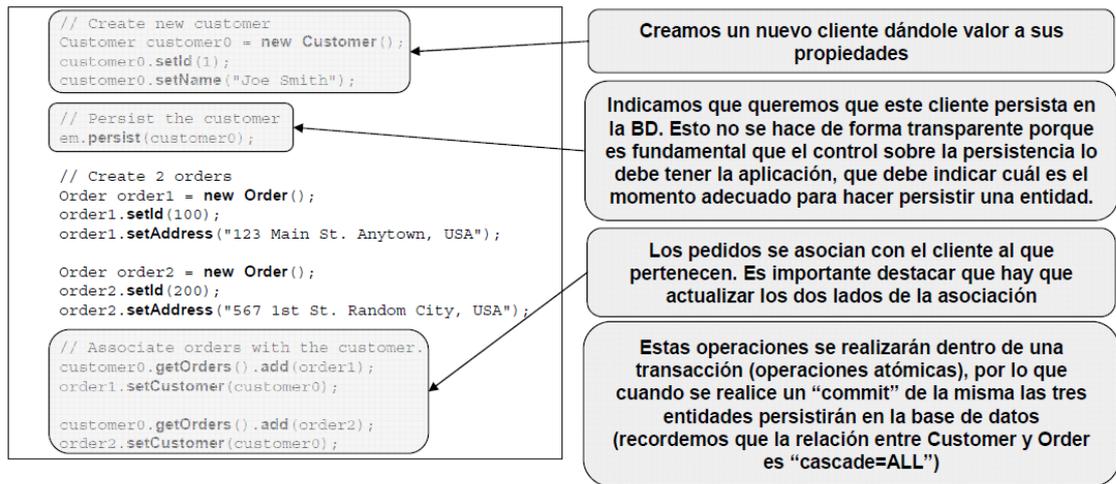


Figura 56: Ejemplo JPA - Manipulación de las clases

Para realizar las consultas se hacen a través del lenguaje JPQL (Java Persistence Query Language). Es un lenguaje independiente de la BD ya que opera sobre el modelo lógico y no sobre el modelo físico. Utiliza una sintaxis muy similar al SQL que facilita su utilización.

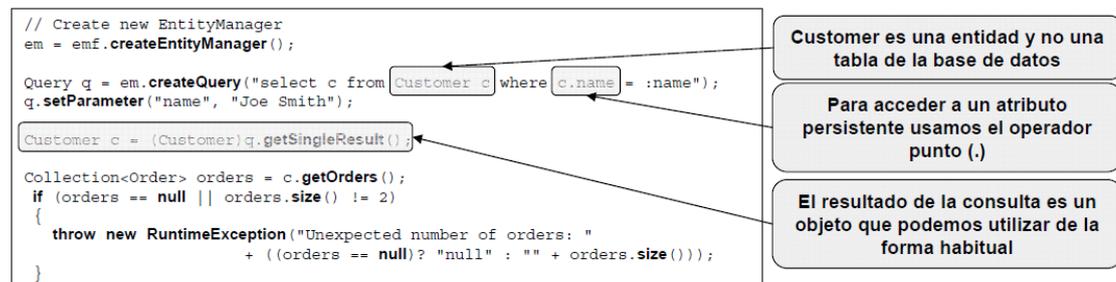


Figura 57: Ejemplo de JPA - Consultas con JPQL

**EntityManager** es una interfaz que es implementado por el proveedor de persistencia. Las operaciones de persistencia se delegan fundamentalmente en él. Para obtener instancias se utiliza la clase EntityManagerFactory que hace referencia a una unidad de persistencia (Persistence Unit) que contiene aspectos de configuración (librería usada, conexión JDBC,...), clases de entidades que utilizarán los EntityManagers,..

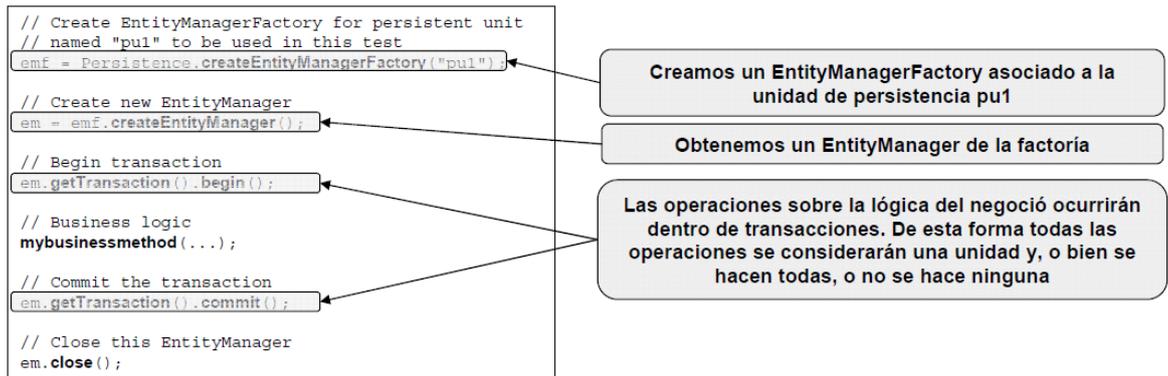


Figura 58: Ejemplo de EntityManager

#### 5.2.1.4. Librería JFreeChart

JFreeChart <sup>1</sup>es un marco de software open source para el lenguaje de programación Java, el cual permite la creación de gráficos complejos de forma simple.

JFreeChart también trabaja con GNU Classpath, una implementación en software libre de la norma estándar de biblioteca de clases para el lenguaje de programación Java.

JFreeChart es compatible con una serie de gráficas diferentes, incluyendo cuadros combinados. Después de tipos de gráficos son compatibles:

- ✗ Gráficos XY (línea, spline y dispersión). Es posible usar un eje del tiempo.
- ✗ Gráfico circular.
- ✗ Diagrama de Gantt.
- ✗ Gráficos de barras (horizontal y vertical, apiladas e independientes). También tiene incorporado un dibujador de histogramas.
- ✗ Single valued (termómetro, brújula, indicador de velocidad) que luego se pueden colocar sobre el mapa.
- ✗ Varias gráficas específicas (tabla de viento, gráfica polar, burbujas de diferentes tamaños, etc.)

Además los gráficos, es posible colocar varios marcadores en el área de gráfica.

<sup>1</sup> <http://www.jfree.org/jfreechart/>

JFreeChart dibuja automáticamente las escalas de los ejes y leyendas. Con el ratón informático se puede hacer zoom en la interfaz de la gráfica automáticamente y cambiar algunos ajustes a través del menú local. Las tablas existentes pueden actualizarse fácilmente a través de los oyentes (listeners) que la biblioteca tiene en sus colecciones de datos.

Para el uso de JFreeChart en Netbeans tenemos que añadir las librerías:

- ✗ jfreechart-1.0.14.jar
- ✗ jcommon-1.0.18.jar

## 5.2.2. Herramientas de desarrollo utilizadas

### 5.2.2.1. *IDE Netbeans*

El IDE NetBeans <sup>2</sup>es un entorno de desarrollo integrado - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. Se trata de un producto libre y gratuito sin restricciones de uso. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

---

<sup>2</sup> [https://netbeans.org/index\\_es.html](https://netbeans.org/index_es.html)

---

---

La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están:

- Administración de las interfaces de usuario (en este proyecto para el desarrollo de la interfaz gráfica se ha utilizado la API Swing).
- Administración de las configuraciones del usuario.
- Administración del almacenamiento.
- Administración de ventanas.
- Framework basado en asistentes.

### Conexión de Netbeans con MySQL y JPA

En primer lugar añadiremos al proyectos las librerías necesarias:

- *MySQL JDBC Driver* para la conexión con la BD.
- El resto están relacionadas con la implementación de la API de JPA que se va a utilizar que en este caso que es *EclipseLink*.

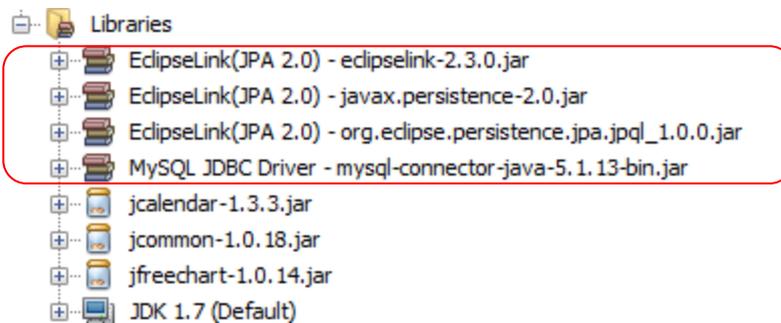


Figura 59: Librerías necesarias para la conexión con la BD y motor de persistencia JPA

A continuación crearemos la conexión con la base de datos:

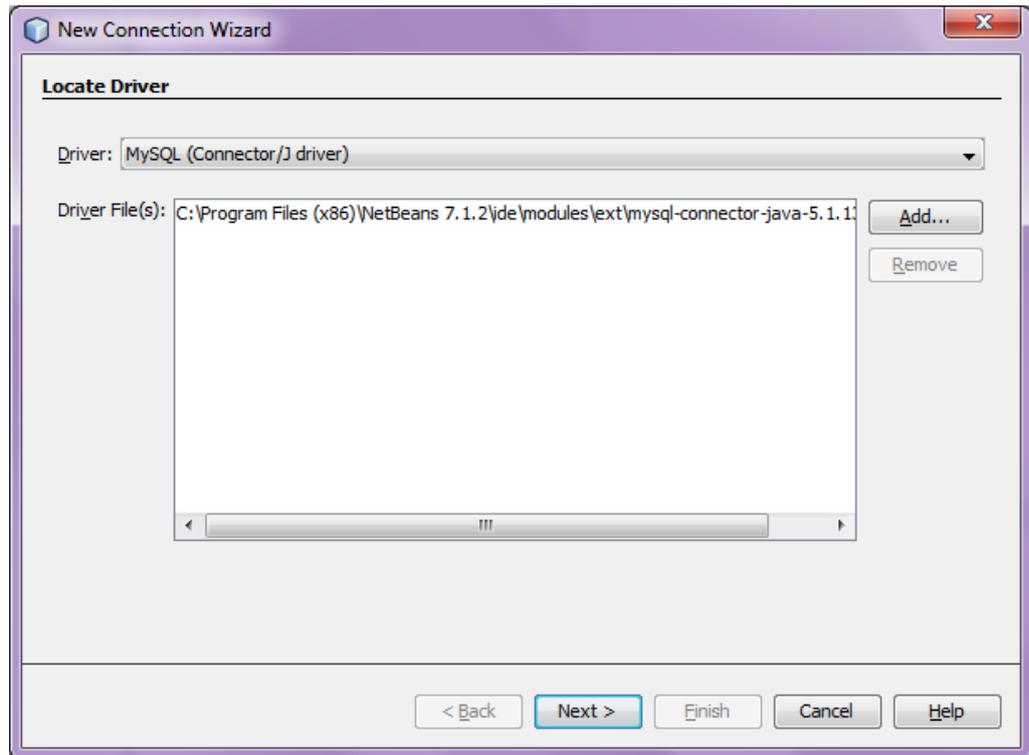


Figura 60: Conexión con el servidor de BD MySQL I

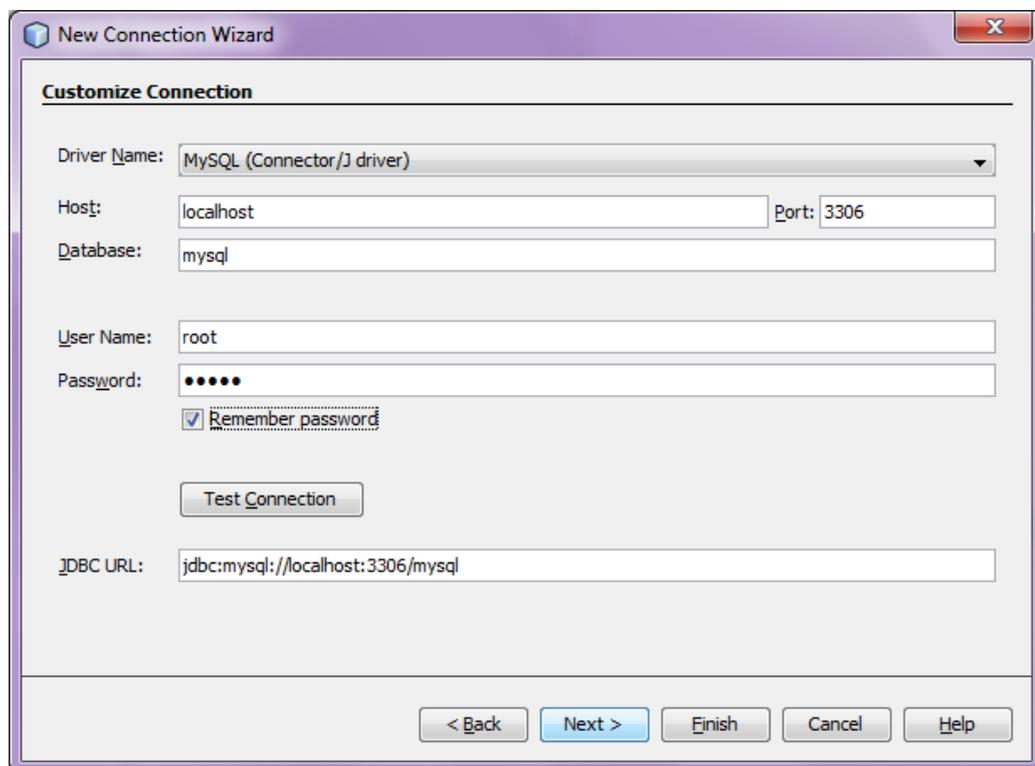


Figura 61: Conexión con el servidor de BD MySQL II

El siguiente paso será crear la unidad de persistencia donde seleccionaremos la conexión creada en el apartado anterior:

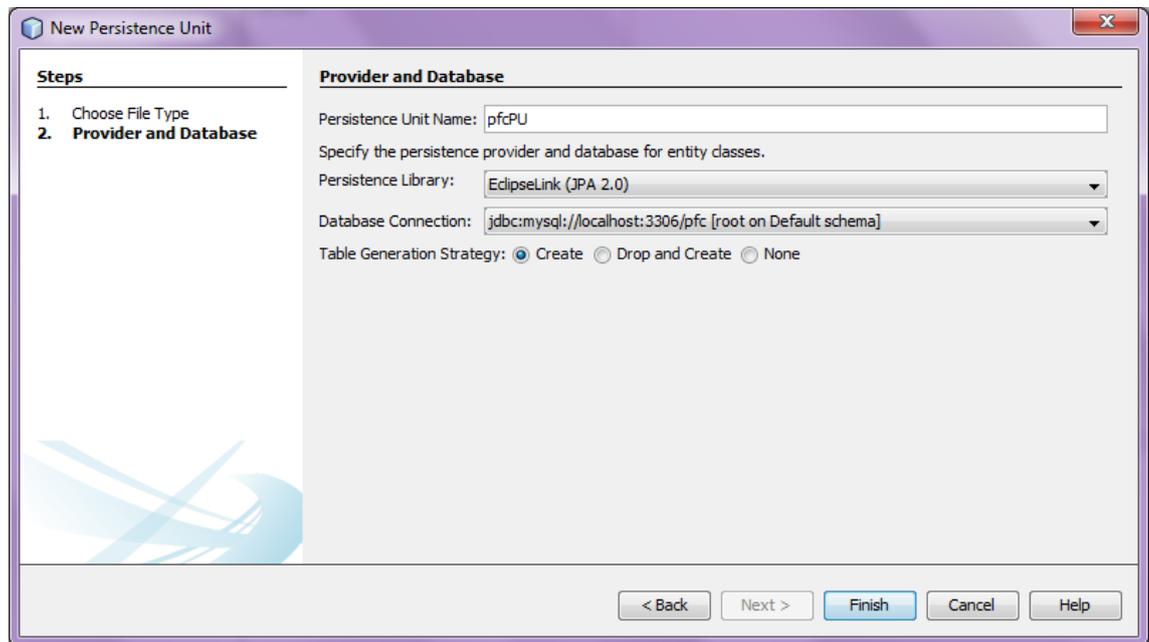


Figura 62: Crear unidad de persistencia

La unidad de persistencia es un archivo de configuración que se encargará de abrir y cerrar las conexiones necesarias.

Luego crearemos las clases asociadas a las tablas existentes en la base de datos:

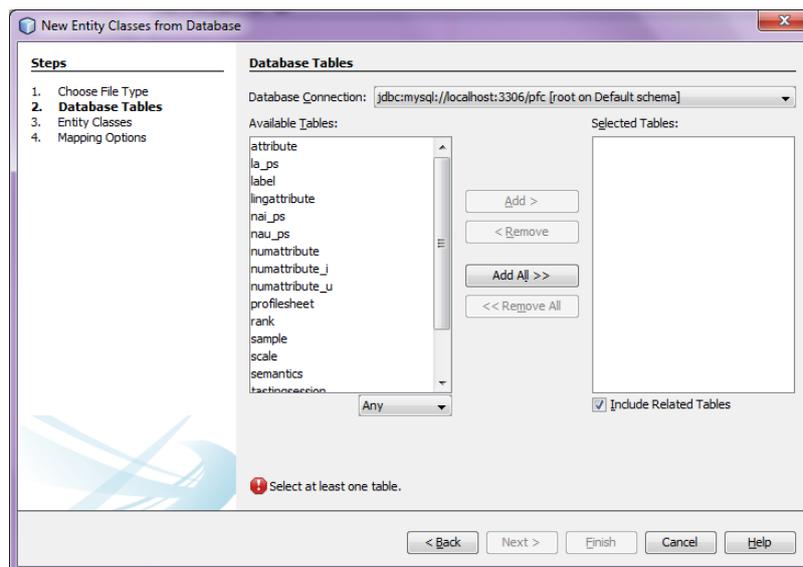


Figura 63: Crear las clases desde la BD I

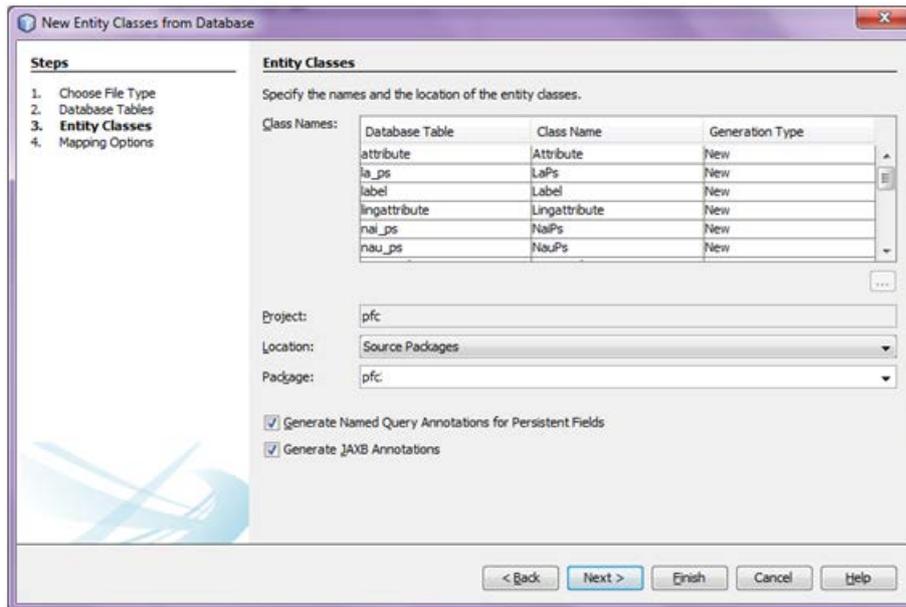


Figura 64: Crear las clases desde la BD II

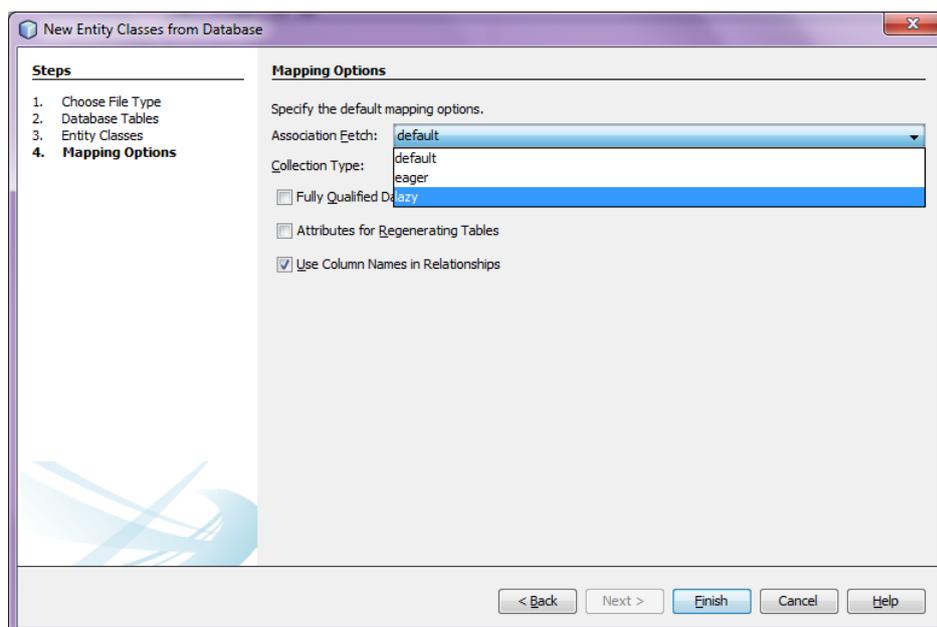


Figura 65: Crear las clases desde la BD III

Association fetch:

- **Eager:** Carga todas las asociaciones con sus datos que se tengan mapeadas. Tiene como inconveniente que puede cargar en memoria toda una base de datos si no se usa correctamente.

- **Lazy:** carga en memoria solo los datos pertinentes a lo que se pidió, si se requieren búsquedas a mas profundidad, JPA administra eso para cargar en memoria más datos cuando sean requeridos.

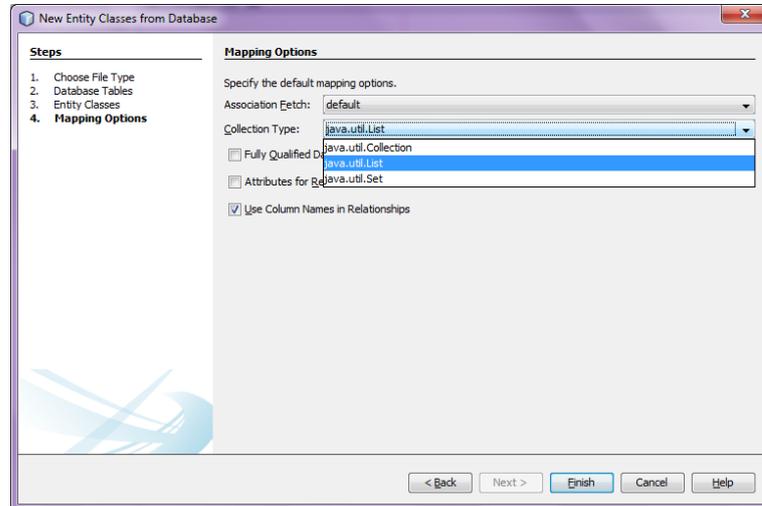


Figura 66: Crear las clases desde la BD IV

Respecto a la forma en la que se presentarán los objetos recuperados podemos elegir las siguientes opciones:

- **Set** : cuyos elementos tienen un id que no se puede repetir.
- **Collection** : Colección de objetos.
- **List**: Colección de objetos (Diferente implementación y métodos que Collection).

Por último los controladores jpa para cada una de las clases creadas en el paso anterior:

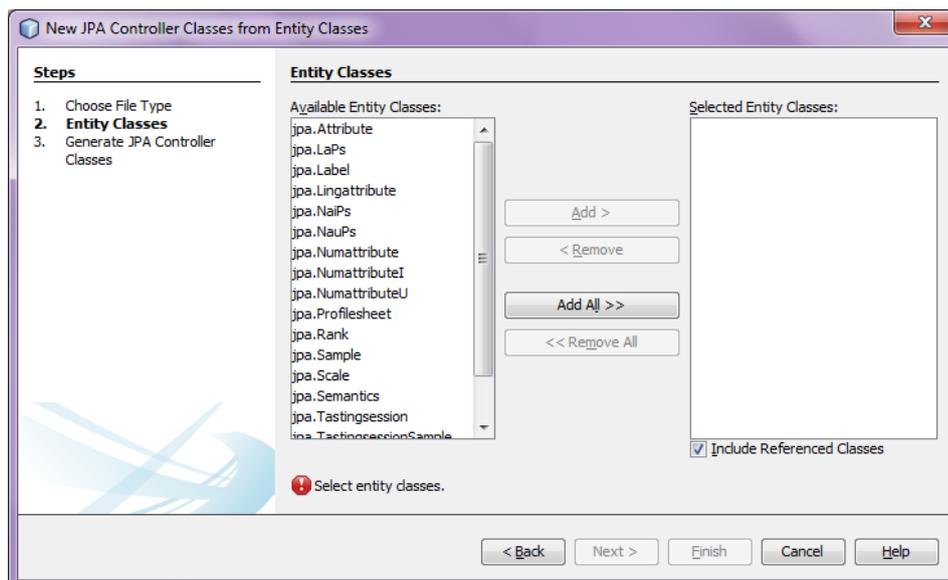


Figura 67: Crear clases JPA controller I

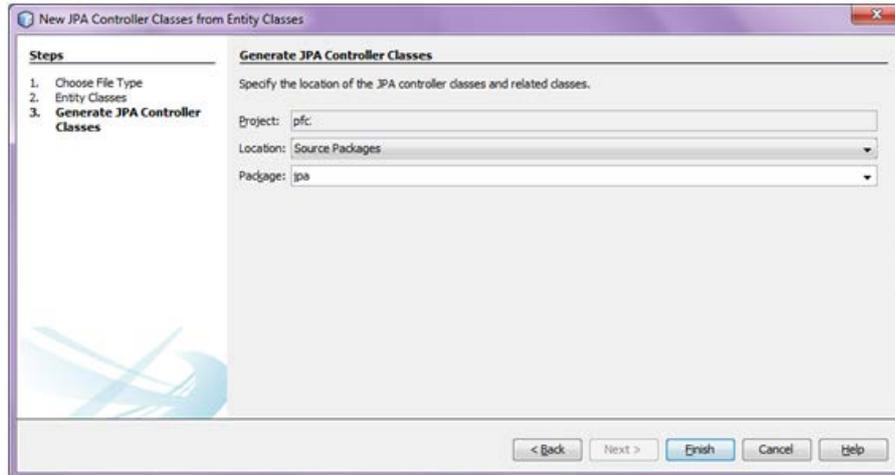


Figura 68: Crear clases JPA controller II

#### 5.2.2.2. *MySQL WorkBench*<sup>3</sup>

Se trata de una herramienta gráfica que permite trabajar con MySQL Server 5.1 (y superior) . Su lanzamiento sustituyó a dos herramientas de trabajo MySQL Administrator y Query Browser. Tiene tres áreas principales de funcionalidad:

- ✦ **SQL para el desarrollo** – Permite crear y administrar las conexiones a los servidores de base de datos. Permite configurar los parámetros de conexión, MySQL Workbench proporciona la capacidad para ejecutar consultas SQL en base a un editor de SQL.
- ✦ **Modelado de datos** – Permite la creación de modelos gráficos de esquema de base de datos, ingeniería directa e inversa, el editor de tablas permite a parte de editar las tablas, las columnas, los índices, los disparadores, privilegios o vistas.
- ✦ **Administración del Servidor** – Permite crear y administrar instancias de servidor.

Algunas de las características más interesantes de MySQL Workbench son:

- ✓ Edición de de diagramas basada en Cairo, con posibilidad de realizar una salida en los formatos como OpenGL, Win32, X11, Quartz, PostScript, PDF...

<sup>3</sup> <http://dev.mysql.com/doc/workbench/en/index.html>

- ✓ Proporciona una representación visual de las tablas, vistas, procedimientos y funciones almacenadas y claves foráneas.
- ✓ Permite acceso a bases de datos e ingeniería inversa de las mismas para crear los SQL de creación.
- ✓ Ofrece sincronización con la base de datos y el modelo.
- ✓ Permite generar los scripts SQL a partir del modelo creado.
- ✓ Ofrece una arquitectura extensible.
- ✓ Tiene soporte para exportar los datos como script SQL CREATE.
- ✓ Permite importar modelos de DBDesigner4.
- ✓ Ofrece soporte completo a las características de MySQL 5.

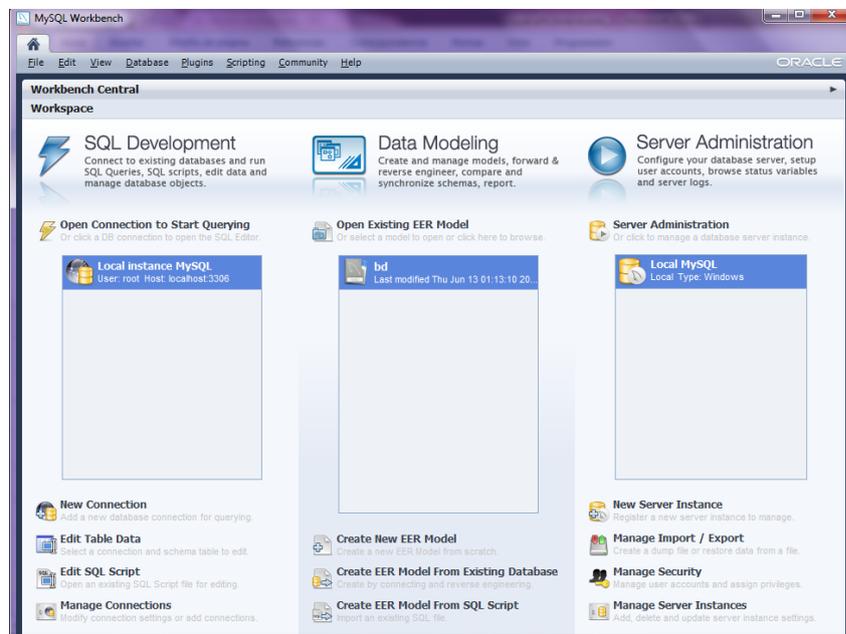


Figura 69: Herramienta MySQL WorkBench

## 6. Pruebas

El objetivo de esta fase es realizar un conjunto de pruebas sobre el sistema. Con esto intentaremos conseguir llegar a un sistema sin errores garantizando, como hemos dicho, la calidad del software. Para comprobar esto realizaremos unas pruebas de sistema.

### 6.1. Casos de test

Los casos de test diseñados son los siguientes:

#### Test 1: Dar de alta una muestra de aceite

<b>Requisitos testeados</b>	RQ-04
<b>Precondiciones</b>	Usuario administrador identificado, se encuentra en el menú del administrador.
<b>CONTENIDO</b>	
<b>Acción</b>	Accede a: <i>Gestionar muestras</i> .
<b>Checkpoint 1</b>	El sistema debe mostrar las muestras que ya están dadas de alta en una tabla y bajo esta un formulario para introducir los datos de la nueva muestra.
<b>Acción</b>	El usuario rellena los datos de la muestra (nombre, identificador, cliente) y pulsa en <i>Guardar</i> .
<b>Checkpoint 2</b>	El sistema debe actualizar la tabla que contiene las muestras dadas de alta y desaparecerá el formulario que había bajo ella.

Tabla 43: Test 1

Test 2: Configurar una sesión de cata

<b>Requisitos testeados</b>	RQ-09
<b>Precondiciones</b>	Usuario administrador identificado, se encuentra en el menú del administrador.
<b>CONTENIDO</b>	
<b>Acción</b>	Accede a: <i>Gestionar sesiones de cata</i> .
<b>Checkpoint 1</b>	El sistema debe mostrar un listado con las sesiones de cata que ya están dadas de alta y bajo este un formulario para introducir los datos de la nueva sesión.
<b>Acción</b>	El usuario rellena los datos de la sesión: nombre, identificador, muestra, selección de catadores y de atributos.
<b>Checkpoint 2</b>	El sistema debe mostrar un icono junto al botón de agregar catadores y agregar atributos indicando que se han configurado dichos puntos.
<b>Acción</b>	El usuario seleccionar el tipo de dominio de expresión con el que quiere trabajar: numérico, lingüístico, difuso, 2-tupla.
<b>Checkpoint 3</b>	El sistema debe actualizar las opciones correspondientes a cada opción.
<b>Acción</b>	El usuario pulsa el botón <i>Guardar</i> .
<b>Checkpoint 4</b>	La nueva sesión de cata debe aparecer en el listado que hay en la parte superior de la pantalla y desaparecerá el formulario.

Tabla 44: Test 2

Test 3: Completar una hoja de perfil

<b>Requisitos testeados</b>	RQ-11
<b>Precondiciones</b>	Usuario catador identificado, se encuentra en el menú del catador.
<b>CONTENIDO</b>	
<b>Acción</b>	Accede a: <i>Listar hojas de perfil pendientes.</i>
<b>Checkpoint 1</b>	El sistema debe mostrar un listado con las hojas de perfil que tiene asignadas dicho catador.
<b>Acción</b>	El usuario selecciona la hoja de perfil que desea completar.
<b>Checkpoint 2</b>	El sistema mostrará la hoja de perfil correspondiente.
<b>Acción</b>	El usuario asignar la valoración correspondiente a cada atributo y pulsa el botón <i>Guardar.</i>
<b>Checkpoint 3</b>	El sistema muestra un mensaje de advertencia al usuario.
<b>Acción</b>	El usuario confirma que desea guardar la información.
<b>Checkpoint 4</b>	El sistema graba la información en la base de datos, la pantalla de la hoja de perfil desaparecerá y volverá a mostrar la pantalla donde quedarán las hojas de perfil pendientes entre las que ya no estará la que se acaba de completar.

Tabla 45: Test 3

**Test 4: Clasificar una muestra de aceite**

<b>Requisitos testeados</b>	RQ-10
<b>Precondiciones</b>	Usuario administrador identificado, se encuentra en el menú de administración.
<b>CONTENIDO</b>	
<b>Acción</b>	Accede a: <i>Gestionar resultados</i> .
<b>Checkpoint 1</b>	El sistema debe mostrar una pantalla con tres bloques en función de estado de la sesión de cata.
<b>Acción</b>	El usuario selecciona la sesión de cata interesada y pulsa en <i>Obtener resultado</i> .
<b>Checkpoint 2</b>	El sistema mostrará un mensaje con el tipo de aceite que corresponderá a la muestra.
<b>Acción</b>	El usuario pulsará aceptar en el mensaje una vez recibida la información.
<b>Checkpoint 3</b>	El sistema moverá de categoría la sesión de cata afectada y pasará a mostrarse en la de sesiones de cata finalizadas.

Tabla 46: Test 4

**Test 5: Consultar una sesión de cata**

<b>Requisitos testeados</b>	RQ-05
<b>Precondiciones</b>	Usuario administrador identificado, se encuentra en el menú de administración.
<b>CONTENIDO</b>	
<b>Acción</b>	Accede a: <i>Gestionar muestras</i> .
<b>Checkpoint 1</b>	El sistema debe mostrar una pantalla con una tabla con

	todas muestras que hay dadas de alta.
<b>Acción</b>	El usuario selecciona la muestra interesada y pulsa en <i>Ver sesiones de cata</i> .
<b>Checkpoint 2</b>	El sistema mostrará bajo esta tabla una subtabla con todas las sesiones de cata donde interviene la muestra.
<b>Acción</b>	El usuario podrá seleccionar la hoja de perfil que desea consultar de la sesión de cata que desee y pulsar <i>Ver hoja de perfil</i> .
<b>Checkpoint 3</b>	El sistema mostrará la hoja de perfil seleccionada.

Tabla 47: Test 5

## 6.2. Resultados obtenidos

Mostramos la hoja de resultados de los test diseñados una vez ya realizados sobre el sistema:

TEST	RESULTADO
Test 1: Dar de alta una muestra de aceite.	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Test 2: Configurar una sesión de cata.	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK

---

Test 3: Completar una hoja de perfil.	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Checkpoint 4	OK
Test 4: Clasificar una muestra de aceite.	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK
Test 5: Consultar una sesión de cata.	
Checkpoint 1	OK
Checkpoint 2	OK
Checkpoint 3	OK

Tabla 48: Resultados obtenidos en los tests

## CAPÍTULO 4 - CONCLUSIONES

### 1. Conclusiones finales

---

El objetivo de este proyecto fin de carrera es realizar una aplicación de escritorio para la clasificación de las muestras de aceite de oliva que ofrezca una flexibilización a la hora de expresar las valoraciones de los atributos sensoriales del aceite de oliva a través de diferentes dominios de expresión. Para ello, se ha dado la posibilidad de trabajar con los siguientes dominios de expresión: *numérico* e *intervalar*, tanto para valores reales como naturales positivos, y a través de variables lingüísticas, modelando dicha información a través de diferentes metodologías como la lógica difusa, enfoque lingüístico difuso y la 2-tupla lingüística.

Con este proyecto cubrimos el vacío que existe actualmente en el mercado en cuanto a aplicaciones informáticas enfocadas a la evaluación sensorial del aceite de oliva. En este contexto, debido a la la incertidumbre e imprecisión de la información que se maneja en estos procesos de evaluación sensorial, se han incluido dominios de expresión lingüísticos, suavizando la elevada precisión que se le exige al catador para valorar las características sensoriales del aceite en una escala no estructura de 10 cm de longitud, tal y como marca actualmente la normativa.

Con la aplicación desarrollada, en cada sesión de cata se podrá decidir qué dominio de expresión de los anteriormente descritos se va a utilizar para elaborar las hojas de perfil con las que trabajarán los catadores. Además, ésta se apoya en una base de datos que permitirá a los usuarios administradores poder consultar en cualquier momento las valoraciones realizadas por los diferentes catadores pertenecientes a las sesiones de cata, consultar la clasificación que se ha obtenido de la muestra de aceite de oliva, consultar la hoja de perfil agregada, ...

## 2. Trabajos futuros

---

Dado que nos encontramos en un tiempo en que cada vez llevamos a cabo más tareas a través de Internet una mejora de este proyecto es desarrollar una aplicación web a la que se pudiese acceder desde diversos dispositivos móviles, desarrollando para ello una interfaz flexible siguiendo las últimas tendencias en *responsive design*, utilizando para ello estándares web como HTML5 y CSS3.

Otra mejora posible es extender el campo de trabajo de esta aplicación a la evaluación de otros productos ya que en cualquier momento es posible editar atributos, usuarios participantes, muestras del producto, dominios de expresión, etc..

## BIBLIOGRAFÍA

- [1] L. Martínez, M. Espinilla, J. Liu, L.G. Pérez, P.J. Sánchez, An Evaluation Model with Unbalanced Linguistic Information: Applied to Olive Oil Sensory Evaluation. *Journal of Multiple-Valued Logic and Soft Computing*, vol. 15, issue 2, pp. 229-251, 2009.
- [2] J. Antes, L. Campen, U. Derigs, C Titze, and G.D.Wolle. A model-based decision support system for the evaluation of flight schedules for cargo airlines. *Decision Support Systems*, 22(4):307-323, 1998.
- [3] A. Anzaldúa-Morales and J.G. Brennan. La medición de la textura de frutas y verduras i. frutas y verduras frescas. *Tecnología de los alimentos*, 19(2):22-30, 1984.
- [4] B. Arfi. Fuzzy decision making in politics: a linguistic fuzzy-set approach (LFSA). *Political Analysis*, 13(1):23-56, 2005.
- [5] K.J. Arrow. *Social choice and individual values*. Yale University Press, New Haven, 1963.
- [6] K. Atanassov and G. Gargov. Interval valued intuitionistic fuzzy-sets. *Fuzzy Sets and Systems*, 31(3):343-349, 1989.
- [7] A. Bechara, D. Tranel, and H. Damasio. Characterization of the decision-making deficit of patients with ventromedial prefrontal cortex lesions. *Brain*, 123:2189-2202, 2005.
- [8] G. Beliakov, R. Mesiar, and L. Valaskova. Fitting generated aggregation operators to empirical data. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(2):219-236, 2004.
- [9] G. Beliakov and J. Warren. Appropriate choice of aggregation operators in fuzzy decision support systems. *IEEE Transactions on Fuzzy Systems*, 9(6):773-784, 2001.
- [10] D. Ben-Arieh and Z. Chen. Linguistic group decision-making: opinion aggregation and measures of consensus. *Fuzzy Optimization and Decision Making*, 5(4):371- 386, 2006.
- [11] D. Ben-Arieh and Z. Chen. Linguistic-labels aggregation and consensus measure for autocratic decision making using group recommendations. *IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans*, 36(3):558-568, 2006.
- [12] T. Bilgiç. Interval-valued preference structures. *European Journal of Operational Research*, 105(1):162-183, 1998.
- [13] D. Bouyssou, T. Marchant, M. Pirlot, P. Perny, and A. Tsoukia`s. *Evaluation and Decision Models: A critical perspective*. Kluwer Academic Publishers, 2000.
-

- [14] P. Burillo and H. Bustince. Construction theorems for intuitionistic fuzzy sets. *Fuzzy Sets and Systems*, 84(3):271-281, 1996.
- [15] H. Bustince, E. Barrenechea, and V. Mohedano. Intuitionistic fuzzy implication operators: an expression and main properties. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(3):387-406, 2004.
- [16] H. Bustince and P. Burillo. Mathematical analysis of interval-valued fuzzy relations: application to approximate reasoning. *Fuzzy Sets and Systems*, 113(2):205-219, 2000.
- [17] H. Bustince and P. Burillo. Perturbation of intuitionistic fuzzy relations. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(1):81-103, 2001.
- [18] T. Calvo, R. Mesiar, and R.R. Yager. Quantitative weight and aggregation. *IEEE Transactions on Fuzzy Systems*, 12(1):62-69, 2004.
- [19] C.H. Cheng and Y. Lin. Evaluating the best main battle tank using fuzzy decision theory with linguistic criteria evaluation. *European Journal of Operational Research*, 142(1):174-186, 2002.
- [20] H. Chernoff. *Elementary decision theory*. Dover Publications, New York, 1987.
- [21] F. Chiclana, F. Herrera, and E. Herrera-Viedma. Integrating three representation models in fuzzy multipurpose decision making based on fuzzy preference relations. *Fuzzy Sets and Systems*, 97(1):33-48, 1998.
- [22] D.H. Choi, B.S. Ahn, and S.H. Kim. Multicriteria group decision making under incomplete preference judgments: using fuzzy logic with a linguistic quantier. *International Journal of Intelligent Systems*, 22(6):641-660, 2007.
- [23] S.J. Chuu. Fuzzy multi-attribute decision-making for evaluating manufacturing flexibility. *Production Planning and Control*, 16(3):323-335, 2005.
- [24] R.T. Clemen. *Making Hard Decisions. An Introduction to Decision Analysis*. Duxbury Press, 1995.
- [25] V. Cutello and J. Montero. Hierarchies of aggregation operators. *International Journal of Intelligent Systems*, 9(11):1025-1045, 1994.
- [26] J. Dombi. *Fuzzy logic and soft computing*, chapter A general framework for the utility-based and outranking methods, pages 202-208. In: B.B. Bouchon (Ed.). World Scientific, London, 1995.
- [27] D. Dubois and H. Prade. *Fuzzy sets and systems: theory and applications*. Academic Press, New York, 1980.
- [28] D. Dubois and H. Prade. Rough fuzzy-sets and fuzzy rough sets. *International Journal of General Systems*, 13(2-3):191-209, 1990.
-

- 
- [29] R. Duncan and H. Raiffa. *Games and decision. Introduction and critical survey*. Dover Publications, New York, 1985.
- [30] J. Fodor and M. Roubens. *Fuzzy preference modelling and multicriteria decision support*. Kluwer Academic Publishers, Dordrecht, 1994.
- [31] G. Fu. A fuzzy optimization method for multicriteria decision making: an application to reservoir flood control operation. *Expert Systems with Applications*, 34(1):145-149, 2008.
- [32] R.A. Gheorghe, A. Bufardi, and P. Xirouchakis. Fuzzy multicriteria decision aid method for conceptual design. *Cirp Annals-Manufacturing Technology*, 54(1):151- 154, 2005.
- [33] D. Gómez and J. Montero. A discussion on aggregation operators. *Kybernetika*, 40(1):107-120, 2004.
- [34] S. Greco, B. Matarazzo, and R. Slowinski. Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research*, 129(1):1-47, 2001.
- [35] F. Herrera and E. Herrera-Viedma. Choice functions and mechanisms for linguistic preference relations. *European Journal of Operational Research*, 120(1):144-161, 2000.
- [36] F. Herrera and E. Herrera-Viedma. Linguistic decision analysis: steps for solving decision problems under linguistic information. *Fuzzy Sets and Systems*, 115(1):67-82, 2000.
- [37] F. Herrera, E. Herrera-Viedma, and J.L. Verdegay. A sequential selection process in group decision making with a linguistic assessment approach. *Information Sciences*, 85(4):223-239, 1995.
- [38] F. Herrera, L. Martínez, and P.J. Sánchez. Managing non-homogeneous information in group decision making. *European Journal of Operational Research*, 166(1):115-132, 2005.
- [39] E. Herrera-Viedma, S. Alonso, F. Chiclana, and F. Herrera. A consensus model for group decision making with incomplete fuzzy preference relations. *IEEE Transactions on Fuzzy Systems*, 15(5):863-877, 2007.
- [40] E. Herrera-Viedma, F. Chiclana, F. Herrera, and S. Alonso. A group decision-making model with incomplete fuzzy preference relations based on additive consistency. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*, 37(1):176-189, 2007.
- [41] E. Herrera-Viedma, F. Herrera, and F. Chiclana. A consensus model for multiperson decision making with different preference structures. *IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans*, 32(3):394-402, 2002.
-

- 
- [42] E. Herrera-Viedma, L. Martínez, F. Mata, and F. Chiclana. A consensus support system model for group decision-making problems with multi-granular linguistic preference relations. *IEEE Transactions on Fuzzy Systems*, 13(5):644-658, 2005.
- [43] E. Herrera-Viedma, E. Peis, J.M. Morales del Castillo, S. Alonso, and E.K. Anaya. A fuzzy linguistic model to evaluate the quality of web sites that store xml documents. *International Journal of Approximate Reasoning*, 46(1):226-253, 2007.
- [44] M. Hsu, M. Baht, R. Adolfs, D. Tranel, and C.F. Camarero. Neural systems responding to degrees of uncertainty in human decision-making. *Science*, 310:1680-1683, 2005.
- [45] M. Inuiguchi. Generalizations of rough sets: from crisp to fuzzy cases. *Lecture Notes in Artificial Intelligence*, 3066:26-37, 2004.
- [46] A. Jiménez, S. Ríos-Insúa, and A. Mateos. A decision support system for multiattribute utility evaluation based on imprecise assignments. *Decision Support Systems*, 36(1):65-79, 2003.
- [47] J. Kacprzyk. Group decision making with a fuzzy linguistic majority. *Fuzzy Sets and Systems*, 18:105-118, 1986.
- [48] J. Kacprzyk and M. Fedrizzi. *Multiperson decision making models using fuzzy sets and possibility theory*. Kluwer Academic Publishers, Dordrecht, 1990.
- [49] R.L. Keeney and H. Raiffa. *Decisions with multiple objectives: preferences and value tradeoffs*. Cambridge University Press, Cambridge, 1993.
- [50] S. Kundu. Min-transitivity of fuzzy leftness relationship and its application to decision making. *Fuzzy Sets and Systems*, 86:357-367, 1997.
- [51] J. Lu, G. Zhang, and D. Ruan. Intelligent multi-criteria fuzzy group decision-making for situation assessments. *Soft Computing*, 12(3):289-299, 2008.
- [52] J. Ma, D. Ruan, Y. Xu, and G. Zhang. A fuzzy-set approach to treat determinacy and consistency of linguistic terms in multi-criteria decision making. *International Journal of Approximate Reasoning*, 44(2):165-181, 2007.
- [53] A.C. Marquez and C. Blanchar. A decision support system for evaluating operations investments in high-technology business. *Decision Support Systems*, 41(2):472-487, 2006.
- [54] L. Martínez, M. Espinilla, and L. G. Pérez. A linguistic multigranular sensory evaluation model for olive oil. *International Journal of Computational Intelligence Systems*, 1(2):148-158, 2008.
- [55] L. Martínez, J. Liu, J.B. Yang, and F. Herrera. A multi-granular hierarchical linguistic model for design evaluation based on safety and cost analysis. *International Journal of Intelligent Systems*, 20(12):1161-1194, 2005.
-

- [56] J. Montero. Arrow's theorem under fuzzy rationality. *Behavioral Science*, 32(4):267-273, 1987.
- [57] H. Nurmi. *Non-conventional preference relations in decision making*, chapter Assumptions of individual preferences in theory of voting procedures, pages 142-155. In: J. Kacprzyk and M. Roubens (Eds.). Springer-Verlag, Berlín, 1988.
- [58] S.A. Orlovski. Decision-making with fuzzy preference relations. *Fuzzy Sets and Systems*, 1(3):155-167, 1978.
- [59] A.D. Pearman, J. Montero, and J. Tejada. Fuzzy multicriteria techniques: An application to transport planning. *Proceedings Lecture Notes In Computer Science*, 521:509-519, 2006.
- [60] R.A. Ribeiro. Fuzzy multiple attribute decision making: A review and new preference elicitation techniques. *Fuzzy Sets and Systems*, 78(2):155-181, 1996.
- [61] S. Ríos, C. Bielza, and A. Mateos. *Fundamentos de los Sistemas de Ayuda a la Decisión*. Ra-Ma, Madrid, 2002.
- [62] C. Romero. *Teoría de la decisión multicriterio: conceptos, técnicas y aplicaciones*. Alianza Universidad, Madrid, 1993.
- [63] M. Roubens. Some properties of choice functions based on valued binary relations. *European Journal of Operational Research*, 40(3):309-321, 1989.
- [64] M. Roubens. Fuzzy sets and decision analysis. *Fuzzy Sets and Systems*, 90(2):199-206, 1997.
- [65] M. Roubens and Ph. Vincke. *Preference modelling*. Springer-Verlag, Berlín/Heidelberg, 1985.
- [66] S. Saint and J.R. Lawson. *Rules for reaching consensus. A modern approach to decision making*. Jossey-Bass, San Francisco, 1994.
- [67] P. Salminen, J. Hokkanen, and R. Lahdelma. Comparing multicriteria methods in the context of environmental problems. *European Journal of Operational Research*, 104(3):485-496, 1998.
- [68] S. Seo and M. Sakawa. Fuzzy multiattribute utility analysis for collective choice. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):45-53, 1985.
- [69] E. Szmidszt and J. Kacprzyk. A consensus-reaching process under intuitionistic fuzzy preference relations. *International Journal of Intelligent Systems*, 18(7):837-852, 2003.
- [70] Y. Tang. A collective decision model involving vague concepts and linguistic expressions. *IEEE Transactions on Systems, Man, and Cybernetics. Part B: Cybernetics*, 38(2):421-428, 2008.
-

- 
- [71] T. Tanino. Fuzzy preference orderings in group decision making. *Fuzzy Sets and Systems*, 12(2):117-131, 1984.
- [72] T. Tanino. *On group decision making under fuzzy preferences*, pages 172-185. in: J. Kacprzyk and M. Fedrizzi, Eds., *Multiperson Decision Making Using Fuzzy Sets and Possibility Theory*. Kluwer Academic Publishers, 1990.
- [73] E. Triantaphyllou. *Multi-criteria decision making methods: a comparative study*. Kluwer Academic Publishers, Dordrecht/Boston/London, 2000.
- [74] R.C. Wang and S.J. Chuu. Group decision-making using a fuzzy linguistic approach for evaluating the flexibility in a manufacturing system. *European Journal of Operational Research*, 153(3):563-572, 2004.
- [75] Z.S. Xu. Goal programming models for obtaining the priority vector of incomplete fuzzy preference relation. *International Journal of Approximate Reasoning*, 36(3):261-270, 2004.
- [76] Z.S. Xu. A method based on linguistic aggregation operators for group decision making with linguistic preference relations. *Information Sciences*, 166(1-4):19-30, 2004.
- [77] Z.S. Xu. On compatibility of interval fuzzy preference relations. *Fuzzy Optimization and Decision Making*, 3(3):217-225, 2004.
- [78] Z.S. Xu. An approach based on the uncertain LOWG and induced uncertain LOWG operators to group decision making with uncertain multiplicative linguistic preference relations. *Decision Support Systems*, 41(6):488-499, 2006.
- [79] Z.S. Xu. Intuitionistic preference relations and their application in group decision making. *Information Sciences*, 177(11):2363-2379, 2007.
- [80] Z.S. Xu. Group decision making based on multiple types of linguistic preference relations. *Information Sciences*, 178(2):452-467, 2008.
- [81] R.R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183-190, 1988.
- [82] R.R. Yager. Aggregation operators and fuzzy system modelling. *Fuzzy Sets and Systems*, 67(2):129-145, 1993.
- [83] R.R. Yager. Aggregation of ordinal information. *Fuzzy Optimization and Decision Making*, 6(3):199-219, 2007.
- [162] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338-353, 1965.
- [163] L.A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. *Information Sciences, Part I, II, III*, 8,8,9:199-249,301-357,43-80, 1975.
- [164] M. Keith & M. Schincariol, Pro EJB 3: Java Persistence API, 2006
- [165] Bill Burke & Richard Monson-Haefel, "Enterprise JavaBeans 3.0", 2006.
-

# ANEXO I

## Manual de instalación

Este Anexo está dedicado a realizar las configuraciones necesarias para poner en marcha nuestra aplicación. Las únicas consideraciones previas que han de tenerse en cuenta son que durante todo este manual se ha supuesto que la unidad principal de disco duro es C:, que la unidad principal de disco flexible es D:.

Hemos de decir que nuestro sistema trabajara con un servidor basado en la plataforma Microsoft Windows, por lo que todas estas instalaciones y configuraciones hay que realizarlas teniendo en cuenta dicho Sistema Operativo.

Todo el material necesario para instalar y dejar operativo el servidor se encuentra disponible en el CD que acompaña a esta memoria. Vaya a dispositivo de CD-ROM D:\SistemaDeControlCataDeAceite y compruebe que se encuentran los siguientes archivos:

- ⊕ jre-7u21-windows-i586
- ⊕ mysql-installer-community-5.6.11.0
- ⊕ pfc-oil.rar
  - pfc-oil.jar
  - lib

### **Paso 1: Instalar Máquina Virtual Java**

Para comprobar la versión del entorno de ejecución instalado en el equipo ejecutar el siguiente test :

<http://www.java.com/es/download/installed.jsp>

Si no dispone de Java o la versión es anterior a la versión JRE 7, deberá instalar el archivo *jre-7u21-windows-i586.exe*. Para ello vaya al CD-ROM, ejecute dicho archivo y siga los sencillos pasos de instalación.

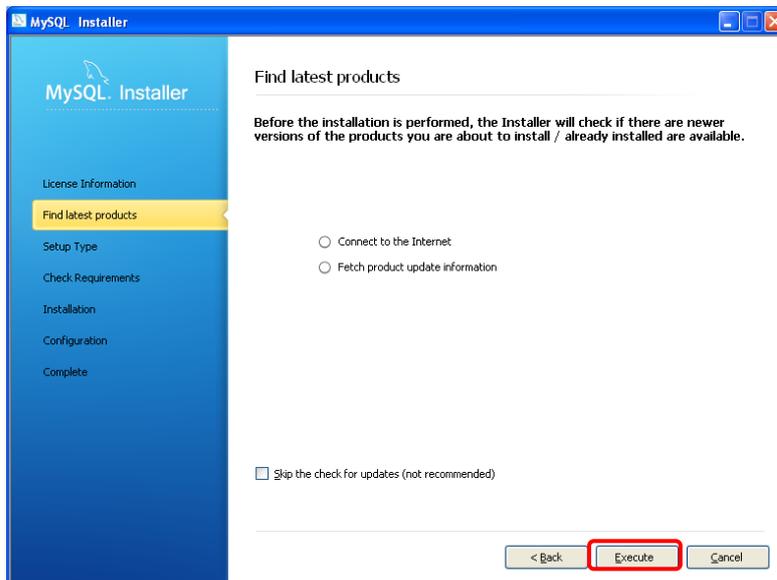
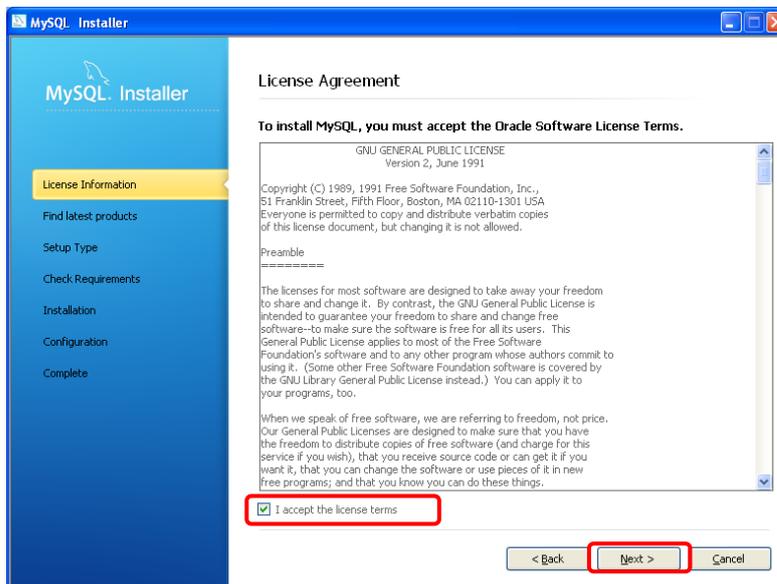
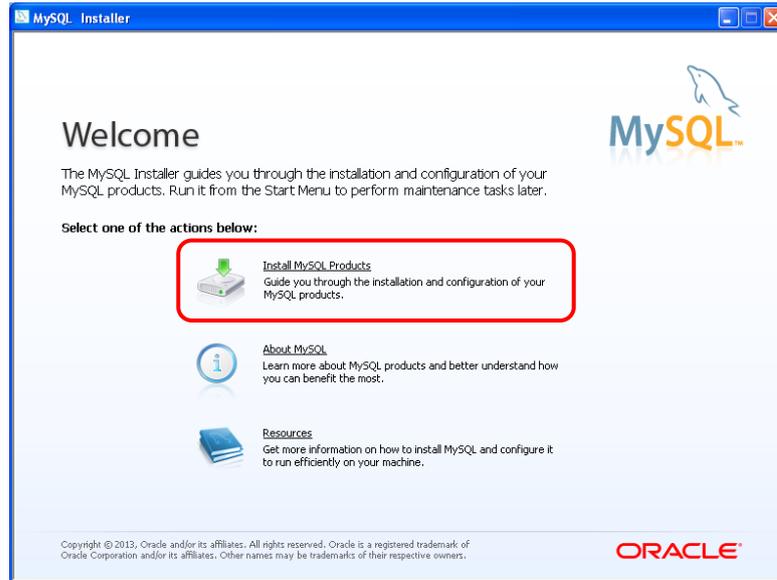
### **Paso 2: Instalar servidor de base de datos MySQL**

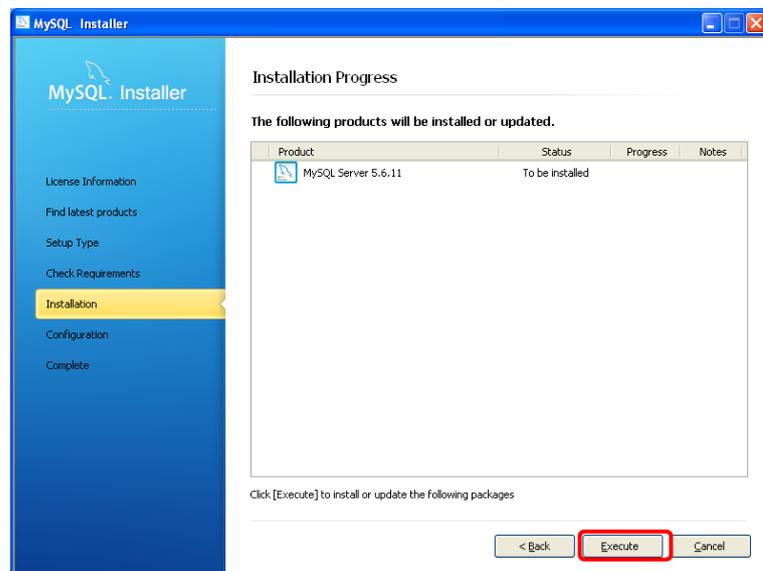
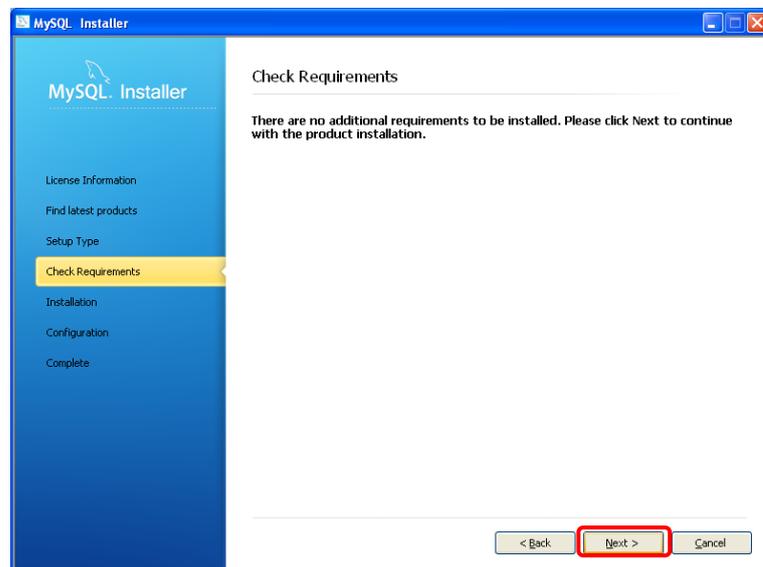
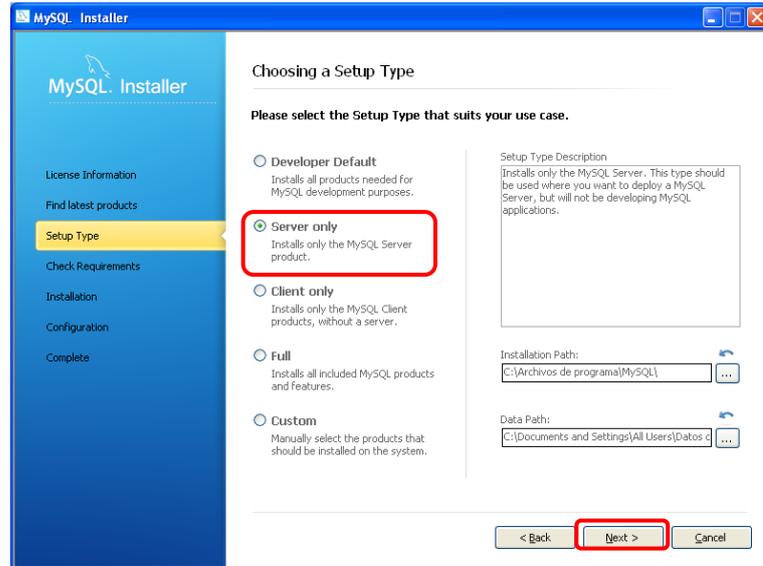
Ejecutamos el instalador mysql-installer-community-5.6.11.0, si nos detecta que falta configuraciones previas, instalar los siguientes ejecutables:

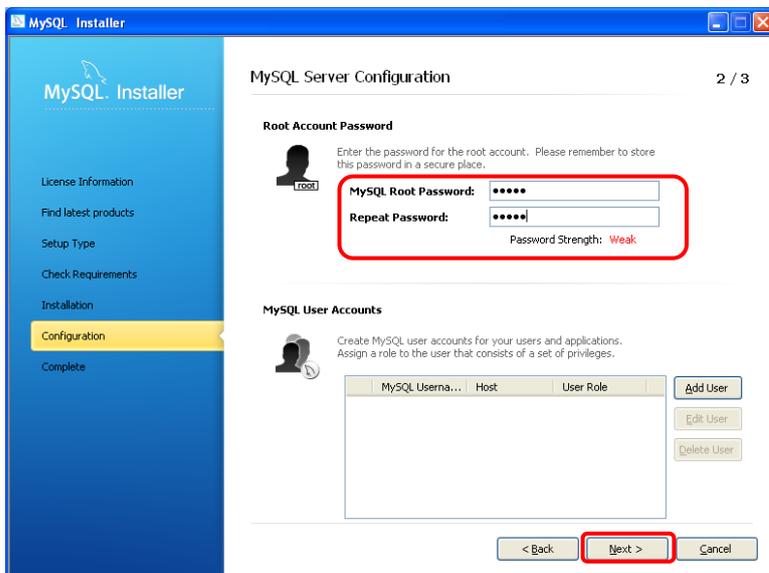
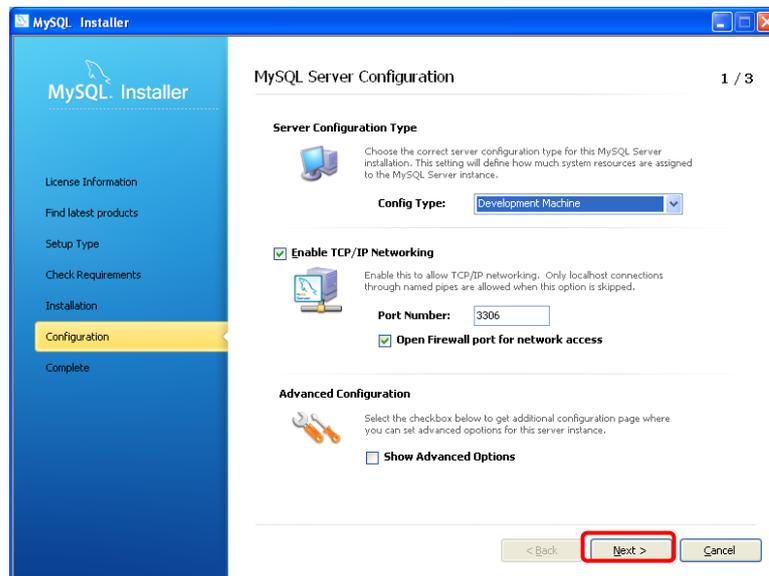
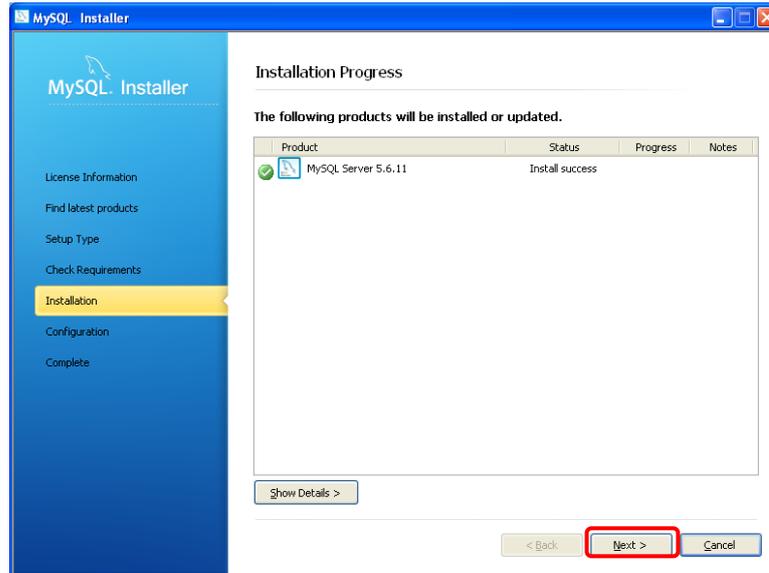
- dotNetFx40\_Client\_setup.exe
- vcredist\_x86.exe

Realizados los pasos anteriores, realizar la instalación tal y como se muestra en las siguientes pantallas:

---







Introducir una contraseña para el usuario root: **admin**

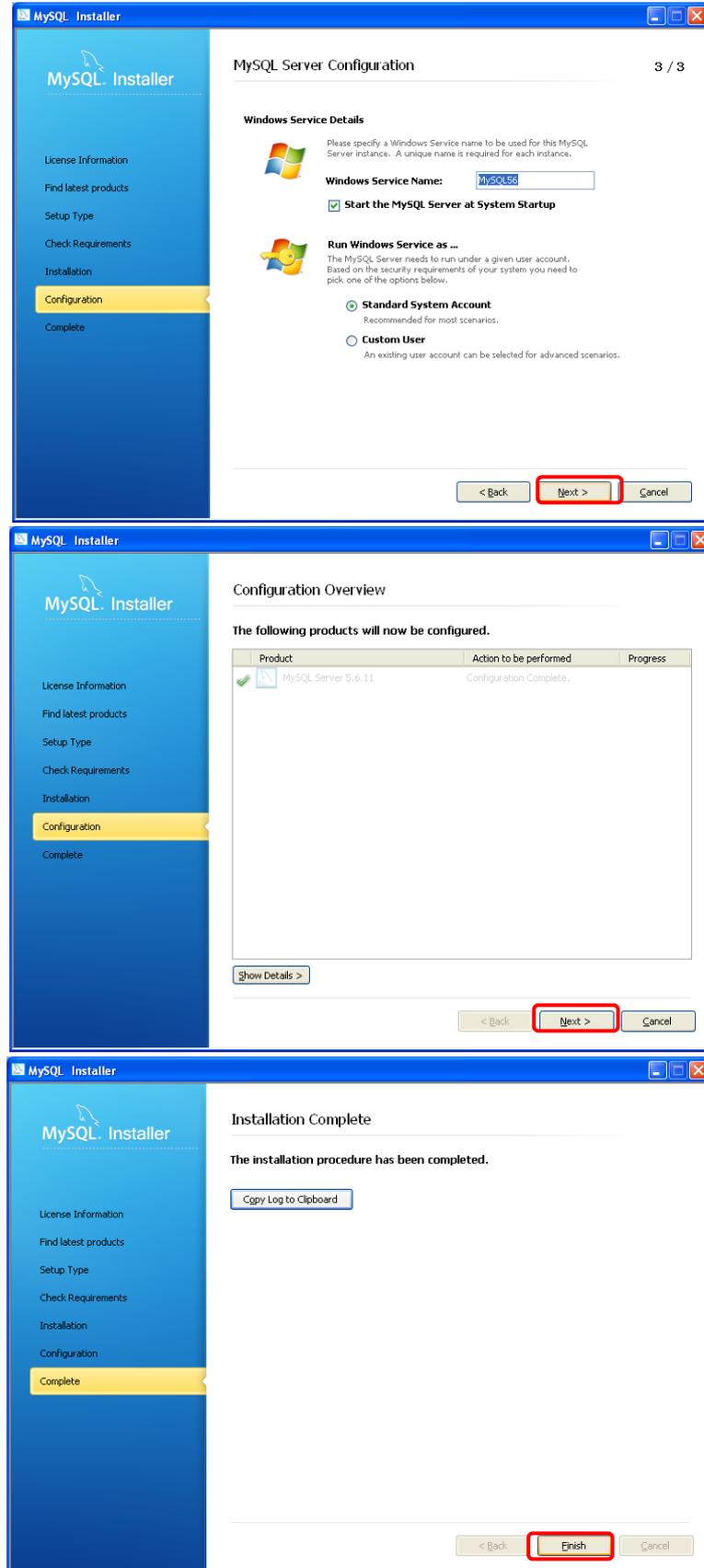
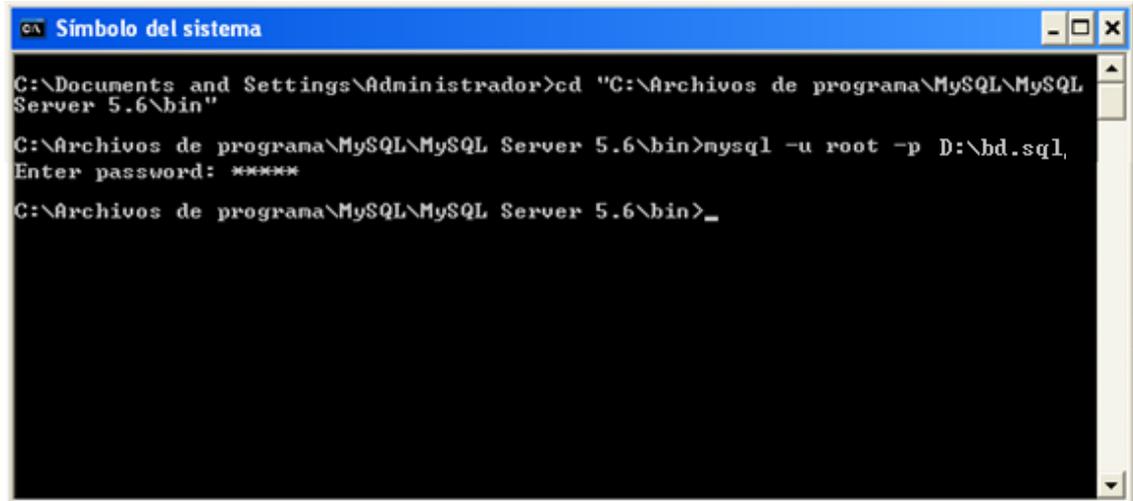


Ilustración 70: Instalación del servidor de bases de datos MySQL

Instalado el servidor de base de datos, llega el momento de ejecutar el script que instala la base de datos desarrollada para la aplicación. Para ello introduciremos en el símbolo de sistema los siguientes comandos:



```
Símbolo del sistema
C:\Documents and Settings\Administrador>cd "C:\Archivos de programa\MySQL\MySQL Server 5.6\bin"
C:\Archivos de programa\MySQL\MySQL Server 5.6\bin>mysql -u root -p D:\bd.sql
Enter password: *****
C:\Archivos de programa\MySQL\MySQL Server 5.6\bin>_
```

Ilustración 71: Ejecución del script que contiene la base de datos

La contraseña que tenemos que introducir es '**admin**' tal y como indicamos durante la instalación.

### Paso 3: Puesta en marcha de la aplicación

Ya solo nos queda copiar el directorio pfc-oil.rar en el equipo y hacer doble clic sobre el fichero .jar que hay dentro para ejecutar la aplicación.

# Anexo II

## Manual de usuario: Administrador

## 1. Gestión de usuarios catadores / administradores

Iniciamos sesión en el sistema como un usuario administrador y seleccionamos la opción de *Gestión de usuarios*, bien desde el menú principal o desde el icono que hay en la pantalla.



Figura 72: Gestión de usuarios

### a. *Alta de usuarios*

Rellenamos la información que nos pide el formulario y pulsamos en *Guardar*.

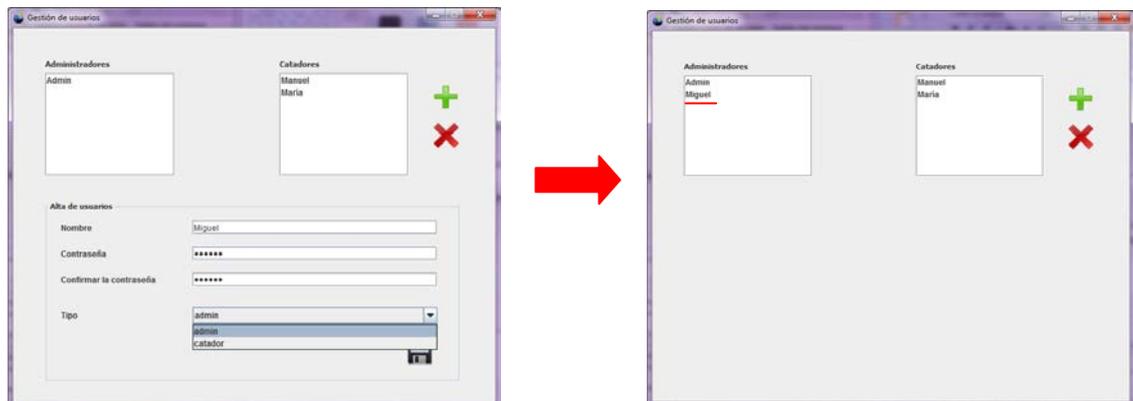


Figura 73: Alta de usuarios

A la hora de introducir la contraseña del usuario, el sistema exigirá que tenga como mínimo 6 caracteres y que se introduzca por duplicado de forma correcta.

Para volver a introducir un usuario pulsar en el botón *Añadir usuario* (cruz verde) que está en la parte superior derecha de la pantalla.

### b. Baja de usuarios

En primer lugar seleccionamos el nombre del usuario que se desea eliminar y pulsamos posteriormente en el botón *Eliminar usuario* (equis roja).

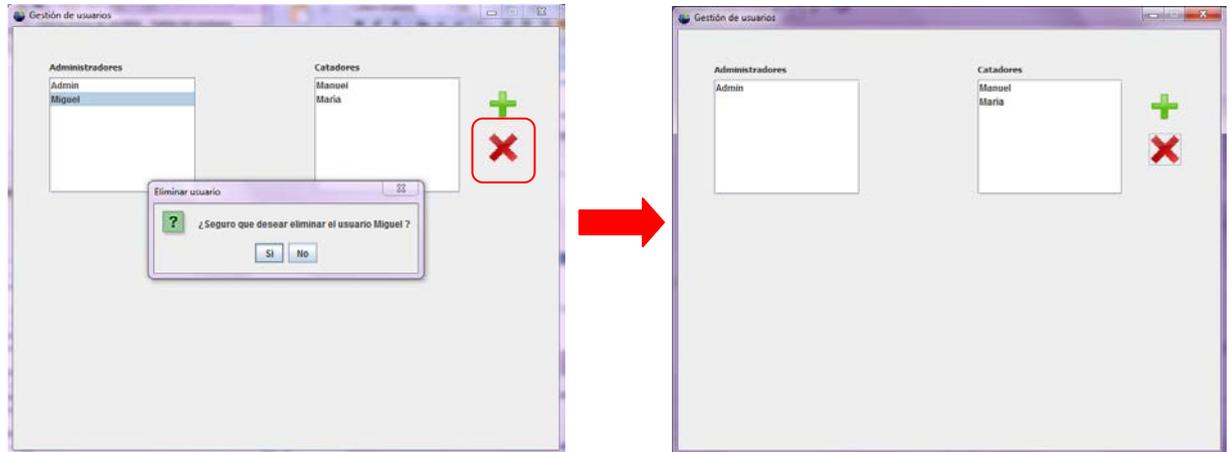


Figura 74: Baja de usuarios

La aplicación en ningún momento nos dejará eliminar un usuario si hay seleccionado más de uno a la vez.

## 2. Gestión de atributos

Iniciamos sesión en el sistema como un usuario administrador y seleccionamos la opción de *Gestión de atributos*, bien desde el menú principal o desde el icono que hay en la pantalla.

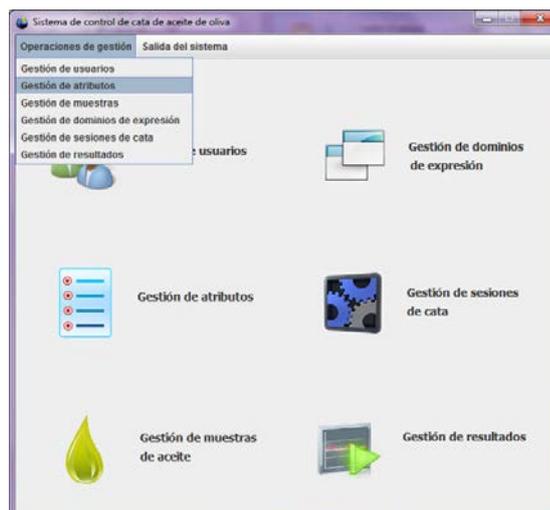


Figura 75: Gestión de atributos

### a. Alta de atributos

Rellenamos el nombre del atributo, seleccionamos el tipo (negativo o positivo) y pulsamos en *Guardar*.

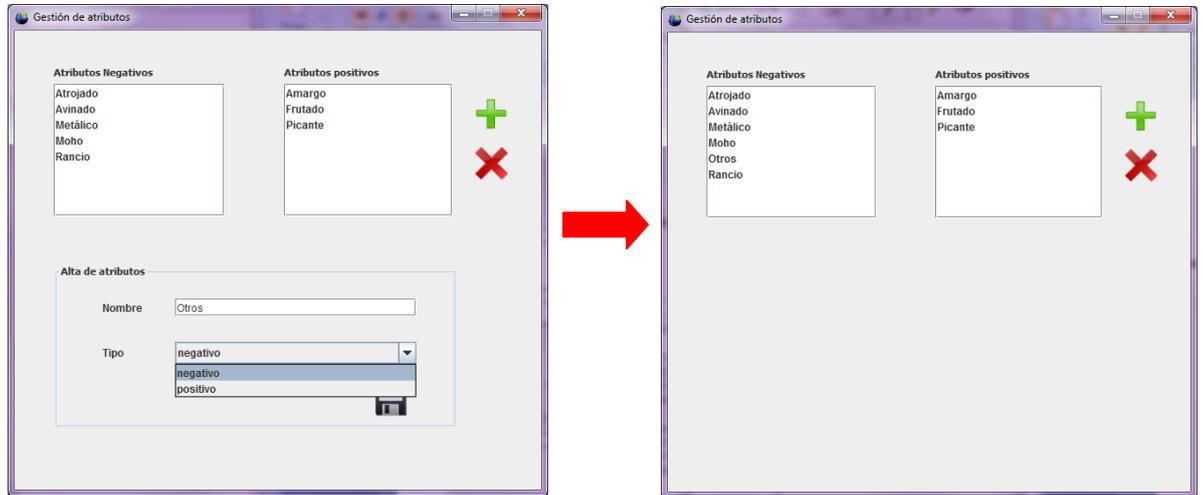


Figura 76: Alta de atributos

Para volver a introducir un atributo pulsar en el botón *Añadir atributo* (cruz verde) que está en la parte superior derecha de la pantalla.

### b. Baja de atributos

Seleccionamos el nombre del atributo que se desea eliminar y a continuación pulsamos en el botón *Eliminar usuario* (equis roja).

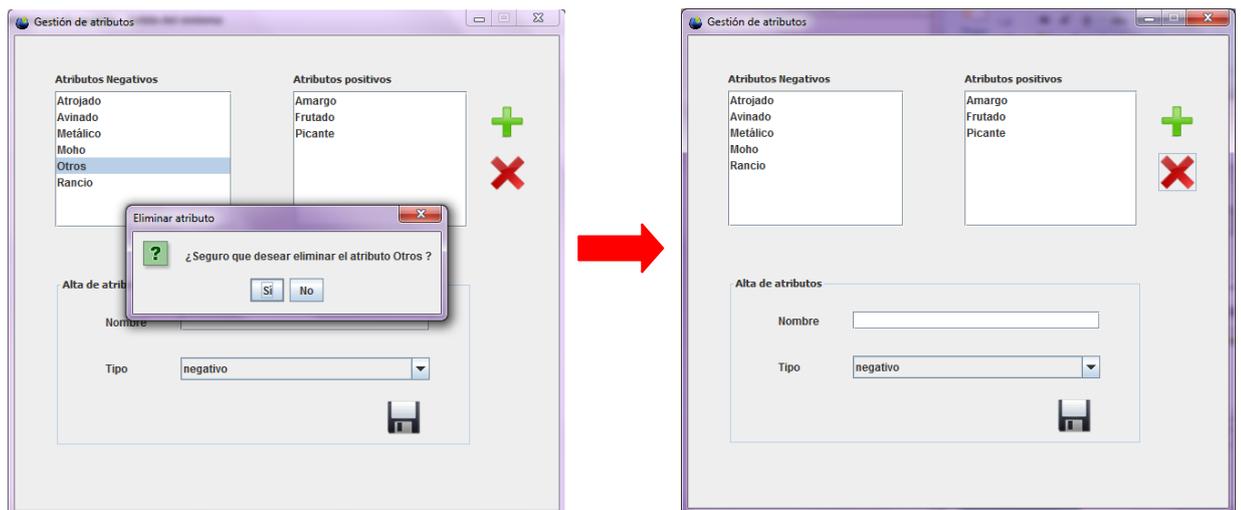


Figura 77: Baja de atributos

La aplicación en ningún momento nos dejará eliminar un atributo si hay seleccionado más de uno a la vez.

#### 4. Gestión de muestras de muestras de aceite de oliva

Iniciamos sesión en el sistema como un usuario administrador y seleccionamos la opción de *Gestión de muestras*, bien desde el menú principal o desde el icono que hay en la pantalla.

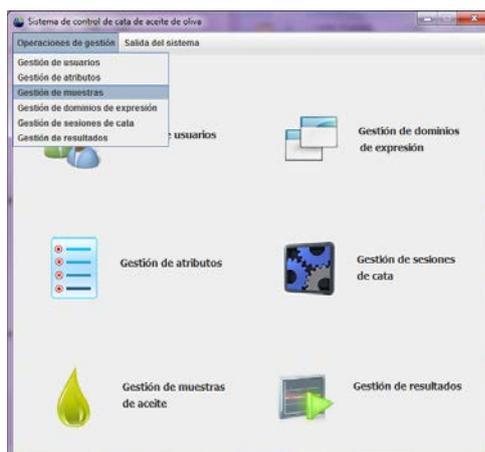


Figura 78: Gestión de muestras

##### a. *Alta de muestras*

Para la creación de una nueva muestra introducimos su nombre, identificador y el nombre del cliente propietario de la misma y pulsamos en *Guardar*.



Figura 79: Alta de muestras de aceite

Para volver a introducir una muestra pulsar en el botón *Añadir muestra* (cruz verde) que está en la parte superior derecha de la pantalla.

### b. Baja de muestras

Seleccionamos la muestra que se desea eliminar y pulsamos en el botón *Eliminar muestra*.

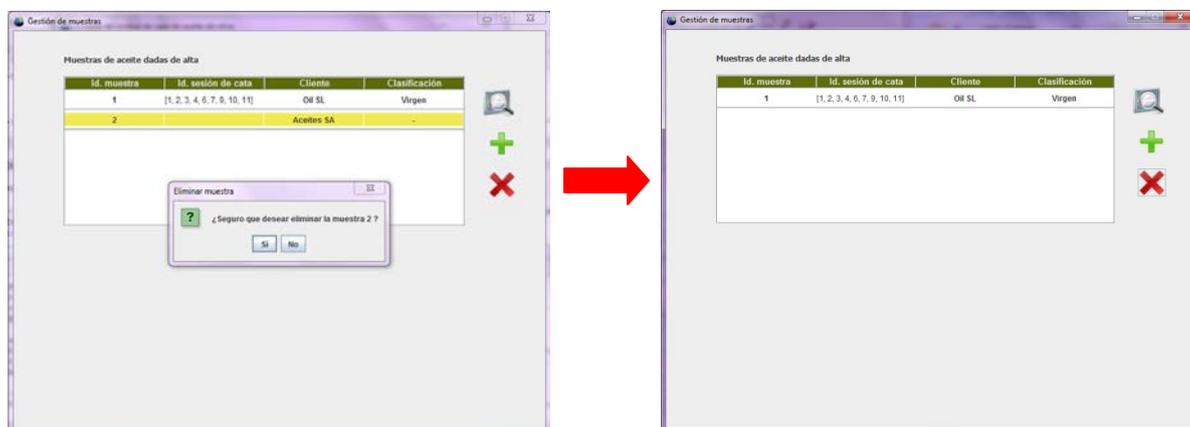
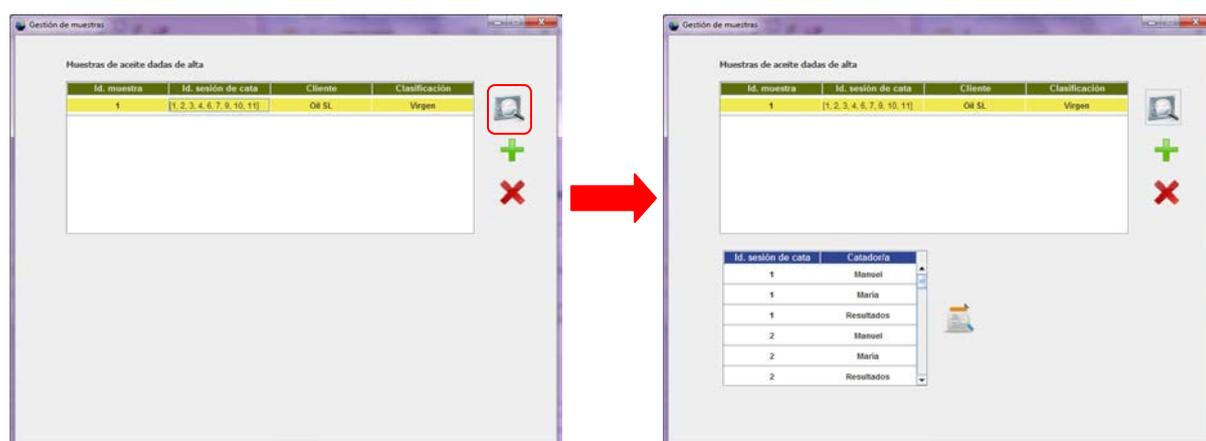


Figura 80: Baja de muestras de aceite

Si la muestra está asociada a alguna sesión de cata, no podremos eliminarla.

### c. Consultar sesiones de cata asociadas a una muestra de aceite

Para saber a cuántas sesiones de cata está asociada una muestra de aceite, debemos seleccionar la muestra interesada y pulsar en el botón *Ver sesiones de cata*. A continuación se mostrará bajo la tabla principal una subtabla con las sesiones de cata y los catados que participan en ella.



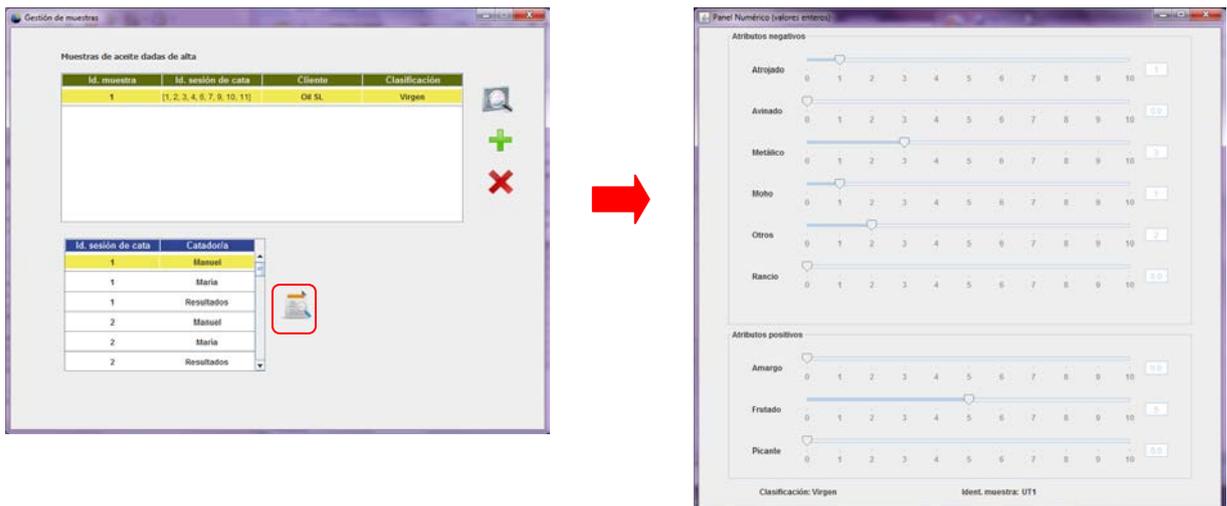


Figura 81: Consultar las sesiones de cata que están asociadas a una determinada muestra de aceite

Si deseamos conocer la valoración de cada catador en cada sesión de cata, seleccionamos la línea correspondiente y pulsamos en *Ver hoja de perfil*.

**d. Consultar hoja de perfil agregada**

Para consultar la hoja de perfil agregada seguiremos los mismos pasos que en el apartado anterior solo que esta vez seleccionaremos la opción de Resultados.

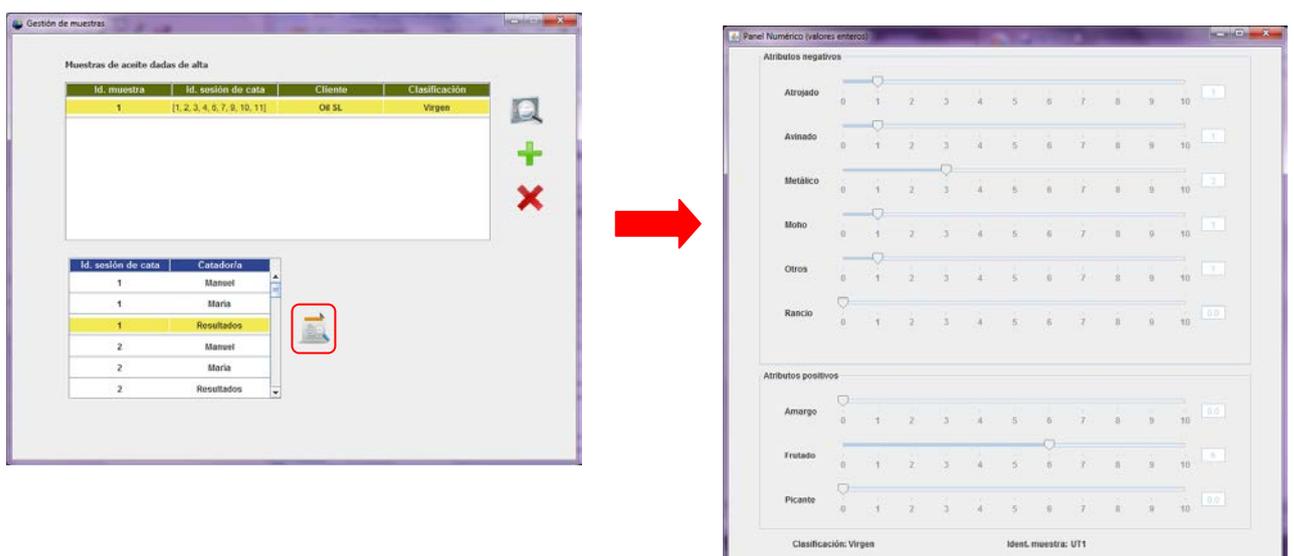


Figura 82: Consultar hoja de perfil agregada

## 5. Gestión de dominios de expresión

Iniciamos sesión en el sistema como un usuario administrador y seleccionamos la opción de *Gestión de dominios de expresión*, bien desde el menú principal o desde el icono que hay en la pantalla.

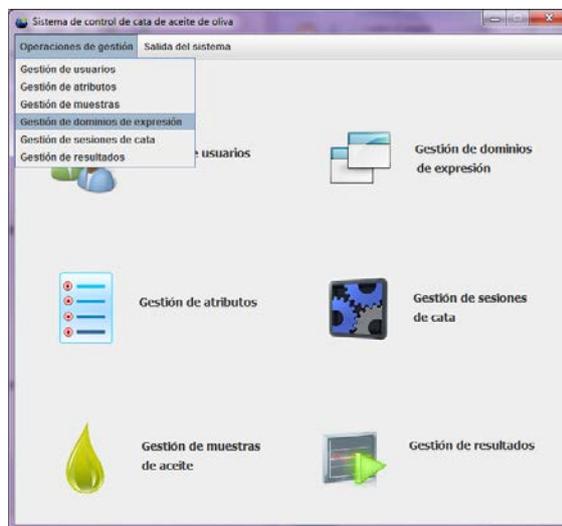


Figura 83: Gestión de dominios de expresión

### a. *Dominios de expresión numéricos*

#### i. *Alta de rangos numéricos*

Para añadir un nuevo rango, rellenamos en el formulario el valor mínimo y máximo y pulsamos en *Guardar*.

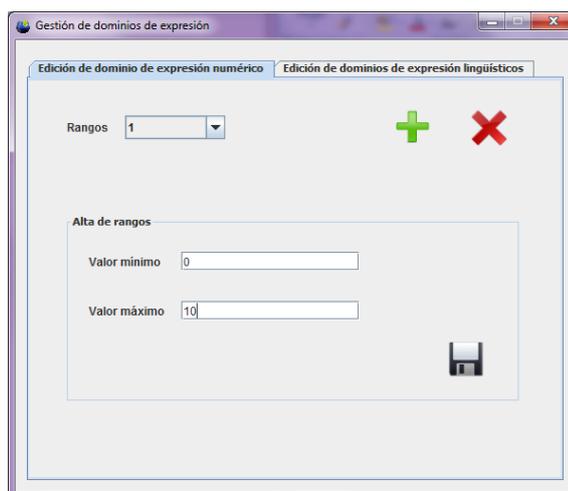


Figura 84: Alta de rangos numéricos

Para dar de alta otro rango, pulsamos en *Añadir rango numérico* (cruz verde situada en la parte superior derecha).

### ii. Baja de rangos numéricos

Si lo que deseamos es eliminar un rango, debemos seleccionar entre los existentes el que vamos suprimir y pulsamos en *Eliminar Rango*.

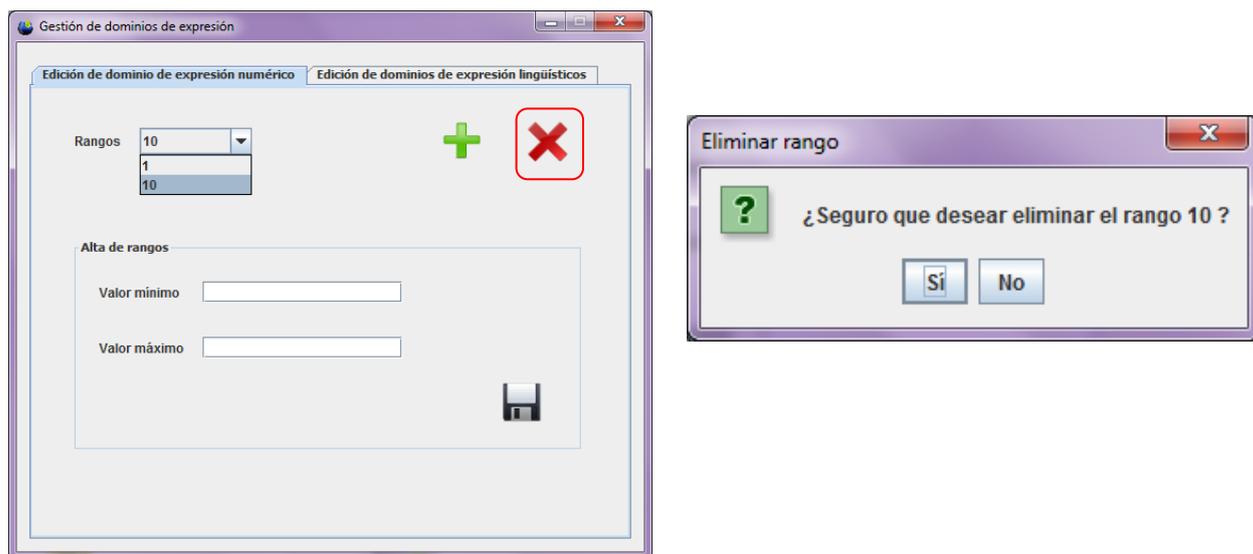


Figura 85: Baja de rango numérico

No podremos eliminar rangos que estén siendo usados por sesiones de cata.

## b. Dominios de expresión lingüísticos

### i. Etiquetas lingüísticas

Para añadir un juego de etiquetas lingüísticas tenemos que seleccionar el tipo de dominio, en este caso lingüístico, el número de etiquetas que formará el juego y pulsar en *Añadir juego de etiquetas*. Introducimos los nombres de las etiquetas pulsamos en *Guardar*.

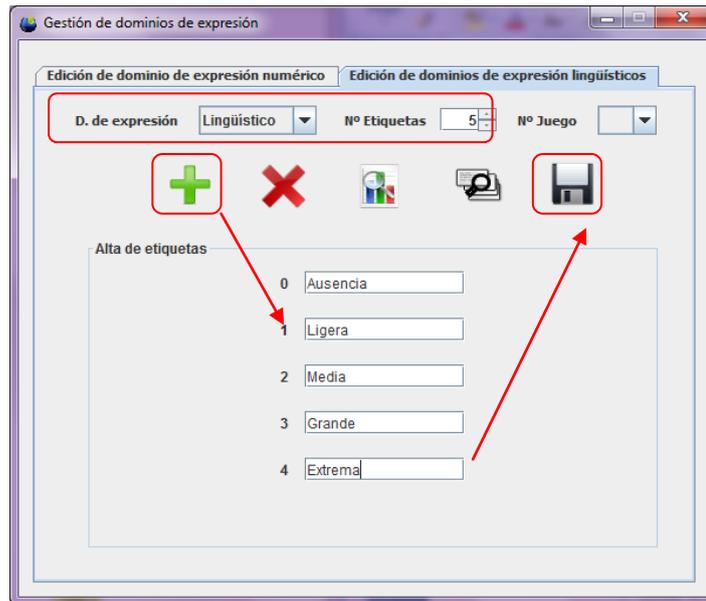


Figura 86: Alta de un juego con 5 etiquetas lingüísticas

Para eliminar un juego de etiquetas tenemos que seleccionar el tipo de dominio de expresión, el número de etiquetas por el que está formado y el número de juego del que se trata. Realizadas estas selecciones pulsamos en *Eliminar juego de etiquetas*.

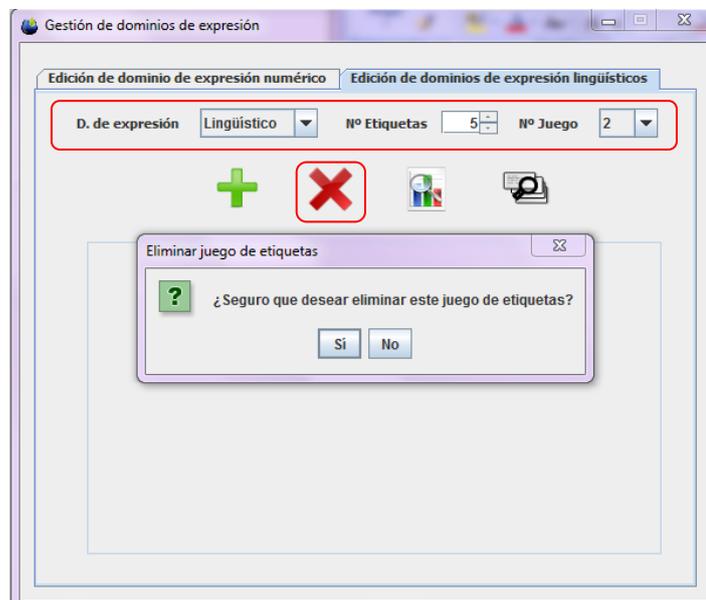


Figura 87: Baja de un juego de etiquetas

### ii. Etiquetas difusas

El proceso de creación es igual al de las etiquetas difusas, solo que hay que añadir la información semántica para cada etiqueta.

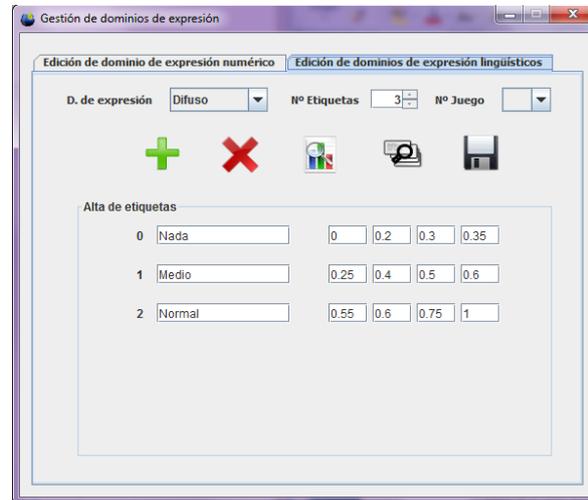


Figura 88: Alta de un juego con 3 etiquetas difusas

### iii. Etiquetas 2-tupla

Se trabaja de la misma forma que con las etiquetas lingüísticas, la única diferencia es que solo podremos crear juegos con un número de etiquetas impar.

### iv. Consultar escalas existentes

Si queremos conocer todas las escalas que hay en el sistema pulsamos en el botón *Mostrar las escalas creadas*.

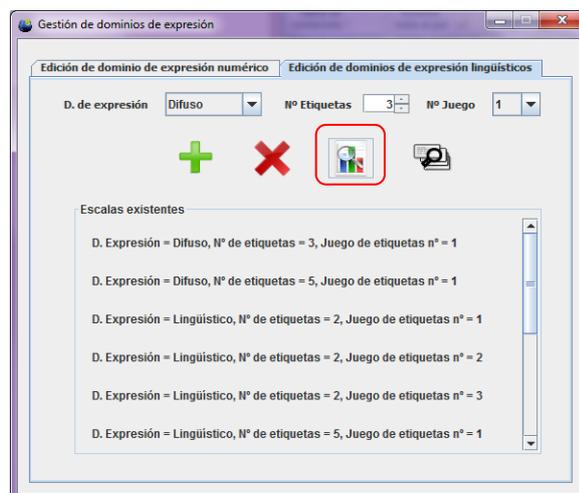


Figura 89: Consultar las escalas existentes en el sistema

v. Consultar las etiquetas de una determinada escala

Para conocer con más detalle la información de cada juego de etiquetas pulsar en *Mostrar juego de etiquetas*.

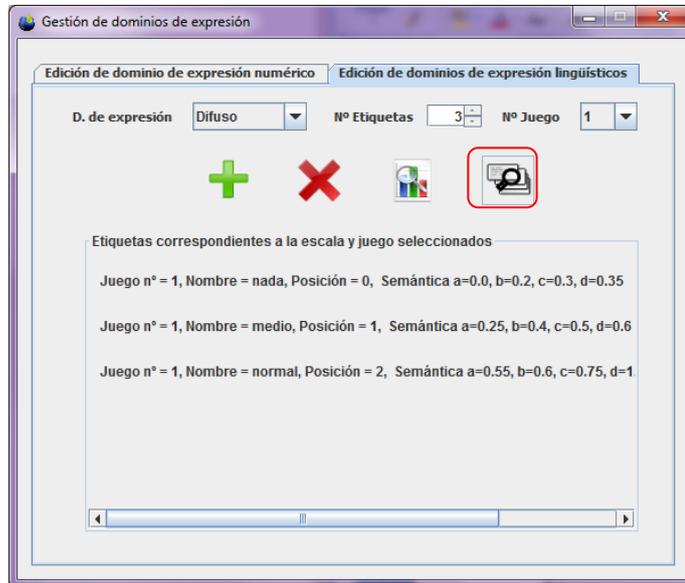


Figura 90: Consultar las etiquetas de un determinado juego y escala

6. Gestión de sesiones de cata

a. *Alta de sesiones de cata*

Para crear una nueva sesión de cata es obligatorio rellenar todos los campos que se encuentran en el formulario. Realizado ésto, pulsamos en *Guardar*.

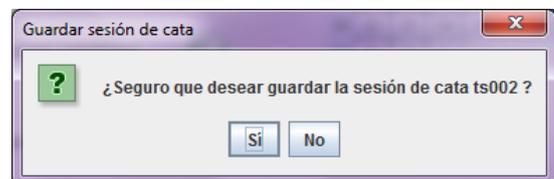
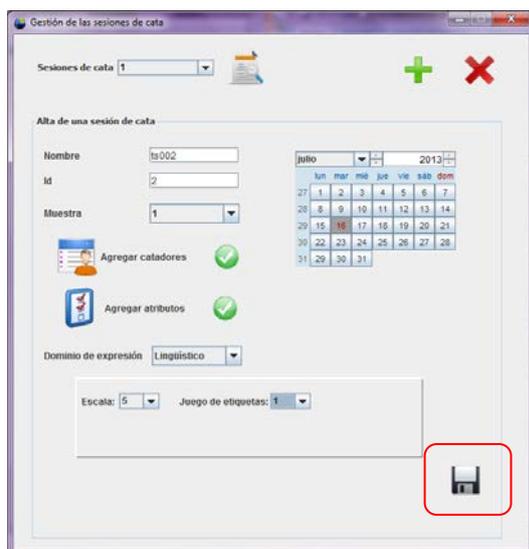


Figura 91: Alta de una sesión de cata

### b. Baja de sesiones de cata

Para dar de baja una sesión de cata solo tenemos que seleccionar el identificador correspondiente a la sesión pulsar en *Eliminar sesión de cata*.

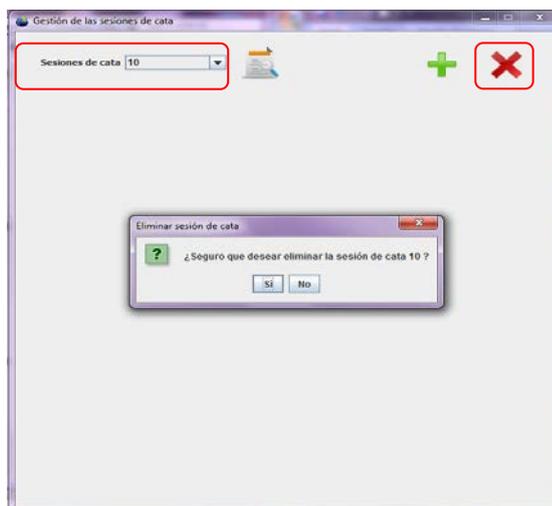


Figura 92: Eliminar una sesión de cata

## 7. Gestión de resultados

Iniciamos sesión en el sistema como un usuario administrador y seleccionamos la opción de *Gestión de resultados*, bien desde el menú principal o desde el icono que hay en la pantalla.

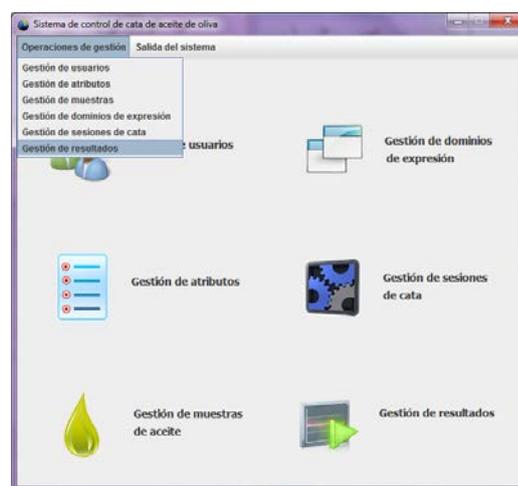


Figura 93: Gestión de resultados

### a. Consultar estado de las sesiones de cata

Las sesiones de cata se pueden encontrar en los siguientes estados:

- ✗ Pendientes de evaluar por los catadores.
- ✗ Pendientes de agregar valoraciones.
- ✗ Finalizadas.

Basta con buscar el identificador de la sesión en las tres categorías existentes.

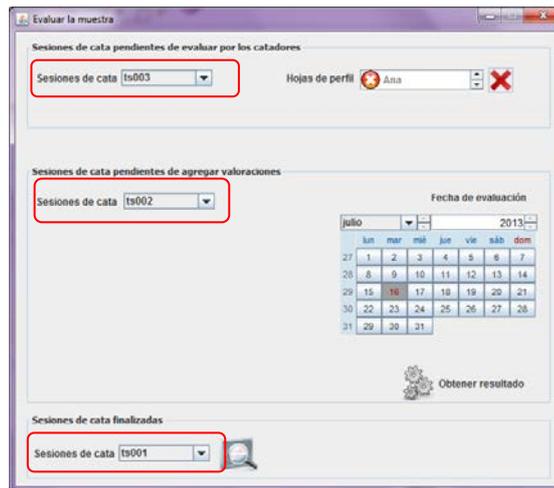


Figura 94: Estado de las sesiones de cata

### b. Eliminar un catador de una sesión de cata

Solo podremos eliminar un catador de una sesión que está en la categoría de "Pendientes de evaluar por los catadores" y debe de cumplir como condiciones que el catador no haya evaluado ya la hoja de perfil y que no sea el único catador asociado a la sesión de cata.

En el caso de cumplir dichas condiciones, lo único que debemos hacer es seleccionar el nombre del catador y pulsar en el botón *Eliminar hoja de perfil pendiente*.

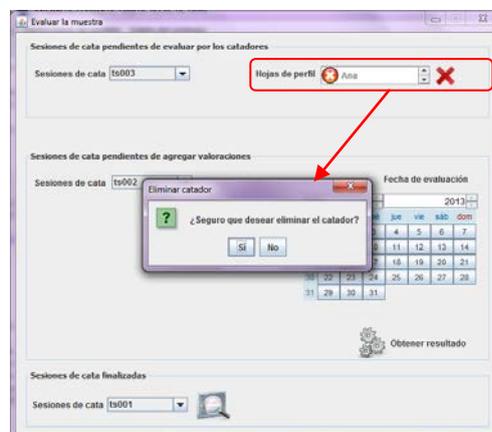


Figura 95: Eliminar catador/a de una sesión de cata

### c. Obtener clasificación de la muestra de aceite

Para obtener la clasificación de una muestra de aceite de oliva nos tenemos que dirigir a la categoría de "Sesiones de cata pendientes de agregar valoraciones". Allí seleccionamos el identificador de la sesión interesada y pulsamos en el botón *Obtener resultado*.

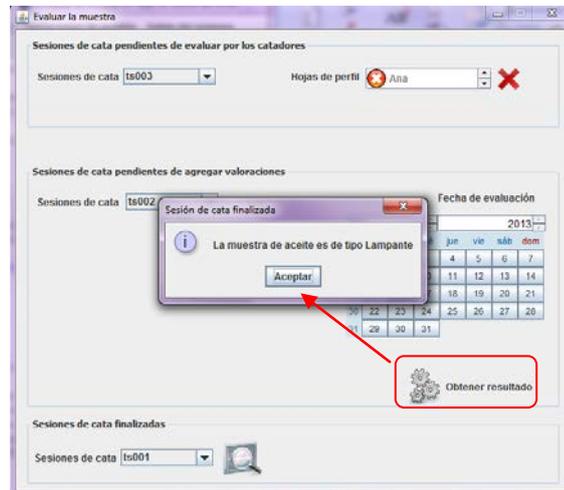


Figura 96: Clasificación de la muestra de aceite de oliva

### d. Consultar resultados de las sesiones de cata finalizadas

Si deseamos conocer una información resumida de los resultados obtenidos en una sesión de cata nos tenemos que dirigir a la categoría "Sesiones de cata finalizadas", seleccionamos el identificador de la cata interesada y pulsamos en el botón *Mostrar medianas*.

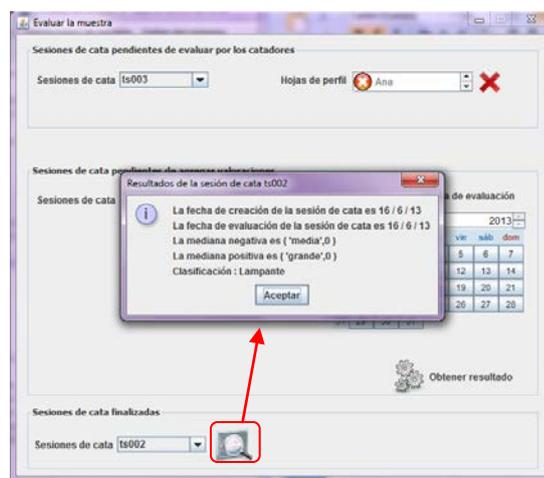


Figura 97: Resultados de las sesiones de cata finalizadas

## 8. Salida del sistema. Cambio de usuario

Para cambiar de usuario sin cerrar la aplicación, seleccionamos en el menú principal la opción *Salir del sistema* --> *Cerrar sesión*.

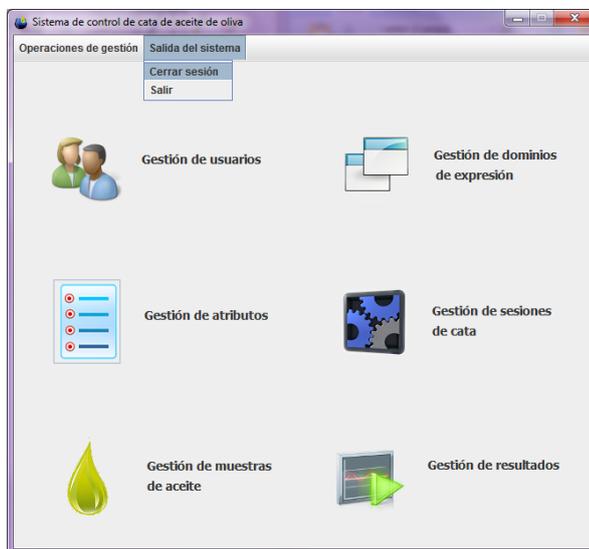


Figura 98: Cerrar sesión

Si por el contrario se desea salir definitivamente de éste, podemos hacerlo de dos formas:

- Mediante el menú principal: *Salida del sistema* --> *Salir*.
- O pulsando el botón cerrar de la pantalla.

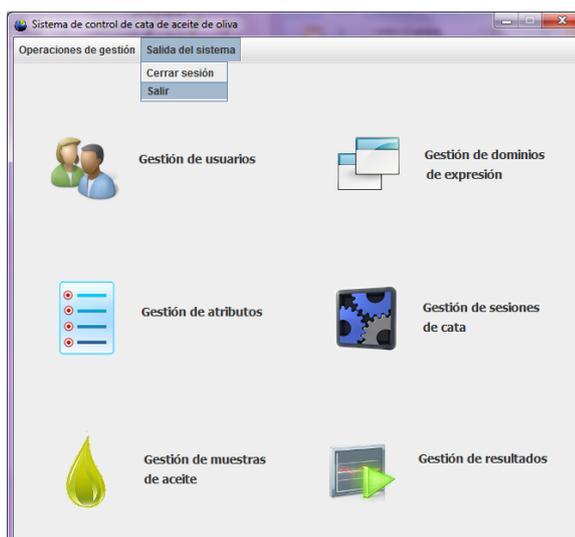


Figura 99: Cerrar la aplicación

# Anexo III

## Manual de usuario: Catador

## 1. Modificar contraseña

Una vez iniciada sesión como usuario catador, pulsamos en la opción *Cambio de contraseña* a través del menú principal o del icono ubicado en la parte superior derecha de la pantalla.

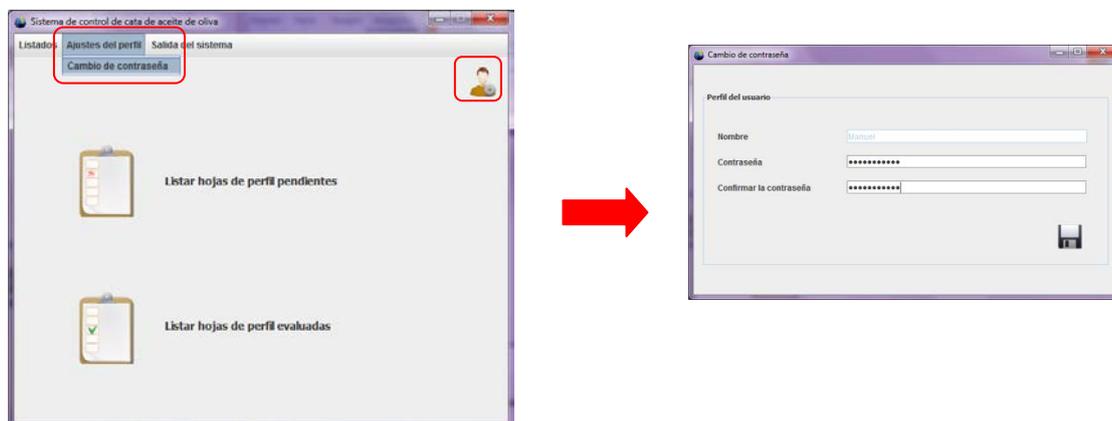
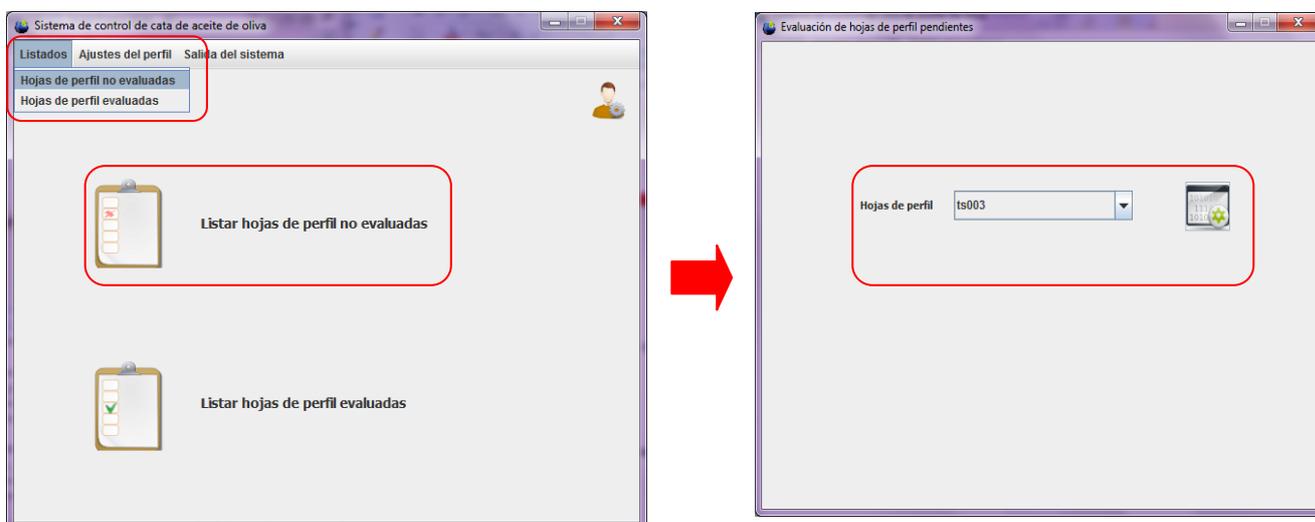


Figura 100: Modificar contraseña de un catador

El usuario introduce la nueva contraseña por duplicado y pulsa en *Guardar*.

## 2. Evaluar hojas de perfil pendientes

Para realizar esta operación tendríamos que seleccionar la opción de *Listar hojas de perfil no evaluadas*, elegimos el identificador de la sesión de cata interesada y pulsamos en el botón *Evaluar hojas de perfil*. Una vez realizada la valoración de la hoja de perfil pulsamos en *Guardar*.



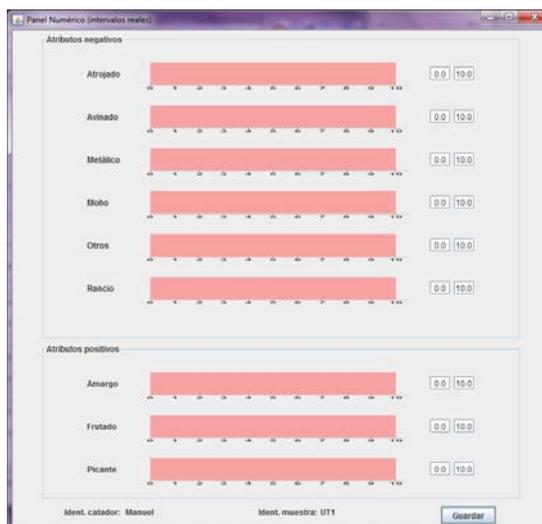


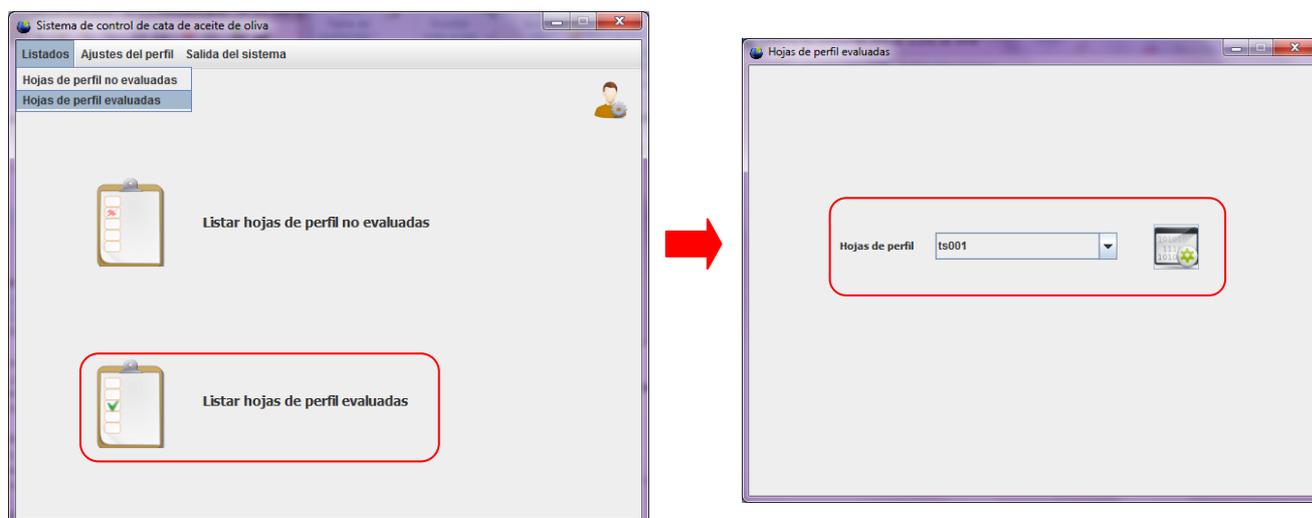
Figura 101: Consultar hojas de perfil sin evaluar

### 3. Consultar y/o modificar hojas de perfil evaluadas

Para ello tendríamos que seleccionar la opción de *Listar hojas de perfil evaluadas*, elegimos el identificador de la sesión de cata interesada y pulsamos en el botón *Consultar hojas de perfil evaluadas*.

Nos podemos encontrar con dos situaciones:

- ⊕ La sesión de cata aún no haya finalizado, la hoja de perfil sigue activa y el catador tiene la posibilidad de modificar sus valoraciones.
- ⊕ La sesión de cata ya ha finalizado y el catador ya solo podría consultar sus valoraciones pero no modificarlas.



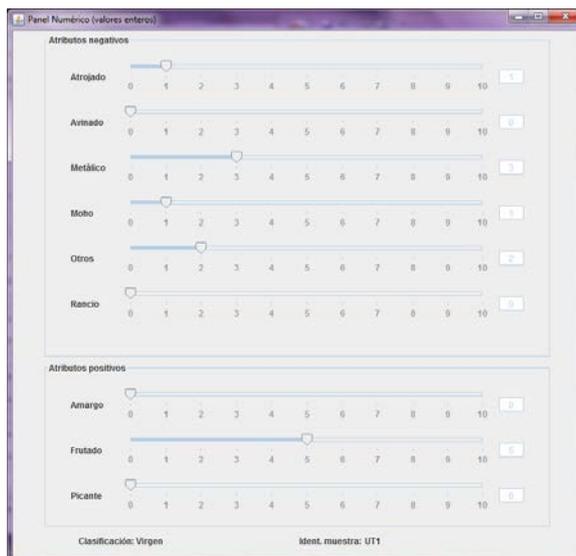


Figura 102: Consultar hojas de perfil evaluadas

#### 4. Salida del sistema. Cambio de usuario

Para cambiar de usuario sin cerrar la aplicación, seleccionamos en el menú principal la opción *Salir del sistema* --> *Cerrar sesión*.

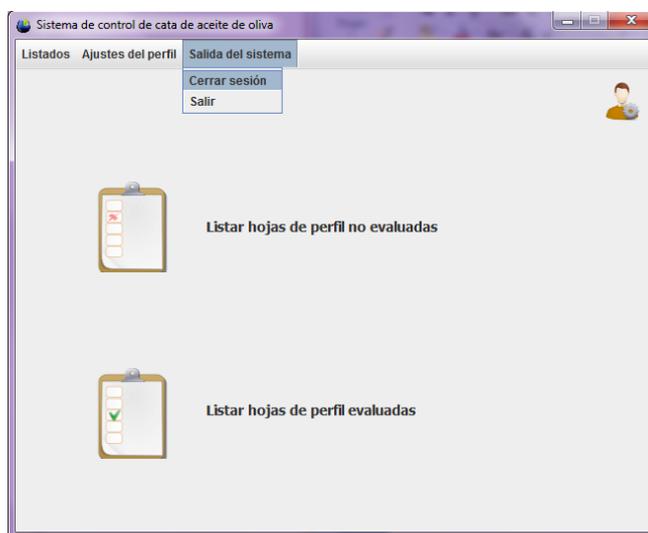


Figura 103: Cerrar sesión catador

Si por el contrario se desea salir definitivamente de éste, podemos hacerlo de dos formas:

- Mediante el menú principal: *Salida del sistema* --> *Salir*.
- O pulsando el botón cerrar de la pantalla.

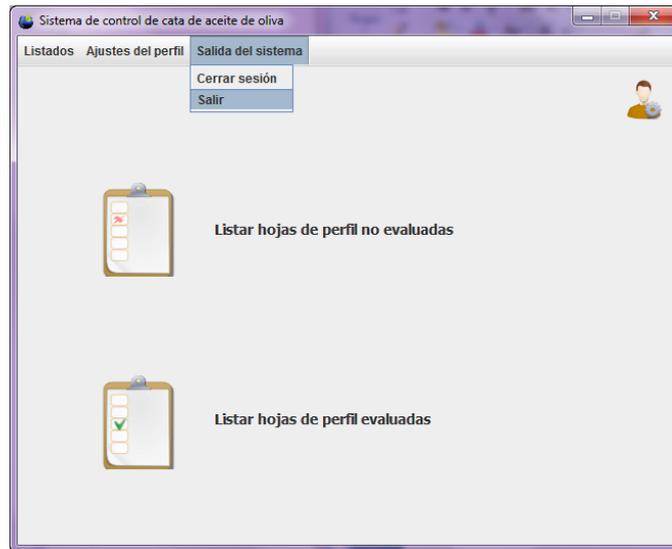


Figura 104: Cerrar la aplicación catador

# Anexo IV

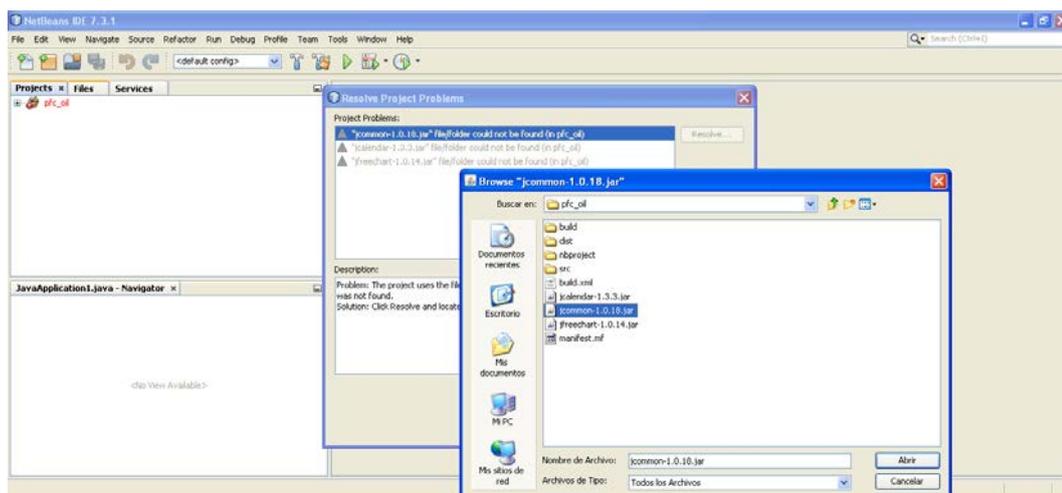
## Manual del desarrollador

Para comenzar procedemos a la instalación de Netbeans junto con el kit para desarrolladores JDK7, basta con seguir las diferentes pantallas dejando la información que aparece por defecto. El software se encuentran en el cd en el directorio D:\NETBEANS,JDK,JRE y se llama *jdk-7u25-nb-7\_3\_1-windows-i586*.

El siguiente paso que tenemos que hacer es instalar el servidor de bases de datos MySQL. Los pasos de este proceso están detallados en el apartado " **Paso 2: Instalar servidor de base de datos MySQL**" del manual de instalación que podemos encontrar en el anexo nº 1 de esta memoria.

A continuación pasamos a copiar el proyecto (*pfc-oil*) que se encuentra dentro del directorio D:\PROYECTO NETBEANS a C:\Documents and Settings\Administrador\Mis documentos\NetBeansProjects. (La primera vez que se usa la herramienta, el directorio NetBeansProjects no existe, crear uno para que se cree automáticamente). Tras copiarlo, pasamos a abrirlo desde Netbeans.

Si nos aparecieran algunas advertencias relacionadas con problemas en la localización de algunas librerías del proyecto, para solucionarlas basta con indicar donde se encuentran cada una de ellas, en este caso, dentro del propio proyecto.



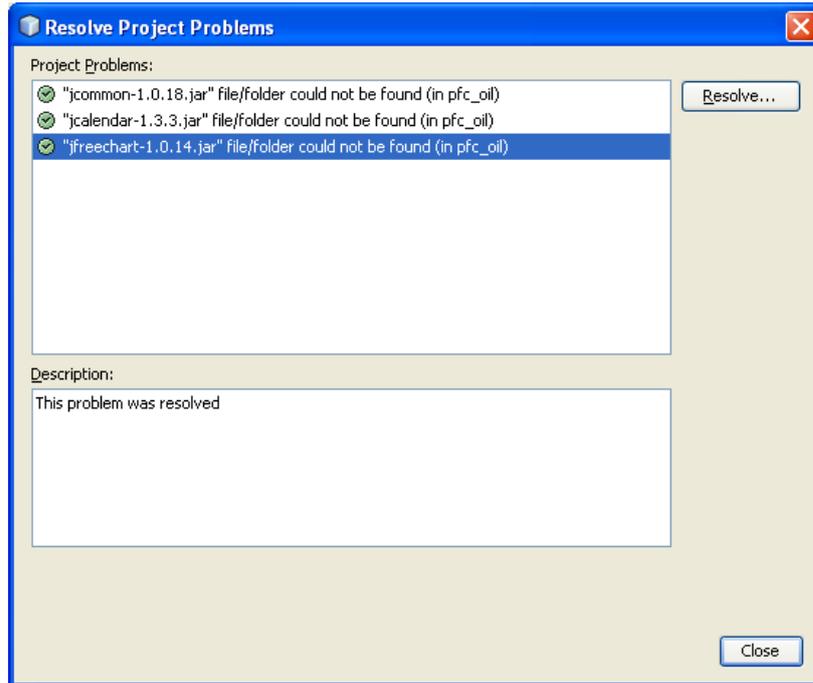


Figura 105: Reparación de problemas asociados a las librerías de un proyecto en Netbeans

Solucionados estos problemas podemos pasar a ejecutar el proyecto.

NOTA: Si se desea conocer como se realiza la conexión de la base de datos con Netbeans y/o cómo se generan los ficheros jpa, consultar la página nº 149 de esta memoria.