



UNIVERSIDAD
POLITECNICA
DE VALENCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

UNIVERSIDAD POLITÉCNICA DE VALENCIA
ESCUELA TÉCNICA SUPERIOR DE INFORMÁTICA APLICADA

*Aplicación práctica de MDA con Olivanova para
una aplicación de evaluación de recursos
humanos*

PROYECTO FIN DE CARRERA

Autor:
Solá Gómez, Diego

Directores:
Pastor López, Óscar
Giachetti Herrera, Giovanni
Marín Campusano, Beatriz

13 de Julio de 2010

Contenido

1. INTRODUCCIÓN	15
1.1 Motivación	15
1.2. Problema	16
1.3. Objetivos	17
1.4. Estructura del documento.....	18
2. HERRAMIENTAS DE RECURSOS HUMANOS EN LA ADMINISTRACIÓN PÚBLICA	20
2.1. El modelo Excarpus.....	23
2.1. 1. Valoración de puntos por factor	24
3. CONTEXTO TECNOLÓGICO	27
3.1. MDD	27
3.2. OO-Method	29
3.3 El modelado conceptual con Olivanova	31
3.3.1 El modelo de objetos.....	31
3.3.2 El Modelo Dinámico	33
3.3.3 El Modelo Funcional.....	34
3.3.4 El Modelo de Presentación	34
4. ESPACIO DEL PROBLEMA.....	38
4.1. <i>Resumen de las aplicaciones</i>	38
4.1.1 SIESMACRe	38
4.1.2 SIEFAP	40
4.2. <i>Especificación de Requisitos</i>	42
4.2.1. Diagrama de Casos de Uso	42
4.2.2. Casos de Uso SIESMACRe	44

4.2.3 Casos de Uso SIEFAP.....	63
4.3. <i>Mapeo Casos de Uso – Modelo</i>	99
4.3.1 Mapeo de SIESMACRe	101
4.3.1 Mapeo de SIEFAP	110
5. ESPACIO DE LA SOLUCIÓN.....	146
5.1. Transformación de PIM a PSM en OO-Method	146
5.2 Obtención de código con el sistema Olivanova STAR	150
5.3. Preparación de Base de datos.....	154
5.4. Compilación del código y puesta en marcha	156
5.5. El modelo de Implementación	160
6. CONCLUSIONES	190
6.1 OO-Method y Olivanova.....	190
6.2 Situaciones detectadas y posibles mejoras.....	191
6.3 Cumplimiento de objetivos y ampliaciones del PFC	194
Anexo I. Inventario del mapeo entre casos de uso y elementos conceptuales.....	199

Índice de Figuras

Figura 1.	Ficha de valoración por puntos de Requisitos Intelectuales.	26
Figura 2.	Esquema de desarrollo MDA	28
Figura 3.	Proceso de Desarrollo de Software utilizando OO-Method.	30
Figura 4.	Ejemplo del Diagrama de clases	32
Figura 5.	Ejemplo del Diagrama de Transición de Estados	33
Figura 6.	Niveles y elementos del modelo de presentación [22].	35
Figura 7.	Diagrama de casos de uso de SIESMACRe – Vista General.	44
Figura 8.	Diagrama de casos de uso para la gestión de categoría.....	45
Figura 9.	Diagrama de casos de uso para la gestión de complemento de destino.	47
Figura 10.	Diagrama de casos de uso para la gestión de puesto de trabajo.	50
Figura 11.	Diagrama de casos de uso para la gestión de Comunidad Autónoma.	54
Figura 12.	Diagrama de casos de uso para la gestión de número de habitantes.....	56
Figura 13.	Diagrama de casos de uso para la gestión de plantilla de trabajadores.	59
Figura 14.	Diagrama de casos de uso para obtener resultados.	61
Figura 15.	Diagrama general de casos de uso de SIEFAP.	64
Figura 16.	Diagrama de casos de uso para la gestión de empleados.....	65
Figura 17.	Diagrama de casos de uso para la gestión de usuarios.	69
Figura 18.	Diagrama de casos de uso para la gestión de departamentos.....	71
Figura 19.	Diagrama de casos de uso para la gestión de Administraciones.....	73
Figura 20.	Diagrama de casos de uso para la gestión de conocimientos.....	76
Figura 21.	Diagrama de casos de uso para la gestión de preguntas.	78
Figura 22.	Diagrama de casos de uso para la gestión de preguntas.	82
Figura 23.	Diagrama de casos de uso para la gestión de evaluaciones.....	86

Figura 24.	Diagrama de casos de uso para la gestión de evaluaciones.....	90
Figura 25.	Diagrama de casos de uso para realizar evaluación.....	93
Figura 26.	Diagrama de casos de uso para validación de cuestionarios.	98
Figura 27.	Esquema de mapeo SIESMACRe.....	100
Figura 28.	Esquema de mapeo SIEFAP.	100
Figura 29.	Modelo de objetos de SIESMACRe.....	101
Figura 30.	Modelo de objetos de SIEFAP.....	111
Figura 31.	Clases del modelo de objetos de SIEFAP.....	112
Figura 32.	Modelo de objetos para la Gestión de Empleados.....	113
Figura 33.	Modelo de objetos para la Gestión de Usuarios.	116
Figura 34.	Modelo de objetos para la Gestión de Departamentos.	118
Figura 35.	Modelo de objetos para la Gestión de Conocimientos.	120
Figura 36.	Relación de asociación de Subconocimientos.....	121
Figura 37.	Modelo de objetos para la Gestión de Preguntas.....	123
Figura 38.	Modelo de objetos para la Gestión de Plantillas de Cuestionario.	126
Figura 39.	Modelo de objetos para la Gestión de Evaluaciones.	129
Figura 40.	Derivación del atributo NivelObtenido de la clase Valoracion_Conocimiento. ...	134
Figura 41.	Derivación del atributo PuntacionObtenida de la clase Valoracion_Conocimiento.	135
Figura 42.	Modelo de objetos para la Gestión de Cuestionarios.	136
Figura 43.	Modelo de objetos para la Realización de la Evaluación.....	140
Figura 44.	Definición de la <i>MDIU Pregunta_PlantillaPIU</i>	143
Figura 45.	Esquema de la arquitectura de tres capas [33].....	148
Figura 46.	Esquema OLIVANOVA Transformation Engine [34].....	150
Figura 47.	Propiedades generales del Sistema Star.....	151
Figura 48.	Perfiles disponibles en Olivanova STAR Client.....	152

Figura 49.	Opciones de transformación de C# .NET 2.0.....	153
Figura 50.	Opciones de transformación de Desktop C# .NET 2.0.....	153
Figura 51.	Contenido del paquete de C# .NET 2.0.....	154
Figura 52.	Ejecución de script de creación de tablas.....	155
Figura 53.	Administrador de orígenes ODBC con vptDB.	156
Figura 54.	Dependencias del proyecto.	157
Figura 55.	Dependencias del proyecto.	158
Figura 56.	Solución VPT.	158
Figura 57.	Configuración de la compilación para plataforma .Net.....	159
Figura 58.	Explorador de componentes COM+ con “VPTrabajoSrv”.....	159
Figura 59.	Ventana de identificación de la aplicación.	160
Figura 60.	Árbol de Jerarquía de Acciones (HAT).	161
Figura 61.	Menú inicial para el agente AdministradorSalarios.....	161
Figura 62.	Menú inicial para el agente AdministradorValoracion.....	161
Figura 63.	Argumentos de entrada del servicio <i>create_instance</i> de la clase <i>Empleado</i>	163
Figura 64.	Entrada del servicio de creación de empleado.....	163
Figura 65.	Salida del servicio de creación de empleado.....	164
Figura 66.	Argumento de salida del servicio <i>create_instance</i> de la clase <i>Empleado</i>	164
Figura 67.	<i>Display Set Empleado</i>	165
Figura 68.	<i>Action Pattern Empleado</i>	165
Figura 69.	<i>Navigational Pattern Empleado</i>	166
Figura 70.	IIU Empleado.	166
Figura 71.	IIU Departamento.....	167
Figura 72.	PIU Empleado.	168
Figura 73.	Definición de <i>PIU Empleado</i> en el esquema conceptual.	168
Figura 74.	Definición del Filtro <i>Nombre</i> de la clase <i>Empleado</i> en el esquema conceptual...	169

Figura 75.	Definición de <i>OC EmpleadoFNacAsc</i> de la clase <i>Empleado</i> en el esquema conceptual.	169
Figura 76.	Atributo de entrada <i>i_rid_Respuesta3</i> del servicio <i>CREA_PREGUNTA</i> con posible valor nulo.	170
Figura 77.	Event-Condition-Action de <i>SIU CREA_PREGUNTA</i>	171
Figura 78.	Entrada de <i>SIU CREA_PREGUNTA</i>	171
Figura 79.	Especificación de la navegación condicional de la <i>SIU CREA_PREGUNTA</i>	172
Figura 80.	Representación de la <i>Conditional Navigation</i> de la <i>SIU CREA_PREGUNTA</i>	173
Figura 81.	Entrada al servicio de creación de respuesta.	173
Figura 82.	Especificación de <i>MDIU Pregunta</i>	174
Figura 83.	Representación de la <i>MDIU Pregunta</i>	174
Figura 84.	Entrada del servicio de creación de una evaluación.	175
Figura 85.	Salida del servicio de creación de una evaluación.	175
Figura 86.	Visibilidad de <i>PIU Conocimientos</i> para <i>ResponsableAdministracion</i>	176
Figura 87.	Visibilidad de <i>PIU Conocimientos</i> para <i>AdministradorValoracion</i>	176
Figura 88.	Definición de la visibilidad del agente <i>ResponsableAdministracion</i> sobre la clase <i>Conocimiento</i>	177
Figura 89.	Entrada de servicio de creación de evaluación de conocimiento.	177
Figura 90.	Salida de servicio de creación de evaluación de conocimiento.	178
Figura 91.	<i>MDIU Valoracion</i>	178
Figura 92.	Derivación del atributo <i>PuntosConocimiento</i> de la clase <i>Valoracion_Conocimiento</i>	179
Figura 93.	Restricción incumplida al ejecutar servicio de edición de evaluación de conocimiento.	179
	Esta restricción está especificada en el esquema conceptual en la clase <i>Valoracion_Conocimiento</i> , como se observa en la Figura 94.	180
Figura 94.	Restricción definida en la clase <i>Valoracion_Conocimiento</i>	180
Figura 95.	Entrada de servicio de creación de cuestionario.	181

Figura 96.	IIU Cuestionario.	181
Figura 97.	PIU Cuestionario para el empleado 12345678A.	182
Figura 98.	MDIU Plantilla_cuestionarioAcc.	183
Figura 99.	Entrada del servicio responder.	183
Figura 100.	MDIU Plantilla_cuestionarioAcc con respuesta seleccionada.	184
Figura 101.	Modelado funcional del servicio <i>validar_cuestionario</i>	184
Figura 102.	Ejecución de servicio de validación de cuestionario.	185
Figura 103.	MDIU Valoracion.	186
Figura 104.	MDIU Valoracion_ConocimientoCo.	186
Figura 105.	IIU Cuestionario.	187
Figura 106.	Precondiciones para el servicio <i>terminar_cuestionario</i> de la clase <i>Cuestionario</i>	188
Figura 107.	Precondición incumplida para SIU <i>terminar_cuestionario</i>	188
Figura 108.	Defecto de modelado en evaluación de atributos por situación.	192

Índice de Tablas

Tabla 1.	Resultado de la Encuesta de Estructura salarial del INE de 2006 del salario medio por estudios, organizado por sexos y nacionalidad.....	21
Tabla 2.	Caso de uso Alta de Categoría	45
Tabla 3.	Caso de uso Baja de Categoría.....	46
Tabla 4.	Caso de uso Modificación de salario de Categoría	46
Tabla 5.	Caso de uso Actualización de salario de Categoría.....	46
Tabla 6.	Caso de uso consulta de salario de Categoría	47
Tabla 7.	Caso de uso Alta de Complemento de Destino	48
Tabla 8.	Caso de uso Baja de Complemento de Destino.....	48
Tabla 9.	Caso de uso Modificación de valor económico	48
Tabla 10.	Caso de uso Actualización de valor económico.....	49
Tabla 11.	Caso de uso Consulta de valor económico	49
Tabla 12.	Caso de uso Alta de Puesto de Trabajo	50
Tabla 13.	Caso de uso Baja de puesto de Trabajo.....	51
Tabla 14.	Caso de uso Modificación de nombre de Puesto de Trabajo	51
Tabla 15.	Caso de uso Modificación de la media del salario del Puesto de Trabajo.....	51
Tabla 16.	Caso de uso Modificación de los percentiles del salario del Puesto de Trabajo	52
Tabla 17.	Caso de uso Actualización de la media del salario	52
Tabla 18.	Caso de uso Actualización de los percentiles del salario.....	53
Tabla 19.	Caso de uso Consulta de la media del salario.....	53
Tabla 20.	Caso de uso Consulta de los percentiles del salario	53
Tabla 21.	Caso de uso Alta de Comunidad Autónoma.....	54
Tabla 22.	Caso de uso Baja de Comunidad Autónoma	55

Tabla 23.	Caso de uso Modificación de nombre de Comunidad Autónoma.....	55
Tabla 24.	Caso de uso Modificación del factor de Comunidad Autónoma.....	55
Tabla 25.	Caso de uso Consulta del factor de Comunidad Autónoma.....	55
Tabla 26.	Caso de uso Alta de intervalo de Número de Habitantes.....	57
Tabla 27.	Caso de uso Baja de intervalo de Número de Habitantes.....	57
Tabla 28.	Caso de uso Modificación de intervalo de Número de Habitantes.....	57
Tabla 29.	Caso de uso Modificación del factor de intervalo de Número de Habitantes.....	58
Tabla 30.	Caso de uso Consulta del factor del intervalo de Número de habitantes.....	58
Tabla 31.	Caso de uso Alta de intervalo de plantilla.....	59
Tabla 32.	Caso de uso Baja de intervalo de Plantilla.....	60
Tabla 33.	Caso de uso Modificación de intervalo de Plantilla.....	60
Tabla 34.	Caso de uso Modificación de intervalo de Plantilla.....	60
Tabla 35.	Caso de uso Consulta del factor del intervalo de Plantilla.....	61
Tabla 36.	Caso de uso Obtención de informe.....	62
Tabla 37.	Caso de uso Cálculo del salario.....	63
Tabla 38.	Caso de uso de Obtención de estadísticas.....	63
Tabla 39.	Caso de uso Alta de Empleado.....	65
Tabla 40.	Caso de uso Activar Empleado.....	66
Tabla 41.	Caso de uso Desactivar Empleado.....	66
Tabla 42.	Caso de uso Baja de Empleado.....	66
Tabla 43.	Caso de uso Modificación de detalles de Empleado.....	67
Tabla 44.	Caso de uso Convertir Empleado en Supervisor.....	67
Tabla 45.	Caso de uso Abandonar rol de Supervisor.....	67
Tabla 46.	Caso de uso Convertir Empleado en Responsable.....	68
Tabla 47.	Caso de uso Abandonar rol de Responsable.....	68
Tabla 48.	Caso de uso Crear Usuario.....	69

Tabla 49.	Caso de uso Borrar Usuario	69
Tabla 50.	Caso de uso Convertir usuario en Administrador.....	70
Tabla 51.	Caso de uso Abandonar rol de Administrador	70
Tabla 52.	Caso de uso Convertir Usuario en Editor.....	70
Tabla 53.	Caso de uso Abandonar rol de Editor	70
Tabla 54.	Caso de uso Alta de Departamento.....	71
Tabla 55.	Caso de uso Baja de Departamento	72
Tabla 56.	Caso de uso Modificación del nombre del Departamento.....	72
Tabla 57.	Caso de uso Asignar presupuesto a Departamento	72
Tabla 58.	Caso de uso Consulta de presupuesto del Departamento	72
Tabla 59.	Caso de uso Consulta de empleados del Departamento.....	73
Tabla 60.	Caso de uso Alta de Administración	74
Tabla 61.	Caso de uso Baja de Administración.....	74
Tabla 62.	Caso de uso Modificación del nombre de Administración	74
Tabla 63.	Caso de uso Asignar presupuesto a Administración.....	74
Tabla 64.	Caso de uso Consulta de presupuesto de la Administración.....	75
Tabla 65.	Caso de uso Consulta de empleados de la Administración	75
Tabla 66.	Caso de uso Crear Conocimiento.....	76
Tabla 67.	Caso de uso Modificar Conocimiento	76
Tabla 68.	Caso de uso Eliminar Conocimiento	77
Tabla 69.	Caso de uso Crear Subconocimiento	77
Tabla 70.	Caso de uso Crear Pregunta.....	78
Tabla 71.	Caso de uso Modificar Pregunta.....	79
Tabla 72.	Caso de uso Crear respuesta	79
Tabla 73.	Caso de uso Modificar Respuesta.....	80
Tabla 74.	Caso de uso Eliminar Respuesta	80

Tabla 75.	Caso de uso Asignar Pregunta a Conocimiento	80
Tabla 76.	Caso de uso Desasignar Pregunta de Conocimiento	81
Tabla 77.	Caso de uso Crear Plantilla de Cuestionario	82
Tabla 78.	Caso de uso Modificar Plantilla de Cuestionario	83
Tabla 79.	Caso de uso Asignar Pregunta a Plantilla de Cuestionario	84
Tabla 80.	Caso de uso Desasignar Pregunta de Plantilla de Cuestionario	84
Tabla 81.	Caso de uso Cambiar valor de Pregunta.....	84
Tabla 82.	Caso de uso Asignar Plantilla de cuestionario a Conocimiento.....	85
Tabla 83.	Caso de uso Crear Evaluación	86
Tabla 84.	Caso de uso Crear Evacuación de Conocimiento.....	87
Tabla 85.	Caso de uso Editar Evaluación	87
Tabla 86.	Caso de uso Cambiar estado.....	88
Tabla 87.	Caso de uso Editar Evacuación de Conocimiento	88
Tabla 88.	Caso de uso Eliminar Evaluación	88
Tabla 89.	Caso de uso Eliminar Evaluación de Conocimiento	89
Tabla 90.	Caso de uso Crear Cuestionario.....	90
Tabla 91.	Caso de uso Asignar Cuestionario a Evaluación de Conocimiento	91
Tabla 92.	Caso de uso Desasignar Cuestionario de Evaluación de Conocimiento	91
Tabla 93.	Caso de uso Editar Cuestionario	91
Tabla 94.	Caso de uso Dejar Cuestionario Preparado	92
Tabla 95.	Caso de uso Asignar Supervisor a Cuestionario.....	92
Tabla 96.	Caso de uso Eliminar Cuestionario	92
Tabla 97.	Caso de uso Responder Cuestionario	94
Tabla 98.	Caso de uso Comenzar Cuestionario	94
Tabla 99.	Caso de uso Responder Pregunta de Cuestionario	95
Tabla 100.	Caso de uso Terminar Cuestionario.....	95

Tabla 101.	Caso de uso Consultar Detalles Cuestionario	96
Tabla 102.	Caso de uso Consultar Detalles de Evaluación	96
Tabla 103.	Caso de uso Consultar Detalles de Evaluación Del Conocimiento	97
Tabla 104.	Caso de uso Validar Cuestionario	98
Tabla 105.	Caso de uso Comenzar Respuesta de Empleado	98
Tabla 106.	Caso de uso Solicitar corrección Cuestionario.....	99
Tabla 107.	Correspondencias entre clase Categoria y Gestión de Categorías	102
Tabla 108.	Correspondencias entre clase ComplementoDestino y Gestión de Complementos de Destino.....	103
Tabla 109.	Correspondencias entre clase Puesto y Gestión de Puestos de Trabajo.....	104
Tabla 110.	Correspondencias entre clase ComunidadAutonoma y Gestión de Comunidad Autónoma	105
Tabla 111.	Correspondencias entre clase NHabitantes y Gestión de Número de Habitantes	106
Tabla 112.	Correspondencias entre clase Plantilla y Gestión de Plantilla de Trabajadores...	107
Tabla 113.	Correspondencias entre clase Calculadora y Obtención de Resultados	108
Tabla 114.	Correspondencias entre clase Empleado y Gestión de Empleados.....	113
Tabla 115.	Correspondencias entre clase EmpleadoSupervisor y Gestión de Empleados	114
Tabla 116.	Correspondencias entre clase ResponsableAdministracion y Gestión de Empleados	115
Tabla 117.	Correspondencias entre clase Usuario y Gestión de Usuarios	116
Tabla 118.	Correspondencias entre clase Usuario_editor y Gestión de Usuarios	117
Tabla 119.	Correspondencias entre clase AdministradorValoracion y Gestión de Usuarios.	117
Tabla 120.	Correspondencias entre clase AdministradorSalarios y Gestión de Usuarios.....	117
Tabla 121.	Correspondencias entre clase Departamento y Gestión de Departamentos	118
Tabla 122.	Correspondencias entre clase Administracion y Gestión de Administraciones...	119
Tabla 123.	Correspondencias entre clase Conocimiento y Gestión de Conocimientos.....	122

Tabla 124.	Correspondencias entre clase Pregunta y Gestión de Preguntas.....	123
Tabla 125.	Correspondencias entre clase Respuesta y Gestión de Preguntas.....	124
Tabla 126.	Correspondencias entre clase Conocimiento y Gestión de Preguntas.....	125
Tabla 127.	Correspondencias entre clase Plantilla_Cuestionario y Gestión de Plantillas de Cuestionario.....	126
Tabla 128.	Correspondencias entre clase Pregunta_Plantilla y Gestión de Plantillas de Cuestionario.....	128
Tabla 129.	Correspondencias entre clase Valoracion y Gestión de Evaluaciones	130
Tabla 130.	Correspondencias entre clase Valoracion_Conocimiento y Gestión de Evaluaciones	131
Tabla 131.	Correspondencias entre clase Cuestionario y Gestión de Cuestionarios	137
Tabla 132.	Correspondencias entre clase PesoCuestionario y Gestión de Cuestionarios	139
Tabla 133.	Correspondencias entre clase Cuestionario y Realización de Evaluaciones	141
Tabla 134.	Correspondencias entre clase Respuesta_Empleado y Realización de Evaluaciones	141
Tabla 135.	Correspondencias entre Instancias del modelo de presentación y Realización de Evaluaciones	144
Tabla 136.	Correspondencias entre clase Cuestionario y Validación de Cuestionarios.....	144
Tabla 137.	Correspondencias entre clase Respuesta_Empleado y Validación de Cuestionarios	145
Tabla 138.	Casos de uso de SIESMACRe y clases, relaciones, atributos derivados y restricciones que participan en su diseño.	199
Tabla 139.	Casos de uso de SIESMACRe y servicios con sus entradas, salidas y precondiciones que participan en su diseño.....	204
Tabla 140.	Casos de uso de SIESMACRe, agentes que los pueden lanzar y elementos del modelo de presentación que participan en su diseño.	209
Tabla 141.	Casos de uso de SIEFAP y clases, relaciones, atributos derivados y restricciones que participan en su diseño.....	213
Tabla 142.	Casos de uso de SIEFAP y servicios con sus entradas, salidas y precondiciones que participan en su diseño.....	221

Tabla 143. Casos de uso de SIEFAP, agentes que los pueden lanzar y elementos del modelo de presentación que participan en su diseño.....	232
---	-----

1. INTRODUCCIÓN

En la sociedad actual, donde la información y el conocimiento han adquirido roles fundamentales para desempeñar prácticamente cualquier actividad generadora de valor, la correcta gestión de los recursos humanos toma cada vez más importancia por sobre otros recursos físicos e intangibles. En particular, en la Administración Pública, donde los recursos humanos se transforman en una interfaz directa hacia los contribuyentes y constituyen la cara visible de las instituciones públicas, contar con una buena gestión de estos recursos para asegurar su buen desempeño y satisfacción por parte de los ciudadanos se torna fundamental. Por este motivo, en este Proyecto Final de Carrera se intenta proveer una solución innovadora para la adecuada gestión de los recursos humanos asociado a la Administración Pública. En esta solución se han integrado las últimas técnicas para el desarrollo de sistemas de información siguiendo una filosofía de desarrollo de software dirigido por modelos.

A continuación se presenta en más detalle la motivación de este trabajo, se enmarcan los problemas detectados en el contexto de la gestión de recursos humanos en instituciones públicas, y se presentan los objetivos del trabajo asociado a este proyecto. Finalmente, se presenta la estructura general de este documento.

1.1 Motivación

En los Estados europeos del bienestar [1], los ciudadanos tienen unos derechos básicos garantizados. Los Estados son los encargados de velar por que los derechos se cumplan, bien supervisando que las necesidades queden satisfechas por el mercado, o bien proporcionando directamente los servicios que el mercado no llegue a cubrir.

El Estado español, a través de las distintas Administraciones Públicas, proporciona algunos de estos servicios básicos. Los recursos económicos para que las Administraciones Públicas puedan cumplir con su papel provienen del dinero de los ciudadanos; destinando la mayor parte de éstos a gastos de personal o Recursos Humanos.

La Administración Pública, al igual que las organizaciones privadas, se enfrenta al desafío de asignar las retribuciones de los empleados de forma que satisfagan la equidad externa y el equilibrio interno. Se entiende por equidad externa la similitud entre los mismos cargos de organizaciones diferentes que actúan en el mercado; y por equilibrio interno la comparativa salarial entre los distintos cargos de la misma organización [2].

El concepto de la remuneración salarial se ha abordado, con mayor o menor éxito, para el mercado de trabajo de la actividad privada; existe gran cantidad de consultoras de recursos humanos, como por ejemplo ICSA [¡Error! No se encuentra el origen de la referencia.] o Deloitte [¡Error! No se encuentra el origen de la referencia.], que ofrecen asesoramiento y

cuentan con estudios y herramientas sobre este tema, como encuestas salariales, referencias salariales, etc. Por contra, para en el ámbito público no se dispone de estas herramientas.

Por este motivo, contar con una herramienta adecuada para la gestión salarial en el sector público sería de gran importancia para mejorar la gestión de los recursos económicos del erario público, así como para dotar de una mayor equidad y transparencia a la retribución de los empleados de la Administración Pública.

En este trabajo de final de carrera nos proponemos elaborar una solución informática que apunta en esta dirección. Para esto se utilizarán tecnologías de desarrollo dirigido por modelos, que permiten acercarse a una mejor representación del espacio del problema y al mismo tiempo, permiten generar de forma automática los productos de software necesarios para dar soporte a nuestra solución. Así, es posible abordar un tema tan complejo como es la gestión de recursos humanos mediante herramientas y tecnologías de desarrollo de software de vanguardia, asegurando un resultado más cercano a las necesidades de información del sector público.

1.2. Problema

En la Administración Pública es necesario considerar diferentes elementos al momento de definir criterios adecuados de retribución a los empleados. Estos criterios pueden depender del nivel de estudios, de la experiencia o años de antigüedad en el puesto, del nivel de responsabilidad, de la peligrosidad del trabajo, del tipo de jornada laboral, etc.

Es necesario que estos criterios se establezcan de forma transparente y precisa para alcanzar los objetivos de equidad externa y equilibrio interno. La equidad externa se consigue cuando los empleados perciben que dos puestos iguales en dos Administraciones distintas están igualmente remunerados. En caso que se perciba distinta remuneración para dos puestos aparentemente iguales, se deben establecer y divulgar claramente las diferencias en características y competencias exigidas para acceder a cada puesto. El equilibrio interno se alcanza cuando los empleados entienden que las retribuciones de los distintos puestos de trabajo existentes dentro de una misma organización se explican por las diferencias entre sus competencias y funciones; para conseguirlo, las competencias y funciones de cada puesto de trabajo deben ser definidas de forma objetiva.

En definitiva, la consecución de estos dos objetivos necesita de dispositivos que permitan establecer comparaciones tanto entre distintas Administraciones, como entre los puestos de trabajo de la misma Administración.

Actualmente, no se conocen mecanismos que permitan a los empleados públicos (o aspirantes a serlo) conocer la información salarial de puestos de trabajo equivalentes en

distintas Administraciones, para poder valorar si se cumple el criterio de equidad externa. Esta carencia informativa puede provocar el descontento de empleados públicos que, pagados por encima del mercado, consideren que su sueldo es injusto, al no disponer de referencias salariales. También provoca que el aspirante a trabajar en la Administración no tenga medios suficientes para elegir a qué convocatoria presentarse o simplemente decidir si le interesa más el mercado privado de trabajo o el público.

La Administración Pública en general también adolece de una metodología de asignación de tareas y competencias a los puestos de trabajo, y de la evaluación de las habilidades de cada empleado para determinar el nivel de adaptación de sus empleados a las exigencias de los puestos de trabajo. Este problema lleva a situaciones en que se dispone de empleados sobrecapacitados (en la mayor parte de los casos) o infracapacitados para las tareas que desempeñan. Además, en muchos casos se puede percibir que la asignación salarial y los premios o compensaciones laborales se deben a criterios subjetivos, no especificados u opacos, lo que crea un clima laboral de desconfianza y desmotivación.

Estos dos problemas, la falta de mecanismos para la gestión transparente de salarios en distintas Administraciones Públicas y la inexistencia de una metodología para caracterizar y evaluar objetivamente los puestos de trabajo, provocan descontento y desmotivación en los empleados, lo que afecta notablemente a la productividad y eficiencia de la Administración Pública. A su vez, genera en la ciudadanía un clima de desconfianza en la gestión económica de los servicios que ofrece.

1.3. Objetivos

De acuerdo a los antecedentes presentados anteriormente, este Proyecto Final de Carrera tiene como meta principal mejorar la eficiencia en la gestión de los recursos humanos de la Administración Pública, y ayudar de este modo, a mejorar la percepción ciudadana del gasto que se realiza del erario público.

Para alcanzar la meta propuesta, es necesario alcanzar los siguientes objetivos específicos:

- Facilitar la información necesaria al empleado de la Administración Pública para que pueda comprobar si se cumple el criterio de equidad externa.
- Poner a disposición de cualquier ciudadano las estadísticas salariales de la Administración Pública con el fin de dotar de transparencia los gastos destinados al funcionariado.
- Permitir desarrollar un modelo de evaluación de empleados de acuerdo a sus conocimientos.

- Facilitar unos criterios objetivos para la asignación salarial y promoción dentro de la Administración Pública.
- Favorecer información que ayude a evaluar si se cumple el criterio de equilibrio interno.

Para satisfacer los objetivos específicos propuestos nos proponemos elaborar dos sistemas de información que den un soporte adecuado a la gestión de las retribuciones. Estos sistemas de información serán generados mediante las herramientas tecnológicas que actualmente existen para el desarrollo de sistemas de información complejos, como son la definición de modelos conceptuales y la generación automática de productos de software a partir de estos modelos.

El primero de los sistemas de información permitirá la publicación de estadísticas de salarios de la Administración Pública local, con el objetivo facilitar a los empleados la comprobación de si se cumple el criterio de equidad externa y dotar de transparencia el gasto del erario público empleado en recursos humanos.

El segundo de los sistemas de información permitirá la evaluación de conocimientos o competencias de los empleados de una Administración. Este sistema de información será diseñado con el objetivo de que pueda ser utilizado para un proyecto innovador dirigido a Administraciones Públicas nacionales denominado ©Excarpsus[5]. Excarpsus presenta un modelo de valoración de puestos de trabajo en la Administración Pública local. Los objetivos específicos que pretende cubrir el sistema de información dentro del proyecto Excarpsus son los siguientes:

- Ser la herramienta informática que permita desarrollar el modelo de evaluación de empleados Excarpsus.
- Facilitar y definir unos criterios objetivos para la asignación salarial y promoción dentro de la Administración Pública.
- Proporcionar información que ayude a evaluar si se cumple el criterio de equilibrio interno.

1.4. Estructura del documento

Esta memoria de Proyecto Final de Carrera (PFC) se estructura en seis capítulos y un anexo que se describen a continuación:

- En el capítulo uno se explica la motivación que hace surgir este PFC y el problema que se intenta resolver con el mismo, además de los objetivos que se desean alcanzar.

- En el capítulo dos se presenta el contexto actual de herramientas de gestión de recursos humanos en la Administración Pública. Además, en este capítulo se presenta el modelo Excarsus, el cual es el modelo de gestión de recursos humanos para el que se construyen los sistemas de información de este PFC.
- En el capítulo tres se presenta el Desarrollo Dirigido por Modelos (MDD). MDD es el método de desarrollo elegido para la construcción de estos sistemas, siendo OO-Method la tecnología concreta aplicada a través del conjunto de herramientas Olivanova.
- En el capítulo cuatro se hace un análisis del espacio del problema. En este capítulo se especifican los requisitos funcionales de los sistemas de información desarrollados mediante casos de uso y se explica la correspondencia entre cada caso de uso y el elemento del modelo conceptual involucrado en su representación.
- En el capítulo cinco se describe el espacio de la solución. En este capítulo se explica el proceso de transformación desde el modelo conceptual hasta el producto final; y posteriormente se muestra cómo se obtiene el código fuente mediante los sistemas Olivanova, los pasos de preparación de la base de datos y cómo se realiza la compilación del código fuente. Finalmente, se muestra cómo la aplicación final representa los conceptos diseñados en el modelo conceptual.
- En el capítulo seis se presentan las conclusiones del PFC y las posibles extensiones al trabajo realizado.
- En el anexo I se presenta el detalle de los casos de uso y los elementos del modelo conceptual que se utilizan para su representación.

2. HERRAMIENTAS DE RECURSOS HUMANOS EN LA ADMINISTRACIÓN PÚBLICA

La sociedad capitalista se organiza en mercados regidos por la ley de la oferta y la demanda. En estos mercados, los productos en los que su oferta sea inferior a la demanda tienden a tener un precio más alto que aquellos en los que la oferta supera a su demanda. El laboral, no es más que otro mercado que se rige, en principio sobre las leyes de la oferta y la demanda. En el mercado laboral, el producto que se comercia es el esfuerzo o conocimiento de los empleados que suele cuantificarse en horas o jornadas de trabajo.

Desde un punto de vista económico y siguiendo estrictamente este criterio, el precio de la jornada de trabajo estará dado por el equilibrio entre el hecho de que una organización no dispone de una alternativa más rentable que su empleado y el empleado no encuentra un puesto laboral más rentable que el de la organización en la que trabaja. En otras palabras, “si una empresa paga salarios más altos que el mercado, pierde competitividad. Si paga salarios más bajos que el mercado, pierde a sus empleados” [6].

Desde un punto de vista social, el salario debe ser suficiente para que el trabajador pueda cumplir con sus necesidades básicas. En el caso concreto de España, la Constitución expresa que “todos los españoles tienen el deber de trabajar y el derecho al trabajo, a la libre elección de profesión u oficio, a la promoción a través del trabajo y a una remuneración suficiente para satisfacer sus necesidades y las de su familia, sin que en ningún caso pueda hacerse discriminación por razón de sexo” [7].

Teniendo en cuenta ambas perspectivas, es necesario tener herramientas para conocer el valor nominal de los salarios para que se ajusten al mercado laboral y que, a su vez, sirvan para satisfacer las necesidades básicas de los individuos de la sociedad. Una de estas herramientas es la referencia salarial. Históricamente, la primera referencia salarial conocida es el salario mínimo, establecido por primera vez en el siglo XIX en Australia y Nueva Zelanda [8].

En España, el salario mínimo (llamado *Salario Mínimo Interprofesional*, en adelante SMI) representa el sueldo mínimo legal que un trabajador puede cobrar independientemente de su dedicación profesional. Además del SMI, existen otras referencias salariales en España, como son las aparecidas en los convenios colectivos¹. Estas son referencias que están fijadas por ley o amparadas en una negociación colectiva. Sin embargo, además existen referencias y estudios

¹ Convenio colectivo: “Un Convenio Colectivo es el acuerdo suscrito entre los representantes de los trabajadores y de los empresarios para fijar las condiciones de trabajo y productividad en un ámbito laboral determinado” [9].

de mercado, que sin ser fijadas por ley, están dirigidas a que la empresa privada conozca el valor de los salarios.

Para el mercado privado, las empresas consultoras de recursos humanos ofrecen servicios como los estudios de mercado mediante encuestas salariales u otros como la auditoría salarial. La encuesta salarial consiste en recabar información en relación con el nivel de remuneraciones acorde con las tendencias económicas de una región [10]. Ejemplos de esta herramienta son el “Observatorio Salarial” de la consultora de recursos humanos *ICSA* [11] o las encuestas salariales que ofrecen empresas de internet como *salarios.net* [**¡Error! No se encuentra el origen de la referencia.**] o *tusalarario.es* [12]. También el Instituto Nacional de Estadística (INE) realiza una “Encuesta de Estructura Salarial” [13] con carácter cuatrienal. La **¡Error! No se encuentra el origen de la referencia.** presenta el resultado de una estadística de media salarial de los trabajadores en España según su nacionalidad, sexo y nivel de estudios publicada por el INE. La auditoría salarial, por su parte, se emplea para conocer la política de remuneraciones aplicada a los distintos niveles de una organización y compararlas con el mercado que compite. Estos servicios también son proporcionados por empresas consultoras de recursos humanos como *ICSA* o *Deloitte* [4].

Tabla 1. Resultado de la Encuesta de Estructura salarial del INE de 2006 del salario medio por estudios, organizado por sexos y nacionalidad.

	I. Sin estudios	II. Educación primaria	III. Educación secundaria I	IV. Educación secundaria II	V. Formación profesional de grado medio	VI. Formación profesional de grado superior	VII. Diplomados universitarios o equivalente	VIII. Licenciados, ingenieros superiores y doctores
Ambos sexos								
Total	14.363,99	16.115,33	15.839,69	20.732,53	18.079,05	19.962,21	25.166,90	32.307,43
España	14.804,93	16.500,73	16.045,85	21.139,15	18.180,63	20.064,22	25.351,87	32.481,92
Varones								
Total	15.996,55	17.971,76	17.718,74	24.418,31	21.436,54	23.313,75	30.474,84	38.598,16
España	16.617,20	18.591,08	18.007,96	25.059,56	21.612,15	23.491,10	30.892,86	38.852,04
Mujeres								
Total	11.159,21	11.949,30	12.399,34	16.530,53	14.678,69	15.637,70	21.222,51	25.441,05
España	11.457,53	12.092,31	12.499,78	16.691,17	14.752,70	15.678,13	21.310,23	25.593,51

Para el mercado de trabajo público, sin embargo, no existen herramientas que permitan conocer el salario que se paga en las Administraciones. Es necesario y deseable contar con este tipo de información y que esté disponible tanto para empleados, como para gestores de recursos humanos o para candidatos a trabajar en la Administración Pública.

Para lograrlo, se deben tener en cuenta los conceptos salariales propios de la Administración Pública, que se fijan en la *Ley 7/2007, de 12 de abril, del Estatuto Básico del*

Empleado Público [14]. Los conceptos que se consideran de relevancia para la realización de la referencia salarial que motiva este Proyecto Final de Carrera son los de los funcionarios de carrera. Los artículos 23 y 24 de la citada Ley exponen estos conceptos:

Artículo 23. Retribuciones básicas.

Las retribuciones básicas, que se fijan en la Ley de Presupuestos Generales del Estado, estarán integradas única y exclusivamente por:

- a) *El sueldo asignado a cada Subgrupo o Grupo de clasificación profesional, en el supuesto de que éste no tenga Subgrupo.*
- b) *Los trienios, que consisten en una cantidad, que será igual para cada Subgrupo o Grupo de clasificación profesional, en el supuesto de que éste no tenga Subgrupo, por cada tres años de servicio.*

Artículo 24. Retribuciones complementarias.

La cuantía y estructura de las retribuciones complementarias de los funcionarios se establecerán por las correspondientes leyes de cada Administración Pública atendiendo, entre otros, a los siguientes factores:

- a) *La progresión alcanzada por el funcionario dentro del sistema de carrera administrativa.*
- b) *La especial dificultad técnica, responsabilidad, dedicación, incompatibilidad exigible para el desempeño de determinados puestos de trabajo o las condiciones en que se desarrolla el trabajo.*
- c) *El grado de interés, iniciativa o esfuerzo con que el funcionario desempeña su trabajo y el rendimiento o resultados obtenidos.*
- d) *Los servicios extraordinarios prestados fuera de la jornada normal de trabajo.*

Por su parte, la dinámica de la negociación colectiva en cada una de las Administraciones, regulada por la *Ley 9/1987 de Órganos de Representación al Servicio de las Administraciones Públicas [15]*, ha hecho que mediante diversas valoraciones o actualizaciones de los salarios fluctúen en cada una de ellas, por lo que las referencias entre comunidades y municipios se ven alteradas.

Este Proyecto Final de Carrera pretende ofrecer al conjunto de la ciudadanía una referencia salarial para la Administración Pública local. Ésta debe tener en cuenta los diferentes de los conceptos salariales mencionados en la *Ley 7/2007 del Estatuto Básico del Empleado* y las variaciones de dichos valores dependiendo de las características de la localidad del puesto de trabajo. Para lograrlo, se parte de una referencia salarial inicial de la *Asociación Valenciana de Técnicos de Personal (ATPCV) [16]*.

El sistema de información que da soporte al uso de esta referencia salarial se llama ***Sistema de Información de Estimación del Salario mediante el Análisis de Conceptos Retributivos*** (en adelante, **SIEMACRe**). Durante la vida útil de este sistema de información, la

referencia salarial debe ser actualizada para no quedar desfasada, cumplir los objetivos propuestos y ser de utilidad para la sociedad.

La *Ley 7/2007 del Estatuto Básico del Empleado Público* también expresa en su artículo 1 que la Administración Pública debe cumplir el fundamento de “*igualdad, mérito y capacidad en el acceso y en la promoción profesional*”. Este principio, unido a los puntos *a, b* y *c* de retribuciones complementarias, plantea la necesidad de conseguir un sistema objetivo de evaluación de empleados para la asignación retributiva.

Para satisfacer esa necesidad, se creó el modelo ©Excarpsus [5] para la valoración de puestos de trabajo en la Administración Pública española. Este modelo tiene dos referentes históricos. El primero de ellos es un proyecto del Departamento de Trabajo de Nueva Zelanda orientado a la función pública en el que se pretendía realizar cuestionarios a los empleados para conocer el nivel en el que se encontraban de acuerdo a unos factores de los puestos de trabajo previamente definidos. El objetivo de este proyecto era garantizar la no discriminación, por razón de género o nacionalidad, en el proceso de valoración de los puestos de trabajo. El segundo referente histórico es el denominado proyecto ISOS [17], de la Universidad Politécnica de Catalunya, que importó a España el trabajo neozelandés, pero lo adaptó a la empresa privada, dejando vacante la aplicación del proyecto para la función pública.

Excarpsus propone la adaptación del modelo neozelandés para su uso en la Administración Pública española. Para realizar la valoración de puestos de trabajo en una Administración Pública determinada siguiendo el modelo Excarpsus, se deben definir las competencias de los puestos de trabajo y las habilidades necesarias para llevarlas a cabo. Una vez definidas, se evalúa a los empleados de la Administración a través de cuestionarios de sus habilidades para obtener su adecuación al puesto de trabajo.

En el Proyecto Final de Carrera objeto de este documento, se pretende crear también un sistema de información que dé soporte a la aplicación práctica del modelo Excarpsus, concretamente a la gestión de evaluaciones de los empleados. Por tanto, debe gestionar la información relativa a los requisitos de los puestos de trabajo, la creación y asignación de los cuestionarios a los empleados correspondientes y las respuestas y obtención de puntos por dichos empleados. Esta herramienta se denomina **Sistema de Información de la Evaluación de Funcionarios de Administraciones Públicas** (en adelante, **SIEFAP**).

2.1. El modelo Excarpsus

El Modelo Excarpsus [5] es un método de clasificación de puestos de trabajo de Administraciones Públicas locales y valoración de sus empleados. Es un mecanismo completo para conocer o establecer la política retributiva de las Administraciones en las que se implanta

y evaluar el rendimiento y la adecuación de los empleados a su puesto de trabajo con respecto al salario que perciben.

La aplicación del Modelo Excarpus en una Administración comienza con el análisis y clasificación de los puestos de trabajo de la Administración local particular. Las características mínimas que se necesitan de cada puesto de trabajo son [18]:

- Denominación del puesto.
- Requisitos exigidos para su desempeño.
- Nivel de complemento de destino y específico.
- Categoría profesional o régimen jurídico.

Entre diferentes Administraciones Públicas locales suele haber diferencias, entre otras, en políticas de retribución y situación económica. Por tanto, además de la clasificación de los puestos de trabajo, se deben conocer otros aspectos importantes de la Administración local. Entre estos se destacan [18]:

- Definición exhaustiva de los conceptos retributivos y su percepción; así como establecimiento de los parámetros básicos del mismo.
- Determinación de la situación actual y la masa salarial disponible.
- Determinación del abanico salarial y su adaptación al mercado.
- Fluctuaciones de la masa salarial en función de parámetros determinados por variaciones en el desarrollo de la organización (población, plantilla, competencias, excelencia, etc).
- Establecimiento de mecanismos de compensación y actualización de disonancias retributivas tras la alteración de los contenidos.

Una vez conocidas las características económicas y de política laboral de la Administración, se debe definir los conocimientos o habilidades exigibles a los puestos de trabajo resultantes de la clasificación. La evaluación a los empleados se realiza en base a estas habilidades.

2.1. 1. Valoración de puntos por factor

El método que utiliza el modelo Excarpus en su valoración es el de *valoración de puntos por factor*. Excarpus tiene dos referentes históricos: un proyecto de evaluación de empleados de la función pública realizado por el Departamento de Trabajo de Nueva Zelanda, y el proyecto ISOS [17] de la Universidad Politécnica de Catalunya. Ambos trabajos utilizan también el método de valoración de puntos por factor.

Este método fue creado por el norteamericano Merrill R. Lott y es el más perfeccionado y utilizado de los métodos para la evaluación de cargos. Es una técnica cuantitativa en donde se asignan valores numéricos (puntos) a cada elemento o aspecto del cargo y se obtiene un valor total por la suma de valores numéricos.

El método de evaluación por puntos se basa en un trabajo previo de análisis de cargos y exige las siguientes etapas:

- ❖ Elección de factores de evaluación. La identificación de los factores está directamente relacionada con los tipos de cargos que van a evaluarse. Más que de factores individuales, se habla de cuatro grupos de factores: a. Requisitos intelectuales b. Requisitos físicos c. Responsabilidades implicadas d. Condiciones de trabajo.
- ❖ Ponderación de los factores de evaluación .La ponderación de los factores de evaluación se hace de acuerdo con su importancia relativa, una vez que éstos no sean idénticos en su contribución al desempeño de los cargos, requiriendo ajustes compensatorios.
 1. Montaje de la escala de puntos. Es la atribución de valores numéricos (puntos) a los diversos grados de cada factor. Por lo general, el grado más bajo de cada factor (grado A) corresponde al valor del porcentaje de ponderación. Es decir, los valores ponderados sirven como base para la evaluación de escalas de puntos para cada factor y constituirán el valor en puntos para el grado A de cada factor. Establecidos los valores numéricos (puntos) para el grado A de cada factor, el siguiente paso es la asignación de puntos a los grados B, C, D y así sucesivamente.
 2. Montaje del manual de evaluación de cargos. Una vez efectuado el montaje de la escala de puntos, se procede a definir el significado de cada uno de los factores de evaluación. Se trata del montaje del manual de evaluación de cargos, una especie de guía o patrón de comparación entre los diversos grados de cada factor y sus respectivos valores en puntos.
 3. Agrupar los puestos similares en grados de remuneración.
- ❖ Designación del Comité de valoración y aplicación del procedimiento establecido en el manual de valoración.

De todo este proceso resulta esencial acordar por la Corporación y su equipo técnico los siguientes elementos:

1. Determinación de los factores y sus pesos en términos porcentuales
2. Determinación de los grados de cada factor
3. La determinación de la aplicación del manual a los puestos, comité de valoración
4. La aplicación del nuevo régimen retributivo.

La Figura 1 muestra un ejemplo de desglose de puntos para los Requisitos Intelectuales. En este ejemplo se aprecia que el número de puntos totales es de 2000 y que las habilidades “Conocimientos, aptitudes y comprensión” representan el 40 % del total. Como se observa, se puede dividir uno de los Requisitos en nuevas habilidades. Sobre estas últimas, se definen los grados y se asigna un valor porcentaje de la puntuación de la habilidad a cada uno de los grados.

PUNTOS		2000	
1.-Conocimientos, aptitudes y comprensión.	PESO %	PUNTOS	
	40	800,0	
1.1. Equipos y máquinas.	PESO %	PUNTOS	
	12	96,0	96,0
1.2. Matemáticas.	13	104,0	104,0
1.3. Lectura y comprensión.	13	104,0	104,0
1.4. Procesamiento de información.	12	96,0	96,0
1.5. Ofimática	12	96,0	96,0

Figura 1. Ficha de valoración por puntos de Requisitos Intelectuales.

La evaluación de los Requisitos o habilidades de los empleados se realiza mediante cuestionarios específicos. La puntuación obtenida en los cuestionarios de una habilidad es la que determina el nivel obtenido por el empleado en la misma.

A día de hoy, este modelo no tiene herramientas informáticas específicas que faciliten su implantación práctica en Administraciones Públicas locales. En este PFC, se persigue el desarrollo de un Sistema de Información que gestione la evaluación de empleados y el mecanismo de valoración de puntos por factor.

Para el desarrollo de estos dos sistemas de información (**SIESMACRe** y **SIEFAP**) se opta por una tecnología de última generación, como es OO-Method. Esta tecnología permite el desarrollo dirigido por modelos y ha sido elegida para este Proyecto Final de Carrera por las ventajas de diseño a un nivel más próximo al usuario, pues los requerimientos del sistema para funcionar en la Administración Pública son complejos y es más fácil el entendimiento entre usuario y desarrollador con tecnologías como la empleada. En el siguiente capítulo, se explica en detalle el contexto tecnológico de este Proyecto Final de Carrera.

3. CONTEXTO TECNOLÓGICO

La motivación de este Proyecto Final de Carrera es el desarrollo de Sistemas de Información orientados a la mejora de la gestión de Recursos Humanos en la Administración Pública. El proyecto está basado en un modelo de evaluación de competencias de empleados de Administraciones Públicas denominado Excarpsus. Debido a la complejidad que presenta este modelo de gestión de recursos humanos, se decide emplear técnicas de desarrollo dirigido por modelos, que nos permiten acercarnos a la solución informática centrandolo el esfuerzo en la representación del espacio del Problema.

En este capítulo, se explica brevemente el enfoque MDD, la metodología de desarrollo de software OO-Method empleada para lograr los sistemas de información que se implementan en el PFC y la herramienta que da soporte a OO-Method.

3.1. MDD

La historia en el desarrollo de software puede verse como una sucesión de incrementos en el nivel de abstracción. Desde ordenadores que se programaban en lenguaje máquina, pasando por la aparición de lenguajes ensamblador, métodos de programación estructurada o modelos orientados a objetos, cada paso en la evolución pretende acercar la producción de software al Espacio del Problema.

El último avance en este proceso evolutivo es el Desarrollo Dirigido por Modelos (MDD, por sus siglas en inglés: Model Driven Development). MDD propone que el desarrollo de software se centre en el Espacio del problema, empleando modelos conceptuales, en lugar de hacerlo en el Espacio de la solución, que centra el esfuerzo en cómo conseguir el producto software.

Los beneficios perseguidos por MDD, al igual que con el resto de metodologías de desarrollo de software, son la reducción en los tiempos de desarrollo y la disminución de costos asociados al mantenimiento de aplicaciones, además de conseguir aplicaciones correctas y software de calidad. El hecho de que el modelo sea la parte central del desarrollo proporciona dos importantes ventajas para lograr estos objetivos. El esfuerzo se centra en los primeros pasos del ciclo de vida de producción del software, con lo que se consigue que los errores o problemas sean descubiertos en fases tempranas del desarrollo y se resuelvan independientemente de la tecnología empleada. La segunda ventaja es que facilita que los interesados en el producto software se involucren en el diseño, ya que no se precisan conocimientos de la tecnología final para entender el modelo conceptual [19].

El conjunto de tecnologías que dan soporte a la producción de software centrado en el uso de modelos se conoce como Ingeniería Dirigida por Modelos (MDE, de las siglas de Model Driven Engineering en inglés). Forma parte de esta ingeniería la Arquitectura Dirigida por Modelos (MDA, siglas de Model Driven Architecture), que es la propuesta del Object Management Group (OMG) y la arquitectura más utilizada actualmente en entornos MDD [20] [21].

MDA propone que la lógica de la aplicación se debe separar de los aspectos tecnológicos asociados a la plataforma de la implementación. Para ello, define estos cuatro niveles de abstracción: modelos independientes de computación (CIM), modelos independientes de plataforma (PIM), modelos específicos de plataforma (PSM) y modelos de implementación (IM), este último modelo corresponde a la aplicación final. El proceso de desarrollo consiste, por tanto, en ir transformando el modelo de nivel de abstracción superior en un modelo del nivel inmediatamente inferior sucesivamente hasta llegar al nivel correspondiente a la aplicación final [20] [22] [23]. La Figura 2 representa este esquema de desarrollo.

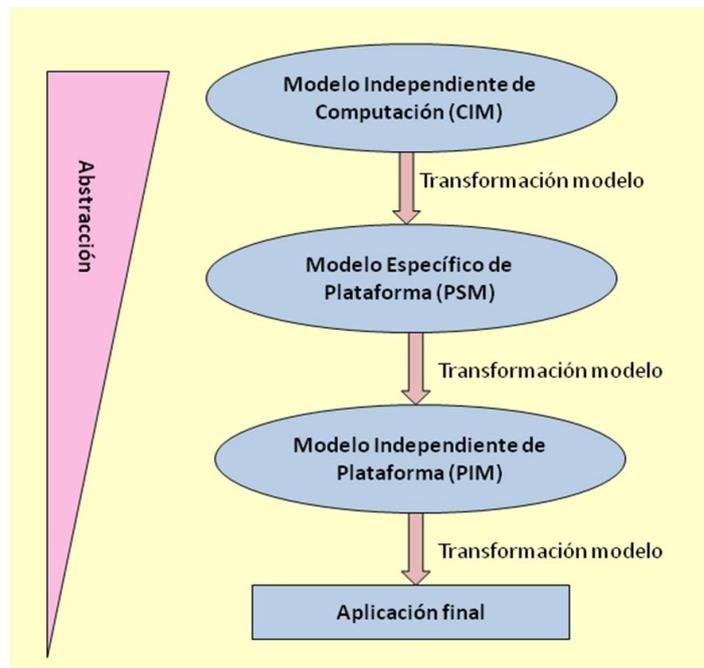


Figura 2. Esquema de desarrollo MDA

En el contexto MDE, es necesario contar con un lenguaje de modelado que tenga un conjunto finito de patrones conceptuales y cuya semántica sea precisa para así poder obtener modelos conceptuales no ambiguos [19]. MDA recomienda el uso del Lenguaje de Modelado Unificado UML (Unified Modeling Language), pero este lenguaje no tiene una semántica completamente precisa, por lo que se hace necesario extenderlo.

El modo de realizar estas extensiones no está definido. Diversas soluciones MDE han optado por definir lenguajes de modelado específicos para su dominio de aplicación,

conocidos como DSML (Domain Specific Modeling Languages en inglés). Un DSML define sin ambigüedad los modelos conceptuales asociados a propuestas MDE, permitiendo realizar de forma correcta las transformaciones de modelos destinadas a generar los productos finales [20].

3.2. OO-Method

OO-Method es un método de producción automática de software. Pone en práctica la propuesta MDA de separación entre la lógica de las aplicaciones y la plataforma tecnológica de implementación. Por tanto, permite centrar el esfuerzo del desarrollo del software en lo que hace el sistema (Espacio del Problema) en lugar de la implementación de la solución (Espacio de la Solución).

OO-Method se basa en el paradigma orientado a objetos. El DSML parte de UML, pero acota sus conceptos para crear unos patrones conceptuales bien definidos y precisos. Cada uno de los patrones conceptuales del DSML de OO-Method tiene una representación gráfica igual o similar a la propuesta en UML, pero su formalismo se especifica mediante el lenguaje formal OASIS (Open and Active Specification of Information Systems) [24].

El proceso de producción de software propuesto por OO-Method se divide en 4 fases, que se corresponden con los niveles de abstracción definidos por MDA: modelos independientes de computación (CIM), modelos independientes de plataforma (PIM), modelos específicos de plataforma (PSM) y modelos de implementación (IM).

En la primera fase, OO-Method utiliza el *Modelo de Requisitos* para describir los modelos independientes de computación (CIM). Este modelo es realizado por el analista del sistema con la ayuda de la herramienta *RETO* [25].

En la segunda fase, OO-Method emplea el *Modelo Conceptual* para describir los modelos independientes de plataforma (PIM). La transformación del Modelo de Requisitos en Modelos Conceptuales se realiza manualmente; el analista crea el Modelo con la ayuda de herramientas CASE. La empresa CARE Technologies, junto con el apoyo del grupo de investigación OO-Method, ha desarrollado un conjunto de herramientas para el diseño de Modelos Conceptuales denominado Olivanova Suite [26].

En la tercera fase, se emplea el Modelo de Ejecución para la creación de los modelos específicos de plataforma (PSM). La transformación del modelo anterior al de Ejecución se realiza de forma automática por el compilador de modelos *The Olivanova Programming Machine* desarrollado por la empresa CARE Technologies [26]. Con este compilador también se obtiene de forma automática el Modelo de Implementación (IM), que es la aplicación final.

En esta tercera fase, también se especifica la forma en que será utilizado el sistema de manera abstracta en el modelo de ejecución; la cual permite especificar los usuarios que accederán al sistema, qué acciones tendrán disponibles al acceder y la secuencia de ejecución entre ellas. Esto se consigue con el uso del lenguaje formal OASIS, que permite la especificación exacta de las características de implementación de los objetos del sistema de acuerdo a la forma en que serán utilizados estos objetos.

En el esquema OO-Method, la obtención de forma automática del modelo de implementación a partir del modelo de ejecución representa la transformación del Espacio del Problema (representado por el modelo conceptual) al Espacio de la Solución (producto software correspondiente) [20] [22].

La Figura 3 muestra el esquema del proceso de producción de software planteado por OO-Method. En ella se pueden ver los modelos que lo componen y sus correspondencias con la propuesta MDA, además de las transformaciones entre modelos.

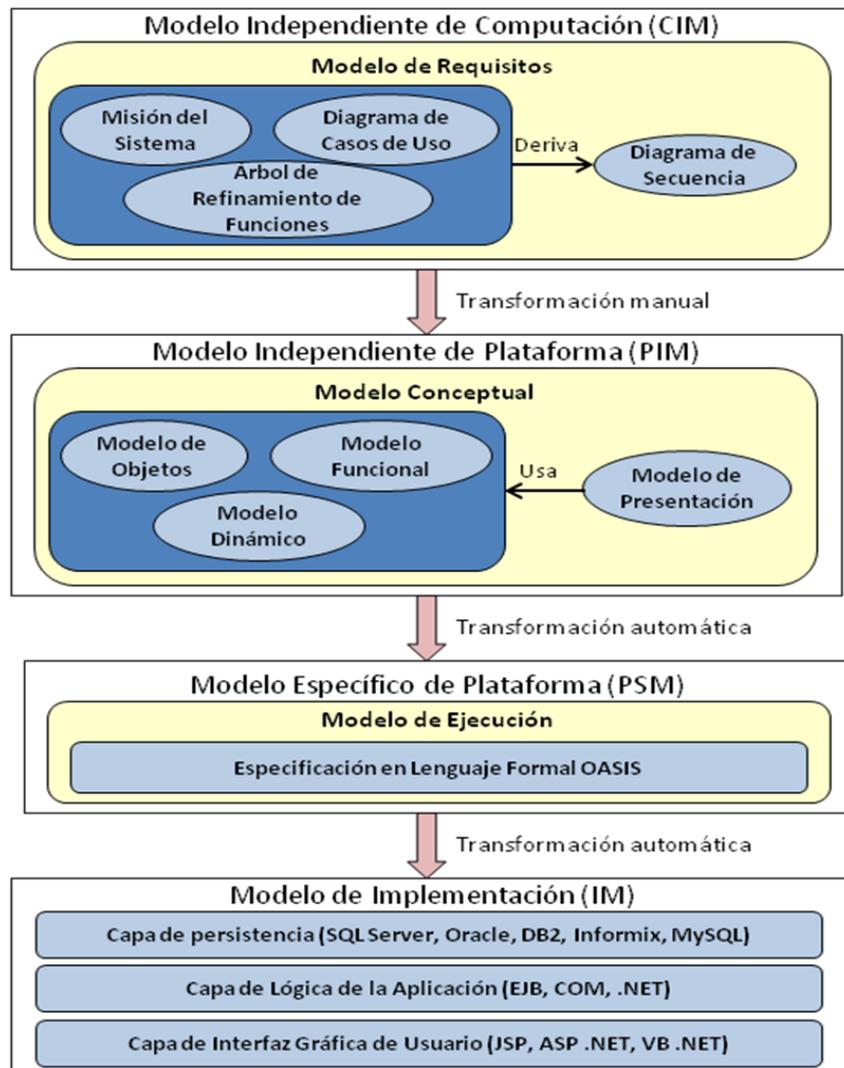


Figura 3. Proceso de Desarrollo de Software utilizando OO-Method.

3.3 El modelado conceptual con Olivanova

Olivanova Modeler es la herramienta creada por CARE Technologies para el Modelado Conceptual orientado a objetos. Permite la creación, edición y validación formal de modelos conceptuales de alto nivel.

El objetivo principal de OLIVANOVA Modeler es construir una especificación formal de un sistema de información de forma que dicha especificación sea validable. La especificación se basa en el lenguaje formal de especificación de sistemas de información orientado a objetos denominado OASIS. El uso de este lenguaje permite asegurar la consistencia, corrección y no ambigüedad de los modelos conceptuales creados, y por lo tanto, de los sistemas de información generados posteriormente de forma automática a partir de los mismos.

Para ello dispone de los siguientes modelos que nos permitirán definir un modelo conceptual con el máximo detalle:

- Modelo de objetos
- Modelo Dinámico
- Modelo Funcional
- Modelo de Presentación

Para definir cada uno de estos modelos, OLIVANOVA Modeler proporciona una serie de patrones de modelado conceptual, un conjunto de diagramas UML y una serie de fórmulas bien formadas, mediante los cuales se puede definir de manera sencilla el modelo conceptual, haciendo que la complejidad de la especificación formal construida sea totalmente transparente al analista.

A continuación, se explica brevemente cómo crear un modelo conceptual mediante esta herramienta, viendo los diversos modelos que la componen.

3.3.1 El modelo de objetos

El punto de partida para especificar un modelo conceptual mediante OLIVANOVA Modeler es el Modelo de objetos, el cual recoge las propiedades estáticas del sistema de información. En los sistemas de información, podremos identificar diferentes entidades con una estructura y un comportamiento denominados objetos, los cuales pueden ser agrupados en clases en base a sus propiedades. El Modelo de objetos estará formado por las propiedades estáticas de los objetos que lo componen y las relaciones entre ellos.

Para especificar el Modelo de Objetos, OLIVANOVA Modeler utiliza una notación gráfica bien definida que se emplea en el Diagrama de Clases. La Figura 4 muestra un ejemplo de un modelo de objetos.

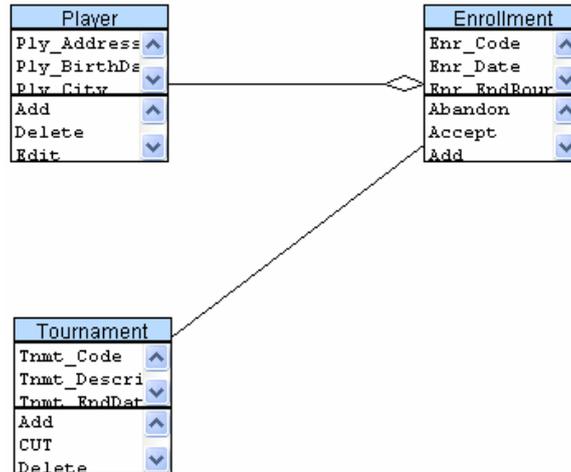


Figura 4. Ejemplo del Diagrama de clases

En el Diagrama de Clases se definen tanto las clases identificadas en el sistema, como las relaciones entre dichas clases. Las clases son las unidades básicas de modelado, y en ellas se describe las propiedades estáticas de los objetos (atributos, servicios y restricciones de integridad). Mientras que las relaciones pueden ser de tres tipos: relaciones de agregación, relaciones de especialización y relaciones de agente. Se trata de relaciones estructurales entre las clases.

Todas las propiedades de las clases son accesibles desde un mismo diálogo, de forma que se facilita al analista la edición de las mismas. Entre las más importantes cabe destacar los atributos, el conjunto de valores de los cuales definen el estado de cada objeto de esa clase.

Por otro lado tenemos los servicios, que pueden ser de tres tipos:

- Eventos: son unidades de proceso atómicas, ocurren en un momento específico de tiempo y producen cambios de estado en los objetos.
- Transacciones locales: son unidades de proceso que pueden estar compuestas por varios servicios. Cuando se inicia una transacción, tendrá que continuar hasta que termine o volver al estado inicial en el caso de que se produzca algún error.
- Operaciones locales: son idénticas a las transacciones locales, pero en caso de fallo no vuelven al estado inicial.

Por último, también se pueden definir restricciones de integridad en las clases, se trata de condiciones que se deben cumplir en cualquier instante de tiempo.

Olivanova Modeler emplea fórmulas bien formadas para definir propiedades del sistema en todos los modelos que componen el Modelo Conceptual. En el Diagrama de Clases, además de la notación gráfica, se dispone de fórmulas bien formadas para especificar restricciones de

integridad, transacciones, operaciones, valores por defecto de los atributos, valores de atributos derivados, precondiciones y otros elementos [27].

3.3.2 El Modelo Dinámico

El Modelo Dinámico recoge las propiedades dinámicas de los objetos. En el Modelo Dinámico se definen los estados válidos de los objetos y las secuencias de servicios que producen los cambios entre dichos estados. En este modelo, también se especifica las interacciones entre los distintos objetos. Para ello disponemos de dos diagramas:

- Diagrama de Transición de Estados
- Diagrama de Interacción de Objetos

Diagrama de Transición de Estados

El Diagrama de Transición de Estados permite especificar el comportamiento correcto de los objetos, es decir en este diagrama podremos ver los diferentes estados en los que un objeto de una clase puede encontrarse desde que es creado hasta que se destruye, así como las transiciones que ocurren para llegar a ellos. Cada una de dichas transiciones entre estados, estará asociada a un servicio del propio objeto. La Figura 5 muestra un ejemplo de Diagrama de Transición de Estados.

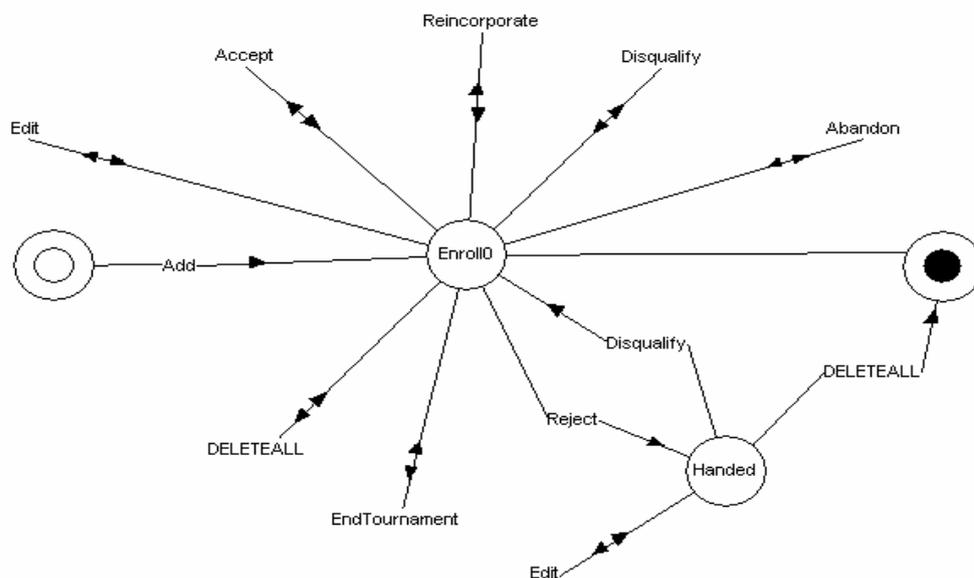


Figura 5. Ejemplo del Diagrama de Transición de Estados

Para poder especificar el Diagrama de Transición de Estados de una clase, es necesario tener definidos previamente los servicios, puesto que son estos los que provocan las transiciones. Así pues, este diagrama se realiza en un instante posterior a la elaboración del Diagrama de Clases.

Diagrama de Interacción de Objetos

En el Diagrama de Interacción de Objetos recoge la manera de interactuar que tienen los objetos en el sistema. Mediante este diagrama, el usuario de la herramienta especifica:

- **Disparos.** Cuando se satisface una determinada condición sobre el estado de cierto objeto, se puede provocar la ejecución de servicios de este mismo objeto o de otros objetos del sistema. A éste método de interacción entre objetos se le denomina disparo. Para su definición se deberá seleccionar el objeto de origen o definición del disparo, el objeto u objetos destino del disparo y el servicio que se ejecutará al activarse el disparo.
- **Servicios Globales.** Pueden ser Transacciones Globales u Operaciones Globales. Es posible definir transacciones u operaciones que engloben a servicios de diversos objetos (usualmente, de clases distintas no relacionadas), para ello se dispone de los servicios globales. Dichos servicios globales se definen mediante una fórmula en la cual se especifica la secuencia de servicios que deben ser ejecutados. [27]

3.3.3 El Modelo Funcional

El Modelo Funcional especifica la forma en que la ejecución de eventos modifica el estado de los objetos mediante evaluaciones (fórmulas bien formadas que expresan cómo cambia el valor de un atributo variable ante la ocurrencia de un evento). El Modelo Funcional contiene las evaluaciones para un atributo variable y un evento de una determinada clase. Las evaluaciones únicamente se pueden definir para eventos, nunca para transacciones y operaciones, puesto que los servicios moleculares producen el cambio de estado en los objetos al enlazar la ejecución de diversos eventos.

Un atributo podrá ser modificado por varios eventos y un evento podrá modificar a varios atributos. El evento puede modificar al atributo de diversas formas, puede tener en cuenta el valor anterior del mismo, puede incrementar, decrementar o inicializar su valor, o también puede asignarle un valor directamente. Al tener dependencia de las propiedades estáticas de las clases, este modelo también se define con posterioridad al Diagrama de Clases [27].

3.3.4 El Modelo de Presentación

OLIVANOVA Modeler proporciona la posibilidad de especificar la interfaz de usuario del sistema mediante un cuarto modelo, el Modelo de Presentación. Este modelo, que no está incluido en UML y que diferencia positivamente a OLIVANOVA Modeler de las otras herramientas de modelado, tiene como finalidad proporcionar al analista un medio para la captura de los requisitos de interfaz de usuario. De esta manera, permite especificar interfaces abstractas y homogéneas, independientes de consideraciones de diseño.

El Modelo de presentación ofrece los patrones de interfaz necesarios para especificar cualquier aplicación de gestión. Mediante estos patrones se recogerán todos los requisitos de la interfaz:

- **Presentación.** El formato en que los datos se muestran al usuario.
- **Navegación.** Información relacionada al medio de acceso a los objetos
- **Búsqueda.** Facilidad para encontrar la información
- **Visibilidad.** Los permisos de los usuarios para acceder a la información.
- **Acceso.** Los permisos asociados a las acciones que proporciona el sistema.

La combinación de estos patrones permite definir la interfaz de la aplicación tal y como la utilizará el usuario final, de una manera intuitiva y sencilla.

El modelo de presentación se estructura en tres niveles de trabajo organizados, de lo general a lo particular. La Figura 6 esquematiza el Modelo de Presentación; se pueden observar los distintos niveles, los elementos que los componen y qué elementos de pueden formar parte de otros de nivel superior [22] [27].

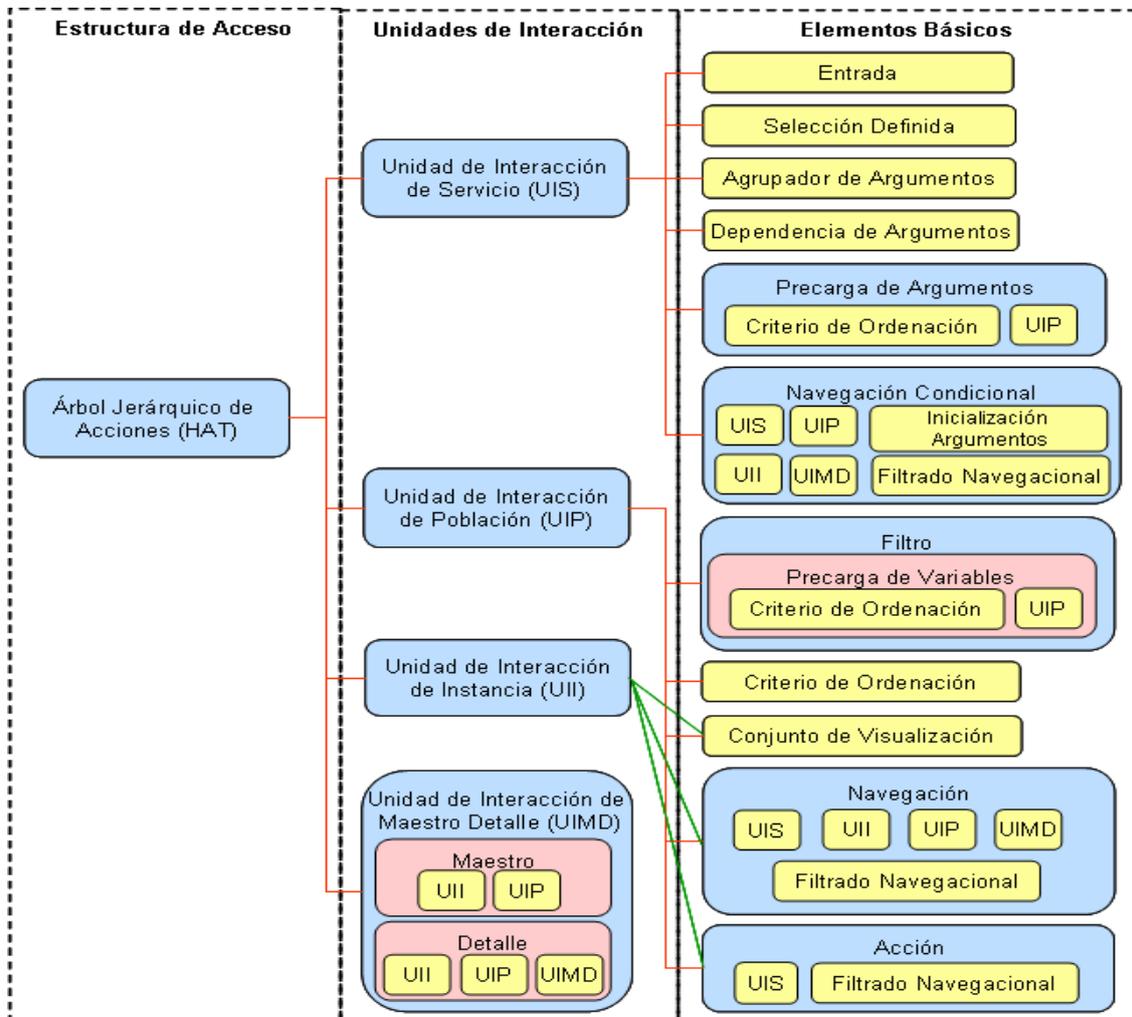


Figura 6. Niveles y elementos del modelo de presentación [22].

NIVEL 1. El primer nivel contiene el Árbol de Jerarquía de Acciones (Hierarchy Action Tree – HAT en inglés), que permite organizar la manera en que el usuario accederá a la aplicación. Mediante el Árbol de Jerarquía de Acciones se podrá definir el menú de acceso de los diversos tipos de usuarios de la aplicación.

NIVEL 2. El segundo nivel contiene las Unidades de Interacción del sistema, los cuales especifican las características de los diálogos de la aplicación generada. La interfaz de usuario se descompone en diversos escenarios que permitirán al usuario ejecutar sus tareas. Las tareas están representadas por patrones del primer nivel de trabajo y son los elementos con los que interactuará el usuario, pero se agrupan en unidades de interacción para presentar la información de forma compacta. Las unidades de interacción pueden ser:

- *Unidades de Interacción de Servicio (UIS):* Representan la interacción entre el usuario de la aplicación y la ejecución de un servicio del sistema.
- *Unidades de Interacción de Instancia (UIS):* Representa la interacción entre el usuario de la aplicación y la información de un objeto específico.
- *Unidades de Interacción de Población (UIP):* Representa la interacción entre el usuario de la aplicación y la información de varios objetos de una clase.
- *Unidades de Interacción de Maestro Detalle (UIMD):* Representa la interacción entre el usuario de la aplicación y un grupo de unidades de interacción del sistema, siguiendo una estructura de maestro – detalle.

NIVEL 3. Patrones de presentación. Son las piezas básicas sobre las que se construyen las Unidades de Interacción simples. Mediante estos patrones, se pueden expresar requisitos de interfaz precisos. Cada uno de estos patrones, puede participar en diversas Unidades de Interacción (Vea la Figura 6). Estos elementos pueden ser:

- *Entrada:* Este elemento permite capturar los datos que deben ser ingresados por el usuario del sistema.
- *Selección definida:* Este elemento permite al usuario seleccionar un valor desde una colección de datos existente.
- *Agrupador de argumentos:* Este elemento permite definir la forma en que los argumentos de los servicios serán presentados al usuario.
- *Dependencia de argumentos:* Permite definir relaciones de dependencias entre los valores o estado de un argumento de entrada de un servicio y los valores o estado de otros argumentos de entrada del mismo servicio. La dependencia de argumentos utiliza reglas de tipo ECA (Evento – Condición – Acción) para cada argumento de entrada de un servicio, que tienen la siguiente semántica: cuando un evento de

interfaz ocurre en un argumento de entrada (por ejemplo cuando el usuario ingresa un valor) se realiza una acción si se cumple una determinada condición.

- *Precarga de argumentos*: Este elemento permite definir un conjunto de objetos que pueden ser seleccionados como argumentos de entrada de un servicio.
- *Filtro*: Este elemento permite definir una condición de selección sobre una población de objetos de una clase. El filtro puede tener variables dato-valuadas y variables objeto-valuadas que pueden tener definido un valor por defecto, una población asociada para seleccionar los valores de las variables objeto-valuadas y precarga de los valores de las variables objeto-valuadas.
- *Criterio de ordenación*: Este elemento permite definir el orden de los objetos presentados en una población, considerando el orden ascendente/descendente sobre los valores de los atributos de los objetos presentes en la población.
- *Conjunto de visualización*: Permite especificar las propiedades de una clase que serán visibles al usuario, y el orden en que se mostrarán dichas propiedades.
- *Navegación*: Permite la visualización de información de los objetos de las clases relacionadas a un objeto de una clase.
- *Acción*: Permite la ejecución de los servicios definidos en las clases sobre un objeto de la clase.
- *Filtrado Navegacional*: Permite acotar la navegación hacia los objetos de las clases relacionadas a un objeto según una condición de filtro.
- *Inicialización de Argumentos*: Permite asignar un valor inicial a los argumentos de entrada de un servicio que se accede directamente desde otro servicio.
- *Navegación Condicional*: Permite navegar a las unidades de interacción luego de la ejecución de un servicio que haya terminado de manera exitosa o fallida. Para seleccionar la unidad de interacción a la que se navegará, es necesario especificar una condición que debe cumplirse cuando se haya ejecutado el servicio.

4. ESPACIO DEL PROBLEMA

En este proyecto se desea plantear una solución genérica a la gestión de recursos humanos de Administraciones Públicas. Esta solución se divide en dos sistemas de información, el primero de los cuales es una herramienta que permite realizar una estimación del salario de un funcionario dependiendo de ciertos parámetros. Su nombre es **Sistema de Información de Estimación del Salario Mediante el Análisis de Conceptos Retributivos** (en adelante, **SIESMACRe**). El segundo sistema de información es un método de evaluación del funcionariado de una Administración Pública, que recibe el nombre de **Sistema de Información para la Evaluación de Funcionarios de las Administraciones Públicas** (en adelante, **SIEFAP**).

SIESMACRe va dirigido a usuarios anónimos que deseen conocer el salario que tendrían si fuesen funcionarios, dependiendo de parámetros tales como el tipo de puesto, la Comunidad Autónoma y el tamaño de la población para la que trabaja la Administración o el tamaño de su plantilla de trabajadores.

SIEFAP consta de un modelo de evaluación de los empleados de una Administración, con el objetivo de establecer un método de asignación automática de la masa salarial de la Administración. El método de evaluación se basa en la realización de cuestionarios sobre conocimientos por parte del empleado; el método de asignación salarial debe gestionar la información relativa al presupuesto de la Administración y el sueldo que debería tener el empleado de acuerdo con su evaluación de conocimientos. Los usuarios de este segundo sistema son los empleados que se desea evaluar de la Administración, usuarios editores de los cuestionarios, y usuario/s administrador/es.

4.1. Resumen de las aplicaciones

A continuación, se explican brevemente los requisitos que deben cumplir los dos sistemas de información.

4.1.1 SIESMACRe

La aplicación de la estimación del salario de un funcionario debe gestionar la información relativa a los factores relevantes para el cálculo del salario, los cuales tienen las siguientes características:

La **CATEGORÍA** del funcionario indica el salario bruto tanto anual como mensual que se recibe. La categoría puede tomar los valores A, B, C, D o E.

El **COMPLEMENTO DE DESTINO** es otro concepto que tiene un valor en términos de salario anual y mensual que percibe el funcionario. El complemento de destino puede tomar valores de 5 a 30. Dependiendo de la categoría del funcionario este rango puede ser más restringido.

El **COMPLEMENTO ESPECÍFICO** es un concepto salarial más que representa una cantidad extra a recibir por parte del funcionario.

El valor correspondiente a la CATEGORÍA, el correspondiente al COMPLEMENTO DE DESTINO y el COMPLEMENTO ESPECÍFICO se suman para formar el salario del funcionario.

La aplicación resultante informa al usuario del salario calculado y de estadísticas de salarios de trabajadores con puestos similares al suyo, con el objetivo de que pueda comparar sus ingresos con los de funcionarios que desempeñan funciones semejantes. Los parámetros estadísticos son el salario medio, el percentil 10, percentil 25, percentil 50, percentil 75 y percentil 90.

Para poder obtener estas estadísticas, el usuario debe introducir la denominación del **PUESTO DE TRABAJO**, la **COMUNIDAD AUTÓNOMA** y el **NÚMERO DE HABITANTES** de la población para la que trabaja la administración y el tamaño de la **PLANTILLA** de empleados

Del PUESTO DE TRABAJO se debe conocer su nombre, la media del salario y los percentiles 10, 25, 50, 75 y 90.

El resto de parámetros que se requieren no son valores absolutos, sino que representan un porcentaje sobre las cantidades de cada uno de los parámetros estadísticos ya comentados.

La COMUNIDAD AUTÓNOMA tiene un nombre que la identifica y el valor del factor de variación sobre las cantidades absolutas.

El NÚMERO DE HABITANTES se representa como un rango con su límite inferior y superior. A cada rango le corresponde un factor de variación sobre las cantidades absolutas.

La PLANTILLA se representa también como un rango de empleados, en el que a cada rango le corresponde un factor de variación.

Esta aplicación se diseña con la intención de que cualquier usuario anónimo pueda seleccionar los valores que desee de los parámetros anteriores y conocer la estimación y estadísticas del salario a partir de lo introducido.

Para que esto sea posible, debe existir una funcionalidad extra para usuario/s administrador/es que permita dar de alta, dar de baja o modificar propiedades de CATEGORÍAS, COMPLEMENTOS DE DESTINO, PUESTOS DE TRABAJO, COMUNIDADES AUTÓNOMAS, NÚMERO DE HABITANTES Y tamaño de las PLANTILLAS de empleados.

4.1.2 SIEFAP

El Sistema de Información para la Evaluación de Funcionarios de la Administración Pública (SIEFAP) está definido de acuerdo a la siguiente descripción de características y funcionalidades:

Los **EMPLEADOS** se identifican por su NIF y se necesita al menos la información de su nombre, apellidos, fecha de nacimiento y sexo. El EMPLEADO trabaja para un **DEPARTAMENTO**, que a su vez pertenece a una **ADMINISTRACIÓN**.

La información que se almacena del DEPARTAMENTO es su nombre, el presupuesto asignado para Recursos Humanos y el número de EMPLEADOS que trabajan en él. La información que se tiene de la ADMINISTRACIÓN es su nombre, el presupuesto del que dispone para Recursos Humanos, el número de DEPARTAMENTOS en los que se divide y el número de EMPLEADOS que trabajan en ella.

A un EMPLEADO se le pueden realizar distintas **EVALUACIONES** de sus conocimientos. Una EVALUACIÓN contiene el número de puntos máximo y los obtenidos por el EMPLEADO, la fecha de creación, la fecha de finalización y un indicador de si la EVALUACIÓN está finalizada.

Para representar un **CONOCIMIENTO** simplemente es necesario su nombre y descripción, pero un CONOCIMIENTO puede tener **SUBCONOCIMIENTOS** asociados. Por ejemplo, podría existir un CONOCIMIENTO llamado "Aptitudes generales" que se dividiese en los SUBCONOCIMIENTOS "Matemáticas", "Lectura y comprensión" e "Informática".

La evaluación de los CONOCIMIENTOS de los EMPLEADOS se realiza por medio de **CUESTIONARIOS**. Los CUESTIONARIOS se responden en base a una **PLANTILLA DE CUESTIONARIO** que debe estar creada en el sistema antes que se empiece a responder. Cada CUESTIONARIO lo realiza un EMPLEADO y sus respuestas las debe validar un **EMPLEADO SUPERVISOR**.

El EMPLEADO SUPERVISOR tiene las mismas características que un EMPLEADO, pero es necesario que valide y/o corrija las respuestas de los CUESTIONARIOS de los EMPLEADOS que tiene a cargo.

La información que se almacena del CUESTIONARIO es la fecha de creación, la fecha en que el EMPLEADO comienza a realizar el CUESTIONARIO, la fecha en la que lo finaliza y las fechas en las que el EMPLEADO SUPERVISOR sugiere una corrección o valida las respuestas. También se debe conocer el estado en el que se encuentra el cuestionario, el peso que tiene dentro de la evaluación del conocimiento, la puntuación obtenida por el EMPLEADO que lo realiza y el porcentaje de puntos obtenidos respecto del total del CUESTIONARIO.

Los estados pueden ser "En Edición", en el que se puede asociar el CUESTIONARIO a la VALORACIÓN; "Preparado", que indica que está listo para comenzarse; "En Respuesta", en el que el EMPLEADO responde a las preguntas; "Finalizado", en el que el EMPLEADO SUPERVISOR

puede validarlo o solicitar una corrección; y “Validado”, cuando el resultado del CUESTIONARIO ya es definitivo.

Como ya se ha comentado, un CUESTIONARIO se basa en una PLANTILLA DE CUESTIONARIO. Una PLANTILLA siempre pertenece a un CONOCIMIENTO o SUBCONOCIMIENTO. La PLANTILLA DE CUESTIONARIO está formada por **PREGUNTAS** que contienen **RESPUESTAS** entre las que el EMPLEADO puede elegir. Las PREGUNTAS, a su vez, también deben pertenecer a uno o varios CONOCIMIENTOS o SUBCONOCIMIENTOS, con el objetivo de que puedan ser reutilizadas para diversas PLANTILLAS DE CUESTIONARIO.

Las propiedades de una PLANTILLA DE CUESTIONARIO son su nombre, descripción, el número de PREGUNTAS que contiene, la puntuación máxima que se puede obtener y un estado que indica si la PLANTILLA está empleándose para CUESTIONARIOS comenzados.

Las propiedades de una PREGUNTA son el texto de la pregunta, la puntuación máxima que se puede obtener con la pregunta, el número de RESPUESTAS que tiene asignadas, el número de CONOCIMIENTOS a los que pertenece y un estado que indica si la PREGUNTA se está empleando en CUESTIONARIOS.

En cuanto a las RESPUESTAS, se debe conocer su texto y los puntos que proporciona su elección.

Las PREGUNTAS pueden pertenecer a PLANTILLAS DE CUESTIONARIO, pero no es obligatorio. Por tanto, es posible tener PREGUNTAS creadas en el sistema asociadas a un CONOCIMIENTO que no se estén empleando en PLANTILLAS DE CUESTIONARIO.

Lo mismo sucede con las PLANTILLAS DE CUESTIONARIO. Pueden ser creadas, pero no necesariamente deben estar empleándose en algún CUESTIONARIO.

Cuando una PREGUNTA se incluye en una PLANTILLA DE CUESTIONARIO, se debe decidir el peso que tiene dicha PREGUNTA en la PLANTILLA. De igual manera, cuando una PLANTILLA DE CUESTIONARIO, se emplea para una EVALUACIÓN vía CUESTIONARIO, se debe decidir el peso que tiene dicho CUESTIONARIO dentro de la EVALUACIÓN.

Para facilitar la EVALUACIÓN de un EMPLEADO, ésta debe ser la suma de tantas **EVALUACIONES DE CONOCIMIENTO** como CONOCIMIENTOS se deseen evaluar. De las EVALUACIONES DE CONOCIMIENTO, se debe conocer el número de CUESTIONARIOS totales que se necesitan, el número de CUESTIONARIOS pendientes, finalizados y validados. También se debe conocer el porcentaje de puntos que representa la EVALUACIÓN DEL CONOCIMIENTO dentro de la EVALUACIÓN global y los puntos máximos que se pueden obtener en ella.

En una EVALUACIÓN DE CONOCIMIENTO, el EMPLEADO obtiene un nivel que depende del porcentaje de puntos conseguidos en los CUESTIONARIOS con respecto a los totales. Este nivel, a su vez, se traduce en los **PUNTOS** que se suman para obtener la EVALUACIÓN global.

En esta aplicación puede haber estos 4 tipos de usuarios:

- El EMPLEADO, que es el que se quiere evaluar y responde a los CUESTIONARIOS relativos a sus CONOCIMIENTOS.
- El EMPLEADO puede convertirse en EMPLEADO SUPERVISOR de otros. Su función es validar que las RESPUESTAS de los EMPLEADOS que tiene a su cargo son verídicas.
- El **USUARIO EDITOR** es quien prepara el contenido de los CUESTIONARIOS, es decir, las PREGUNTAS y PLANTILLAS DE CUESTIONARIO asociados a CONOCIMIENTOS.
- El USUARIO ADMINISTRADOR tiene las funciones clásicas de un administrador del sistema de información en cuanto a la gestión de la información que contiene. Además, es el encargado de preparar las EVALUACIONES, asignar los CUESTIONARIOS que corresponden a cada EMPLEADO.

4.2. Especificación de Requisitos

En este apartado se desarrolla la Especificación de Requisitos de los dos Sistemas que Información que se construyen en el proyecto. La técnica elegida para realizarlo son los Casos de Uso.

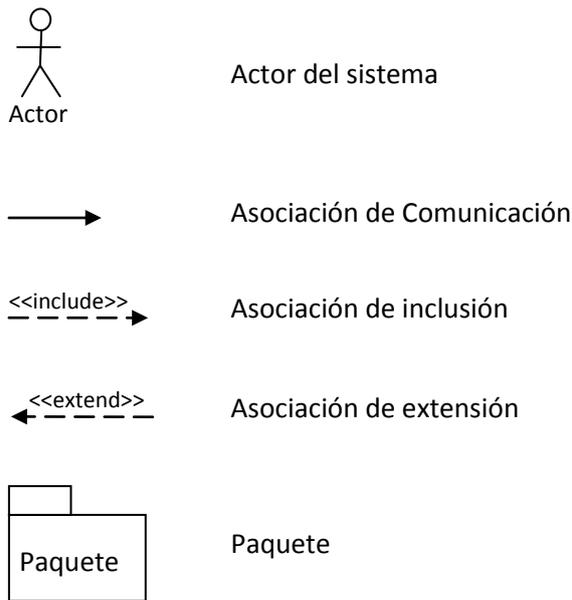
En el primer punto del apartado, se explica la notación utilizada en los Casos de Uso. El segundo punto es la Especificación de Requisitos del **Sistema de Información de Estimación del Salario Mediante Análisis de Conceptos Retributivos (SIESMACRe)**. Finalmente, el tercer punto aborda la Especificación de Requisitos del **Sistema de Información de la Evaluación de Funcionarios de Administraciones Públicas (SIEFAP)**.

4.2.1. Diagrama de Casos de Uso

Para realizar una Especificación de Requisitos de forma precisa, se va a emplear la técnica de Casos de Uso. Los casos de uso son típicamente utilizados para indicar los usos de un sistema [28], es decir, para indicar lo que debe hacer el sistema. Así, el Diagrama de casos de uso permite dar una vista general del funcionamiento del Sistema de Información.

La notación que se sigue en los diagramas de Casos de Uso es la siguiente:





Un **Caso de Uso** representa una unidad de funcionalidad del sistema. [29]

Un **actor** es una entidad que interactúa con el sistema para llevar a cabo una acción completa que le ayuda a conseguir un objetivo. [30]

Las relaciones en los diagramas de casos de uso pueden ser de tres tipos:

- La **comunicación** representa la relación existente entre un actor y un caso de uso, que denota la participación de dicho actor en el caso de uso. [31]
- La **inclusión** representa la dependencia entre dos casos de uso e indica que el comportamiento de un caso de uso se incluye en el otro. [31]
- La **extensión** representa también dependencia entre dos casos de uso, pero indica que uno de los casos de uso es una especialización del otro, es decir, que puede ocurrir bajo ciertas circunstancias, pero no de forma obligatoria. [31]

Finalmente, el **Paquete** es un mecanismo que engloba partes del diagrama de Casos de Uso que guardan relación entre ellos [29], con el objetivo de simplificar y facilitar el entendimiento del sistema que se describe.

En este documento, los casos de uso se especifican a través de plantillas, que especifican los usuarios que intervienen en cada uno de los casos, el propósito de los mismos, la forma en que cada caso de uso interactúa con el sistema por medio de sus entradas y salidas, las precondiciones que se tienen que cumplir para que suceda el caso de uso y las postcondiciones o efectos que tiene el caso de uso sobre el sistema.

4.2.2. Casos de Uso SIESMACRe

SIESMACRe (Sistema de Información de Estimación del Salario Mediante Análisis de Conceptos Retributivos) se diseña con el objetivo de que cualquier persona pueda consultar el Salario aproximado que tendría si decidiese trabajar para la Administración Pública, concretamente en ayuntamientos.

La Especificación de requisitos comprende la parte funcional del usuario final y la parte correspondiente al usuario administrador del sistema.

En la Figura 7 se presentan los actores del sistema: Usuario Administrador y Usuario Anónimo, y los paquetes de Casos de Uso del sistema.

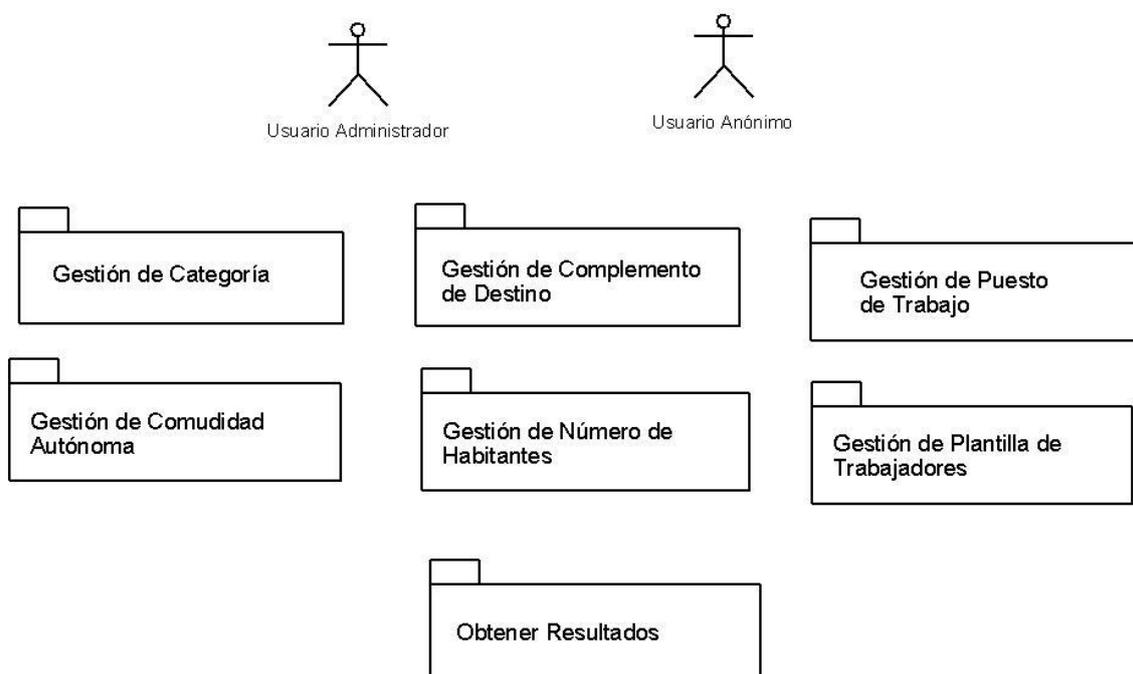


Figura 7. Diagrama de casos de uso de SIESMACRe – Vista General.

Una vez introducido el esquema general, se detallan cada uno de los paquetes que conforman **SIESMACRe**.

CASOS DE USO REFERENTES A LA GESTIÓN DE CATEGORÍA

La categoría o grado es la principal característica que diferencia a los funcionarios de las administraciones públicas. Usualmente, la categoría se corresponde con el nivel de estudios exigido para optar al puesto de trabajo y se dividen en estos 5 tipos:

- A: exige una Licenciatura o una Ingeniería.
- B: exige una Diplomatura o una Ingeniería técnica
- C: exige el Bachillerato

- D: exige el Graduado escolar
- E: no exige estudios

La Especificación de Requisitos para la gestión de información referente a la categoría se define en el paquete “Gestión de Categoría”. La Figura 8 presenta el diagrama de casos de uso de la gestión de categoría.

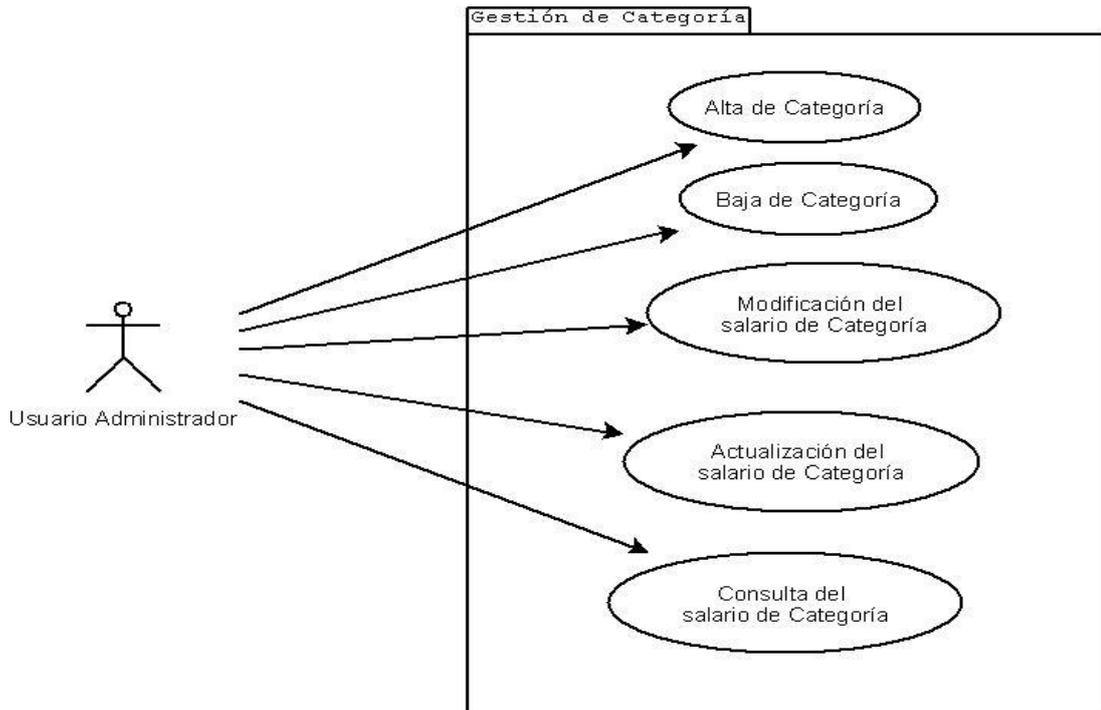


Figura 8. Diagrama de casos de uso para la gestión de categoría.

Tabla 2. Caso de uso Alta de Categoría

CASO DE USO	1. Alta de Categoría.
ACTORES	Usuario Administrador.
PROPÓSITO	Crear una categoría.
RESUMEN	Se crea una categoría con las propiedades de tipo, salario anual y salario mensual
PRECONDICIONES	- El salario mensual debe ser mayor que cero.
POSTCONDICIONES	- Los datos relativos a la categoría quedan almacenados en el sistema. - El salario anual se calcula a partir del salario mensual de entrada.
ENTRADAS	El tipo de categoría y el salario mensual que le corresponde.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 3. Caso de uso Baja de Categoría

CASO DE USO	2. Baja de Categoría.
ACTORES	Usuario Administrador.
PROPÓSITO	Dar de baja una categoría.
RESUMEN	La categoría se elimina del sistema.
PRECONDICIONES	La categoría debe existir en el sistema.
POSTCONDICIONES	Los datos relativos a la categoría se eliminan del sistema.
ENTRADAS	El tipo de la categoría.
SALIDAS	Mensaje de confirmación de la baja.

Tabla 4. Caso de uso Modificación de salario de Categoría

CASO DE USO	3. Modificación de salario de Categoría.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar las propiedades del salario de la categoría
RESUMEN	Se modifican el salario mensual y el salario anual de la categoría.
PRECONDICIONES	<ul style="list-style-type: none"> - La categoría debe existir en el sistema. - El salario mensual de entrada debe ser mayor que cero.
POSTCONDICIONES	<ul style="list-style-type: none"> - El salario mensual toma el valor de la entrada. - El salario anual se calcula a partir del salario mensual.
ENTRADAS	El tipo de la categoría y el nuevo valor del salario mensual.
SALIDAS	Ninguna.

Tabla 5. Caso de uso Actualización de salario de Categoría

CASO DE USO	4. Actualización de salario de Categoría.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar de las propiedades del salario mensual y anual de la categoría
RESUMEN	Se modifican los salarios anual y mensual mediante un porcentaje al alza o a la baja.
PRECONDICIONES	<ul style="list-style-type: none"> - La categoría debe existir en el sistema y los salarios mensual y anual deben tener valor. - El valor del porcentaje debe ser mayor que cero.
POSTCONDICIONES	<ul style="list-style-type: none"> - El salario mensual y el anual modifican su valor. - El salario mensual y el anual resultantes deben ser mayor que cero.
ENTRADAS	El tipo de la categoría, el porcentaje de variación y un indicador de si es al alza o a la baja.
SALIDAS	Ninguna.

Tabla 6. Caso de uso consulta de salario de Categoría

CASO DE USO	5. Consulta de salario de Categoría.
ACTORES	Usuario Administrador.
PROPÓSITO	Consultar las propiedades del salario mensual o anual de la categoría.
RESUMEN	Se consulta el salario mensual o el salario anual de la categoría.
PRECONDICIONES	La categoría debe existir en el sistema.
POSTCONDICIONES	Ninguna.
ENTRADAS	El tipo de la categoría.
SALIDAS	El valor del salario mensual o anual de la categoría.

CASOS DE USO REFERENTES A LA GESTIÓN DE COMPLEMENTO DE DESTINO

El Complemento del Destino es otro de los factores que componen el sueldo de un funcionario. El funcionario, además de su Categoría, tiene un nivel de complemento de destino, dependiendo de la experiencia adquirida o de las competencias específicas para el trabajo que desempeña.

La **Figura 9** muestra los casos de uso de la gestión del complemento de destino.

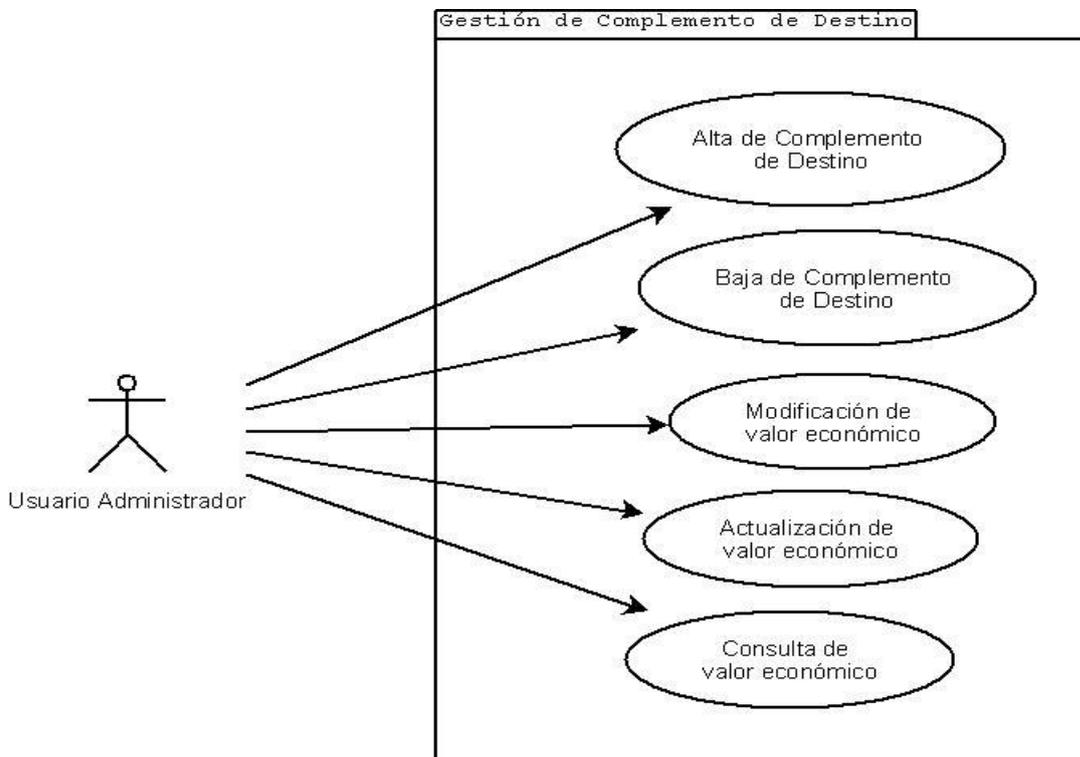


Figura 9. Diagrama de casos de uso para la gestión de complemento de destino.

Tabla 7. Caso de uso Alta de Complemento de Destino

CASO DE USO	6. Alta de Complemento de Destino.
ACTORES	Usuario Administrador.
PROPÓSITO	Crear un complemento de destino.
RESUMEN	Se crea un complemento de destino con las propiedades de tipo, valor económico mensual y valor económico anual.
PRECONDICIONES	El valor económico mensual debe ser mayor que cero.
POSTCONDICIONES	<ul style="list-style-type: none"> - Los datos relativos al complemento de destino quedan almacenados en el sistema. - El valor económico anual se calcula a partir del mensual.
ENTRADAS	El tipo y el valor mensual correspondiente al tipo de complemento de destino.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 8. Caso de uso Baja de Complemento de Destino

CASO DE USO	7. Baja de Complemento de Destino.
ACTORES	Usuario Administrador.
PROPÓSITO	Dar de baja un complemento de destino.
RESUMEN	El complemento de destino se elimina del sistema.
PRECONDICIONES	El complemento de destino debe existir en el sistema.
POSTCONDICIONES	Los datos relativos al complemento de destino se eliminan del sistema.
ENTRADAS	El tipo del complemento de destino.
SALIDAS	Mensaje de confirmación de la baja.

Tabla 9. Caso de uso Modificación de valor económico

CASO DE USO	8. Modificación de valor económico.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar las propiedades del valor económico del complemento de destino
RESUMEN	Se modifican el valor económico mensual el anual del complemento de destino.
PRECONDICIONES	<ul style="list-style-type: none"> - El complemento de destino debe existir en el sistema. - El valor económico mensual de entrada debe ser mayor que cero
POSTCONDICIONES	<ul style="list-style-type: none"> - El valor económico mensual toma el valor de la entrada. - El valor económico anual se calcula a partir del mensual.
ENTRADAS	El tipo del complemento de destino y el nuevo valor mensual.
SALIDAS	Ninguna.

Tabla 10. Caso de uso Actualización de valor económico

CASO DE USO	9. Actualización de valor económico.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar las propiedades del valor económico.
RESUMEN	Se modifica el valor económico anual y mensual mediante un porcentaje al alza o a la baja
PRECONDICIONES	<ul style="list-style-type: none"> - El complemento de destino debe existir en el sistema y el valor mensual y anual no puede ser cero. - El valor del porcentaje debe ser mayor que cero.
POSTCONDICIONES	<ul style="list-style-type: none"> - El valor mensual y el anual se modifican. - El valor mensual y el anual resultantes deben ser mayores que cero.
ENTRADAS	El tipo del complemento de destino, el porcentaje de variación y un indicador de si es al alza o a la baja.
SALIDAS	Ninguna.

Tabla 11. Caso de uso Consulta de valor económico

CASO DE USO	10. Consulta de valor económico.
ACTORES	Usuario Administrador.
PROPÓSITO	Consultar las propiedades del valor económico mensual o anual del complemento de destino.
RESUMEN	Se consulta el valor económico mensual o anual del complemento de destino.
PRECONDICIONES	El complemento de destino debe existir en el sistema.
POSTCONDICIONES	Ninguna.
ENTRADAS	El tipo del complemento de destino.
SALIDAS	El valor económico anual o mensual del complemento de destino.

CASOS DE USO REFERENTES A LA GESTIÓN DE PUESTO DE TRABAJO

El puesto de trabajo es uno de los parámetros en los que se basa la estimación del salario. El sistema debe almacenar el nombre y valores estadísticos del salario.

Para gestionar esa información, se especifican los casos de uso de la **Figura 10**.

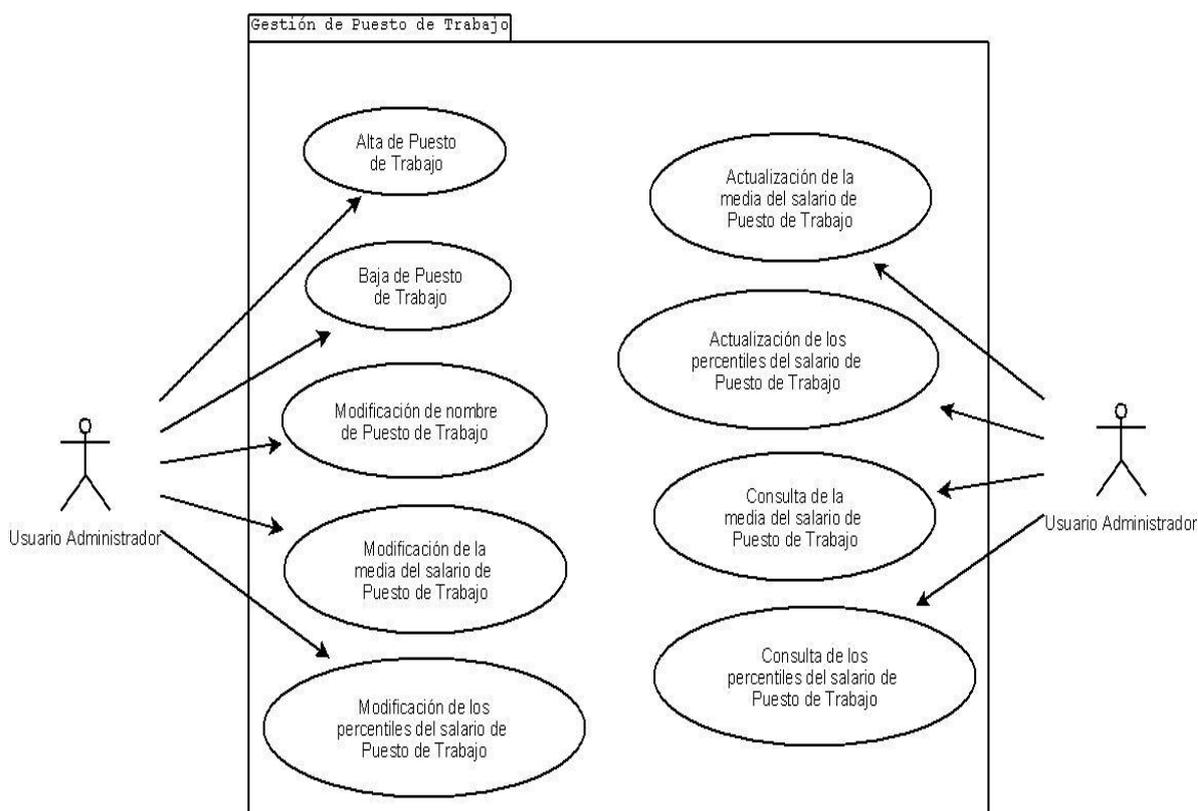


Figura 10. Diagrama de casos de uso para la gestión de puesto de trabajo.

Tabla 12. Caso de uso Alta de Puesto de Trabajo

CASO DE USO	11. Alta de Puesto de Trabajo.
ACTORES	Usuario Administrador.
PROPÓSITO	Crear un puesto de trabajo.
RESUMEN	Se crea un puesto de trabajo con las propiedades de nombre, media de salario y los percentiles 10, 25, 50, 75 y 90 del salario.
PRECONDICIONES	Ninguna.
POSTCONDICIONES	<ul style="list-style-type: none"> - Los datos relativos al Puesto de Trabajo quedan almacenados en el sistema. - Media > 0 - Percentil 10 > 0 - Percentil 10 <= Percentil 25 - Percentil 25 <= Percentil 50 - Percentil 50 <= Percentil 75 - Percentil 75 <= Percentil 90
ENTRADAS	El nombre, la media del salario y los percentiles 10, 25, 50, 75 y 90 del salario.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 13. Caso de uso Baja de puesto de Trabajo

CASOS DE USO	12. Baja de Puesto de Trabajo.
ACTORES	Usuario Administrador.
PROPÓSITO	Dar de baja un puesto de trabajo.
RESUMEN	El puesto de trabajo se elimina del sistema.
PRECONDICIONES	El puesto de trabajo debe existir en el sistema.
POSTCONDICIONES	Los datos relativos al puesto de trabajo se eliminan del sistema.
ENTRADAS	El Puesto de Trabajo.
SALIDAS	Mensaje de confirmación de la baja.

Tabla 14. Caso de uso Modificación de nombre de Puesto de Trabajo

CASO DE USO	13. Modificación de nombre de Puesto de Trabajo
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar el nombre de puesto de trabajo
RESUMEN	Se modifica el nombre del puesto de trabajo.
PRECONDICIONES	El puesto de trabajo debe existir en el sistema.
POSTCONDICIONES	El nombre del puesto de trabajo cambia por el nuevo.
ENTRADAS	El puesto de trabajo y su nuevo nombre.
SALIDAS	Mensaje de confirmación del cambio.

Tabla 15. Caso de uso Modificación de la media del salario del Puesto de Trabajo

CASO DE USO	14. Modificación de la media del salario del Puesto de Trabajo.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar la media del salario del puesto de trabajo
RESUMEN	Se modifica la media del salario del puesto de trabajo
PRECONDICIONES	El puesto de trabajo debe existir en el sistema.
POSTCONDICIONES	- La media del salario toma el valor introducido. - La media del salario debe ser mayor que cero
ENTRADAS	El puesto de trabajo y la nueva media del salario.
SALIDAS	Mensaje de confirmación del cambio.

Tabla 16. Caso de uso Modificación de los percentiles del salario del Puesto de Trabajo

CASO DE USO	15. Modificación de los percentiles del salario del Puesto de Trabajo.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar los percentiles del salario del puesto de trabajo
RESUMEN	Se modifican los percentiles 10, 25, 50, 75 y 90 del salario del puesto de trabajo.
PRECONDICIONES	El puesto de trabajo debe existir en el sistema.
POSTCONDICIONES	<ul style="list-style-type: none"> - Los nuevos valores de los percentiles quedan almacenados en el sistema. - Percentil 10 > 0 - Percentil 10 <= Percentil 25 - Percentil 25 <= Percentil 50 - Percentil 50 <= Percentil 75 - Percentil 75 <= Percentil 90
ENTRADAS	El Puesto de Trabajo y los nuevos valores de los percentiles.
SALIDAS	Ninguna.

Tabla 17. Caso de uso Actualización de la media del salario

CASO DE USO	16. Actualización de la media del salario.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar la media del salario del puesto de trabajo.
RESUMEN	Se modifica la media según un porcentaje al alza o a la baja.
PRECONDICIONES	<ul style="list-style-type: none"> - El puesto de trabajo debe existir en el sistema. - El porcentaje debe ser mayor que cero.
POSTCONDICIONES	<ul style="list-style-type: none"> - La media del salario se modifica. - La media del salario debe ser mayor que cero.
ENTRADAS	El puesto de trabajo, el porcentaje de variación y un indicador de si es al alza o a la baja.
SALIDAS	Ninguna.

Tabla 18. Caso de uso Actualización de los percentiles del salario

CASO DE USO	17. Actualización de los percentiles del salario.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar los percentiles del salario del puesto de trabajo.
RESUMEN	Se modifican los percentiles según un porcentaje al alza o a la baja.
PRECONDICIONES	- El puesto de trabajo debe existir en el sistema. - El porcentaje debe ser mayor que cero.
POSTCONDICIONES	- Los percentiles del salario se modifican. - Los percentiles deben ser mayor que cero.
ENTRADAS	El Puesto de Trabajo, el porcentaje de variación y un indicador de si es al alza o a la baja.
SALIDAS	Ninguna.

Tabla 19. Caso de uso Consulta de la media del salario

CASO DE USO	18. Consulta de la media del salario
ACTORES	Usuario Administrador.
PROPÓSITO	Consultar la media del salario.
RESUMEN	Se consulta la media del salario del puesto de trabajo.
PRECONDICIONES	El puesto de trabajo debe existir en el sistema.
POSTCONDICIONES	Ninguna.
ENTRADAS	El puesto de trabajo.
SALIDAS	La media del salario del puesto de trabajo.

Tabla 20. Caso de uso Consulta de los percentiles del salario

CASO DE USO	19. Consulta de los percentiles del salario
ACTORES	Usuario Administrador.
PROPÓSITO	Consultar los percentiles del salario.
RESUMEN	Se consultan los percentiles del salario del puesto de trabajo.
PRECONDICIONES	El puesto de trabajo debe existir en el sistema.
POSTCONDICIONES	Ninguna.
ENTRADAS	El puesto de trabajo.
SALIDAS	Los percentiles del salario del puesto de trabajo.

CASOS DE USO REFERENTES A LA GESTIÓN DE COMUNIDAD AUTÓNOMA

La Comunidad Autónoma en la que se trabaja añade un factor que hace variar la estimación del salario. El sistema debe almacenar el nombre y el valor de dicho factor de variación.

Para gestionar esa información, se especifican los casos de uso de la **Figura 11**.

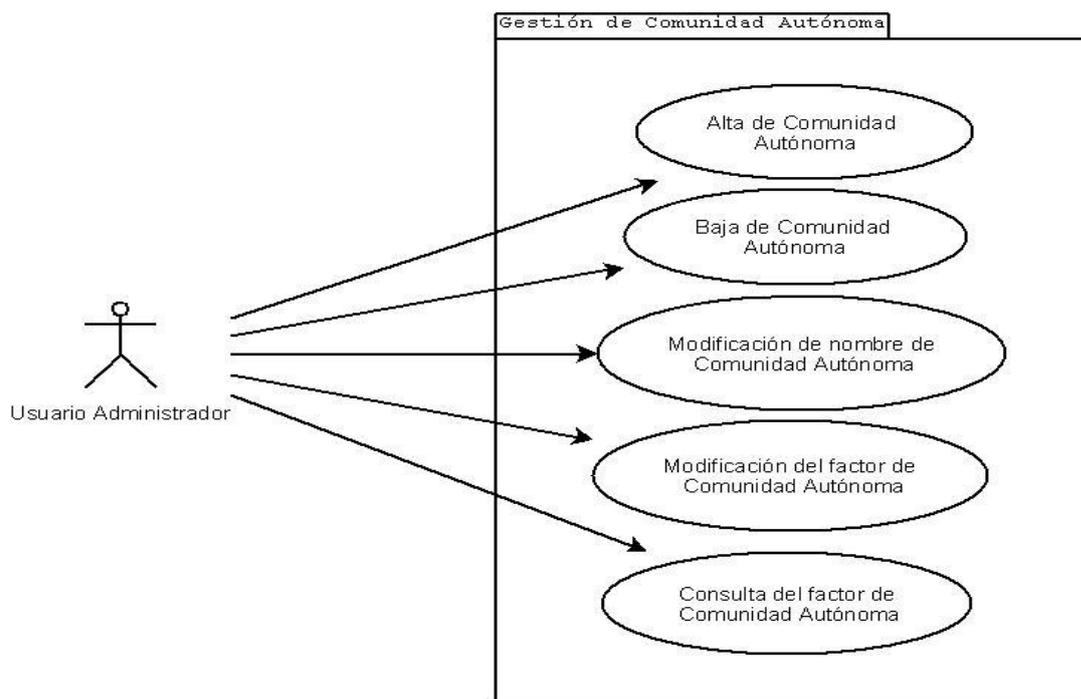


Figura 11. Diagrama de casos de uso para la gestión de Comunidad Autónoma.

Tabla 21. Caso de uso Alta de Comunidad Autónoma

CASO DE USO	20. Alta de Comunidad Autónoma.
ACTORES	Usuario Administrador.
PROPÓSITO	Crear una Comunidad Autónoma.
RESUMEN	Se crea una Comunidad Autónoma con su nombre y el factor de variación sobre el salario.
PRECONDICIONES	Ninguna.
POSTCONDICIONES	Los datos relativos a la Comunidad Autónoma quedan almacenados en el sistema.
ENTRADAS	El nombre y el factor de variación.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 22. Caso de uso Baja de Comunidad Autónoma

CASO DE USO	21. Baja de Comunidad Autónoma.
ACTORES	Usuario Administrador.
PROPÓSITO	Dar de baja una Comunidad Autónoma.
RESUMEN	La Comunidad Autónoma se elimina del sistema.
PRECONDICIONES	La Comunidad Autónoma debe existir en el sistema.
POSTCONDICIONES	Los datos relativos a la Comunidad Autónoma se eliminan del sistema.
ENTRADAS	La Comunidad Autónoma.
SALIDAS	Mensaje de confirmación de la baja.

Tabla 23. Caso de uso Modificación de nombre de Comunidad Autónoma

CASO DE USO	22. Modificación de nombre de Comunidad Autónoma.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar el nombre de Comunidad Autónoma.
RESUMEN	Se modifica el nombre de la Comunidad Autónoma.
PRECONDICIONES	La Comunidad Autónoma debe existir en el sistema.
POSTCONDICIONES	El nombre de la C.A cambia por el nuevo.
ENTRADAS	La Comunidad Autónoma y el nuevo nombre.
SALIDAS	Mensaje de confirmación del cambio.

Tabla 24. Caso de uso Modificación del factor de Comunidad Autónoma

CASO DE USO	23. Modificación del factor de Comunidad Autónoma.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar el factor de variación de la Comunidad Autónoma.
RESUMEN	Se modifica el factor de variación de la Comunidad Autónoma.
PRECONDICIONES	La Comunidad Autónoma debe existir en el sistema.
POSTCONDICIONES	El factor de variación toma el valor introducido.
ENTRADAS	La Comunidad Autónoma y el nuevo factor de variación.
SALIDAS	Ninguna.

Tabla 25. Caso de uso Consulta del factor de Comunidad Autónoma

CASO DE USO	24. Consulta del factor de Comunidad Autónoma.
ACTORES	Usuario Administrador.
PROPÓSITO	Consultar el factor de variación de la Comunidad Autónoma.
RESUMEN	Se consulta el factor de variación de la Comunidad Autónoma.
PRECONDICIONES	La Comunidad Autónoma debe existir en el sistema.
POSTCONDICIONES	Ninguna.
ENTRADAS	La Comunidad Autónoma.
SALIDAS	El valor del factor de variación de la Comunidad Autónoma.

CASOS DE USO REFERENTES A LA GESTIÓN DE NÚMERO DE HABITANTES

El número de habitantes de la población para la que se trabaja añade un factor que hace variar la estimación del salario. El sistema debe almacenar intervalos de número de habitantes y el valor de dicho factor de variación.

Para gestionar esa información, se especifican los casos de uso de la **Figura 12**.

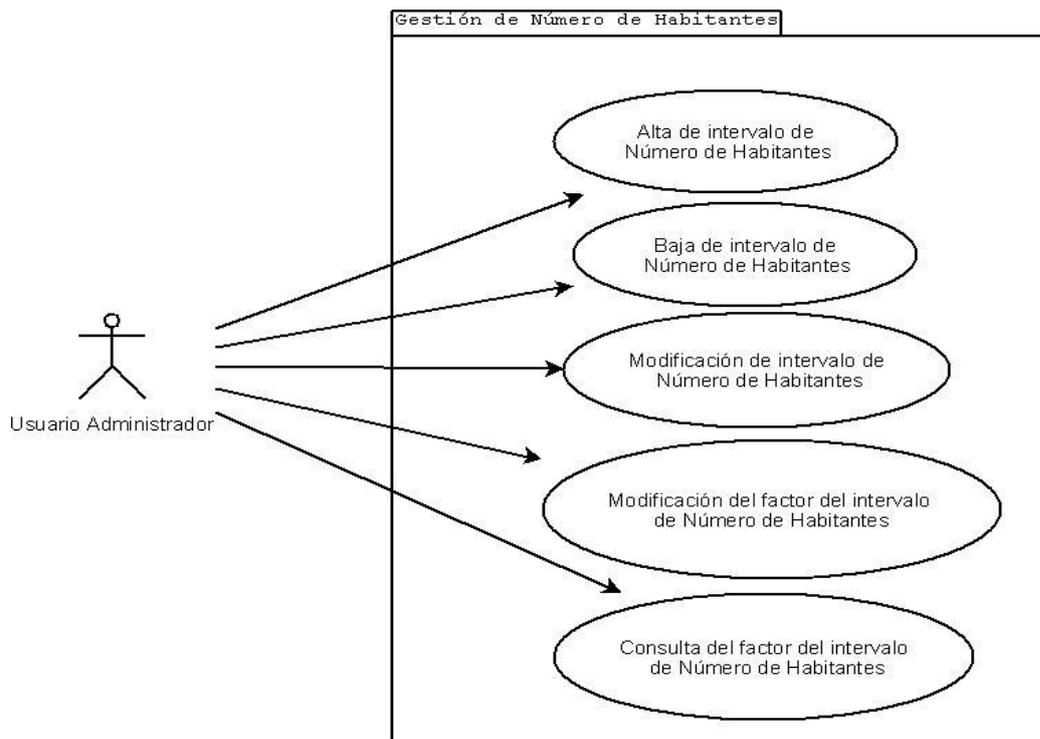


Figura 12. Diagrama de casos de uso para la gestión de número de habitantes.

Tabla 26. Caso de uso Alta de intervalo de Número de Habitantes

CASO DE USO	25. Alta de intervalo de Número de Habitantes.
ACTORES	Usuario Administrador.
PROPÓSITO	Crear un intervalo de Número de Habitantes.
RESUMEN	Se crea el intervalo del Número de Habitantes con las cantidades del límite inferior, el límite superior y el factor de variación.
PRECONDICIONES	Ninguna.
POSTCONDICIONES	<ul style="list-style-type: none"> - Los datos relativos al intervalo del Número de Habitantes quedan almacenados en el sistema. - Límite inferior <= Límite superior - Límite inferior >= 0
ENTRADAS	Los límites inferior y superior del Número de Habitantes y el factor de variación.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 27. Caso de uso Baja de intervalo de Número de Habitantes

CASO DE USO	26. Baja de intervalo de Número de Habitantes.
ACTORES	Usuario Administrador.
PROPÓSITO	Dar de baja un intervalo de Número de Habitantes.
RESUMEN	El intervalo de Número de Habitantes se elimina del sistema.
PRECONDICIONES	El intervalo de Número de Habitantes debe existir en el sistema.
POSTCONDICIONES	Los datos relativos al intervalo del Número de Habitantes se eliminan del sistema.
ENTRADAS	El intervalo del Número de Habitantes.
SALIDAS	Mensaje de confirmación de la baja.

Tabla 28. Caso de uso Modificación de intervalo de Número de Habitantes

CASO DE USO	27. Modificación de intervalo de Número de Habitantes.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar los límites del intervalo del Número de Habitantes.
RESUMEN	Se modifican los límites del intervalo del número de habitantes.
PRECONDICIONES	El intervalo del Número de Habitantes debe existir en el sistema.
POSTCONDICIONES	<ul style="list-style-type: none"> - Los nuevos límites del intervalo del Número de Habitantes quedan almacenados en el sistema. - Límite inferior <= Límite superior - Límite inferior >= 0
ENTRADAS	El intervalo del Número de Habitantes y los nuevos valores de los límites inferior y superior.
SALIDAS	Mensaje de confirmación del cambio.

Tabla 29. Caso de uso Modificación del factor de intervalo de Número de Habitantes

CASO DE USO	28. Modificación del factor de intervalo de Número de Habitantes
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar el factor de variación del intervalo de Número de Habitantes.
RESUMEN	Se modifica el factor de variación del intervalo del Número de Habitantes.
PRECONDICIONES	El intervalo de Número de Habitantes debe existir en el sistema.
POSTCONDICIONES	El factor de variación toma el valor introducido.
ENTRADAS	El intervalo del Número de Habitantes y el nuevo factor de variación.
SALIDAS	Ninguna.

Tabla 30. Caso de uso Consulta del factor del intervalo de Número de habitantes

CASO DE USO	29. Consulta del factor del intervalo de Número de Habitantes.
ACTORES	Usuario Administrador.
PROPÓSITO	Consultar el factor de variación del intervalo del Número de Habitantes
RESUMEN	Se consulta el factor de variación del intervalo del Número de Habitantes.
PRECONDICIONES	El intervalo del Número de Habitantes debe existir en el sistema.
POSTCONDICIONES	Ninguna.
ENTRADAS	El intervalo del Número de Habitantes
SALIDAS	El valor del factor de variación del intervalo del Número de Habitantes.

CASOS DE USO REFERENTES A LA GESTIÓN DE PLANTILLA DE TRABAJADORES

La cantidad de trabajadores que se tiene a cargo dentro de una administración pública añade un factor que hace variar la estimación del salario. El sistema debe almacenar intervalos de número de trabajadores y el valor de dicho factor de variación.

Para gestionar esa información, se especifican los casos de uso de la **Figura 13**.

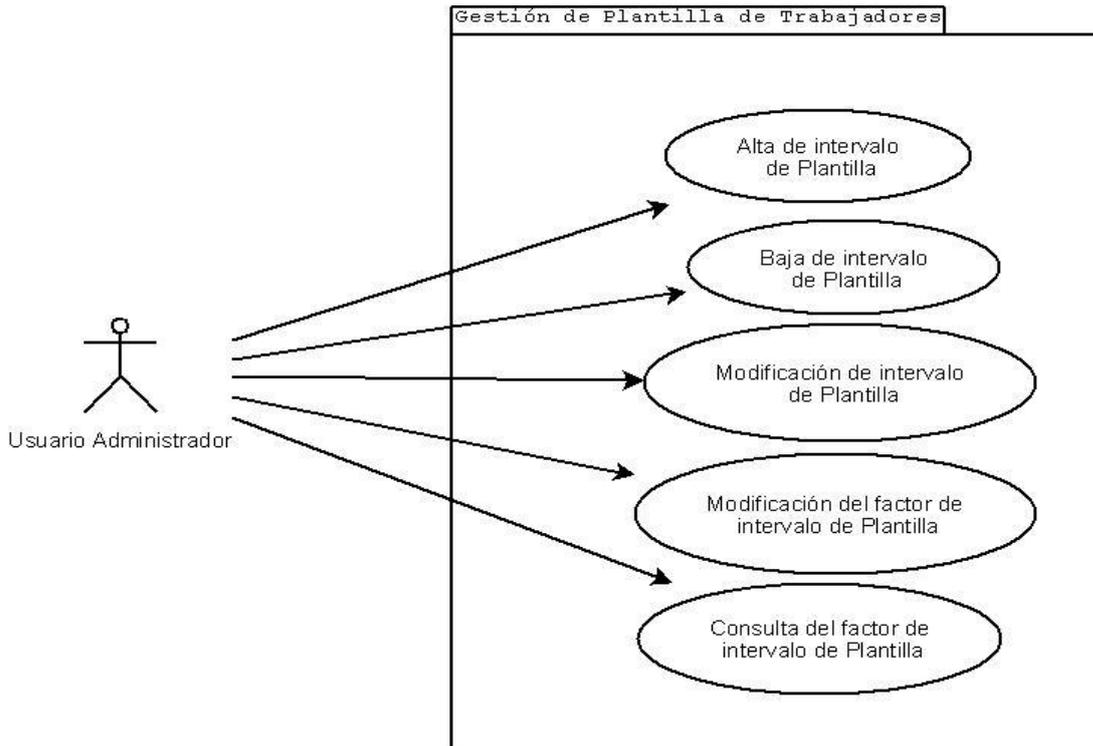


Figura 13. Diagrama de casos de uso para la gestión de plantilla de trabajadores.

Tabla 31. Caso de uso Alta de intervalo de plantilla

CASO DE USO	30. Alta de intervalo de Plantilla.
ACTORES	Usuario Administrador.
PROPÓSITO	Crear un intervalo de Plantilla de trabajadores.
RESUMEN	Se crea el intervalo de la Plantilla de trabajadores con las cantidades del límite inferior, el límite superior y el factor de variación.
PRECONDICIONES	Ninguna.
POSTCONDICIONES	<ul style="list-style-type: none"> - Los datos relativos al intervalo de la Plantilla quedan almacenados en el sistema. - Límite inferior \leq Límite superior - Límite inferior \geq 0
ENTRADAS	Los límites inferior y superior de la Plantilla y el factor de variación.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 32. Caso de uso Baja de intervalo de Plantilla

CASO DE USO	31. Baja de intervalo de Plantilla.
ACTORES	Usuario Administrador.
PROPÓSITO	Dar de baja un intervalo de Plantilla de trabajadores.
RESUMEN	El intervalo de Plantilla de trabajadores se elimina del sistema.
PRECONDICIONES	El intervalo de Plantilla debe existir en el sistema.
POSTCONDICIONES	Los datos relativos al intervalo de Plantilla se eliminan del sistema.
ENTRADAS	El intervalo de Plantilla.
SALIDAS	Mensaje de confirmación de la baja.

Tabla 33. Caso de uso Modificación de intervalo de Plantilla

CASO DE USO	32. Modificación de intervalo de Plantilla.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar los límites del intervalo de Plantilla de trabajadores.
RESUMEN	Se modifican los límites del intervalo de la Plantilla de trabajadores.
PRECONDICIONES	El intervalo de Plantilla debe existir en el sistema.
POSTCONDICIONES	<ul style="list-style-type: none"> - Los nuevos límites del intervalo de Plantilla quedan almacenados en el sistema. - Límite inferior \leq Límite superior - Límite inferior \geq 0
ENTRADAS	El intervalo de Plantilla y los nuevos valores de los límites inferior y superior.
SALIDAS	Mensaje de confirmación del cambio.

Tabla 34. Caso de uso Modificación de intervalo de Plantilla

CASO DE USO	33. Modificación del factor de intervalo de Plantilla.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar el factor de variación del intervalo de Plantilla de trabajadores.
RESUMEN	Se modifica el factor de variación del intervalo de la Plantilla de trabajadores.
PRECONDICIONES	El intervalo de Plantilla debe existir en el sistema.
POSTCONDICIONES	El factor de variación toma el valor introducido.
ENTRADAS	El intervalo de Plantilla y el nuevo factor de variación.
SALIDAS	Ninguna.

Tabla 35. Caso de uso Consulta del factor del intervalo de Plantilla

CASO DE USO	34. Consulta del factor del intervalo de Plantilla.
ACTORES	Usuario Administrador.
PROPÓSITO	Consultar el factor de variación del intervalo de Plantilla de trabajadores.
RESUMEN	Se consulta el factor de variación del intervalo de la Plantilla de trabajadores.
PRECONDICIONES	El intervalo de Plantilla debe existir en el sistema.
POSTCONDICIONES	Ninguna.
ENTRADAS	El intervalo de Plantilla
SALIDAS	El valor del factor de variación del intervalo de Plantilla.

CASOS DE USO REFERENTES A LA OBTENCIÓN DE RESULTADOS

La funcionalidad que ofrece **SIESMACRe** para un usuario anónimo es la de la obtención de un informe con el cálculo de salarios y estadísticas de salario para puestos de trabajo similares.

Los casos de uso de la **Figura 14** recogen la especificación de requisitos de la obtención del informe.

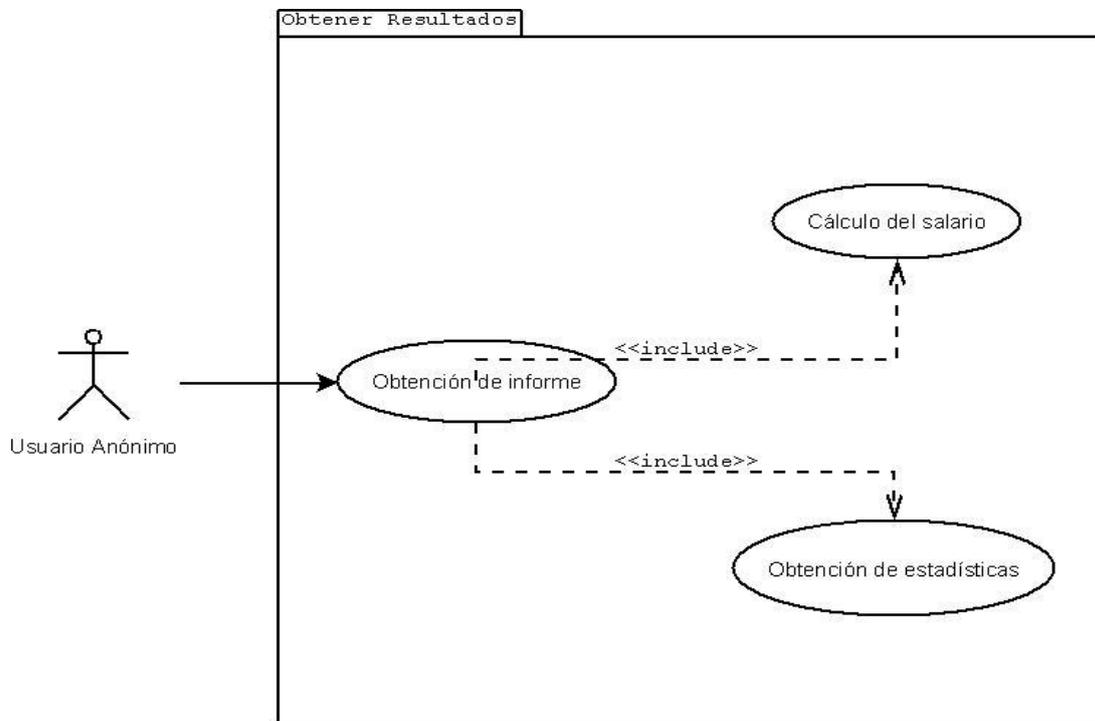


Figura 14. Diagrama de casos de uso para obtener resultados.

Tabla 36. Caso de uso Obtención de informe

CASO DE USO	35. Obtención de informe.
ACTORES	Usuario Anónimo.
PROPÓSITO	Obtener el informe del salario.
RESUMEN	Se obtiene el cálculo del salario y las estadísticas de salario de puestos similares.
PRECONDICIONES	<ul style="list-style-type: none"> - La Categoría debe existir en el sistema. - El Complemento de destino debe existir en el sistema. - El Puesto de Trabajo debe existir en el sistema. - La Comunidad Autónoma debe existir en el sistema. - El intervalo del Número de Habitantes debe existir en el sistema. - El intervalo de Plantilla de trabajadores debe existir en el sistema. - El valor del Complemento Específico de entrada debe ser mayor o igual a cero.
POSTCONDICIONES	Ninguna.
INCLUYE	<ul style="list-style-type: none"> - Caso de Uso 36: Cálculo del salario - Caso de Uso 37: Obtención de estadísticas
ENTRADAS	<p>La Categoría, el Complemento de Destino y el valor del Complemento Específico.</p> <p>El Puesto de Trabajo, la Comunidad Autónoma, el intervalo de Número de Habitantes y el intervalo de Plantilla de trabajadores.</p>
SALIDAS	<p>1.-Informe con el cálculo del salario:</p> <ul style="list-style-type: none"> - Valor anual y mensual de la Categoría - Valor anual y mensual del Complemento de Destino - Valor anual y mensual del complemento específico - Valor anual de la paga extraordinaria - Salario total anual <p>2.-Estadísticas de salario de puestos similares:</p> <ul style="list-style-type: none"> - El salario medio. - Los percentiles 10, 25, 50, 75 y 90 del salario - Posición sobre el salario, y los percentiles 10, 25, 50, 75 y 90 (porcentaje del salario real obtenido en el caso de uso 36 con respecto al valor estadístico)

Tabla 37. Caso de uso Cálculo del salario

CASO DE USO	36. Cálculo del salario.
ACTORES	Usuario Anónimo.
PROPÓSITO	Calcular el salario del usuario a partir de los parámetros de entrada.
RESUMEN	Se calcula el salario del usuario a partir de la Categoría, el Complemento de Destino y la cantidad percibida como complemento específico
PRECONDICIONES	Debe ocurrir el caso de uso 35: Obtención de informe
POSTCONDICIONES	Ninguna.
ENTRADAS	La Categoría, el Complemento de Destino y el valor del Complemento Específico
SALIDAS	<ul style="list-style-type: none"> - Valor anual y mensual de la Categoría - Valor anual y mensual del Complemento de Destino - Valor anual y mensual del complemento específico - Valor anual de la paga extraordinaria - Salario total anual

Tabla 38. Caso de uso de Obtención de estadísticas

CASO DE USO	37. Obtención de estadísticas.
ACTORES	Usuario Anónimo.
PROPÓSITO	Obtener estadísticas de salario para condiciones de trabajo similares a las de entrada.
RESUMEN	Se calcula la media del salario y los percentiles 10, 25, 50, 75 y 90 dependiendo de los factores de Puesto de Trabajo, Comunidad Autónoma, Número de Habitantes y Plantilla de trabajadores
PRECONDICIONES	Debe ocurrir el caso de uso 35: Obtención de informe
POSTCONDICIONES	Ninguna.
ENTRADAS	El Puesto de Trabajo, la Comunidad Autónoma, el intervalo de Número de Habitantes y el intervalo de Plantilla de trabajadores.
SALIDAS	<ul style="list-style-type: none"> - El salario medio. - Los percentiles 10, 25, 50, 75 y 90 del salario - Posición sobre el salario, y los percentiles 10, 25, 50, 75 y 90 (porcentaje del salario real obtenido en el caso de uso 36 con respecto al valor estadístico)

4.2.3 Casos de Uso SIEFAP

SIEFAP (Sistema de Información para la Evaluación de Funcionarios de Administraciones Públicas) se diseña con el objetivo de evaluar las aptitudes, conocimientos y competencias de

la plantilla de funcionarios de cualquier administración pública. Esta evaluación se realiza a través de cuestionarios específicos.

El sistema debe ser capaz de permitir la gestión del contenido de dichos cuestionarios y su organización en conocimientos, además de ser la plataforma por la que los empleados contesten a los cuestionarios y consulten el estado de sus evaluaciones.

La **Figura 15** presenta el diagrama general de casos de uso del sistema SIEFAP, que muestra los actores y los paquetes de casos de uso.

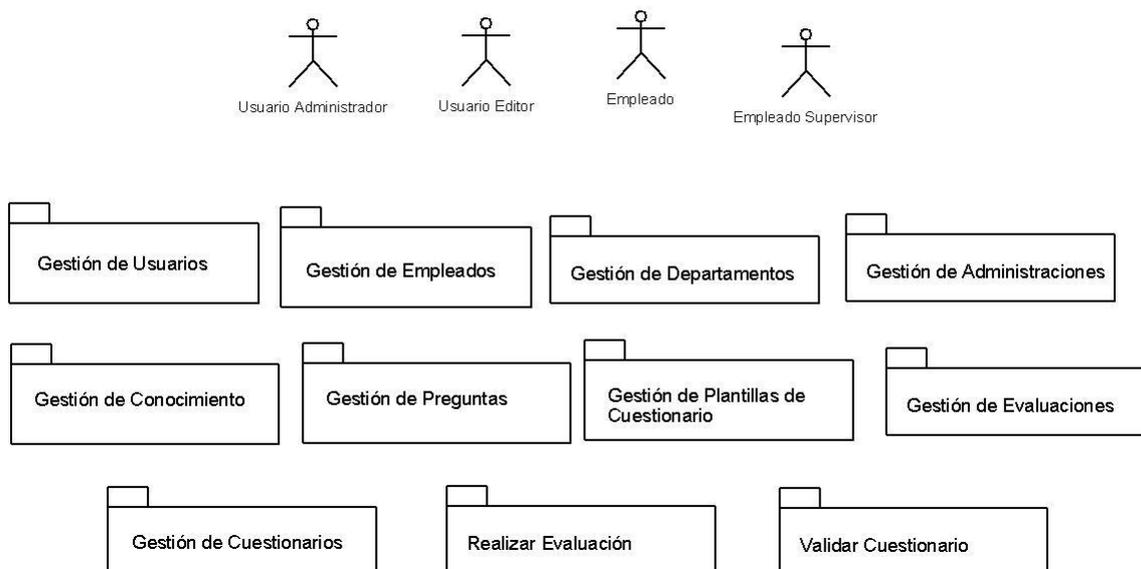


Figura 15. Diagrama general de casos de uso de SIEFAP.

CASOS DE USO REFERENTES A LA GESTIÓN DE EMPLEADOS

Los empleados de la administración deben estar registrados en el sistema. De ellos se debe mantener su NIF, nombre, apellidos, la fecha de nacimiento, el sexo, el salario y el departamento y administración para la que trabaja.

Un empleado puede encontrarse en estado “activo”, que le permite interactuar con el sistema, o en estado “inactivo”. En este estado, el empleado no puede interactuar con el sistema, pero la información relacionada no es eliminada. Cuando el empleado está activo, debe pertenecer a un departamento; por el contrario, cuando su estado es inactivo, no pertenece a ningún departamento.

Por otra parte, el empleado puede realizar funciones de supervisor de cuestionarios y, para ello, debe considerarse “empleado supervisor”. También existe el rol de Responsable de la Administración, que se corresponde con un empleado que tiene los permisos del Usuario Administrador, pero sólo para la Administración en la que trabaja.

La **Figura 16** muestra el diagrama de casos de uso para la gestión de empleados.

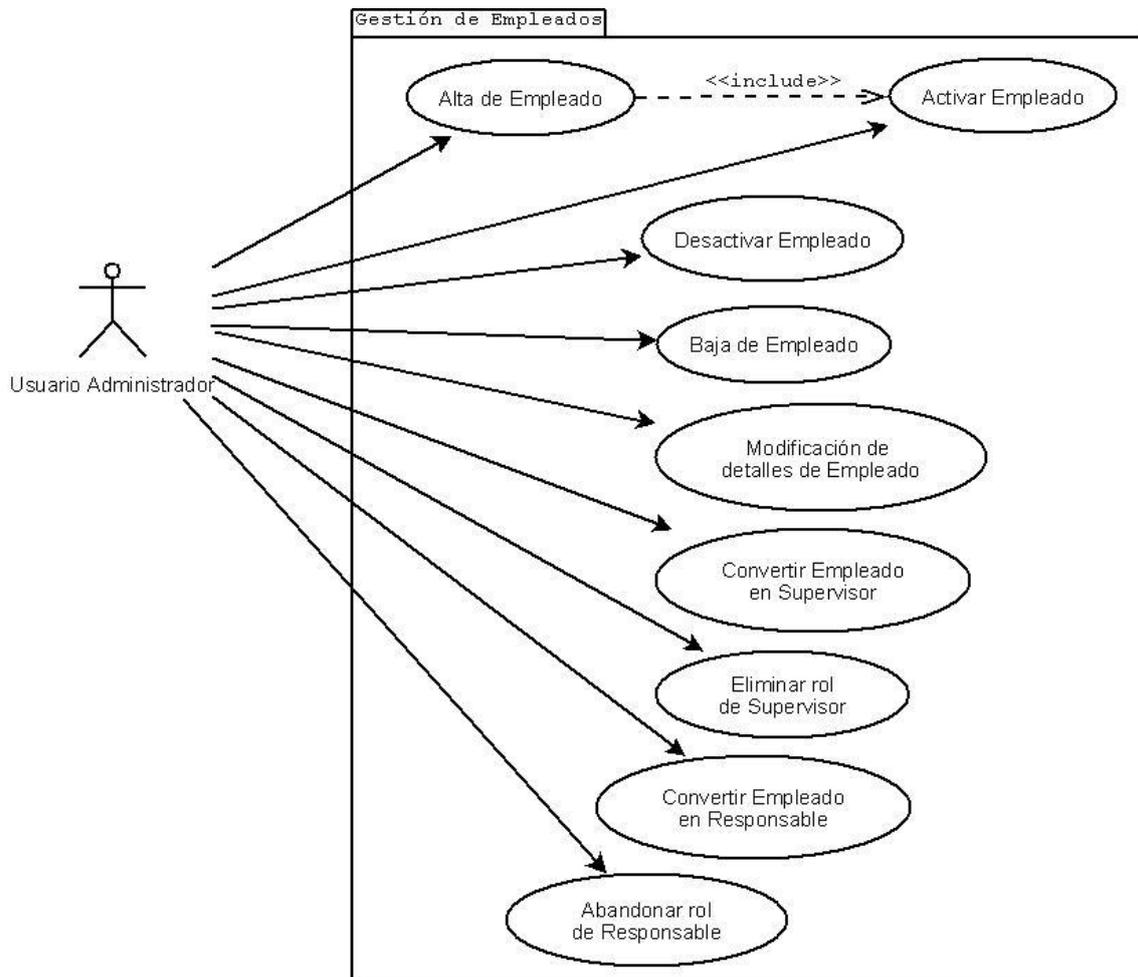


Figura 16. Diagrama de casos de uso para la gestión de empleados.

Tabla 39. Caso de uso Alta de Empleado

CASO DE USO	1. Alta de Empleado.
ACTORES	Usuario Administrador.
PROPÓSITO	Dar de alta un Empleado en el sistema.
RESUMEN	Se crea el empleado en el sistema con sus atributos de nombre, apellidos, NIF, fecha de nacimiento y sexo.
PRECONDICIONES	El Departamento debe existir en el sistema.
POSTCONDICIONES	Los datos relativos al Empleado quedan almacenados en el sistema. El usuario se crea en estado “Activo”: puede interactuar con el sistema.
INCLUYE	- Caso de Uso 2: Activar Empleado
ENTRADAS	El nombre, los apellidos, el NIF del empleado, la fecha de nacimiento, el sexo y el departamento en el que trabaja.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 40. Caso de uso Activar Empleado

CASO DE USO	2. Activar Empleado.
ACTORES	Usuario Administrador.
PROPÓSITO	Activar el empleado para que pueda interactuar con el sistema.
RESUMEN	Se pasa el empleado al estado “Activo”.
PRECONDICIONES	<ul style="list-style-type: none"> - El Empleado debe existir en el sistema y estar en estado “Inactivo”. <li style="padding-left: 20px;">o - Debe ocurrir el caso de uso “1 – Alta de Empleado”. - El Departamento debe existir en el sistema.
POSTCONDICIONES	<ul style="list-style-type: none"> - El Empleado pasa a estar en estado “Activo”, con lo que puede interactuar con el sistema. - El Empleado se relaciona con el Departamento de la entrada.
ENTRADAS	El Empleado y el Departamento.
SALIDAS	Mensaje de confirmación de la activación.

Tabla 41. Caso de uso Desactivar Empleado

CASO DE USO	3. Desactivar Empleado.
ACTORES	Usuario Administrador.
PROPÓSITO	Desactivar el empleado.
RESUMEN	Se pasa el Empleado al estado “Inactivo”.
PRECONDICIONES	<ul style="list-style-type: none"> - El Empleado debe existir en el sistema y estar en estado “Activo”.
POSTCONDICIONES	<ul style="list-style-type: none"> - El Empleado pasa a estar en estado “Inactivo”. - El Empleado deja de pertenecer a ningún Departamento. - El salario del empleado toma el valor 0.
ENTRADAS	El Empleado
SALIDAS	Mensaje de confirmación de la desactivación.

Tabla 42. Caso de uso Baja de Empleado

CASO DE USO	4. Baja de Empleado.
ACTORES	Usuario Administrador.
PROPÓSITO	Dar de baja un Empleado
RESUMEN	Dar de baja un Empleado del sistema, eliminando las relaciones que tenga con las demás entidades.
PRECONDICIONES	El Empleado debe existir en el sistema.
POSTCONDICIONES	<ul style="list-style-type: none"> - Los datos relativos al Empleado se eliminan del sistema. - Se eliminan los Cuestionarios realizados por el empleado, en caso de que los haya. - Se eliminan las Evaluaciones realizadas por el empleado, en caso de que las haya.
ENTRADAS	El Empleado.
SALIDAS	Mensaje de confirmación de la baja.

Tabla 43. Caso de uso Modificación de detalles de Empleado

CASO DE USO	5. Modificación de detalles de Empleado.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar los detalles del Empleado
RESUMEN	Modificación de alguno de los siguientes atributos, o de todos ellos: <ul style="list-style-type: none"> - Nombre - Apellidos - NIF - Fecha de nacimiento - Salario - Sexo
PRECONDICIONES	<ul style="list-style-type: none"> - El Empleado debe existir en el sistema. - El Empleado debe estar en estado “Activo”.
POSTCONDICIONES	Los atributos cambiados toman el nuevo valor.
ENTRADAS	<ul style="list-style-type: none"> - El empleado. - El nuevo valor de los atributos que se deseen cambiar.
SALIDAS	Mensaje de confirmación del cambio.

Tabla 44. Caso de uso Convertir Empleado en Supervisor

CASO DE USO	6. Convertir Empleado en Supervisor.
ACTORES	Usuario Administrador.
PROPÓSITO	Asignar a un empleado el papel de supervisor de Cuestionarios.
RESUMEN	Se asigna a un empleado el rol de supervisor de cuestionarios.
PRECONDICIONES	<ul style="list-style-type: none"> - El Empleado debe existir en el sistema y no debe ser ya supervisor. - El Empleado debe estar en estado “Activo”
POSTCONDICIONES	<ul style="list-style-type: none"> - El Empleado pasa a ser supervisor de Cuestionarios, con los permisos asociados que conlleva. - Se almacena la fecha en que se convierte en Supervisor.
ENTRADAS	El Empleado.
SALIDAS	Mensaje de confirmación de la asignación del rol.

Tabla 45. Caso de uso Abandonar rol de Supervisor

CASO DE USO	7. Abandonar rol de Supervisor.
ACTORES	Usuario Administrador.
PROPÓSITO	Desasignar a un Empleado Supervisor de este rol.
RESUMEN	El Empleado Supervisor deja de serlo.
PRECONDICIONES	El Empleado debe tener el rol de Supervisor.
POSTCONDICIONES	<ul style="list-style-type: none"> - El Empleado deja de ser Supervisor de Cuestionarios. - Se almacena la fecha en que abandona el rol de Supervisor.
ENTRADAS	El Empleado.
SALIDAS	Mensaje de confirmación de la desasignación del rol.

Tabla 46. Caso de uso Convertir Empleado en Responsable

CASO DE USO	8. Convertir Empleado en Responsable.
ACTORES	Usuario Administrador.
PROPÓSITO	Asignar a un empleado el papel de Responsable de la Administración.
RESUMEN	Se asigna a un empleado el rol de responsable de la administración.
PRECONDICIONES	<ul style="list-style-type: none"> - El Empleado debe existir en el sistema y no debe ser ya Responsable. - El Empleado debe estar en estado “Activo”
POSTCONDICIONES	- El Empleado pasa a ser Responsable de la Administración, con los permisos asociados que conlleva.
ENTRADAS	El Empleado.
SALIDAS	Mensaje de confirmación de la asignación del rol.

Tabla 47. Caso de uso Abandonar rol de Responsable

CASO DE USO	9. Abandonar rol de Responsable.
ACTORES	Usuario Administrador.
PROPÓSITO	Desasignar a un Responsable de Administración de este rol.
RESUMEN	El Responsable de Administración deja de serlo.
PRECONDICIONES	El Empleado debe tener el rol de Responsable.
POSTCONDICIONES	- El Empleado deja de ser Responsable de Administración.
ENTRADAS	El Empleado.
SALIDAS	Mensaje de confirmación de la desasignación del rol.

CASOS DE USO REFERENTES A LA GESTIÓN DE USUARIOS

Además de los empleados de la Administración, el sistema debe tener otros roles de Usuario con ciertos permisos para poder realizar acciones de mantenimiento del sistema o de mantenimiento de contenidos de las evaluaciones.

Para ello, se definen los roles de Usuario Administrador y Usuario Editor, y los casos de uso de la gestión de estos roles se presentan en la [Figura 17](#).

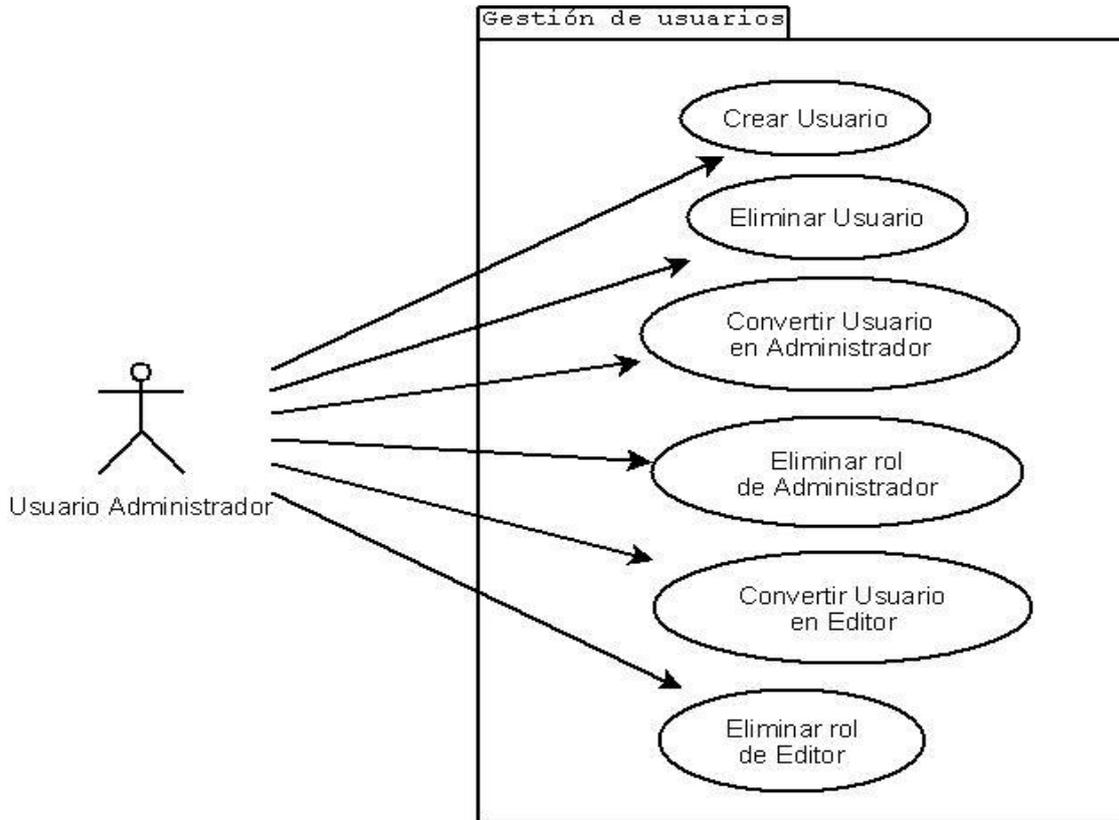


Figura 17. Diagrama de casos de uso para la gestión de usuarios.

Tabla 48. Caso de uso Crear Usuario

CASO DE USO	10. Crear Usuario.
ACTORES	Usuario Administrador.
PROPÓSITO	Crear un nuevo Usuario del sistema.
RESUMEN	Se crea un nuevo Usuario en el sistema.
PRECONDICIONES	El nombre de usuario debe ser único.
POSTCONDICIONES	El nuevo usuario queda registrado en el sistema.
ENTRADAS	El nombre y la contraseña del usuario.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 49. Caso de uso Borrar Usuario

CASO DE USO	11. Borrar Usuario.
ACTORES	Usuario Administrador.
PROPÓSITO	Eliminar un Usuario del sistema.
RESUMEN	Se elimina un Usuario del sistema.
PRECONDICIONES	El Usuario debe existir en el sistema.
POSTCONDICIONES	El Usuario se elimina del sistema
ENTRADAS	El Usuario.
SALIDAS	Mensaje de confirmación del borrado.

Tabla 50. Caso de uso Convertir usuario en Administrador

CASO DE USO	12. Convertir Usuario en Administrador.
ACTORES	Usuario Administrador.
PROPÓSITO	Asignar a un Usuario existente en el sistema el rol de Administrador.
RESUMEN	Se asigna a un Usuario el papel de Administrador.
PRECONDICIONES	El Usuario debe existir en el sistema.
POSTCONDICIONES	El usuario pasa a ser Administrador del sistema, con los permisos asociados que conlleva.
ENTRADAS	El Usuario.
SALIDAS	Mensaje de confirmación de la asignación del rol.

Tabla 51. Caso de uso Abandonar rol de Administrador

CASO DE USO	13. Abandonar rol de Administrador.
ACTORES	Usuario Administrador.
PROPÓSITO	Desasignar a un Usuario Administrador de este rol.
RESUMEN	El Usuario Administrador deja de serlo.
PRECONDICIONES	El Usuario debe existir en el sistema.
POSTCONDICIONES	El Usuario deja de ser Administrador del sistema.
ENTRADAS	El nombre del usuario.
SALIDAS	Mensaje de confirmación de la desasignación del rol.

Tabla 52. Caso de uso Convertir Usuario en Editor

CASO DE USO	14. Convertir Usuario en Editor.
ACTORES	Usuario Administrador.
PROPÓSITO	Asignar a un Usuario existente en el sistema el rol de Editor.
RESUMEN	Se asigna a un Usuario el papel de Editor.
PRECONDICIONES	El Usuario debe existir en el sistema.
POSTCONDICIONES	El usuario pasa a ser Editor, con los permisos asociados que conlleva.
ENTRADAS	El nombre del Usuario.
SALIDAS	Mensaje de confirmación de la asignación del rol.

Tabla 53. Caso de uso Abandonar rol de Editor

CASO DE USO	15. Abandonar rol de Editor.
ACTORES	Usuario Administrador
PROPÓSITO	Desasignar a un Usuario Administrador de este rol.
RESUMEN	El Usuario Editor deja de serlo.
PRECONDICIONES	El Usuario debe existir en el sistema.
POSTCONDICIONES	El Usuario deja de ser Editor.
ENTRADAS	El nombre del usuario.
SALIDAS	Mensaje de confirmación de la desasignación del rol.

CASOS DE USO REFERENTES A LA GESTIÓN DE DEPARTAMENTOS

Los empleados están asignados a Departamentos de la Administración. Cada Departamento debe pertenecer a una Administración.

De los Departamentos se debe guardar información de su nombre, del número de empleados que lo forman, el presupuesto que dispone para recursos humanos y el presupuesto gastado en recursos humanos.

La **Figura 18** presenta los casos de uso de la gestión de departamentos.

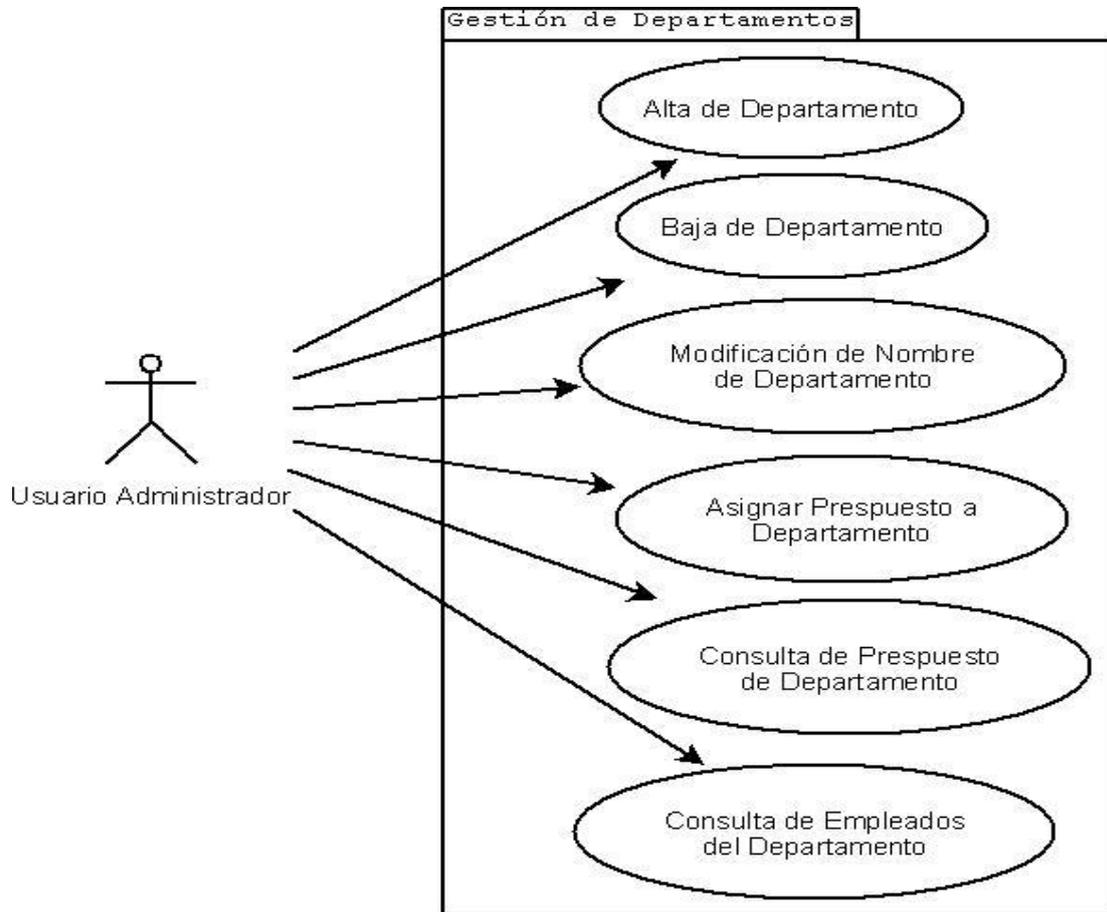


Figura 18. Diagrama de casos de uso para la gestión de departamentos.

Tabla 54. Caso de uso Alta de Departamento

CASO DE USO	16. Alta de Departamento.
ACTORES	Usuario Administrador.
PROPÓSITO	Dar de alta un Departamento en el sistema.
RESUMEN	Se crea un departamento en el sistema con su nombre y la administración a la que pertenece.
PRECONDICIONES	Debe existir la Administración a la que pertenece.
POSTCONDICIONES	Los datos relativos al Departamento quedan almacenados en el sistema.
ENTRADAS	El nombre del departamento y la Administración a la que pertenece.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 55. Caso de uso Baja de Departamento

CASO DE USO	17. Baja de Departamento.
ACTORES	Usuario Administrador.
PROPÓSITO	Dar de baja un Departamento
RESUMEN	Se elimina el Departamento del sistema.
PRECONDICIONES	<ul style="list-style-type: none"> - El Departamento debe existir en el sistema. - El Departamento no puede tener empleados en Estado “Activo”.
POSTCONDICIONES	Los datos relativos al Departamento se eliminan del sistema.
ENTRADAS	El Departamento.
SALIDAS	Mensaje de confirmación de la baja.

Tabla 56. Caso de uso Modificación del nombre del Departamento

CASO DE USO	18. Modificación del nombre del Departamento.
ACTORES	Usuario Administrador.
PROPÓSITO	Cambiar el nombre del Departamento por uno nuevo.
RESUMEN	Se modifica el nombre del Departamento.
PRECONDICIONES	El Departamento debe existir en el sistema.
POSTCONDICIONES	El nombre del Departamento pasa a tomar el nuevo valor.
ENTRADAS	El Departamento y el nuevo nombre.
SALIDAS	Mensaje de confirmación del cambio.

Tabla 57. Caso de uso Asignar presupuesto a Departamento

CASO DE USO	19. Asignar presupuesto a Departamento.
ACTORES	Usuario Administrador.
PROPÓSITO	Asignación del presupuesto del Departamento.
RESUMEN	Se le asigna presupuesto al Departamento.
PRECONDICIONES	<ul style="list-style-type: none"> - El Departamento debe existir en el Sistema. - El presupuesto debe ser mayor que cero.
POSTCONDICIONES	El presupuesto del Departamento toma el valor que se le asigna.
ENTRADAS	El Departamento y el presupuesto.
SALIDAS	Mensaje de confirmación de la asignación.

Tabla 58. Caso de uso Consulta de presupuesto del Departamento

CASO DE USO	20. Consulta de presupuesto del Departamento.
ACTORES	Usuario Administrador.
PROPÓSITO	Consultar el presupuesto que tiene asignado y el dispuesto por el Departamento.
RESUMEN	Se consulta el presupuesto asignado y el dispuesto para el área de recursos humanos del departamento.
PRECONDICIONES	El Departamento debe existir en el sistema.
POSTCONDICIONES	Ninguna
ENTRADAS	El Departamento
SALIDAS	El presupuesto asignado y el presupuesto dispuesto.

Tabla 59. Caso de uso Consulta de empleados del Departamento

CASO DE USO	21. Consulta de empleados del Departamento.
ACTORES	Usuario Administrador.
PROPÓSITO	Consultar el número de empleados de un Departamento.
RESUMEN	Se consulta el número de empleados de un Departamento.
PRECONDICIONES	El Departamento debe existir en el sistema.
POSTCONDICIONES	Ninguna
ENTRADAS	El Departamento
SALIDAS	El número de empleados del Departamento.

CASOS DE USO REFERENTES A LA GESTIÓN DE ADMINISTRACIONES

SIEFAP se diseña para ser utilizado en la Administración Pública. Una única implantación del sistema debe poder gestionar la información de distintas Administraciones, entre las cuales no hay ningún requisito que las relacione. Una administración se organiza en departamentos. Al igual que éstos, de la Administración tiene un nombre, un número de empleados y el presupuesto asignado y dispuesto para recursos humanos.

Los casos de uso para la gestión de administraciones se presentan en la Figura 19.

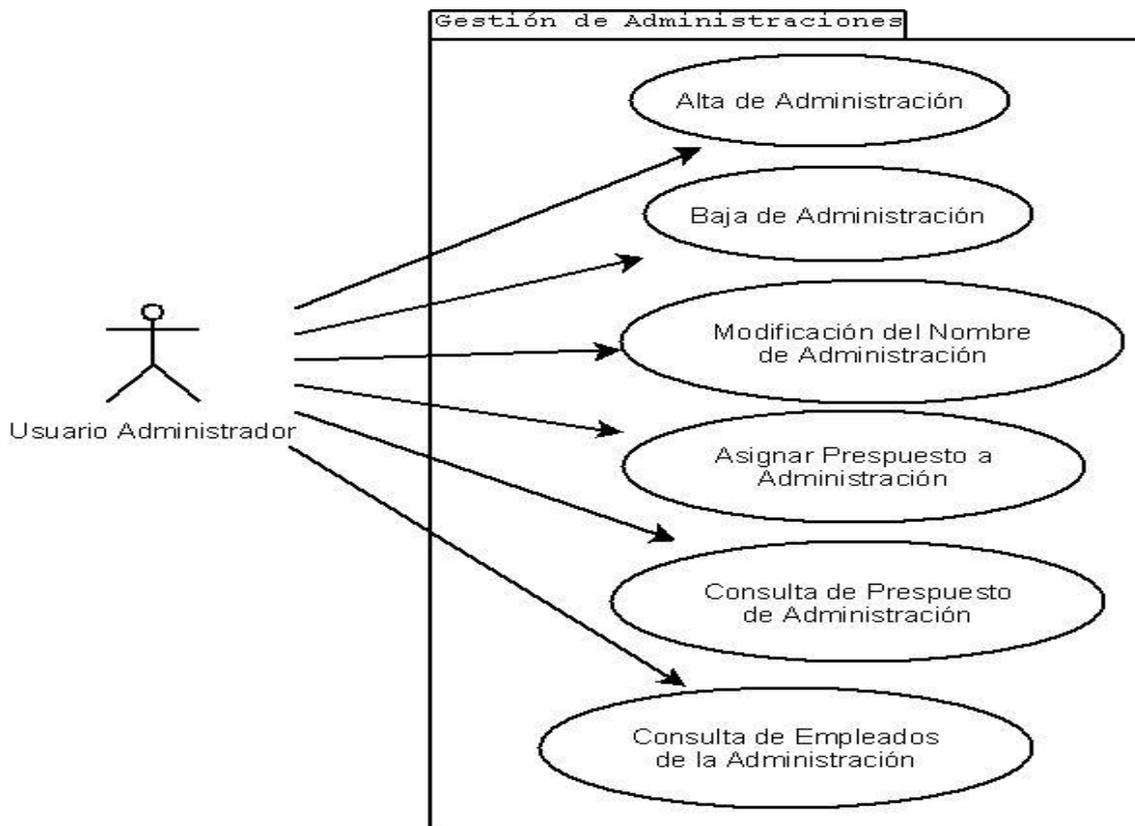


Figura 19. Diagrama de casos de uso para la gestión de Administraciones.

Tabla 60. Caso de uso Alta de Administración

CASO DE USO	22. Alta de Administración.
ACTORES	Usuario Administrador.
PROPÓSITO	Dar de alta una Administración en el sistema.
RESUMEN	Se crea una administración en el sistema con su nombre.
PRECONDICIONES	Ninguna.
POSTCONDICIONES	Los datos relativos a la Administración quedan almacenados en el sistema.
ENTRADAS	El nombre de la administración.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 61. Caso de uso Baja de Administración

CASO DE USO	23. Baja de Administración.
ACTORES	Usuario Administrador.
PROPÓSITO	Dar de baja una Administración.
RESUMEN	La Administración se elimina del sistema.
PRECONDICIONES	<ul style="list-style-type: none"> - La Administración debe existir en el sistema. - La Administración no debe tener Conocimientos asociados.
POSTCONDICIONES	<ul style="list-style-type: none"> - Los datos relativos a la Administración se eliminan del sistema. - Los Departamentos de la Administración se eliminan del sistema. - Los Empleados de la Administración pasan a estado "Inactivo".
ENTRADAS	La Administración.
SALIDAS	Mensaje de confirmación de la baja.

Tabla 62. Caso de uso Modificación del nombre de Administración

CASO DE USO	24. Modificación del nombre de Administración.
ACTORES	Usuario Administrador.
PROPÓSITO	Cambiar el nombre de la Administración por uno nuevo.
RESUMEN	Se modifica el nombre la Administración.
PRECONDICIONES	La Administración debe existir en el sistema.
POSTCONDICIONES	El nombre de la Administración pasa a tomar el nuevo valor.
ENTRADAS	La Administración y el nuevo nombre.
SALIDAS	Mensaje de confirmación del cambio.

Tabla 63. Caso de uso Asignar presupuesto a Administración

CASO DE USO	25. Asignar presupuesto a Administración.
ACTORES	Usuario Administrador.
PROPÓSITO	Asignación del presupuesto de la Administración.
RESUMEN	Se le asigna presupuesto a la Administración.
PRECONDICIONES	<ul style="list-style-type: none"> - La Administración debe existir en el Sistema. - El presupuesto debe ser mayor que cero.
POSTCONDICIONES	El presupuesto de la Administración toma el valor que se le asigna.
ENTRADAS	La Administración y el presupuesto.
SALIDAS	Mensaje de confirmación de la asignación.

Tabla 64. Caso de uso Consulta de presupuesto de la Administración

CASO DE USO	26. Consulta de presupuesto de la Administración.
ACTORES	Usuario Administrador.
PROPÓSITO	Consultar el presupuesto asignado y el dispuesto por una Administración.
RESUMEN	Se consulta el presupuesto asignado y el dispuesto para el área de recursos humanos de la Administración.
PRECONDICIONES	La Administración debe existir en el sistema.
POSTCONDICIONES	Ninguna
ENTRADAS	La Administración
SALIDAS	El presupuesto asignado y el dispuesto por la Administración.

Tabla 65. Caso de uso Consulta de empleados de la Administración

CASO DE USO	27. Consulta de empleados de la Administración.
ACTORES	Usuario Administrador.
PROPÓSITO	Consultar el número de empleados de una Administración.
RESUMEN	Se consulta el número de empleados de una Administración.
PRECONDICIONES	La Administración debe existir en el sistema.
POSTCONDICIONES	Ninguna
ENTRADAS	La Administración
SALIDAS	El número de empleados de la Administración.

CASOS DE USO REFERENTES A LA GESTIÓN DE CONOCIMIENTOS

El objetivo de **SIEFAP** es la evaluación de conocimientos o aptitudes de los funcionarios que conforman determinada Administración Pública. Los conocimientos a evaluar deben existir en el sistema. Estos conocimientos pueden dividirse en subconocimientos, que a su vez, también pueden volver a dividirse en subconocimientos de nivel más bajo.

Los Conocimientos y Subconocimientos tienen las mismas características. Sólo se establecen diferencias en el momento de la creación y, por tanto, se utiliza únicamente el término Conocimiento para referirse a ambos a no ser que se haga una separación explícita.

Los Conocimientos son propios de cada Administración. Esto significa que cada Administración tiene su propio inventario de Conocimientos y éstos deben estar necesariamente relacionados con la Administración a la que pertenecen.

La **Figura 20** presenta los casos de uso de la gestión de conocimientos.

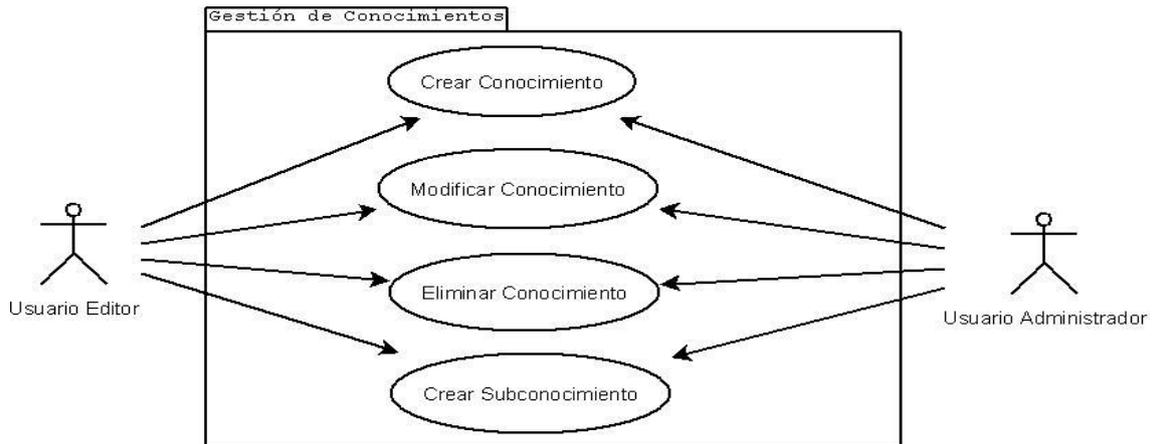


Figura 20. Diagrama de casos de uso para la gestión de conocimientos.

Tabla 66. Caso de uso Crear Conocimiento

CASO DE USO	28. Crear Conocimiento.
ACTORES	Usuarios Editor y Administrador.
PROPÓSITO	Crear un Conocimiento con su nombre y descripción.
RESUMEN	Se crea un conocimiento con su nombre y descripción.
PRECONDICIONES	La Administración debe existir en el sistema.
POSTCONDICIONES	Los datos relativos al Conocimiento quedan almacenados en el sistema.
ENTRADAS	El nombre del Conocimiento, su descripción y la Administración a la que pertenece.
SALIDAS	Mensaje de confirmación de almacenamiento del conocimiento.

Tabla 67. Caso de uso Modificar Conocimiento

CASO DE USO	29. Modificar Conocimiento.
ACTORES	Usuarios Editor y Administrador.
PROPÓSITO	Modificar el nombre y/o la descripción de un Conocimiento existente.
RESUMEN	Se modifica el nombre y/o la descripción de un Conocimiento. Este caso de uso sirve también para Subconocimientos, pues tienen las mismas características que los conocimientos.
PRECONDICIONES	El Conocimiento debe existir en el sistema.
POSTCONDICIONES	El nombre y la descripción del Conocimiento toman el nuevo valor.
ENTRADAS	El Conocimiento, el nuevo nombre y la descripción.
SALIDAS	Mensaje de confirmación del cambio.

Tabla 68. Caso de uso Eliminar Conocimiento

CASO DE USO	30. Eliminar Conocimiento.
ACTORES	Usuarios Editor y Administrador.
PROPÓSITO	Eliminar un Conocimiento existente.
RESUMEN	El Conocimiento se elimina del sistema. Este caso de uso sirve también para Subconocimientos, pues tienen las mismas características que los conocimientos.
PRECONDICIONES	<ul style="list-style-type: none"> - El Conocimiento debe existir en el sistema. - El Conocimiento no puede tener preguntas asociadas. - El Conocimiento no puede tener plantillas de cuestionario asociadas. - El Conocimiento no puede tener evaluaciones de conocimiento asociadas. - El Conocimiento no puede tener Subconocimientos asociados.
POSTCONDICIONES	La información relativa al Conocimiento se elimina del sistema.
ENTRADAS	El Conocimiento.
SALIDAS	Mensaje de confirmación del borrado.

Tabla 69. Caso de uso Crear Subconocimiento

CASO DE USO	31. Crear Subconocimiento.
ACTORES	Usuarios Editor y Administrador.
PROPÓSITO	Crear un Subconocimiento dependiente de un Conocimiento existente.
RESUMEN	Se crea un Subconocimiento asociado a un Conocimiento existente, con su nombre y descripción.
PRECONDICIONES	<ul style="list-style-type: none"> - El Conocimiento debe existir en el sistema. - La Administración debe existir en el sistema. - La administración debe ser la misma que la del Conocimiento ya existente.
POSTCONDICIONES	Los datos relativos al Subconocimiento quedan almacenados en el sistema.
ENTRADAS	El Conocimiento y el nombre del Subconocimiento.
SALIDAS	Mensaje de almacenamiento del Subconocimiento.

CASOS DE USO REFERENTES AL CONTENIDO DE LOS CUESTIONARIOS

Las evaluaciones se realizan sobre cuestionarios que previamente deben estar dados de alta en el sistema.

El contenido de estos cuestionarios consta de “Plantillas de Cuestionario” y de “Preguntas”, que se emplean para formar las “Plantillas de Cuestionario”.

Las “Preguntas” están formadas por su enunciado y por “Respuestas”, que se presentan como opciones de elección y cada una de ellas representa un valor de puntuación.

En el momento en que se crea la pregunta, se debe crear al menos una opción de respuesta y se debe asociar a un conocimiento.

La modificación de una pregunta puede suponer la modificación de su propio enunciado, o bien la creación, modificación o eliminación de opciones de respuesta asociadas.

La **Figura 21** presenta los casos de uso de la gestión de preguntas.

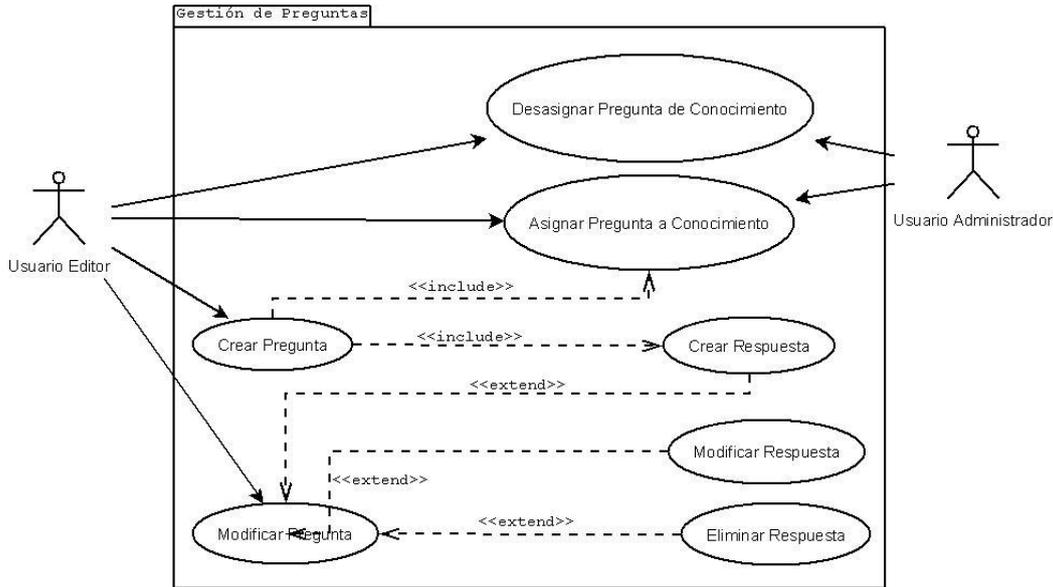


Figura 21. Diagrama de casos de uso para la gestión de preguntas.

Tabla 70. Caso de uso Crear Pregunta

CASO DE USO	32. Crear Pregunta.
ACTORES	Usuario Editor.
PROPÓSITO	Crear una Pregunta con su enunciado y respuestas asociadas.
RESUMEN	Se crea una Pregunta con su enunciado y se crean y asocian con tantas Opciones de Respuesta como se desee.
PRECONDICIONES	Ninguna.
POSTCONDICIONES	<ul style="list-style-type: none"> - Los datos relativos a la Pregunta y sus Opciones de Respuesta quedan almacenados en el sistema. - La Pregunta queda asociada a un Conocimiento.
INCLUYE	Caso de Uso 34: Crear Respuesta. Caso de Uso 37: Asignar Pregunta a Conocimiento
ENTRADAS	El texto de la Pregunta.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 71. Caso de uso Modificar Pregunta

CASO DE USO	33. Modificar Pregunta.
ACTORES	Usuario Editor.
PROPÓSITO	Modificar una pregunta existente.
RESUMEN	La modificación de la pregunta tiene cuatro opciones. 1. Modificar el enunciado de la Pregunta. 2. Modificar el enunciado y/o puntuación de una Respuesta asociada. 3. Añadir una Opción de Respuesta. 4. Eliminar una Opción de Respuesta
PRECONDICIONES	- La Pregunta debe existir en el sistema. - La Pregunta no puede estar siendo utilizada en Cuestionarios. - Para el caso 1 y 3: La pregunta debe existir en el sistema. - Para el caso 2 y 4: La pregunta debe existir en el sistema y tener al menos una opción de Respuesta asociada.
POSTCONDICIONES	- Para el caso 1: El enunciado de la pregunta toma el nuevo valor. - Para el caso 2: El enunciado y la puntuación de la opción de Respuesta toman los nuevos valores. - Para el caso 3: Se añade una nueva opción de respuesta. - Para el caso 4: La opción de Respuesta se elimina.
EXTENSIONES	- Para el caso 2 -> Caso de Uso 33: Modificar Respuesta. - Para el caso 3 -> Caso de Uso 32: Crear Respuesta. - Para el caso 4 -> Caso de Uso 34: Eliminar Respuesta.
ENTRADAS	- Para el caso 1: La pregunta y su nuevo enunciado. - Para el caso 2: La pregunta, la opción de Respuesta, el texto y la puntuación de la Respuesta. - Para el caso 3: La pregunta y la nueva opción de Respuesta. - Para el caso 4: La pregunta y la opción de Respuesta.
SALIDAS	Ninguna.

Tabla 72. Caso de uso Crear respuesta

CASO DE USO	34. Crear Respuesta.
ACTORES	Usuario Editor.
PROPÓSITO	Crear una Opción de respuesta asociada a una pregunta.
RESUMEN	Para una pregunta, se crea una opción de respuesta con su texto y puntuación.
PRECONDICIONES	- La pregunta a la que se le va a asignar no puede estar siendo utilizada en Cuestionarios. - Debe ocurrir el caso de uso “32-Crear Pregunta” o el “33-Modificar Pregunta”
POSTCONDICIONES	La opción de respuesta queda almacenada con su texto y puntuación y se le asigna a la pregunta.
ENTRADAS	La Pregunta, el texto y la puntuación de la Respuesta.
SALIDAS	Ninguna.

Tabla 73. Caso de uso Modificar Respuesta

CASO DE USO	35. Modificar Respuesta.
ACTORES	Usuario Editor.
PROPÓSITO	Modificar una Opción de respuesta asociada a una pregunta.
RESUMEN	Se modifica una opción de respuesta de una pregunta.
PRECONDICIONES	<ul style="list-style-type: none"> - Debe ocurrir el caso de uso “33-Modificar Pregunta”. - La Respuesta debe existir y estar asignada a la Pregunta.
POSTCONDICIONES	La opción de respuesta queda almacenada con su nuevo texto su nueva puntuación.
ENTRADAS	La Pregunta, la Respuesta, el nuevo texto y la nueva puntuación de la Respuesta.
SALIDAS	Ninguna.

Tabla 74. Caso de uso Eliminar Respuesta

CASO DE USO	36. Eliminar Respuesta.
ACTORES	Usuario Editor.
PROPÓSITO	Eliminar una Opción de respuesta asociada a una pregunta.
RESUMEN	Se elimina una opción de respuesta ya asociada a la pregunta.
PRECONDICIONES	<ul style="list-style-type: none"> - Debe ocurrir el caso de uso “33-Modificar Pregunta”. - La Respuesta debe existir y estar asignada a la Pregunta. - La Pregunta debe tener más de una Respuesta asociada; no se puede eliminar la Respuesta si ésta es la única.
POSTCONDICIONES	La opción de respuesta se elimina del sistema.
ENTRADAS	La Pregunta y la Respuesta.
SALIDAS	Ninguna.

Tabla 75. Caso de uso Asignar Pregunta a Conocimiento

CASO DE USO	37. Asignar Pregunta a Conocimiento.
ACTORES	Usuarios Editor o Administrador
PROPÓSITO	Asignar una pregunta a un Conocimiento
RESUMEN	Se asigna una pregunta a un Conocimiento existente.
PRECONDICIONES	<ul style="list-style-type: none"> - El Conocimiento debe existir en el sistema. - La Pregunta debe existir en el sistema o debe ocurrir el caso de uso “32 – Crear Pregunta”.
POSTCONDICIONES	La pregunta queda asociada al Conocimiento.
ENTRADAS	La Pregunta y el Conocimiento.
SALIDAS	Mensaje de confirmación de la asignación.

Tabla 76. Caso de uso Desasignar Pregunta de Conocimiento

CASO DE USO	38. Desasignar Pregunta de Conocimiento
ACTORES	Usuarios Editor o Administrador
PROPÓSITO	Desasignar una pregunta de un Conocimiento
RESUMEN	La pregunta deja de estar relacionada con el conocimiento.
PRECONDICIONES	<ul style="list-style-type: none"> - El Conocimiento y la Pregunta deben estar relacionados. - La Pregunta debe estar relacionada con más de un Conocimiento. - La pregunta no puede estar siendo utilizada en Plantillas de Cuestionario del mismo Conocimiento del que se desea desasignar.
POSTCONDICIONES	Se elimina la relación entre la Pregunta y el Conocimiento
ENTRADAS	La Pregunta y el Conocimiento.
SALIDAS	Ninguna

Por su parte, las “Plantillas de Cuestionario” deben tener un nombre, una descripción y un estado, que indica si la plantilla ya puede ser utilizada en cuestionarios. Cuando la plantilla esté lista para usarse, debe tener también el número de preguntas que la forman y la puntuación máxima que se puede obtener en ella.

La creación de Plantillas de Cuestionario implica asociarlas a un Conocimiento existente y que se le añadan Preguntas ya creadas en el sistema. La modificación de una Plantilla de Cuestionario debe permitir que se le añadan o eliminen Preguntas.

La **Figura 22** presenta los casos de uso de la gestión de plantillas de cuestionario.

Las Preguntas y las Plantillas de Cuestionario se relacionan con los Conocimientos. Mientras que una Pregunta puede pertenecer a uno o más Conocimientos, las Plantillas de Cuestionario deben pertenecer siempre sólo a uno, aunque éste pueda variar.

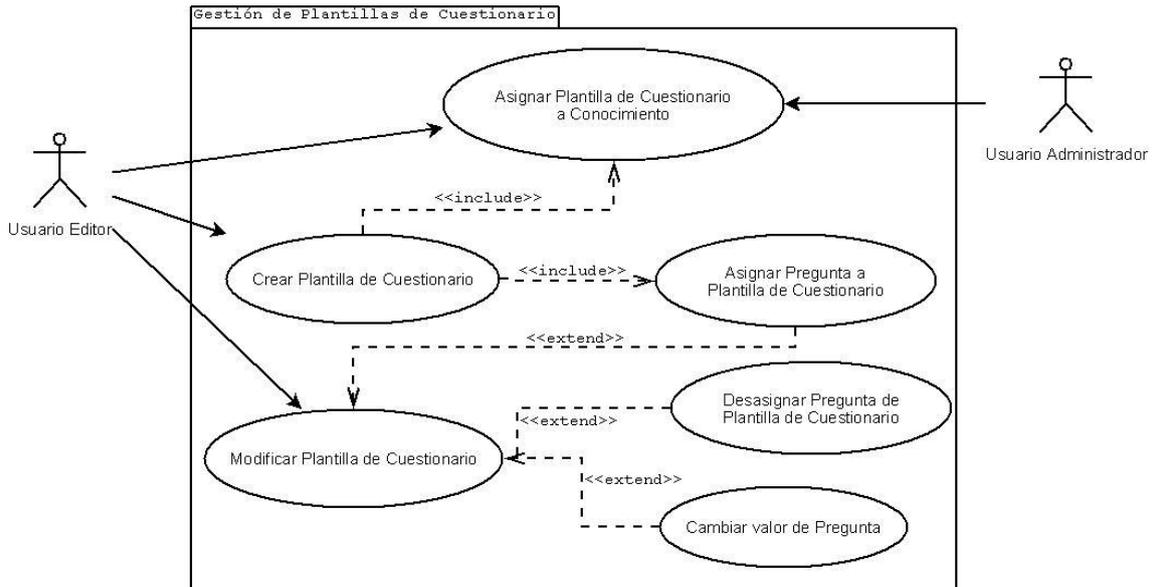


Figura 22. Diagrama de casos de uso para la gestión de preguntas.

Tabla 77. Caso de uso Crear Plantilla de Cuestionario

CASO DE USO	39. Crear Plantilla de Cuestionario.
ACTORES	Usuario Editor.
PROPÓSITO	Crear una Plantilla de Cuestionario.
RESUMEN	Se crea una Plantilla de Cuestionario con su nombre, su descripción y las preguntas que la componen.
PRECONDICIONES	Ninguna.
POSTCONDICIONES	<ul style="list-style-type: none"> - Los datos relativos a la Plantilla de Cuestionario quedan almacenados en el sistema. - La Plantilla de Cuestionario queda asociada a un Conocimiento.
INCLUYE	<ul style="list-style-type: none"> - Caso de Uso 41: Asignar Pregunta a Plantilla de Cuestionario. - Caso de Uso 43: Asignar Plantilla de Cuestionario a Conocimiento.
ENTRADAS	El nombre y la descripción de la Plantilla de Cuestionario.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 78. Caso de uso Modificar Plantilla de Cuestionario

CASO DE USO	40. Modificar Plantilla de Cuestionario.
ACTORES	Usuario Editor.
PROPÓSITO	Modificar una Plantilla de Cuestionario existente.
RESUMEN	Hay cuatro tipos de modificación de la Plantilla de Cuestionario: 1. Modificación del nombre y/o descripción. 2. Inserción de Preguntas. 3. Eliminación de Preguntas. 4. Cambio de valor de la Pregunta.
PRECONDICIONES	<ul style="list-style-type: none"> - La Plantilla de Cuestionario debe existir en el sistema. - La Plantilla no puede estar siendo utilizada en Cuestionarios. - Para el caso 1: La Plantilla de Cuestionario debe existir en el sistema. - Para el caso 2: La Plantilla de Cuestionario y la Pregunta a insertar deben existir en el sistema y estar asociadas al mismo Conocimiento. - Para los casos 3 y 4: La Pregunta debe pertenecer a la Plantilla de Cuestionario.
POSTCONDICIONES	<ul style="list-style-type: none"> - Para el caso 1: El nombre de la Plantilla de Cuestionario toma el nuevo valor. - Para el caso 2: La Pregunta queda asociada a la Plantilla de Cuestionario. - Para el caso 3: La Pregunta deja de estar asociada con la Plantilla de Cuestionario. - Para el caso 4: La Pregunta toma el nuevo valor en la Plantilla de Cuestionario
EXTENSIONES	<ul style="list-style-type: none"> - Para el caso 2 -> Caso de Uso 39: Asignar Pregunta a Plantilla de Cuestionario. - Para el caso 3 -> Caso de Uso 40: Desasignar Pregunta de Plantilla de Cuestionario. - Para el caso 4 -> Caso de Uso 43: Cambiar valor de Pregunta
ENTRADAS	<ul style="list-style-type: none"> - Para el caso 1: La Plantilla de Cuestionario, el nuevo nombre y la nueva descripción. - Para los casos 2 y 3: La Plantilla de Cuestionario y la Pregunta. - Para el caso 4: La Plantilla de Cuestionario, la Pregunta y el nuevo valor
SALIDAS	Ninguna

Tabla 79. Caso de uso Asignar Pregunta a Plantilla de Cuestionario

CASO DE USO	41. Asignar Pregunta a Plantilla de Cuestionario.
ACTORES	Usuario Editor
PROPÓSITO	Añadir una Pregunta a una Plantilla de Cuestionario y asignarle valor.
RESUMEN	Se añade la pregunta seleccionada a la Plantilla de Cuestionario.
PRECONDICIONES	<ul style="list-style-type: none"> - La Pregunta debe existir en el sistema. - El valor de la Pregunta en la Plantilla de Cuestionario no puede ser negativo. - La Pregunta debe estar asociada al Conocimiento de la Plantilla de Cuestionario. - Debe ocurrir el caso de uso “39-Crear Plantilla de Cuestionario” o el “40-Modificar Plantilla de Cuestionario”
POSTCONDICIONES	La Pregunta queda asociada a la Plantilla de Cuestionario
ENTRADAS	La Pregunta, la Plantilla de Cuestionario y el valor.
SALIDAS	Ninguna.

Tabla 80. Caso de uso Desasignar Pregunta de Plantilla de Cuestionario

CASO DE USO	42. Desasignar Pregunta de Plantilla de Cuestionario.
ACTORES	Usuario Editor
PROPÓSITO	Eliminar una Pregunta de una Plantilla de Cuestionario.
RESUMEN	La Pregunta seleccionada se elimina de la Plantilla de Cuestionario.
PRECONDICIONES	<ul style="list-style-type: none"> - La Pregunta debe existir en el sistema y estar relacionada con la Plantilla de Cuestionario. - Debe ocurrir el Caso de Uso “40-Modificar Plantilla de Cuestionario”.
POSTCONDICIONES	Se elimina la Pregunta de la Plantilla de Cuestionario.
ENTRADAS	La Pregunta y la Plantilla de Cuestionario.
SALIDAS	Ninguna.

Tabla 81. Caso de uso Cambiar valor de Pregunta

CASO DE USO	43. Cambiar valor de Pregunta
ACTORES	Usuario Editor
PROPÓSITO	Cambiar el valor de la Pregunta dentro de la Plantilla de Cuestionario
RESUMEN	Se cambia el valor de la Pregunta para la Plantilla de Cuestionario.
PRECONDICIONES	<ul style="list-style-type: none"> - La Pregunta debe existir en el sistema y estar relacionada con la Plantilla de Cuestionario. - Debe ocurrir el Caso de Uso “40-Modificar Plantilla de Cuestionario”.
POSTCONDICIONES	La Pregunta toma el nuevo valor para la Plantilla de Cuestionario seleccionada.
ENTRADAS	La Pregunta, la Plantilla de Cuestionario y el nuevo valor.
SALIDAS	Ninguna.

Tabla 82. Caso de uso Asignar Plantilla de cuestionario a Conocimiento

CASO DE USO	44. Asignar Plantilla de cuestionario a Conocimiento
ACTORES	Usuarios Editor o Administrador
PROPÓSITO	Asignar una Plantilla de cuestionario a un nuevo Conocimiento
RESUMEN	Se asigna una Plantilla de Cuestionario a un Conocimiento existente. Si ya estaba asociada a un Conocimiento, se elimina la relación anterior.
PRECONDICIONES	<ul style="list-style-type: none"> - El Conocimiento debe existir en el sistema. - La Plantilla de Cuestionario no debe tener Preguntas o - Las Preguntas deben estar asociadas al nuevo Conocimiento. - La Plantilla de Cuestionario debe existir en el sistema o debe ocurrir el caso de uso “39 – Crear Plantilla de Cuestionario”.
POSTCONDICIONES	La Plantilla de cuestionario queda asociada al Conocimiento.
ENTRADAS	La Plantilla de cuestionario y el nuevo Conocimiento.
SALIDAS	Ninguna.

CASOS DE USO REFERENTES A LA GESTIÓN DE EVALUACIONES

Las Evaluaciones que se realizan a los empleados se dividen en Evaluaciones de Conocimientos.

Cada Evaluación se asigna a un Empleado y se debe guardar la fecha de creación, una descripción y el número de puntos totales de la evaluación. La evaluación también tiene tres estados: “Edición”, “Activa”, cuando el empleado puede realizarla; y “Terminada”, cuando está cerrada y sólo guarda datos a nivel informativo. Cuando la evaluación está “Terminada”, se debe guardar su fecha de finalización y el número de puntos obtenidos por el empleado.

Las Evaluaciones de Conocimiento que componen la Evaluación global, por su parte, tienen un número de puntos y la suma de ellos debe ser igual a la de la Evaluación global.

Los Cuestionarios se asignan sólo a las Evaluaciones de Conocimiento y, una vez realizados, otorgan al Empleado un nivel determinado en el Conocimiento examinado. Este nivel se convierte en un número de puntos obtenidos. La suma de los puntos obtenidos en todas las Evaluaciones de Conocimiento se suman para formar la puntuación obtenida en la Evaluación global.

La **Figura 23** muestra los casos de uso para la gestión de evaluaciones.

Tabla 83. Caso de uso Crear Evaluación

CASO DE USO	45. Crear Evaluación.
ACTORES	Usuario Administrador.
PROPÓSITO	Crear una Evaluación asignada a un empleado.
RESUMEN	Se crea una Evaluación con un número de puntos totales y se asigna para que la realice un Empleado.
PRECONDICIONES	El Empleado debe existir en el sistema.
POSTCONDICIONES	- Los datos relativos a la Evaluación quedan almacenados en el sistema. - La evaluación queda en estado de “Edición”.
ENTRADAS	El Empleado, la fecha de creación, la descripción y los puntos totales.
SALIDAS	Mensaje de confirmación de la creación.

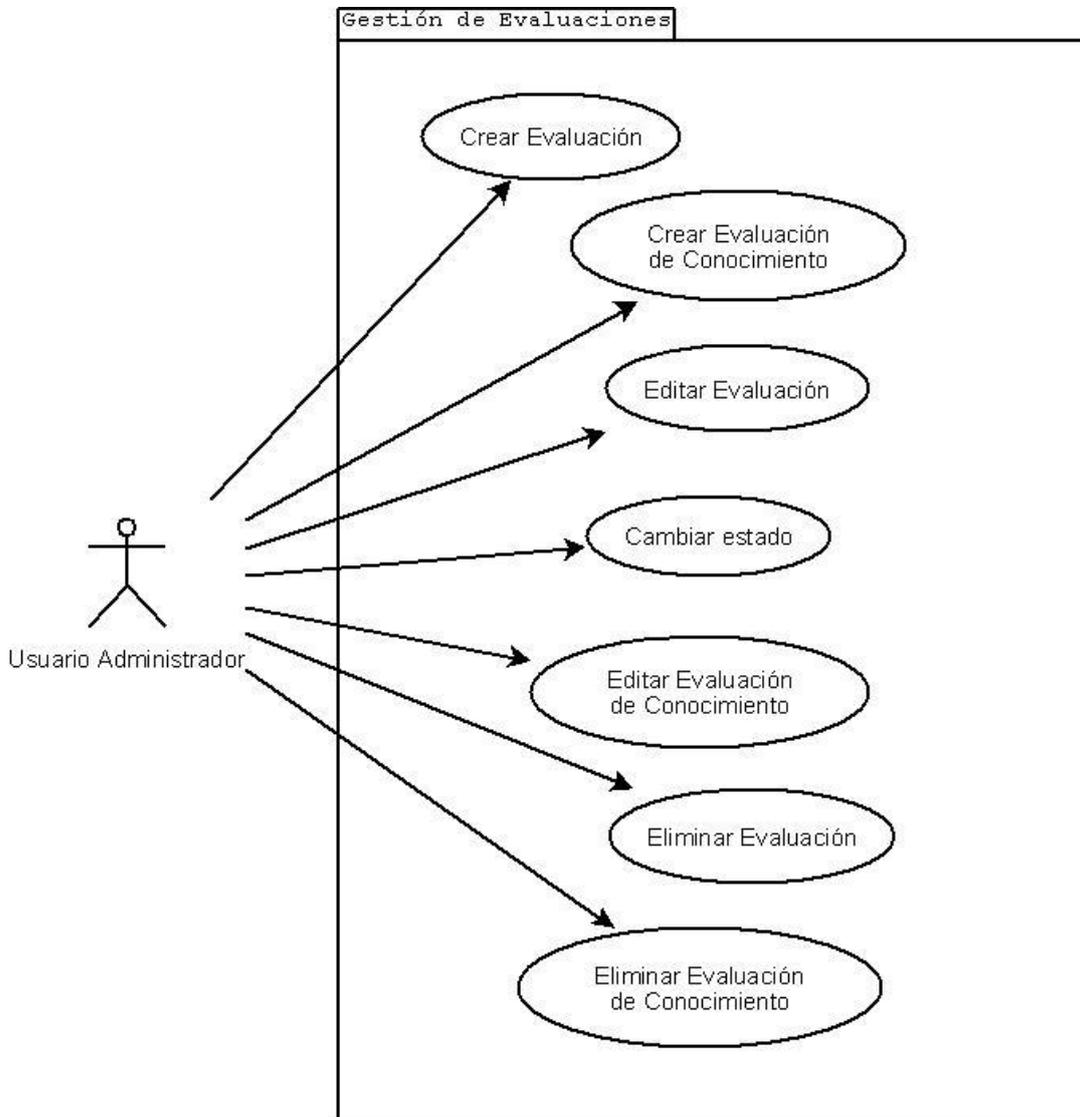


Figura 23. Diagrama de casos de uso para la gestión de evaluaciones.

Tabla 84. Caso de uso Crear Evaluación de Conocimiento

CASO DE USO	46. Crear Evaluación de Conocimiento.
ACTORES	Usuario Administrador.
PROPÓSITO	Crear una Evaluación de Conocimiento.
RESUMEN	Se crea una Evaluación de Conocimiento y se le asigna a una Evaluación global.
PRECONDICIONES	<ul style="list-style-type: none"> - El Conocimiento debe existir en el sistema. - El Empleado debe existir en el sistema. - La Evaluación global a la que se asigna debe existir en el sistema. - La Evaluación global debe estar en estado de "Edición". - Si se la Evaluación creada es de un Subconocimiento, el Superconocimiento relacionado debe existir en la Evaluación Global. - El Empleado y el Conocimiento deben pertenecer a la misma Administración - El Empleado debe ser el mismo que el de la Evaluación global a la que se asocia. - El porcentaje de puntos no puede ser inferior a cero ni superior a 100.
POSTCONDICIONES	<ul style="list-style-type: none"> - Los datos relativos a la Evaluación de Conocimiento quedan almacenados en el sistema. - La Evaluación de Conocimiento se asigna a una Evaluación global existente.
ENTRADAS	El Conocimiento, la Evaluación global, el Empleado y el porcentaje de puntos sobre la Evaluación global o de su Superconocimiento.
SALIDAS	Mensaje de confirmación de almacenamiento.

Tabla 85. Caso de uso Editar Evaluación

CASO DE USO	47. Editar Evaluación.
ACTORES	Usuario Administrador.
PROPÓSITO	Editar una Evaluación.
RESUMEN	Cambiar los puntos totales o la descripción de una Evaluación global.
PRECONDICIONES	<ul style="list-style-type: none"> - La Evaluación debe existir en el sistema. - Sólo se puede modificar en estado de Edición.
POSTCONDICIONES	- Los puntos totales y/o la descripción toman el nuevo valor.
ENTRADAS	- La Evaluación, los puntos totales y la descripción.
SALIDAS	Ninguna.

Tabla 86. Caso de uso Cambiar estado

CASO DE USO	48. Cambiar estado.
ACTORES	Usuario Administrador.
PROPÓSITO	Cambiar el estado de una Evaluación.
RESUMEN	Se cambia el estado de una Evaluación.
PRECONDICIONES	<ul style="list-style-type: none"> - La Evaluación debe existir en el sistema. - Los cambios de estado aceptados son “Edición” -> “Activa” -> “Terminada”. - Si se cambia a estado Activo, la suma del porcentaje de puntos de las Evaluaciones de Conocimiento que forman una Evaluación global debe ser 100.
POSTCONDICIONES	<ul style="list-style-type: none"> - La Evaluación cambia de estado. - Si se cambia a “Terminada”, se debe almacenar la fecha de finalización y se debe calcular el número de puntos obtenidos.
ENTRADAS	La Evaluación.
SALIDAS	Ninguna.

Tabla 87. Caso de uso Editar Evacuación de Conocimiento

CASO DE USO	49. Editar Evaluación de Conocimiento.
ACTORES	Usuario Administrador.
PROPÓSITO	Editar el porcentaje de puntos correspondiente a la Evaluación de Conocimiento.
RESUMEN	Se edita el porcentaje de puntos correspondientes a la Evaluación de Conocimiento.
PRECONDICIONES	<ul style="list-style-type: none"> - La Evaluación de Conocimiento debe existir en el sistema. - La Evaluación global asociada debe estar en estado de “Edición”.
POSTCONDICIONES	El porcentaje de puntos toma el nuevo valor.
ENTRADAS	El porcentaje de puntos y la Evaluación de Conocimiento.
SALIDAS	Ninguna.

Tabla 88. Caso de uso Eliminar Evaluación

CASO DE USO	50. Eliminar Evaluación.
ACTORES	Usuario Administrador.
PROPÓSITO	Eliminar una Evaluación del sistema.
RESUMEN	Se elimina una evaluación del sistema.
PRECONDICIONES	<ul style="list-style-type: none"> - La Evaluación debe estar en estado de “Edición”. o - Debe producirse como consecuencia del caso de uso “4-Baja de Empleado”.
POSTCONDICIONES	<ul style="list-style-type: none"> - La Evaluación deja de existir en el sistema y se eliminan las relaciones que tenga con el Empleado y/o Evaluaciones de Conocimiento.
ENTRADAS	La Evaluación.
SALIDAS	Mensaje de confirmación del borrado.

Tabla 89. Caso de uso Eliminar Evaluación de Conocimiento

CASO DE USO	51. Eliminar Evaluación de Conocimiento.
ACTORES	Usuario Administrador.
PROPÓSITO	Eliminar una Evaluación de Conocimiento del sistema.
RESUMEN	La Evaluación del Conocimiento se elimina del sistema.
PRECONDICIONES	<ul style="list-style-type: none"> - La Evaluación de Conocimiento no puede estar relacionada a alguna otra Evaluación en estados “Activa” o “Terminada”. <li style="padding-left: 20px;">o - Debe producirse como consecuencia del caso de uso “4-Baja de Empleado”. - La Evaluación de Conocimiento que se desea borrar no puede tener Evaluaciones de Subconocimientos asociadas.
POSTCONDICIONES	- La Evaluación de Conocimiento deja de existir en el sistema y se eliminan las relaciones que tenga con el Empleado y/o Evaluaciones de Conocimiento.
ENTRADAS	La Evaluación de Conocimiento.
SALIDAS	Mensaje de confirmación del borrado.

CASOS DE USO REFERENTES A LA GESTIÓN DE CUESTIONARIOS

El Cuestionario es la entidad a través de la cual se evalúa a los Empleados. El Cuestionario se crea partiendo de una Plantilla de Cuestionario y se le asigna a un Empleado y a una Evaluación de Conocimiento. Además, al Cuestionario se le asigna un Empleado Supervisor, cuya misión consiste en validar corrección de las respuestas del Empleado examinado.

El Cuestionario también tiene diversos estados, que son “Edición”, “Preparado”, “En respuesta”, “Terminado” y “Validado”. En este apartado se presentan los casos de uso que forman parte de la Edición de Cuestionarios.

En el momento de “Edición”, del cuestionario se debe guardar la fecha de creación y el peso que tiene en la Evaluación del Conocimiento.

Se debe almacenar también la fecha en que el Empleado inicia el cuestionario y la fecha en que lo termina, así como la fecha en que el Empleado Supervisor lo valida. En caso de que el Empleado Supervisor solicite la corrección del cuestionario, se debe guardar la fecha en que lo hace.

La **Figura 24** presenta los casos de uso de la gestión de cuestionarios.

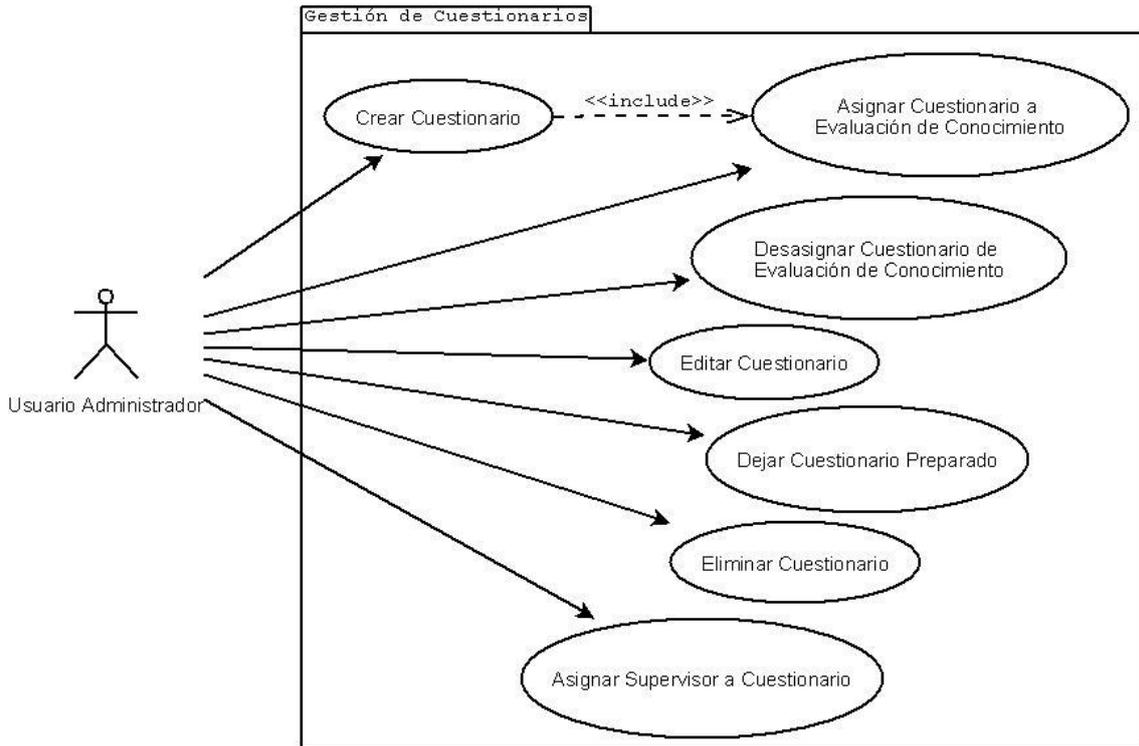


Figura 24. Diagrama de casos de uso para la gestión de evaluaciones.

Tabla 90. Caso de uso Crear Cuestionario

CASO DE USO	52. Crear Cuestionario.
ACTORES	Usuario Administrador.
PROPÓSITO	Crear un Cuestionario y asignarlo a una Evaluación de Conocimiento.
RESUMEN	Se crea un Cuestionario y se asigna a una Evaluación de Conocimiento.
PRECONDICIONES	<ul style="list-style-type: none"> - La Evaluación de Conocimiento, la Plantilla de Cuestionario y el Empleado deben existir en el sistema. - El Empleado Supervisor debe existir y estar Activo.
POSTCONDICIONES	<ul style="list-style-type: none"> - La información relativa al Cuestionario queda almacenada en el sistema. - El Cuestionario se crea en estado de “Edición”.
INCLUYE	Caso de uso 53: Asignar Cuestionario a Evaluación de Conocimiento.
ENTRADAS	<ul style="list-style-type: none"> - La Plantilla del Cuestionario, - El peso del Cuestionario, - El Empleado, - La Evaluación de Conocimiento, - La fecha de creación, - El Empleado Supervisor
SALIDAS	Ninguna.

Tabla 91. Caso de uso Asignar Cuestionario a Evaluación de Conocimiento

CASO DE USO	53. Asignar Cuestionario a Evaluación de Conocimiento.
ACTORES	Usuario Administrador.
PROPÓSITO	Asignar un Cuestionario a una Evaluación de Conocimiento.
RESUMEN	El Cuestionario se asigna a la Evaluación de Conocimiento.
PRECONDICIONES	<ul style="list-style-type: none"> - La Evaluación de Conocimiento debe existir en el sistema. - El Conocimiento no puede tener Subconocimientos. - El Cuestionario debe existir en el sistema o debe ocurrir el caso de uso “52-Crear Cuestionario”. - El peso debe ser mayor que cero. - El Conocimiento del Cuestionario y de la Evaluación de Conocimiento debe ser el mismo. - El Empleado asociado al Cuestionario y a la Evaluación de Conocimiento debe ser el mismo. - La Evaluación a la que se asigna debe estar en estado de “Edición”.
POSTCONDICIONES	El Cuestionario queda relacionado con la Evaluación de Conocimiento.
ENTRADAS	El Cuestionario, la Evaluación de Conocimiento y el peso del cuestionario.
SALIDAS	Ninguna.

Tabla 92. Caso de uso Desasignar Cuestionario de Evaluación de Conocimiento

CASO DE USO	54. Desasignar Cuestionario de Evaluación de Conocimiento.
ACTORES	Usuario Administrador.
PROPÓSITO	Desasignar un Cuestionario de una Evaluación de Conocimiento.
RESUMEN	El Cuestionario se desasigna de la Evaluación de Conocimiento.
PRECONDICIONES	<ul style="list-style-type: none"> - La Evaluación de Conocimiento y el Cuestionario deben existir en el sistema y estar relacionados. - La Evaluación de la que forma parte debe estar en estado de “Edición”.
POSTCONDICIONES	Se elimina la relación entre el Cuestionario y la Evaluación de Conocimiento.
ENTRADAS	El Cuestionario y la Evaluación de Conocimiento.
SALIDAS	Ninguna.

Tabla 93. Caso de uso Editar Cuestionario

CASO DE USO	55. Editar Cuestionario.
ACTORES	Usuario Administrador.
PROPÓSITO	Modificar el peso del Cuestionario.
RESUMEN	Permite modificar el peso del Cuestionario dentro de la Evaluación del Conocimiento.
PRECONDICIONES	El Cuestionario y la Evaluación de Conocimiento deben existir en el sistema y estar relacionados.
POSTCONDICIONES	El peso toma el nuevo valor.
ENTRADAS	El Cuestionario, la Evaluación de Conocimiento y el nuevo peso.
SALIDAS	Mensaje de confirmación del cambio.

Tabla 94. Caso de uso Dejar Cuestionario Preparado

CASO DE USO	56. Dejar Cuestionario Preparado.
ACTORES	Usuario Administrador.
PROPÓSITO	Cambiar el estado de Cuestionario a “Preparado”.
RESUMEN	Permite que el estado del cuestionario pase de “Edición” a “Preparado”.
PRECONDICIONES	<ul style="list-style-type: none"> - El Cuestionario debe existir en el sistema y estar en estado de “Edición”. - El cuestionario debe tener empleado supervisor.
POSTCONDICIONES	El estado pasa a ser “Preparado”.
ENTRADAS	El Cuestionario.
SALIDAS	Mensaje de confirmación del cambio.

Tabla 95. Caso de uso Asignar Supervisor a Cuestionario

CASO DE USO	57. Asignar Supervisor a Cuestionario.
ACTORES	Usuario Administrador.
PROPÓSITO	Asignar un Supervisor a un Cuestionario.
RESUMEN	Asigna un Supervisor a un Cuestionario para que valide las respuestas del Empleado evaluado
PRECONDICIONES	<ul style="list-style-type: none"> - El Cuestionario debe existir en el sistema. - El Supervisor debe existir en el sistema estar activo. - El Cuestionario no puede estar en estado “Validado”.
POSTCONDICIONES	El Supervisor y el Cuestionario quedan relacionados.
ENTRADAS	El Empleado Supervisor y el Cuestionario.
SALIDAS	Mensaje de confirmación de la asignación.

Tabla 96. Caso de uso Eliminar Cuestionario

CASO DE USO	58. Eliminar Cuestionario.
ACTORES	Usuario Administrador.
PROPÓSITO	Eliminar el Cuestionario del sistema.
RESUMEN	El Cuestionario se elimina del sistema.
PRECONDICIONES	<ul style="list-style-type: none"> - Las Evaluaciones en las que participa el Cuestionario deben estar en estado de Edición. <li style="padding-left: 20px;">o - Debe producirse como consecuencia del caso de uso “4-Baja de Empleado”.
POSTCONDICIONES	El Cuestionario se elimina del sistema.
ENTRADAS	El Cuestionario.
SALIDAS	Mensaje del borrado.

CASOS DE USO REFERENTES A LA REALIZACIÓN DE LA EVALUACIÓN

Cuando las evaluaciones y cuestionarios se encuentran en estado “Preparado”, el Empleado al que están asignados puede empezar a responder los Cuestionarios y así obtener su puntuación en los mismos, en las Evaluaciones de Conocimiento y en la Evaluación global.

El Empleado debe elegir la alternativa que considera correcta o más adecuada para cada una de las Preguntas de los Cuestionarios. Cada alternativa tiene una puntuación asociada, que se asigna al resultado final del empleado.

La puntuación de todas las preguntas del cuestionario ponderadas por el peso de cada pregunta en el mismo, forman la puntuación obtenida. A su vez, la puntuación obtenida en los cuestionarios de una evaluación de conocimiento, también ponderada, se corresponden con un nivel en el conocimiento. Los puntos correspondientes a cada nivel de conocimiento se utilizan para calcular la puntuación obtenida en la evaluación global del empleado.

La **Figura 25** presenta los casos de uso de realización de la evaluación.

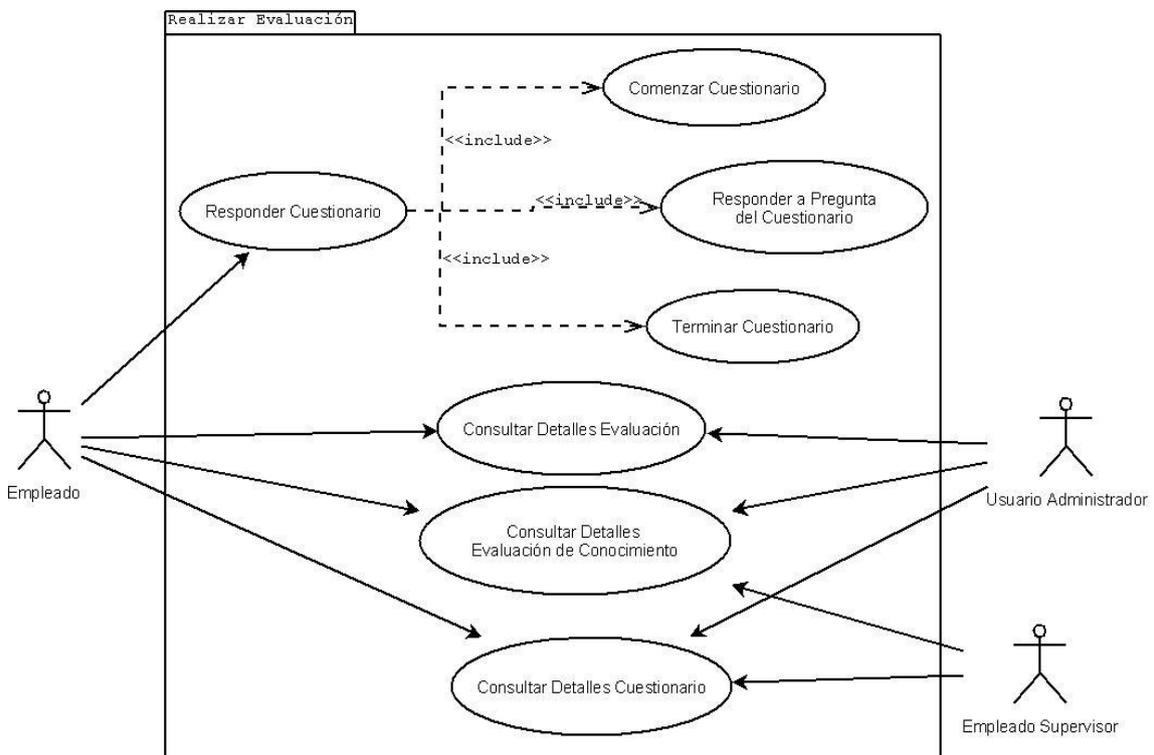


Figura 25. Diagrama de casos de uso para realizar evaluación.

Tabla 97. Caso de uso Responder Cuestionario

CASO DE USO	59. Responder Cuestionario.
ACTORES	Empleado.
PROPÓSITO	El Empleado responde a las Preguntas del Cuestionario que tiene asignado.
RESUMEN	Este caso de uso consta de estas 3 acciones sucesivas: <ol style="list-style-type: none"> 1. El Empleado comienza el Cuestionario. 2. El Empleado responde a una Pregunta, que se repite tantas veces como Preguntas haya. 3. El Empleado finaliza el Cuestionario.
PRECONDICIONES	El Cuestionario debe estar relacionado con el Empleado que lo va a responder y con un Plantilla de Cuestionario.
POSTCONDICIONES	El Cuestionario queda listo para que un Empleado Supervisor lo corrija o ratifique.
INCLUYE	Caso de uso 60: Comenzar Cuestionario. Caso de uso 61: Responder a pregunta del cuestionario. Caso de uso 62: Terminar Cuestionario.
ENTRADAS	El Cuestionario.
SALIDAS	Un mensaje que confirma que el cuestionario está finalizado.

Tabla 98. Caso de uso Comenzar Cuestionario

CASO DE USO	60. Comenzar Cuestionario.
ACTORES	Empleado.
PROPÓSITO	Marcar el cuestionario como comenzado.
RESUMEN	El Empleado comienza el Cuestionario.
PRECONDICIONES	<ul style="list-style-type: none"> - El Cuestionario debe existir en el sistema y estar en estado "Preparado". - El Empleado debe existir en el sistema y estar relacionado al Cuestionario. - Debe ocurrir el caso de uso "59-Responder cuestionario".
POSTCONDICIONES	<ul style="list-style-type: none"> - El Cuestionario pasa al estado "En respuesta". - Se almacena la fecha de comienzo del cuestionario.
ENTRADAS	El Cuestionario.
SALIDAS	Ninguna.

Tabla 99. Caso de uso Responder Pregunta de Cuestionario

CASO DE USO	61. Responder Pregunta de Cuestionario.
ACTORES	Empleado.
PROPÓSITO	El Empleado responde a una Pregunta del Cuestionario.
RESUMEN	El Empleado puede: <ol style="list-style-type: none"> 1. Seleccionar la opción de respuesta que crea más conveniente de la pregunta que aparece en el cuestionario. 2. Cambiar una opción de respuesta elegida previamente. 3. Borrar una opción de respuesta elegida previamente.
PRECONDICIONES	<ul style="list-style-type: none"> - Debe ocurrir el caso de uso “59-Responder Cuestionario”. - El Cuestionario debe estar en estado “En respuesta”. - Para los casos 2 y 3, además, debe haber una opción de respuesta seleccionada.
POSTCONDICIONES	En los casos 1 y 2, se selecciona la opción de respuesta para la pregunta y el empleado. En el caso 3, la opción de respuesta deja de estar seleccionada.
ENTRADAS	El Cuestionario, la Respuesta y la Pregunta.
SALIDAS	Ninguna.

Tabla 100. Caso de uso Terminar Cuestionario

CASO DE USO	62. Terminar Cuestionario.
ACTORES	Empleado.
PROPÓSITO	Marcar el Cuestionario como terminado.
RESUMEN	El Empleado finaliza el Cuestionario.
PRECONDICIONES	<ul style="list-style-type: none"> - Debe ocurrir el caso de uso “59-Responder Cuestionario”. - El Cuestionario debe estar en estado “En respuesta”. - Todas las Preguntas del Cuestionario deben estar respondidas.
POSTCONDICIONES	<ul style="list-style-type: none"> - El Cuestionario pasa a estado “Terminado”. - Se almacena la fecha de finalización del cuestionario.
ENTRADAS	El Cuestionario.
SALIDAS	Un mensaje que confirma que el cuestionario está finalizado.

Tabla 101. Caso de uso Consultar Detalles Cuestionario

CASO DE USO	63. Consultar Detalles Cuestionario.
ACTORES	Empleado, Empleado Supervisor y Usuario Administrador.
PROPÓSITO	Consultar los detalles del Cuestionario.
RESUMEN	El Empleado que responde un Cuestionario, el Empleado Supervisor y el Usuario Administrador pueden consultar la información relativa al mismo.
PRECONDICIONES	<ul style="list-style-type: none"> - El Empleado debe ser el encargado de contestar el Cuestionario que desea consultar. <li style="padding-left: 20px;">o - El Empleado Supervisor debe ser el encargado de validar el Cuestionario.
POSTCONDICIONES	Ninguna.
ENTRADAS	El Cuestionario que se desea consultar.
SALIDAS	<ul style="list-style-type: none"> - Las fechas de comienzo, corrección, validación y fin del Cuestionario. - La Plantilla del Cuestionario. - Las Respuestas dadas a las Preguntas que forman la Plantilla del Cuestionario y los comentarios del Supervisor a esas Respuestas. - El peso del cuestionario dentro de la Evaluación del Conocimiento. - El estado del cuestionario (En Edición, Preparado, En respuesta, Finalizado y Validado). - El porcentaje de Puntuación obtenida y los puntos correspondientes obtenidos.

Tabla 102. Caso de uso Consultar Detalles de Evaluación

CASO DE USO	64. Consultar Detalles de Evaluación.
ACTORES	Empleado y Usuario Administrador
PROPÓSITO	Consultar los detalles de la Evaluación
RESUMEN	El Empleado asignado a la Evaluación y el Usuario Administrador pueden consultar los detalles de la misma.
PRECONDICIONES	El Empleado debe estar asignado a la Evaluación que desea consultar.
POSTCONDICIONES	Ninguna.
ENTRADAS	La Evaluación que se desea consultar.
SALIDAS	<ul style="list-style-type: none"> - Las fechas de creación y finalización de la Evaluación y su estado (edición, en curso, finalizada). - La descripción de la Evaluación. - Los puntos totales y los puntos obtenidos por el Empleado. - Los Conocimientos en los que se divide y la información relativa a la puntuación de cada uno de ellos.

Tabla 103. Caso de uso Consultar Detalles de Evaluación Del Conocimiento

CASO DE USO	65. Consultar Detalles de Evaluación del Conocimiento.
ACTORES	Empleado, Empleado Supervisor y Usuario Administrador
PROPÓSITO	Consultar los detalles de la Evaluación del Conocimiento
RESUMEN	El Empleado asignado a la Evaluación de Conocimiento, el Empleado Supervisor de los Cuestionarios de ese Conocimiento y el Usuario Administrador pueden consultar los detalles de la misma.
PRECONDICIONES	<ul style="list-style-type: none"> - El Empleado debe estar asignado a la Evaluación de Conocimiento que desea consultar. - El Empleado Supervisor debe estar relacionado con alguno de los Cuestionarios de la Evaluación de Conocimiento.
POSTCONDICIONES	Ninguna.
ENTRADAS	La Evaluación de Conocimiento que se desea consultar.
SALIDAS	<ul style="list-style-type: none"> - Los puntos totales de la evaluación del conocimiento y los obtenidos por el empleado. - El número de cuestionarios asignados a la Evaluación del Conocimiento y los que están terminados, validados o en curso. - El porcentaje de puntuación de la Evaluación global - El nivel obtenido por el empleado en la Evaluación del Conocimiento cuando está completa. - Los Conocimientos en los que se divide, si los hay. - Los Cuestionarios que forman la Evaluación de Conocimiento.

CASOS DE USO REFERENTES A LA VALIDACIÓN DE CUESTIONARIOS

En el sistema existe la Figura 26 del Empleado Supervisor, que es el que se asigna a un Cuestionario y tiene la misión de validar o rechazar los cuestionarios que ha realizado otro empleado.

La Figura 26 presenta los casos de uso de la validación de cuestionarios.

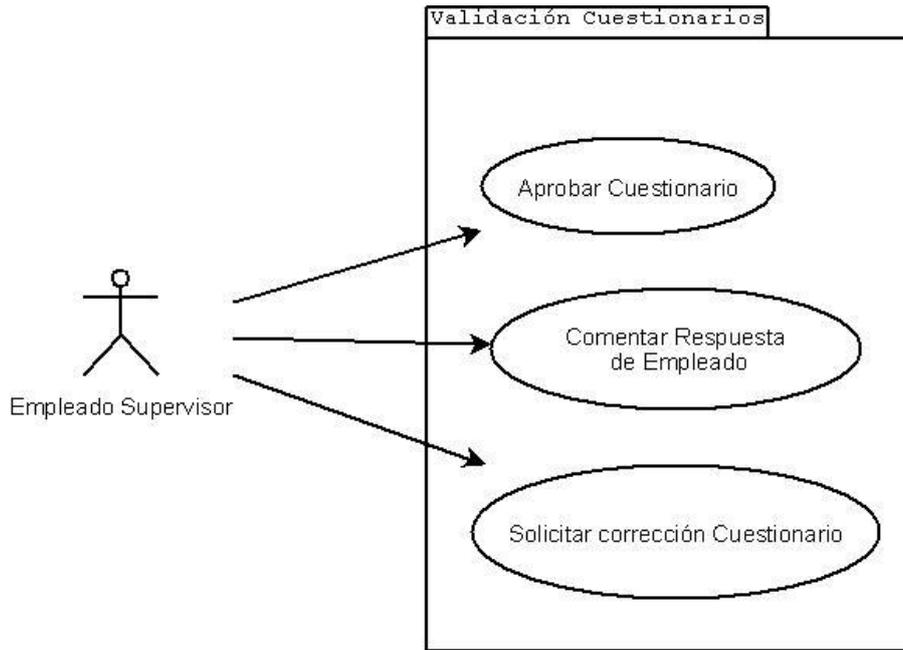


Figura 26. Diagrama de casos de uso para validación de cuestionarios.

Tabla 104. Caso de uso Validar Cuestionario

CASO DE USO	66. Validar Cuestionario.
ACTORES	Empleado Supervisor
PROPÓSITO	Validar un Cuestionario terminado
RESUMEN	El Empleado supervisor valida un cuestionario terminado.
PRECONDICIONES	<ul style="list-style-type: none"> - El Empleado Supervisor debe estar relacionado con el cuestionario. - El Cuestionario debe estar en estado “Terminado”.
POSTCONDICIONES	<ul style="list-style-type: none"> - El Cuestionario pasa a estado “Validado” - Los puntos obtenidos por el empleado son tenidos en cuenta para la evaluación. - Se almacena la fecha de validación
ENTRADAS	El Cuestionario.
SALIDAS	Mensaje de confirmación de la validación.

Tabla 105. Caso de uso Comenzar Respuesta de Empleado

CASO DE USO	67. Comentar Respuesta de Empleado.
ACTORES	Empleado Supervisor
PROPÓSITO	Insertar un comentario a una Respuesta de Empleado.
RESUMEN	El Empleado supervisor inserta un comentario a una Respuesta del Empleado.
PRECONDICIONES	<ul style="list-style-type: none"> - El Empleado Supervisor debe estar relacionado con el cuestionario. - El Cuestionario debe estar en estado “Terminado”.
POSTCONDICIONES	Se inserta el comentario a la Respuesta de Empleado.
ENTRADAS	La Respuesta de Empleado y el comentario.
SALIDAS	Ninguna.

Tabla 106. Caso de uso Solicitar corrección Cuestionario

CASO DE USO	68. Solicitar corrección Cuestionario.
ACTORES	Empleado Supervisor.
PROPÓSITO	Solicitar la corrección de un Cuestionario por parte del Empleado que lo ha realizado.
RESUMEN	El Empleado Supervisor sugiere al Empleado que corrija el cuestionario porque las respuestas no le parecen apropiadas.
PRECONDICIONES	<ul style="list-style-type: none"> - El Empleado Supervisor debe estar relacionado con el cuestionario. - El Cuestionario debe estar en estado “Terminado”.
POSTCONDICIONES	<ul style="list-style-type: none"> - El Cuestionario pasa a estado “Preparado”. - Se almacena la fecha de la solicitud de revisión.
ENTRADAS	El cuestionario.
SALIDAS	Mensaje de confirmación de la solicitud de revisión.

4.3. Mapeo Casos de Uso – Modelo

En el método de desarrollo OO-Method, la transformación entre el modelo de Requisitos y el Modelo conceptual la realiza el analista de forma manual. En el caso que afecta a este PFC, el modelo de Requisitos son los Casos de Uso del apartado 4.2. *Especificación de Requisitos*. En esta sección se explica el mapeo entre los Casos de Uso y sus entidades en el Modelo Conceptual de Olivanova.

En primer lugar, se muestran los esquemas del mapeo de los dos Sistemas de Información que se construyen en este Proyecto. En estos esquemas se muestran los escenarios de Casos de Uso y las clases que del Modelo de Objetos de Olivanova que participan en su diseño. En la Figura 27 se muestra el correspondiente a la aplicación *SIEMACRe*, mientras que en la Figura 28 está representado el mapeo del sistema *SIEFAP*.

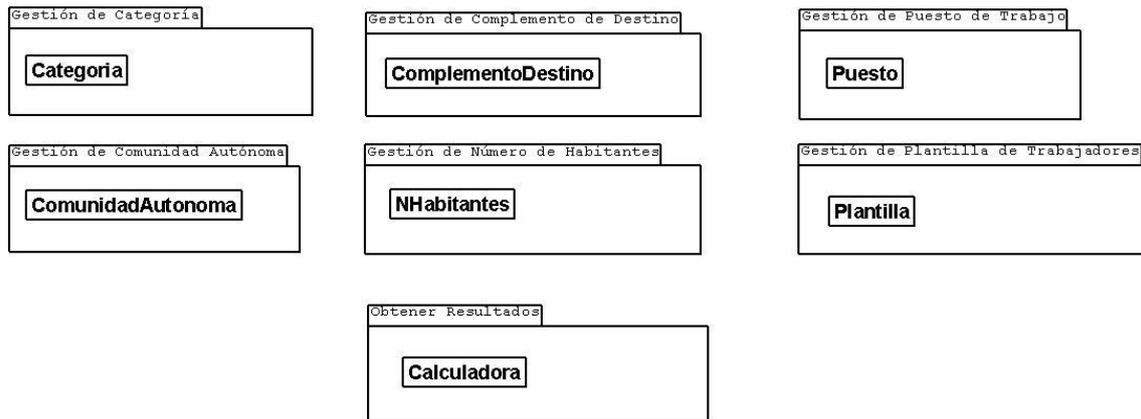


Figura 27. Esquema de mapeo SIESMACRe.

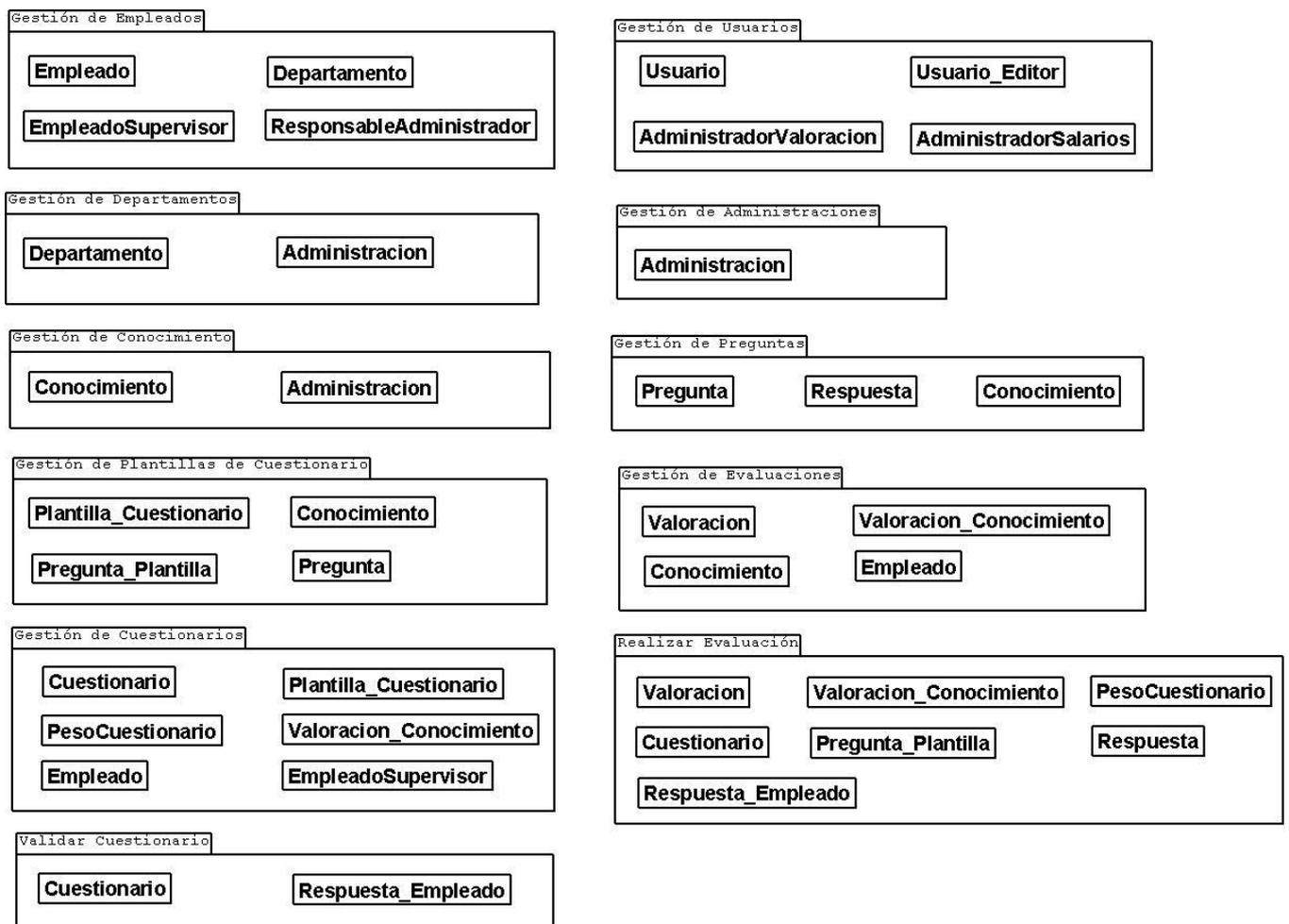


Figura 28. Esquema de mapeo SIEFAP.

Después de una visión general, se detalla el mapeo entre casos de uso y las herramientas de diseño disponibles en Olivanova.

4.3.1 Mapeo de SIESMACRe

Para la aplicación SIESMACRE (Sistema de Información de Estimación del Salario Mediante el Análisis de Conceptos Retributivos), se diseñan nueve clases: *Categoría*, *ComplemetoDestino*, *ComunidadAutonoma*, *Puesto*, *NHabitantes*, *Plantilla*, *Calculadora* y *AdministradorSalarios*. La clase *AdministradorSalarios* se corresponde con el agente *Administrador* de los casos de uso.

En la Figura 29 se observan las relaciones del modelo de Clases. Cada clase gestiona la información de un concepto salarial y todas ellas se relacionan con la clase *Calculadora*. La clase *Calculadora* es la que presenta el informe a partir del valor de los atributos de las instancias seleccionadas de sus clases relacionadas.

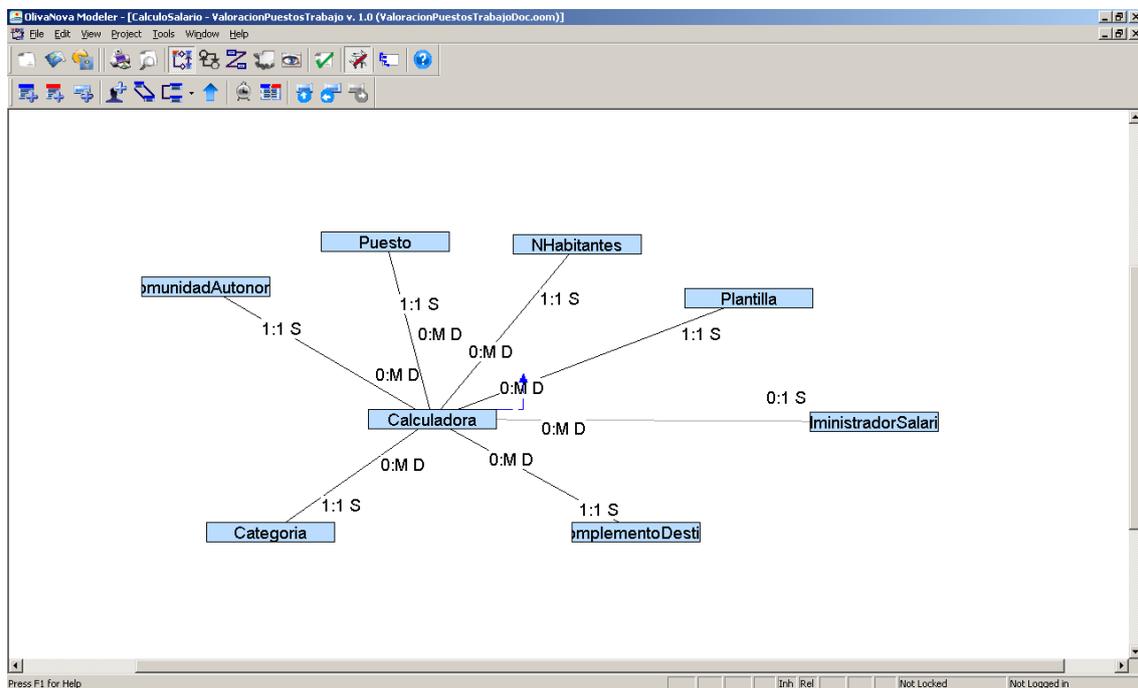


Figura 29. Modelo de objetos de SIESMACRe.

A continuación, se realiza un mapeo entre la Especificación de Requisitos y el Modelo Conceptual. Este mapeo se organiza por los mismos conceptos que los casos de uso presentados en la sección 4.2. *Especificación de Requisitos*. Así, se presenta la clase, los atributos y los servicios que se diseñan en el modelo para cumplir con las exigencias de la aplicación deseada. Los servicios diseñados son eventos, que aparecen en minúsculas en el documento o transacciones, que aparecen en mayúsculas.

Gestión de Categorías

El modelado de los Casos de Uso referentes a la **Gestión de Categorías** se realiza mediante la clase Categoría.

En la siguiente tabla se detallan los atributos y servicios de esta clase y se expone la correspondencia de éstos con los casos de uso.

Tabla 107. Correspondencias entre clase Categoría y Gestión de Categorías

CLASE	Categoría	
ATRIBUTOS	- Tipo	- Atributo identificador de la Categoría; tipo de dato String de dos posiciones.
	- SalarioMensual	- Atributo variable; tipo de dato Real.
	- SalarioAnual	- Atributo derivado a partir del SalarioMensual; tipo de dato Real
SERVICIOS Y SUS CORRESPONDENCIAS	crear_categoria	1. Alta de Categoría
	borrar_categoria	2. Baja de Categoría
	modificar_salario	3. Modificación de salario de Categoría
	actualizar_salario	4. Actualización de salario de Categoría
	consultar_salarios	5. Consulta de salario de Categoría

En los casos de uso, se especifica que el salario anual se calcula en base al mensual. Este cálculo se establece como una Derivación en la clase Categoría en la que el atributo *SalarioAnual* es 12 veces el atributo *SalarioMensual*. A su vez, el atributo *SalarioMensual* tiene una restricción que impide que sea negativo, que es la precondition de 1.-Alta de Categoría.

Para representar que el porcentaje de cambio de 4.-Actualización de salario de Categoría. debe ser mayor que cero, se añade una precondition al servicio *actualizar_salario*.

El agente que tiene visibilidad sobre los servicios de la clase es AdministradorSalarios, que puede acceder a su ejecución a través de la *Unidad de Interacción de Instancia (IIU, por sus siglas en inglés) Categoría* o de la *Unidad de Interacción de Población (PIU) Categoría*.

Gestión de Complemento de Destino

El modelado de los Casos de Uso referentes a la **Gestión de Complementos de Destino** se realiza mediante la clase ComplementoDestino. En la siguiente tabla se detallan los atributos y servicios de esta clase y se expone la correspondencia de éstos con los casos de uso.

Tabla 108. Correspondencias entre clase ComplementoDestino y Gestión de Complementos de Destino

CLASE	ComplementoDestino	
ATRIBUTOS	- Id_ComplementoDestino	- Atributo identificador del Complemento de destino; tipo de dato Natural.
	- ValorMensual	- Atributo variable; tipo de dato Real.
	- ValorAnual	- Atributo derivado a partir del ValorMensual; tipo de dato Real
SERVICIOS Y SUS CORRESPONDENCIAS	crear_complementoDestino	6. Alta de Complemento de Destino
	borrar_complementoDestino	7. Baja de Complemento de Destino
	modificar_valor	8. Modificación de valor económico
	actualizar_valor	9. Actualización de valor económico
	consultar_valores	10. Consulta de valor económico

En los casos de uso se especifica que el valor anual se calcula en base al mensual. Este cálculo se establece como una Derivación en la clase ComplementoDestino en la que el atributo *ValorAnual* es 12 veces el atributo *ValorMensual*. A su vez, el atributo *ValorMensual* tiene una restricción que impide que sea negativo, que es la precondition de 6.-Alta de Complemento de Destino.

El servicio *actualizar_valor* tiene una precondition que obliga a que el porcentaje de cambio del valor sea mayor que cero, como se especifica en 9.-Actualización de valor económico.

El agente que tiene visibilidad sobre los servicios de la clase es *AdministradorSalarios*, que puede acceder a su ejecución a través de la *IU ComplementoDestino* o de la *PIU ComplementoDestino*.

Gestión de Puesto de Trabajo

El modelado de los Casos de Uso referentes a la **Gestión de Puestos de Trabajo** se realiza mediante la clase Puesto. En la siguiente tabla se detallan los atributos y servicios de esta clase y se expone la correspondencia de éstos con los casos de uso.

Tabla 109. Correspondencias entre clase Puesto y Gestión de Puestos de Trabajo

CLASE	Puesto	
ATRIBUTOS	- Id_Puesto	- Atributo identificador del Puesto; tipo de dato Autonumérico.
	- NombrePuesto	- Atributo variable; tipo de dato String de 50 posiciones. Tiene restricción de unicidad
	- Media	- Atributo variable; tipo de dato Real
	- Percentil10	- Atributo variable; tipo de dato Real
	- Percentil25	- Atributo variable; tipo de dato Real
	- Percentil50	- Atributo variable; tipo de dato Real
	- Percentil75	- Atributo variable; tipo de dato Real
	- Percentil90	- Atributo variable; tipo de dato Real
SERVICIOS Y SUS CORRESPONDENCIAS	crear_puesto	11. Alta de Puesto de Trabajo
	borrar_puesto	12. Baja de Puesto de Trabajo
	modificar_nombre	13. Modificación de nombre de Puesto de Trabajo
	modificar_media	14. Modificación de la media del salario del Puesto de Trabajo
	AUMENTAR_PERCENTILES	15. Modificación de los percentiles del salario del Puesto de Trabajo
	DISMINUIR_PERCENTILES	15. Modificación de los percentiles del salario del Puesto de Trabajo
	actualizar_media	16. Actualización de la media del salario
	actualizar_percentiles	17. Actualización de los percentiles del salario
	consultar_media	18. Consulta de la media del salario
	consultar_percentiles	19. Consulta de los percentiles del salario

En los casos de uso 11.-Alta de Puesto de Trabajo., 14.-Modificación de la media del salario del Puesto de Trabajo. y 15.-Modificación de los percentiles del salario del Puesto de Trabajo. se especifica que la media y los percentiles tienen que ser mayor que cero y el orden de los percentiles en cuanto a su valor. Estos requisitos se representan en el modelo mediante restricciones de integridad.

Para realizar la modificación de los percentiles, se diseñan dos transacciones: *AUMENTAR_PERCENTILES*, que se utiliza para aumentar su valor, y *DISMINUIR_PERCENTILES*, que se utiliza para disminuirlo. Ambas transacciones ejecutan los eventos de cambio de cada percentil, pero varían en el orden en que lo hacen para que no se incumplan las restricciones citadas.

El agente que tiene visibilidad sobre los servicios de la clase es *AdministradorSalarios*, que puede acceder a su ejecución a través de la *IIU Puesto* o de la *PIU Puesto*.

Gestión de Comunidad Autónoma

El modelado de los Casos de Uso referentes a la **Gestión de Comunidad Autónoma** se realiza mediante la clase *ComunidadAutonoma*. En la siguiente tabla se detallan los atributos y servicios de esta clase y se expone la correspondencia de éstos con los casos de uso.

Tabla 110. Correspondencias entre clase *ComunidadAutonoma* y Gestión de Comunidad Autónoma

CLASE	ComunidadAutonoma	
ATRIBUTOS	- Id_ComunidadAutonoma	- Atributo identificador de ComunidadAutonoma; tipo de dato Autonumérico.
	- NombreComunidad	- Atributo variable; tipo de dato String de 50 posiciones. Tiene restricción de unicidad.
	- FactorCCAA	- Atributo variable; tipo de dato Entero
SERVICIOS Y SUS CORRESPONDENCIAS	ALTA_COMUNIDADFACTOR	20. Alta de Comunidad Autónoma
	borrar_comunidad	21. Baja de Comunidad Autónoma
	modificar_nombre	22. Modificación de Nombre de Comunidad Autónoma
	modificar_factorCCAA	23. Modificación del factor de Comunidad Autónoma
	consultar_factorCCAA	24. Consulta del factor de Comunidad Autónoma

El atributo *FactorCCAA* se modela como un entero porque puede ser tanto positivo como negativo y se empleará en el cálculo de las estadísticas de los salarios.

El caso de uso 20.- Alta de Comunidad Autónoma. se diseña con la transacción *ALTA_COMUNIDADFACTOR*, que se compone del evento de creación de instancia de la clase *ComunidadAutonoma* y el evento de modificación del factor. El motivo de este diseño es que el evento de creación no contiene el valor del factor.

El agente que tiene visibilidad sobre los servicios de la clase es *AdministradorSalarios*, que puede acceder a su ejecución a través de la *IIU ComplementoDestino* o de la *PIU ComplementoDestino*.

Gestión de Número de Habitantes

El modelado de los Casos de Uso referentes a la **Gestión de Número de Habitantes** se realiza mediante la clase NHabitantes. En la siguiente tabla se detallan los atributos y servicios de esta clase y se expone la correspondencia de éstos con los casos de uso.

Tabla 111. Correspondencias entre clase NHabitantes y Gestión de Número de Habitantes

CLASE	NHabitantes	
ATRIBUTOS	- Id_NHabitantes	- Atributo identificador de NHabitantes; tipo de dato Autonumérico.
	- LimiteInferiorNH	- Atributo variable; tipo de dato Natural.
	- LimiteSuperiorNH	- Atributo variable; tipo de dato Natural.
	- FactorNH	- Atributo variable; tipo de dato Entero.
SERVICIOS Y SUS CORRESPONDENCIAS	ALTA_HABITANTESFACTOR	25. Alta de intervalo del Número de Habitantes
	borrar_NHabitantes	26. Baja de intervalo del Número de Habitantes
	MODIFICAR_LIMITESNH	27. Modificación de Intervalo del Número de Habitantes
	modificar_factorNH	28. Modificación del factor de Intervalo del Número de Habitantes
	consultar_factorNH	29. Consulta del factor de Intervalo del Número de Habitantes

El intervalo de número de habitantes se modela con dos límites naturales, pues representan número de personas. Se añade una Restricción de integridad para que el límite inferior del intervalo no sea mayor que el superior y otra para que el límite inferior sea mayor de cero.

El atributo *FactorNH* se modela como un entero porque puede ser tanto positivo como negativo y se empleará en el cálculo de las estadísticas de los salarios.

El caso de uso 25.- Alta de intervalo de Número de Habitantes. se diseña con la transacción *ALTA_HABITANTESFACTOR*, que se compone del evento de creación de instancia de la clase NHabitantes y el evento de modificación del factor. El motivo de este diseño es que el evento de creación no contiene el valor del factor. También se crea la transacción *MODIFICAR_LIMITESNH* para cambiar el intervalo del número de habitantes. Esta transacción

ejecuta los servicios de cambio del límite superior e inferior, seleccionando el orden correcto de ejecución de estos eventos para que no se incumpla las restricciones de integridad.

El agente que tiene visibilidad sobre los servicios de la clase es *AdministradorSalarios*, que puede acceder a su ejecución a través de la *IU NHabitantes* o de la *PIU NHabitantes*.

Gestión de Plantilla de Trabajadores

El modelado de los Casos de Uso referentes a la **Gestión de Plantilla de Trabajadores** se realiza mediante la clase *Plantilla*. En la siguiente tabla se detallan los atributos y servicios de esta clase y se expone la correspondencia de éstos con los casos de uso.

Tabla 112. Correspondencias entre clase *Plantilla* y Gestión de Plantilla de Trabajadores

CLASE	Plantilla	
ATRIBUTOS	- Id_Plantilla	- Atributo identificador de Plantilla; tipo de dato Autonumérico.
	- LimiteInferiorPlantilla	- Atributo variable; tipo de dato Natural.
	- LimiteSuperiorPlantilla	- Atributo variable; tipo de dato Natural.
	- FactorPlantilla	- Atributo variable; tipo de dato Entero.
SERVICIOS Y SUS CORRESPONDENCIAS	ALTA_PLANTILLAFACTOR	30. Alta de intervalo de Plantilla
	borrar_plantilla	31. Baja de intervalo de Plantilla
	MODIFICAR_LIMITESPLANT	32. Modificación de intervalo de Plantilla
	modificar_factorPlant	33. Modificación del factor de intervalo de Plantilla
	consultar_factorPlant	34. Consulta del factor de intervalo de Plantilla

El intervalo de plantilla se modela con dos límites naturales, pues representan número de empleados. Se añade una Restricción de integridad para que el límite inferior del intervalo no sea mayor que el superior y otra para que el límite inferior sea mayor de cero.

El atributo *FactorPlantilla* se modela como un entero porque puede ser tanto positivo como negativo y se empleará en el cálculo de las estadísticas de los salarios.

El caso de uso 30.- Alta de intervalo de Plantilla. se diseña con la transacción *ALTA_PLANTILLAFACTOR*, que se compone del evento de creación de instancia de la clase *Plantilla* y el evento de modificación del factor. El motivo de este diseño es que el evento de creación no contiene el valor del factor. También se crea la transacción

MODIFICAR_LIMITESPLANT para cambiar el intervalo del número de habitantes. Esta transacción ejecuta los servicios de cambio del límite superior e inferior, seleccionando el orden correcto de ejecución de estos eventos para que no se incumpla la restricción comentada.

El agente que tiene visibilidad sobre los servicios de la clase es *AdministradorSalarios*, que puede acceder a su ejecución a través de la IIU Plantilla o de la PIU Plantilla.

Obtener Resultados

El modelado de los Casos de Uso referentes a la **Obtención de Resultados** se realiza mediante la clase *Calculadora*. En la siguiente tabla se detallan los atributos y servicios de esta clase y se expone la correspondencia de éstos con los casos de uso.

Tabla 113. Correspondencias entre clase *Calculadora* y Obtención de Resultados

CLASE	Calculadora	
ATRIBUTOS	- Id_Calculadora	- Atributo identificador de Calculadora; tipo de dato Autonumérico.
	- SalarioMensual	- Atributo derivado a partir de SalarioMensual de la clase <i>Categoria</i> ; tipo de dato Real.
	- SalarioAnual	- Atributo derivado a partir de SalarioAnual de la clase <i>Categoria</i> ; tipo de dato Real.
	- ValorCDMensual	- Atributo derivado a partir de ValorMensual de la clase <i>ComplementoDestino</i> ; tipo de dato Real.
	- ValorCDAnual	- Atributo derivado a partir de ValorAnual de la clase <i>ComplementoDestino</i> ; tipo de dato Real
	- ValorCEMensual	- Atributo variable; tipo de dato Real
	- ValorCEAnual	- Atributo derivado a partir de ValorCEMensual; tipo de dato Real
	- PagaExtra	- Atributo derivado a partir de SalarioMensual, ValorCDMensual y ValorCEMensual; tipo de dato Real
	- IngresosMensuales	- Atributo derivado a partir de SalarioMensual, ValorCDMensual y ValorCEMensual; tipo de dato Real
	- IngresosAnuales	- Atributo derivado a partir de SalarioAnual, ValorCDAnual, ValorCEAnual y PagaExtra; tipo de dato Real

	<p>- MediaAnual</p> <p>- Percentil10</p> <p>- Percentil25</p> <p>- Percentil50</p> <p>- Percentil75</p> <p>- Percentil90</p>	<p>- Atributo derivado a partir de Media de la clase Puesto, de FactorCCAA de la clase ComunidadAutonoma, de FactorNH de la clase NHabitantes y de FactorPlantilla de la clase Plantilla; tipo de dato Real</p> <p>Atributo derivado a partir de Percentil10 de la clase Puesto, de FactorCCAA de la clase ComunidadAutonoma, de FactorNH de la clase NHabitantes y de FactorPlantilla de la clase Plantilla; tipo de dato Real</p> <p>Atributo derivado a partir de Percentil25 de la clase Puesto, de FactorCCAA de la clase ComunidadAutonoma, de FactorNH de la clase NHabitantes y de FactorPlantilla de la clase Plantilla; tipo de dato Real</p> <p>Atributo derivado a partir de Percentil50 de la clase Puesto, de FactorCCAA de la clase ComunidadAutonoma, de FactorNH de la clase NHabitantes y de FactorPlantilla de la clase Plantilla; tipo de dato Real</p> <p>Atributo derivado a partir de Percentil75 de la clase Puesto, de FactorCCAA de la clase ComunidadAutonoma, de FactorNH de la clase NHabitantes y de FactorPlantilla de la clase Plantilla; tipo de dato Real</p> <p>Atributo derivado a partir de Percentil90 de la clase Puesto, de FactorCCAA de la clase ComunidadAutonoma, de FactorNH de la clase NHabitantes y de FactorPlantilla de la clase Plantilla; tipo de dato Real</p>
SERVICIOS Y SUS CORRESPONDENCIAS	crear_calculadora	35. Obtención de informe

La obtención de resultados se diseña con la clase Calculadora y la forma de realizar el caso de uso es mediante el evento *crear_calculadora*. La entrada de este evento son atributos de tipo de objeto de Categoria, ComplementoDestino, Puesto, ComunidadAutonoma, NHabitantes y Plantilla, además de un argumento de tipo real que es el valor del complemento específico.

La clase Calculadora tiene derivaciones para cada uno de los datos de salida del informe. Con los valores de los argumentos de entrada, estas derivaciones entran en juego para dar

valor a los atributos que luego se muestran como argumentos de salida del cálculo del salario y las estadísticas.

Se crea una clase en lugar de un servicio global por la intención de guardar la información de cada ejecución para estadísticas futuras.

El agente que tiene visibilidad sobre los servicios de la clase es Usuario_Anonimo, que puede acceder a su ejecución a través de la *Unidad de Interacción de Servicio (SIU, por sus siglas en inglés) crear_calculadora*.

4.3.1 Mapeo de SIEFAP

Para la aplicación SIEFAP (Sistema de Información de la Evaluación de Funcionarios de Administraciones Públicas), se diseñan 15 clases. Para facilitar la visión del esquema las clases se dividen en 7 clusters, como se observa en la Figura 30.

Los clusters son los siguientes:

- Administraciones, que contiene las clases Administracion y Departamento.
- Empleados, que contiene las clases Empleado, EmpleadoSupervisor y ResponsableAdministracion.
- Conocimientos, con la clase Conocimiento.
- Preguntas, que contiene las clases Pregunta y Respuesta.
- PCuestionarios, con las clases Plantilla_Cuestionario y Pregunta_Plantilla.
- Evaluaciones, que tiene las clases Valoracion y Valoracion_Conocimiento.
- Cuestionarios, con las clases Cuestionario, PesoCuestionario y Respuesta_Empleado.

Adicionalmente, existen cinco clases que son necesarias para la Gestión de Usuarios de ambos Sistemas de Información. Éstas son Usuario, Usuario_Anonimo, Usuario_Editor, AdministradorSalarios y AdministradorValoracion.

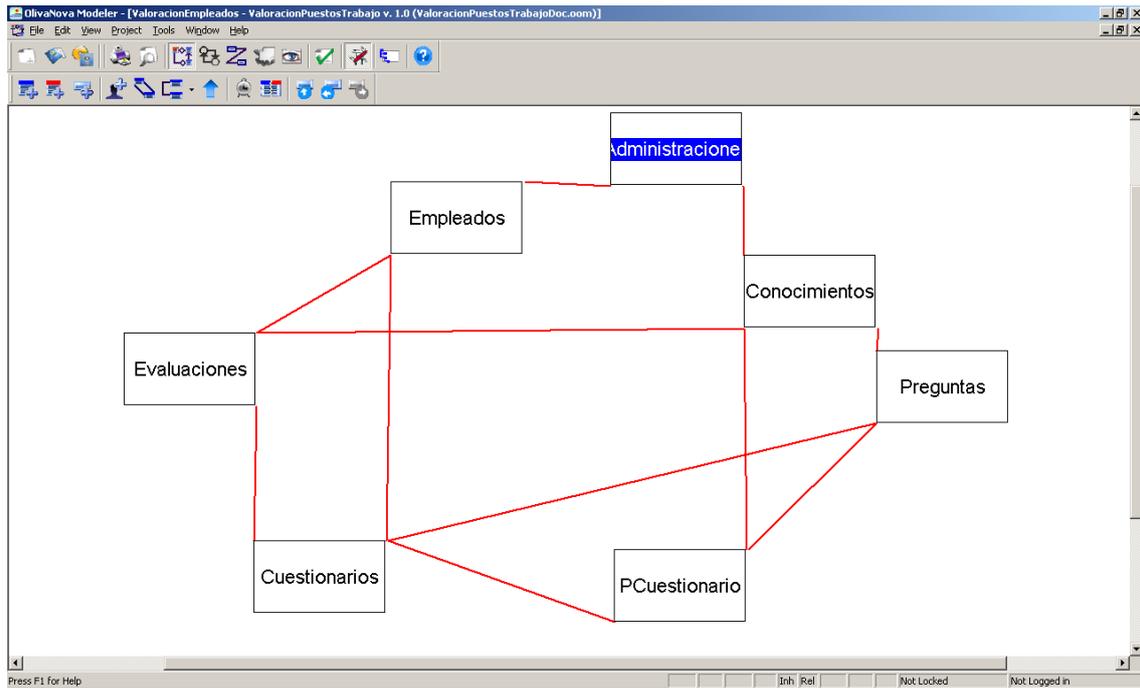


Figura 30. Modelo de objetos de SIEFAP.

El contenido de estos clusters se puede observar en la Figura 31.

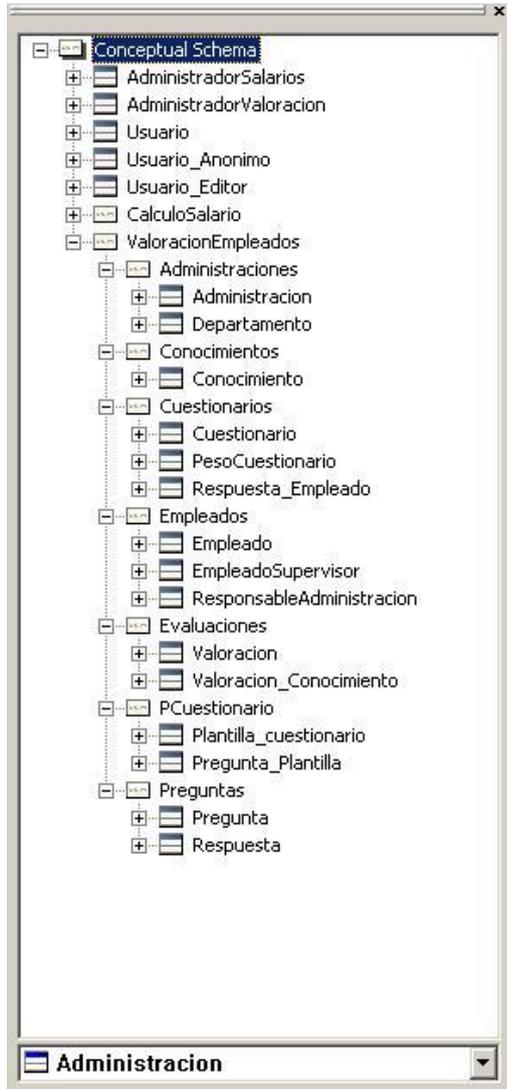


Figura 31. Clases del modelo de objetos de SIEFAP.

En esta sección se realiza un mapeo entre los casos de uso del apartado 4.2. *Especificación de Requisitos* con los elementos del modelo Conceptual que permiten su diseño. Para cada apartado, se explican las clases involucradas, sus atributos y servicios y las relaciones entre clases que intervienen en el diseño. Los servicios diseñados son eventos, que aparecen en minúsculas en el documento o transacciones, que aparecen en mayúsculas.

Gestión de Empleados

El modelado de los Casos de Uso referentes a la **Gestión de Empleados** se realiza mediante las clases (Empleado, EmpleadoSupervisor, ResponsableAdministrador y Departamento. Las relaciones entre ellas se pueden ver en la Figura 32. Entre Empleado y

Departamento existe una relación de asociación porque un Departamento puede tener uno o varios Empleados. Por su parte, el Empleado puede especializarse en EmpleadoSupervisor y en ResponsableAdministracion.

La clase ResponsableAdministracion se diseña para ejercer el Rol de Administrador de una única Administración.

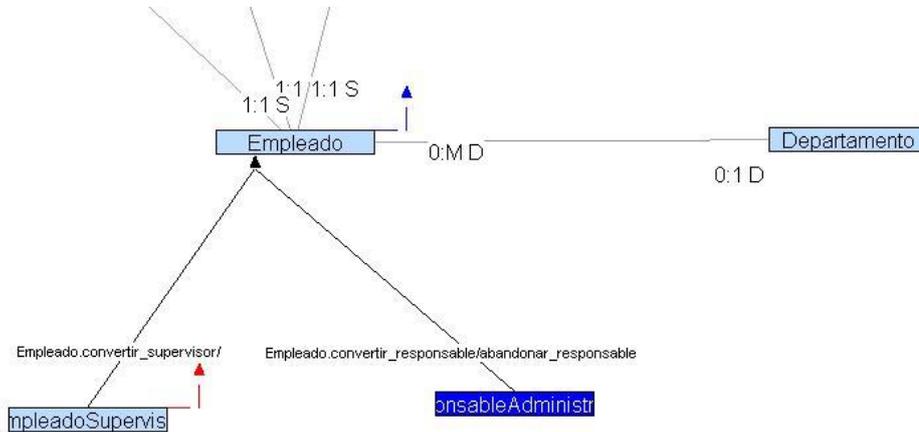


Figura 32. Modelo de objetos para la Gestión de Empleados.

En las siguientes tablas se detallan las clases implicadas en la Gestión de Empleados.

Tabla 114. Correspondencias entre clase Empleado y Gestión de Empleados

CLASE	Empleado	
ATRIBUTOS	<ul style="list-style-type: none"> - NIF - Nombre - Apellidos - FechaNacimiento - Salario - Sexo - Estado 	<ul style="list-style-type: none"> - Atributo identificador de Empleado; tipo de dato String de 9 posiciones. - Atributo variable; tipo de dato String de 40 posiciones. - Atributo variable; tipo de dato String de 100 posiciones. - Atributo variable; tipo de dato Fecha. - Atributo variable; tipo de dato Real. - Atributo variable; tipo de dato String de 1 posición. - Atributo variable; tipo de dato String de 1 posición.
SERVICIOS Y SUS CORRESPONDENCIAS	<ul style="list-style-type: none"> create_instance ACTIVAR_EMPLEADO desactivar_empleado BORRAR_EMPLEADO 	<ul style="list-style-type: none"> 1. Alta de Empleado 2. Activar Empleado 3. Desactivar Empleado 4. Baja de Empleado

	edit_instance	5. Modificación de detalles de Empleado
	convertir_supervisor	6. Convertir Empleado en Supervisor
	convertir_responsable	8. Convertir Empleado en Responsable

En la clase Empleado se crea el atributo *estado*, que puede tomar valores “A” de Activo o “I” de Inactivo. Por defecto, el valor del atributo es “A”. Algunos servicios tienen como precondición que el empleado esté activo, como son *edit_instance*, *convertir_supervisor* y *convertir_responsable*. Para *ACTIVAR_EMPLEADO*, debe estar Inactivo y para *desactivar_empleado* debe estar Activo.

La transacción *ACTIVAR_EMPLEADO* se diseña para realizar el borrado del departamento al que pertenecía el empleado y realizar la inserción en el nuevo departamento.

La transacción *BORRAR_EMPLEADO* se diseña para dar de baja del sistema al empleado. Con esta transacción se llama a los eventos de borrado de las clases *PesoCuestionario*, *Cuestionario*, *Valoracion_Conocimiento* y *Valoracion* para todas las instancias relacionadas con el empleado. Por último se ejecuta el evento de borrado del empleado.

Los agentes que tienen visibilidad sobre los servicios de la clase son *AdministradorValoracion* y *ResponsableAdministrador* de la administración a la que pertenece el empleado, que pueden acceder a su ejecución a través de la *IIU Empleado* o de la *PIU Empleado*. El servicio *create_instance* también es accesible desde la *IIU Departamento* y la *PIU Departamento*.

Tabla 115. Correspondencias entre clase EmpleadoSupervisor y Gestión de Empleados

CLASE	EmpleadoSupervisor	
ATRIBUTOS	Hereda los atributos de la clase Empleado y tiene dos propios. - FechaFin - Atributo variable; tipo de dato fecha. - FechaInicio - Atributo variable; tipo de dato fecha.	
SERVICIOS Y SUS	reconvertir_supervisor	6. Convertir Empleado en Supervisor
CORRESPONDENCIAS	abandonar_supervisor	7. Abandonar Rol de Supervisor

En la clase *EmpleadoSupervisor* se crean los atributos *FechaInicio* y *FechaFin* y los servicios *reconvertir_supervisor* y *abandonar_supervisor* para que un mismo Empleado pueda ser y dejar de ser supervisor tantas veces como sea necesario. La instancia de la especialización no se elimina porque puede estar relacionada con Cuestionarios y se controla si el supervisor está activo por el intervalo de fechas. Para realizar cualquiera de estas dos acciones el empleado debe estar activo.

Los agentes que tienen visibilidad sobre los servicios de la clase son *AdministradorValoracion* y *ResponsableAdministrador* de la administración a la que pertenece

el supervisor, que pueden acceder a su ejecución a través de la *IIU EmpleadoSupervisor* o de la *PIU EmpleadoSupervisor*.

Tabla 116. Correspondencias entre clase ResponsableAdministracion y Gestión de Empleados

CLASE	ResponsableAdministracion	
ATRIBUTOS	Hereda los atributos de la clase Empleado y tiene dos propios.	
SERVICIOS Y SUS CORRESPONDENCIAS	abandonar_responsable	9. Abandonar Rol de Responsable

En la clase ResponsableAdministracion se define una Restricción que obliga a que el empleado esté activo si está especializado en responsable.

Los agentes que tienen visibilidad sobre los servicios de la clase son AdministradorValoracion y ResponsableAdministrador de la administración a la que pertenece el responsable, que pueden acceder a su ejecución a través de la *IIU ResponsableAdministracion* o de la *PIU ResponsableAdministracion*.

Gestión de Usuarios

El modelado de los Casos de Uso referentes a la **Gestión de Usuarios** se realiza mediante las clases Usuario, Usuario_Editor, AdministradorValoracion, AdministradorSalarios y Usuario_Anonimo. Las relaciones entre ellas se pueden ver en la Figura 33.

El Usuario puede especializarse en Usuario_Editor, AdministradorValoracion y AdministradorSalarios. Éstas tres clases y Usuario_Anonimo representan algunos de los roles de los dos Sistemas de Información.

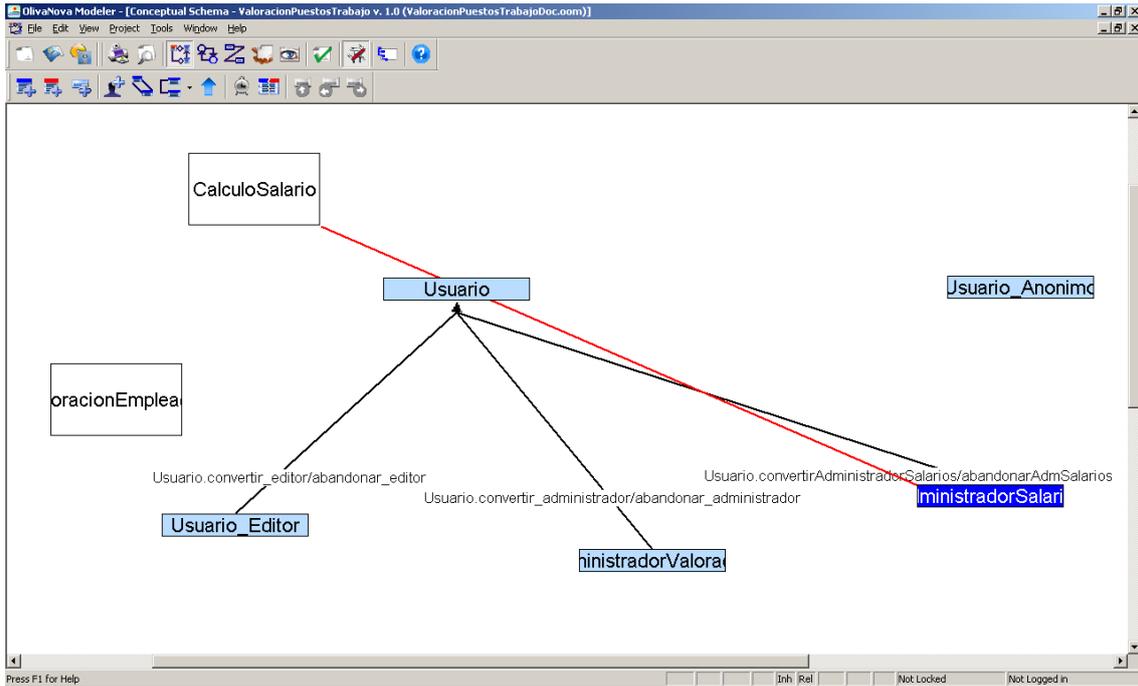


Figura 33. Modelo de objetos para la Gestión de Usuarios.

En las siguientes tablas se detallan las clases implicadas en la Gestión de Usuarios.

Tabla 117. Correspondencias entre clase Usuario y Gestión de Usuarios

CLASE	Usuario	
ATRIBUTOS	- Nombre_usuario	Atributo identificador de Usuario; tipo de dato String de 64 posiciones.
SERVICIOS Y SUS CORRESPONDENCIAS	crear_usuario	10. Crear Usuario
	borrar_usuario	11. Borrar Usuario
	convertir_administrador	12. Convertir Usuario en Administrador (SIEFAC)
	convertirAdministradorSalarios	12. Convertir Usuario en Administrador (SIESMACRe)
	convertir_editor	14. Convertir Usuario en Editor

Los servicios *convertir_administrados* y *convertirAdministradorSalarios* se diseñan para convertir un usuario en administrador de SIEFAP o en administrador de SIESMACRe, respectivamente.

Los agentes que tienen visibilidad sobre los servicios de la clase son *AdministradorValoracion* y *AdministradorSalarios*, que pueden acceder a su ejecución a través de la *IIU Usuario* o de la *PIU Usuario*.

Tabla 118. Correspondencias entre clase Usuario_editor y Gestión de Usuarios

CLASE	Usuario_editor	
ATRIBUTOS	Hereda los atributos de la clase Usuario.	
SERVICIOS Y SUS CORRESPONDENCIAS	abandonar_editor	15. Abandonar Rol de Editor

El agente que tiene visibilidad sobre los servicios de la clase es AdministradorValoracion, que puede acceder a su ejecución a través de la *IIU Usuario_Editor* o de la *PIU Usuario_Editor*.

Tabla 119. Correspondencias entre clase AdministradorValoracion y Gestión de Usuarios

CLASE	AdministradorValoracion	
ATRIBUTOS	Hereda los atributos de la clase Usuario.	
SERVICIOS Y SUS CORRESPONDENCIAS	abandonar_administrador	13. Abandonar Rol de Administrador (SIEFAP)

El agente que tiene visibilidad sobre los servicios de la clase es AdministradorValoracion, que puede acceder a su ejecución a través de la *SIU abandonar_administrador*.

Tabla 120. Correspondencias entre clase AdministradorSalarios y Gestión de Usuarios

CLASE	AdministradorSalarios	
ATRIBUTOS	Hereda los atributos de la clase Usuario.	
SERVICIOS Y SUS CORRESPONDENCIAS	abandonar_administrador	13. Abandonar Rol de Administrador (SIESMACRe)

El agente que tiene visibilidad sobre los servicios de la clase es AdministradorSalarios, que puede acceder a su ejecución a través de la *SIU abandonarAdmSalarios*.

Gestión de Departamentos

El modelado de los Casos de Uso referentes a la **Gestión de Departamentos** se realiza mediante las clases Departamento y Administracion. Las relaciones entre ellas se pueden ver en la Figura 34. Entre Departamento y Administracion existe una relación de asociación porque una Administracion se puede organizar en varios departamentos.



Figura 34. Modelo de objetos para la Gestión de Departamentos.

En la siguiente tabla se detalla la clase implicada en la Gestión de Departamentos.

Tabla 121. Correspondencias entre clase Departamento y Gestión de Departamentos

CLASE	Departamento	
ATRIBUTOS	- id_Departamento	- Atributo identificador de Departamento; tipo de dato Autonumérico.
	- Nombre	- Atributo variable; tipo de dato Texto.
	- NEmpleados	- Atributo derivado a partir de la relación con la clase Empleado; tipo de dato Natural.
	- PresupuestoAsignado	- Atributo variable; tipo de dato Real.
	- PresupuestoDispuesto	- Atributo derivado a partir del atributo Salario de la clase Empleado; tipo de dato Real.
SERVICIOS Y SUS CORRESPONDENCIAS	create_instance	16. Alta de Departamento
	delete_instance	17. Baja de Departamento
	modificar_nombre	18. Modificación del nombre de Departamento
	asignar_presupuesto	19. Asignar Presupuesto a Departamento
	consulta_presupuesto	20. Consulta de Presupuesto de Departamento
	consultar_emleados	21. Consulta de Empleados de Departamento

La clase Departamento se diseña con la Restricción de que el *PresupuestoAsignado* sea mayor o igual que cero y con dos Derivaciones: el número de empleados es la cantidad de instancias de la clase Empleado en estado Activo relacionadas con el Departamento, mientras que el presupuesto dispuesto es la suma del salario de estos empleados. Para el servicio *delete_instance* se ha creado una precondición, que impide que un Departamento se pueda borrar si tiene empleados activos asociados.

Los agentes que tienen visibilidad sobre los servicios de la clase son *AdministradorValoracion* y *ResponsableAdministrador* de la administración a la que pertenece el departamento, que pueden acceder a su ejecución a través de la *IIU Departamento* o de la *PIU Departamento*.

Gestión de Administraciones

El modelado de los Casos de Uso referentes a la **Gestión de Administraciones** se realiza mediante la clase *Administracion*. En la siguiente tabla se detalla la clase implicada en la Gestión de Administraciones.

Tabla 122. Correspondencias entre clase *Administracion* y Gestión de Administraciones

CLASE	Administracion	
ATRIBUTOS	<ul style="list-style-type: none"> - id_Administracion - Nombre - NDepartamentos - NEmpleados - PresupuestoAsignado - PresupuestoDispuesto 	<ul style="list-style-type: none"> - Atributo identificador de Administracion; tipo de dato Autonumérico. - Atributo variable; tipo de dato Texto. - Atributo derivado a partir de la relación con la clase Departamento; tipo de dato Natural. - Atributo derivado a partir del atributo NEmpleados de la clase Departamento; tipo de dato Natural. - Atributo variable; tipo de dato Real. - Atributo derivado a partir del atributo PresupuestoDispuesto de la clase Departamento; tipo de dato Real.
SERVICIOS Y SUS CORRESPONDENCIAS	create_instance	22. Alta de Administración
	BORRAR_ADMINISTRACION	23. Baja de Administración
	modificar_nombre	24. Modificación del nombre de Administración
	asignar_presupuesto	25. Asignar Presupuesto a Administración
	consulta_presupuesto	26. Consulta de Presupuesto de la Administración
	consultar_emleados	27. Consulta de Empleados de la Administración

La clase Administracion se diseña con la Restricción de integridad de que el *PresupuestoAsignado* sea mayor o igual que cero y con tres Derivaciones: el número de departamentos es la cantidad de instancias de la clase Departamento relacionados con la Administracion, el número de empleados es la suma de los empleados de los departamentos relacionados, mientras que el presupuesto dispuesto es la suma del presupuesto dispuesto de los departamentos asociados.

La transacción BORRAR_ADMINISTRACION se diseña para dar de baja del sistema la administración. Con esta transacción se llama a los eventos desactivación de los empleados relacionados, en caso de que los hubiere, y a los eventos de borrado de la clase Departamento de todos los departamentos relacionados con la administración.

El agente AdministradorValoracion tiene visibilidad sobre todos los servicios de la clase Administracion. Por su parte, el agente ResponsableAdministracion tiene visibilidad sobre los servicios *modificar_nombre*, *asignar_presupuesto*, *consulta_presupuesto* y *consultar_empleados* de la administración a la que pertenece. Este último agente no puede crear ni eliminar administraciones. Los servicios de esta clase son accesibles a través de la IIU Administracion y de la PIU Administracion.

Gestión de Conocimiento

El modelado de los Casos de Uso referentes a la **Gestión de Conocimientos** se realiza mediante las clases Conocimiento y Administracion. Las relaciones entre ellas se pueden ver en la Figura 35.

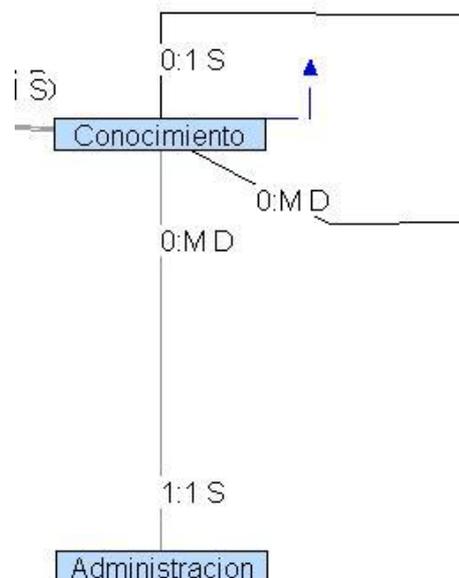


Figura 35. Modelo de objetos para la Gestión de Conocimientos.

Entre Conocimiento y Administracion existe una relación de asociación porque los Conocimientos pertenecen a una Administracion. Existe una relación de asociación entre Conocimientos para representar el concepto de Subconocimiento.

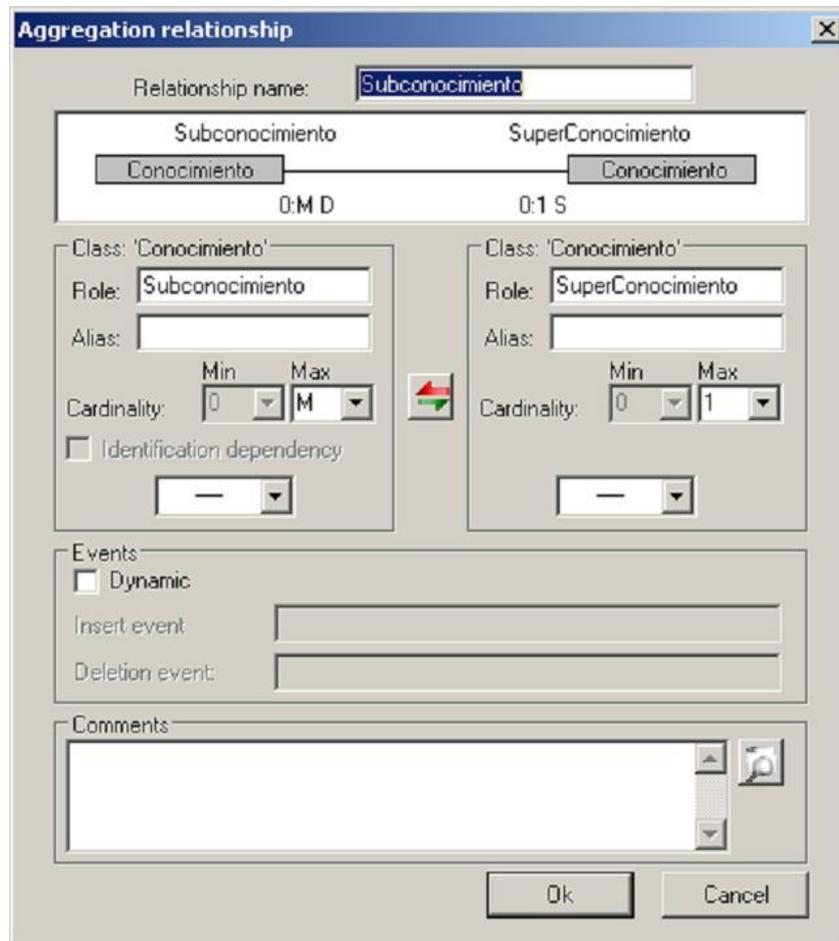


Figura 36. Relación de asociación de Subconocimientos.

En la siguiente tabla se detallan los atributos de la clase Conocimientos y los servicios implicados en la Gestión de Conocimientos.

Tabla 123. Correspondencias entre clase Conocimiento y Gestión de Conocimientos

CLASE	Conocimiento	
ATRIBUTOS	- id_Conocimiento - Atributo identificador de Conocimiento; tipo de dato Autonumérico. - Nombre - Atributo variable; tipo de dato Texto. - Descripcion - Atributo variable; tipo de dato Texto. - NPreguntas - Atributo derivado a partir de la relación con la clase Pregunta; tipo de dato Natural.	
SERVICIOS Y SUS CORRESPONDENCIAS	crear_instancia	28. Crear Conocimiento
	edit_instance	29. Modificar Conocimiento
	delete_instance	30. Eliminar Conocimiento
	crear_instancia	31. Crear Subconocimiento

En la clase Conocimiento se diseña una Derivación para almacenar el número de preguntas asociadas a un conocimiento.

Para el servicio *crear_instancia* se diseña una precondición para que cuando se crea un subconocimiento la administración sea la misma que la del superconocimiento.

En el servicio *delete_instance* se han diseñado 4 precondiciones, que impiden que se ejecute el servicio si tiene preguntas, plantillas de cuestionario, evaluaciones de conocimiento y subconocimientos relacionados.

Los agentes que tienen visibilidad sobre los servicios de la clase son AdministradorValoracion, UsuarioEditor y ResponsableAdministrador de la administración a la que pertenece el conocimiento, que pueden acceder a su ejecución a través de la *IIU Conocimiento* o de la *PIU Conocimiento*.

Gestión de Preguntas

El modelado de los Casos de Uso referentes a la **Gestión de Preguntas** se realiza mediante las clases Pregunta, Respuesta y Conocimiento. Las relaciones entre ellas se pueden ver en la Figura 37. La Respuesta es una clase componente de la Pregunta. Entre la clase Pregunta y la clase Conocimiento existe una relación de asociación porque cada pregunta debe pertenecer a uno o varios Conocimientos.

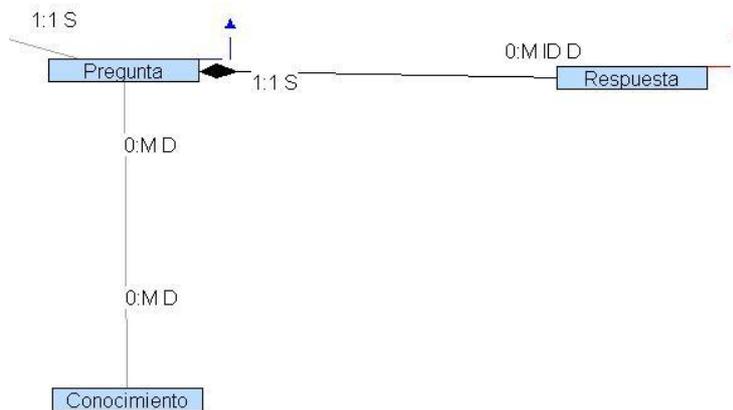


Figura 37. Modelo de objetos para la Gestión de Preguntas.

En la siguiente tabla se detallan los atributos de la clase Pregunta y sus servicios.

Tabla 124. Correspondencias entre clase Pregunta y Gestión de Preguntas

CLASE	Pregunta	
ATRIBUTOS	<ul style="list-style-type: none"> - id_Pregunta - Texto - PuntuacionMaxima - NRespuestas - Estado - NConocimientos 	<ul style="list-style-type: none"> - Atributo identificador de Pregunta; tipo de dato Autonumérico. - Atributo variable; tipo de dato Texto. - Atributo derivado del atributo Puntuacion de las Respuestas que componen la Pregunta; tipo de dato Real. - Atributo derivado de las Respuestas que componen la Pregunta; tipo de dato Natural. - Atributo derivado del atributo Estado de los Cuestionarios en que participa la Pregunta; tipo de dato String de 1 posición. - Atributo derivado de los Conocimientos en que participa la Pregunta; tipo de dato Entero.
SERVICIOS Y SUS CORRESPONDENCIAS	<ul style="list-style-type: none"> CREA_PREGUNTA edit_instance InsPregunta 	<ul style="list-style-type: none"> 32. Crear Pregunta 33. Modificar Pregunta 37. Asignar Pregunta a Conocimiento

En la clase Pregunta se definen cuatro Derivaciones. Dos de estas derivaciones son la cantidad de respuestas y la cantidad de conocimientos asociados a la pregunta. Otra derivación en la puntuación máxima, definida como el máximo valor de las respuestas asociadas. Por último, el atributo *Estado* se emplea para distinguir si la pregunta está siendo

utilizada en cuestionarios: un valor “N” indica que no está siendo utilizado, mientras que si está en uso este atributo toma el valor “S”.

El servicio *InsPregunta* se establece para realizar la asociación de instancias entre preguntas y conocimientos. Para el servicio de modificación de pregunta, se crea una precondición que obliga a que el estado de la pregunta sea “N”.

Por su parte, la transacción *CREA_PREGUNTA* ejecuta los eventos de creación de instancia, de asociación de la pregunta al conocimiento de entrada y de creación de tantas respuestas como se hayan introducido en la entrada, con un mínimo de 1 y un máximo de 5.

En el modelo de presentación, la *Unidad de Interacción de Servicio (SIU) CREA_PREGUNTA* se define con una Navegación Condicional que permite dar de alta más respuestas si se selecciona la opción.

En la tabla se detallan los atributos de la clase Respuesta y sus servicios.

Tabla 125. Correspondencias entre clase Respuesta y Gestión de Preguntas

CLASE	Respuesta	
ATRIBUTOS	- id_Respuesta	- Atributo identificador de Respuesta; tipo de dato Autonumérico.
	- Texto	- Atributo variable; tipo de dato Texto.
	- Puntuacion	- Atributo variable; tipo de dato Real.
SERVICIOS Y SUS CORRESPONDENCIAS	create_instance	34. Crear Respuesta
	edit_instance	35. Modificar Respuesta
	delete_instance	36. Eliminar Respuesta

En la clase Respuesta. Se define una Restricción para que el atributo *Puntuacion* no sea inferior a cero. Para sus servicios, se establece también la precondición de que el estado de la pregunta a la que obligatoriamente están relacionados sea “N”. Además, para el servicio *delete_instance* se establece una precondición que impide su ejecución y, por tanto el borrado de la respuesta, si es la única respuesta asociada a su pregunta, pues una pregunta no se puede quedar sin respuestas.

Para visualizar correctamente la Pregunta y sus Respuestas asociadas se crean dos Unidades de Interacción de Maestro/Detalle (MDIU, por sus siglas en inglés). Éstas son *MDIU Pregunta*, que muestra la información de una pregunta y de la población de respuestas relacionadas, y *MDIU PreguntaPIU*, que muestra un listado de preguntas y la población de sus respuestas.

En la tabla se detallan los servicios de la clase Conocimiento referentes a la Gestión de Preguntas.

Tabla 126. Correspondencias entre clase Conocimiento y Gestión de Preguntas

CLASE	Conocimiento	
SERVICIOS Y SUS CORRESPONDENCIAS	DelPregunta	38. Desasignar Pregunta de Conocimiento

El servicio *DelPregunta* de la clase Conocimiento es el que se emplea para eliminar una relación establecida entre objetos de Conocimiento y de Pregunta. Este servicio tiene dos Precondiciones: no puede existir la pregunta en plantillas de cuestionario del mismo conocimiento y la pregunta debe estar relacionada con otros conocimientos porque no puede quedarse sin relación con ninguno.

Los agentes que pueden realizar acciones de Gestión de Preguntas son *Usuario_Editor*, *AdministradorValoracion* y *ResponsableAdministrador* de la administración a la que pertenecen las preguntas. El *Usuario_Editor* tiene visibilidad sobre todos estos servicios, mientras que los otros dos agentes sólo pueden ejecutar los servicios *InsPregunta* y *DelPregunta*.

Los servicios que se diseñan para la Gestión de Preguntas son accesibles desde la *IU Pregunta*, *PIU Pregunta*, *MDIU Pregunta* y *MDIU PreguntaPIU*. Además, los servicios de asignación y desasignación de preguntas a conocimientos son accesibles desde la *PIU Conocimiento*.

Gestión de Plantillas de Cuestionario

El modelado de los Casos de Uso referentes a la **Gestión de Plantillas de Cuestionario** se realiza mediante las clases *Plantilla_cuestionario*, *Pregunta_Plantilla*, *Pregunta* y *Conocimiento*. Las relaciones entre ellas se pueden ver en la Figura 38. *Pregunta_Plantilla* es una clase componente de la clase *Plantilla_cuestionario* y debe estar relacionada con una *Pregunta*. Entre la clase *Plantilla_cuestionario* y la clase *Conocimiento* existe una relación de asociación porque cada pregunta debe pertenecer un Conocimiento.

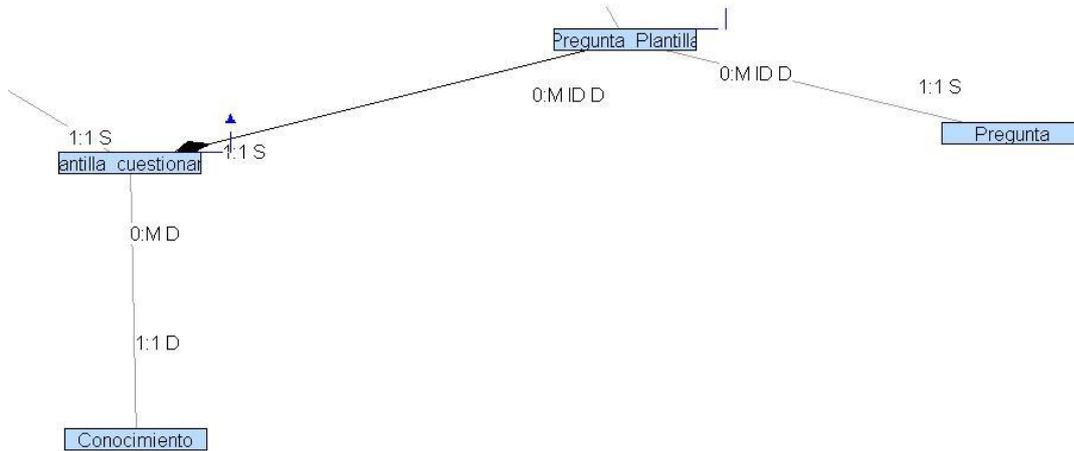


Figura 38. Modelo de objetos para la Gestión de Plantillas de Cuestionario.

En la siguiente tabla se detalla la clase Plantilla_Cuestionario.

Tabla 127. Correspondencias entre clase Plantilla_Cuestionario y Gestión de Plantillas de Cuestionario

CLASE	Plantilla_Cuestionario	
ATRIBUTOS	<ul style="list-style-type: none"> - id_Plantilla_cuestionario - id_Version - Nombre - Descripcion - NPreguntas - ValorTotal - PuntuacionMaxima - Estado 	<ul style="list-style-type: none"> - Atributo identificador de Plantilla_cuestionario; tipo de dato Autonumérico. - Atributo identificador de Plantilla_cuestionario; tipo de dato Autonumérico. - Atributo variable; tipo de dato Texto. - Atributo variable; tipo de dato Texto. - Atributo derivado de la relación con la clase Pregunta_Plantilla; tipo de dato Natural. - Atributo derivado del atributo Valor de la clase Pregunta_Plantilla; tipo de dato Real. - Atributo derivado del atributo PuntosMaximos de la clase Pregunta_Plantilla; tipo de dato Real. - Atributo derivado del atributo Estado de la clase Cuestionario; tipo de dato String de una posición
SERVICIOS Y SUS	create_instance	39. Crear Plantilla de Cuestionario

CORRESPONDENCIAS	edit_instance	40. Modificar Plantilla de Cuestionario
	InsPlatilla_cue	44. Asignar Plantilla de Cuestionario a Conocimiento

En la clase *Plantilla_Cuestionario* se definen cuatro Derivaciones. Una de estas derivaciones es la cantidad de preguntas que contiene la plantilla de cuestionario y que es la cantidad de instancias de la clase *Pregunta_Plantilla* relacionadas. Para el cálculo de la puntuación máxima se crea el atributo *ValorTotal*, que es la suma del atributo *Valor* de la clase *Pregunta_Plantilla*, y el atributo *PuntuacionMaxima*, que es la suma del atributo *PuntosMaximos* de la clase *Pregunta_Plantilla* dividido por el atributo *Valor*. Por último, el atributo *Estado* se emplea para distinguir si la plantilla está siendo utilizada en cuestionarios: un valor "N" indica que no está siendo utilizado, mientras que si está en uso este atributo toma el valor "S".

El servicio *InsPlatilla_cue* se establece para realizar la asociación de instancias entre plantillas de cuestionario y conocimientos. Al ser la asociación de cardinalidad máxima 1, este servicio ofrece un cambio de conocimiento asignado. Para eso, se definen dos precondiciones, que obligan a que la Plantilla no esté siendo utilizada en cuestionarios y que todas las preguntas que lo componen estén también relacionadas con el nuevo conocimiento.

Para el servicio de modificación de la plantilla de cuestionario también se diseña la precondición de que la plantilla esté en estado "N".

Para visualizar correctamente la plantilla de cuestionario se crea la *MDIU Plantilla_cuestionario*, que muestra la información de una plantilla de cuestionario y de la población de preguntas relacionadas.

En la tabla se detalla la clase *Pregunta_Plantilla*.

Tabla 128. Correspondencias entre clase Pregunta_Plantilla y Gestión de Plantillas de Cuestionario

CLASE	Pregunta_Plantilla	
ATRIBUTOS	- id_Plantilla_cuestionario	- Atributo identificador de Pregunta_Plantilla definido en la clase Plantilla_Cuestionario; tipo de dato Autonumérico.
	- id_Version	- Atributo identificador de Pregunta_Plantilla definido en la clase Plantilla_Cuestionario; tipo de dato Autonumérico.
	- id_Pregunta	- Atributo identificador de Pregunta_Plantilla definido en la clase Pregunta; tipo de dato Autonumérico.
	- Valor	- Atributo variable; tipo de dato Real.
	- PuntosMaximos	- Atributo derivado a partir de PuntuacionMaxima de la clase Pregunta y del atributo Valor; tipo de dato Real.
SERVICIOS Y SUS CORRESPONDENCIAS	create_instance	41. Asignar Pregunta a Plantilla de Cuestionario
	edit_instance	42. Desasignar Pregunta de Plantilla de Cuestionario
	edit_instance	43. Cambiar valor de Pregunta

La clase Pregunta_Plantilla tiene sus atributos de identificación definidos en sus clases relacionadas porque sólo puede existir mientras exista la relación entre la plantilla de cuestionario y la pregunta.

Se añade una Restricción para indicar que el atributo *Valor* debe ser mayor o igual que cero. Se diseña también la Derivación del atributo *PuntosMaximos* de forma que sea la puntuación máxima de la pregunta por su *Valor* en la plantilla de cuestionario.

Los tres servicios de la clase tienen la precondition de que el atributo *Estado* de la instancia de la clase *Plantilla_Cuestionario* valga "N" y el servicio de creación tiene además la precondition de que el conocimiento de la plantilla de cuestionario esté entre los conocimientos relacionados a la pregunta.

Los agentes que pueden realizar acciones de Gestión de Plantillas de cuestionario son *Usuario_Editor*, *AdminitradorValoracion* y *ResponsableAdministrador* de la administración a la que pertenecen las plantillas de cuestionario. El *Usuario_Editor* tiene visibilidad sobre todos estos servicios, mientras que los otros dos agentes sólo pueden ejecutar el servicios *InsPlantillaCue*.

Los servicios que se diseñan para la Gestión de Preguntas son accesibles desde la *IIIU Plantilla_Cuestionario*, *PIU Plantilla_Cuestionario* y *MDIU Plantilla_Cuestionario*.

Gestión de Evaluaciones

El modelado de los Casos de Uso referentes a la **Gestión de Evaluaciones** se realiza mediante las clases Valoracion, Valoracion_Conocimiento, Empleado y Conocimiento. Las relaciones entre ellas se pueden ver en la Figura 39. Cada Valoracion se asocia con un Empleado. Las instancias de la clase Valoracion_Conocimiento se deben asociar con un Empleado y un Conocimiento. Entre la clase Valoracion y Valoracion_Conocimiento existe una asociación porque cada Valoracion puede dividirse en una o varias Valoracion_Conocimiento.

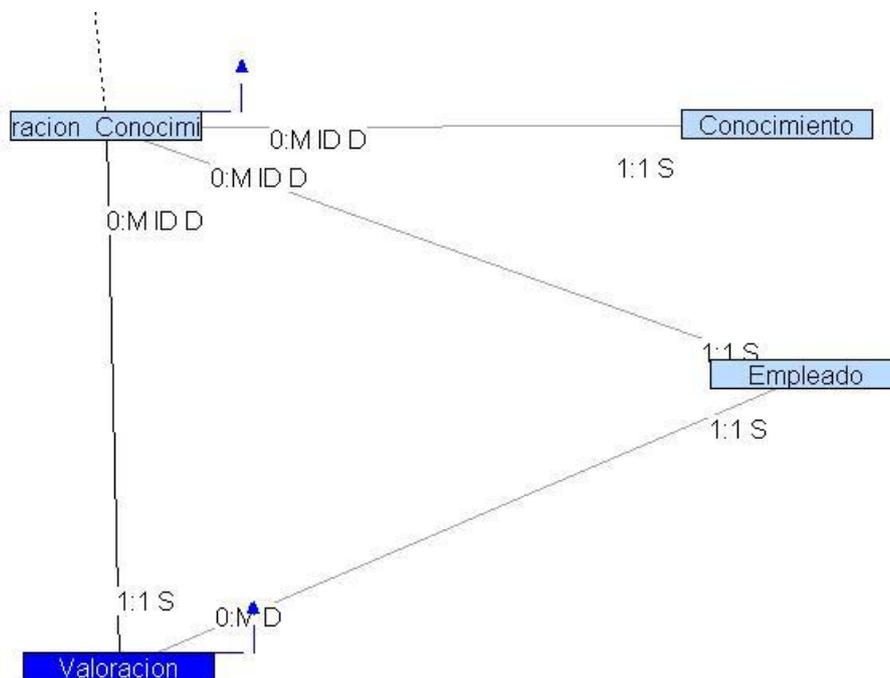


Figura 39. Modelo de objetos para la Gestión de Evaluaciones.

En la siguiente tabla se detalla la clase Valoracion.

Tabla 129. Correspondencias entre clase Valoracion y Gestión de Evaluaciones

CLASE	Valoracion	
ATRIBUTOS	- id_Valoracion	- Atributo identificador de Valoracion; tipo de dato Autonumérico.
	- Puntos	- Atributo variable; tipo de dato Natural.
	- PuntosObtenidos	- Atributo derivado del atributo PuntuacionObtenida de la clase Valoracion_Conocimiento; tipo de dato Real.
	- FechaCreacion	- Atributo constante; tipo de dato Fecha.
	- FechaFin	- Atributo constante; tipo de dato Fecha.
	- Estado	- Atributo variable; tipo de dato String de 1 posición.
	- Descripcion	- Atributo variable; tipo de dato Texto.
SERVICIOS Y SUS CORRESPONDENCIAS	create_instance	45. Crear Evaluación
	edit_instance	47. Editar Evaluación
	cambiar_estado	48. Cambiar estado
	BORRAR_EVALUACION	50. Eliminar Evaluación

En la clase Valoracion, se definen dos Restricciones; una para indicar que el número de puntos totales no puede ser inferior a los obtenidos por el empleado y la otra restringe la asignación de cuestionarios a las evaluaciones de conocimiento relacionadas que no tengan subconocimientos.

El valor del atributo derivado *PuntosObtenidos* se define como la suma del atributo *PuntuacionObtenida* de las evaluaciones de conocimiento relacionadas que no tienen Superconocimiento.

El atributo *Estado* de la clase Valoracion puede tomar valores “E” de “En Edición”, “A” de “Activo” y “F” de “Finalizado”. En los servicios *edit_instance* y *BORRAR_EVALUACION* se crea una Precondición para que sólo se puedan ejecutar en caso de que el estado sea “E”.

El servicio *cambiar_estado* modifica el valor de este atributo mediante evaluaciones de situación en el modelo funcional, de forma que el *Estado* pasa de “E” a “A” o de “A” a “F”. Este servicio tiene como Precondiciones que se ejecute desde los estados “A” o “E” y que si se ejecuta desde el estado “E”, la suma de los atributos *PuntosConocimiento* de las instancias de la clase Valoracion_Conocimiento relacionadas y que no tienen Superconocimientos sea igual al atributo *Puntos*.

La transacción *BORRAR_EVALUACION* borra primero todas las evaluaciones de conocimiento relacionadas y después la evaluación de entrada.

Se crea la *MDIU Valoracion*, para visualizar de forma conjunta una evaluación y la población de evaluaciones de conocimiento que la forman.

En la tabla se detalla la clase Valoracion_Conocimiento.

Tabla 130. Correspondencias entre clase Valoracion_Conocimiento y Gestión de Evaluaciones

CLASE	Valoracion_Conocimiento
ATRIBUTOS	<ul style="list-style-type: none"> - id_Valoracion - Atributo identificador de Valoracion_Conocimiento definido en la clase Valoracion; tipo de dato Autonumérico. - NIF - Atributo identificador de Valoracion_Conocimiento definido en la clase Empleado; tipo de dato String de 9 posiciones. - id_Conocimiento - Atributo identificador de Valoracion_Conocimiento definido en la clase Conocimiento; tipo de dato Autonumérico. - NCuestionariosTotales - Atributo derivado a partir de la relación con la clase PesoCuestionario; tipo de dato Natural. - NCuestionariosPendientes - Atributo derivado a partir de los atributos NCuestionariosTotales, NCuestionariosFinalizados y NCuestionariosValidados; tipo de dato Natural. - NCuestionariosFinalizados - Atributo derivado a partir de la relación con la clase PesoCuestionario; tipo de dato Natural. - NCuestionariosValidados - Atributo derivado a partir de la relación con la clase PesoCuestionario; tipo de dato Natural. - NivelObtenido - Atributo derivado a partir del atributo PorcentajePuntuacionObtenida; tipo de dato Natural.

	<p>- PuntosConocimiento</p> <p>PorcentajePuntuacionObtenida</p> <p>- PuntuacionObtenida</p> <p>- SumaPesos</p> <p>- PorcentajePuntos</p>	<p>- Atributo derivado a partir del atributo Puntos de la clase Valoracion y del atributo PorcentajePuntos; tipo de dato Real.</p> <p>- Atributo derivado a partir del atributo PesoPorcentaje de la clase PesoCuestionario; tipo de dato Real.</p> <p>- Atributo derivado a partir de los atributos PuntosConocimiento y NivelObtenido; tipo de dato Real.</p> <p>- Atributo derivado a partir del atributo Peso de la clase PesoCuestionario; tipo de dato Real.</p> <p>- Atributo variable; tipo de dato Real.</p>
SERVICIOS Y SUS CORRESPONDENCIAS	create_instance	46. Crear Evaluación de Conocimiento
	edit_instance	49. Editar Evaluación de Conocimiento
	BORRAR_VCONOCIMIENTO	51. Eliminar Evaluación de Conocimiento

Esta clase tiene sus atributos de identificación definidos en sus clases relacionadas, pues se asocia a un empleado, un conocimiento y una evaluación en el momento de la creación de forma estática, es decir, que no puede cambiar de objeto asociado.

En la clase Valoracion_Conocimiento, se definen dos Restricciones; una para indicar que el porcentaje de puntos debe estar entre 0 y 100 y para indicar que si el estado de la evaluación no es "E", la suma de los puntos de las evaluaciones de subconocimientos debe ser igual a los puntos de su superconocimiento.

En esta clase se definen las siguientes derivaciones:

- *NcuestionariosTotales* es la cantidad de objetos de la clase PesoCuestionario relacionados y representa al número de cuestionarios de los que se compone una evaluación de conocimiento. En el caso de que existan evaluaciones de subconocimiento, es la suma de los cuestionarios totales de estas evaluaciones.
- *NcuestionariosFinalizados* es la cantidad de objetos de la clase PesoCuestionario relacionados con estado del cuestionario finalizado y representa al

número de cuestionarios finalizados que hay en una evaluación de conocimiento. En el caso de que existan evaluaciones de subconocimiento, es la suma de los cuestionarios finalizados de estas evaluaciones.

- *NcuestionariosValidados* es la cantidad de objetos de la clase *PesoCuestionario* relacionados con estado del cuestionario validado y representa al número de cuestionarios validados por un supervisor en una evaluación de conocimiento. En el caso de que existan evaluaciones de subconocimiento, es la suma de los cuestionarios validados de estas evaluaciones.

- *NcuestionariosPendientes* es la diferencia entre los cuestionarios totales y los finalizados y validados. Representa al número de cuestionarios que aún no se han terminado de responder.

- *PuntosConocimiento* son los puntos máximos que se pueden obtener en la evaluación del conocimiento. Se calcula a partir del atributo *PorcentajePuntos* y el atributo *Puntos* de la instancia de la clase *Valoración* relacionada; en caso de tratarse de un subconocimiento, se calcula a partir del atributo *PuntosConocimiento* de la evaluación del superconocimiento.

- *SumaPesos* es la suma del atributo *Peso* de las instancias relacionadas de la clase *PesoCuestionario*. Este atributo se utiliza en el cálculo del porcentaje de puntos obtenido por un empleado en la evaluación del conocimiento.

- *PorcentajePuntuacionObtenida* se calcula como la suma del atributo *PesoPorcentaje* de las instancias relacionadas de la clase *PesoCuestionario* dividido por el valor del atributo *SumaPesos*. Representa el porcentaje de puntos obtenidos en las respuestas de cuestionarios sobre los puntos totales de la evaluación del conocimiento.

- *NivelObtenido* es un valor entre 0 y 5 que indica la calificación del empleado en la evaluación del conocimiento. Se calcula a partir del atributo *PorcentajePuntuacionObtenida*. La Figura 40 muestra la derivación correspondiente a este atributo.

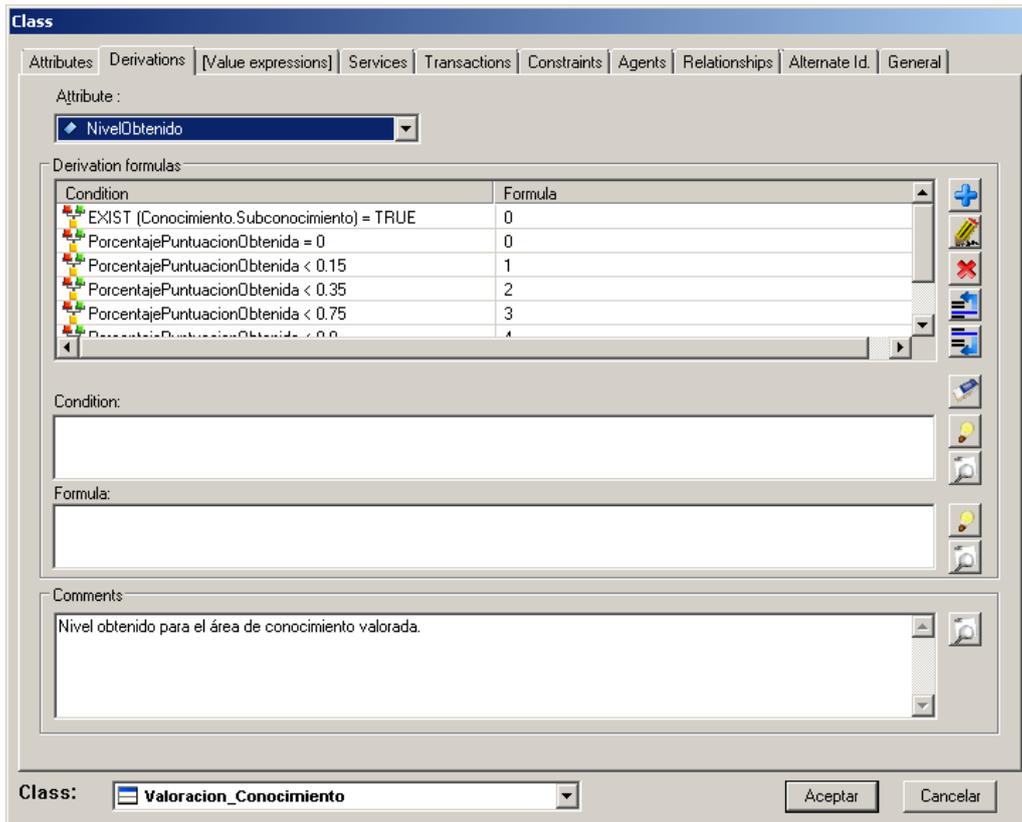


Figura 40. Derivación del atributo NivelObtenido de la clase Valoracion_Conocimiento.

- *PuntuacionObtenida* son los puntos que se le suman a la evaluación global del empleado por parte de este conocimiento. Se calcula aplicando un porcentaje del atributo *PuntosConocimiento* según el valor del atributo *NivelObtenido*. En caso de que la evaluación sea de un Superconocimiento, se toma el valor de la suma del atributo *PuntuacionObtenida* de las evaluaciones de sus subconocimientos. La Figura 41 muestra la derivación correspondiente a este atributo.

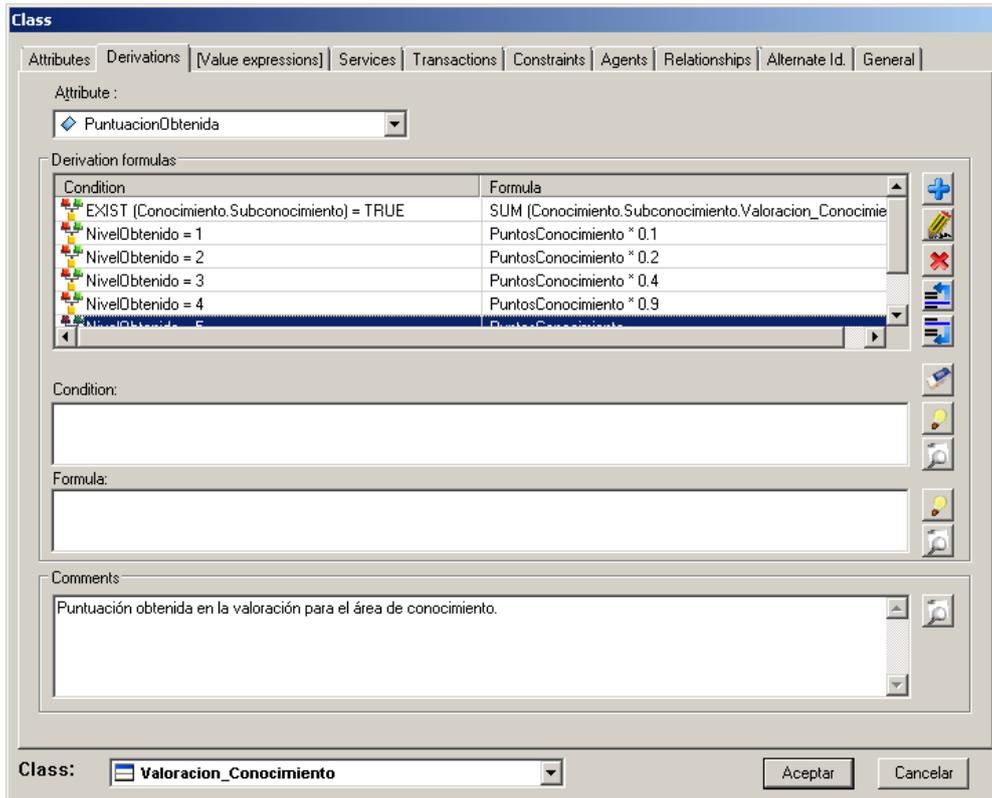


Figura 41. Derivación del atributo PuntuacionObtenida de la clase Valoracion_Conocimiento.

Los servicios de Valoracion_Conocimiento tienen la precondition de que el Estado de la evaluación relacionada sea “En Edición”. Además, el servicio de creación de evaluaciones de conocimiento también comprueba en condiciones que el empleado pertenezca a la administración de entrada y esté relacionado con la evaluación de entrada.

Se define también una precondition en el servicio *create_instance* para asegurar que las evaluaciones de conocimientos que se desean crear tengan ya creadas la evaluación de su superconocimiento. Para el servicio *BORRAR_VCONOCIMIENTO* se diseña una precondition que impide que se borre una evaluación de conocimiento si aún existen evaluaciones de sus subconocimientos.

En cuanto al modelo de presentación, se crea la *MDIU Valoracion_ConocimientoCo*, para visualizar de forma conjunta la población de evaluaciones de conocimiento y los cuestionarios que las forman.

Los agentes que pueden realizar acciones de Gestión de Evaluaciones son *AdminitradorValoracion* y *ResponsableAdministrador* de la administración a la que pertenecen los empleados a evaluar.

Tabla 131. Correspondencias entre clase Cuestionario y Gestión de Cuestionarios

CLASE	Cuestionario	
ATRIBUTOS	<ul style="list-style-type: none"> - NIF - id_Cuestionario - Puntuacion - NRespuestas - Estado - FechaCreacion - FechaComienzo - FechaFin - FechaCorreccion - FechaValidacion - PorcentajeObtenido 	<ul style="list-style-type: none"> - Atributo identificador de Cuestionario definido en la clase Empleado; tipo de dato String de 9 posiciones. - Atributo identificador de Cuestionario; tipo de dato Autonumérico. - Atributo derivado a partir del atributo PuntuacionObtenida de la clase Respuesta_Empleado y del atributo ValorTotal de la clase Plantilla_cuestionario; tipo de dato Real. - Atributo derivado a partir de la relación con la clase Respuesta_Empleado; tipo de dato Natural. - Atributo variable; tipo de dato String de una posición. - Atributo constante; tipo de dato Fecha. - Atributo variable; tipo de dato Fecha. - Atributo derivado a partir del atributo Puntuacion y del atributo PuntuacionMaxima de la clase Plantilla_cuestionario; tipo de dato Real.
SERVICIOS Y SUS CORRESPONDENCIAS	CREAR_CUESTIONARIO	52. Crear Cuestionario
	cambiar_estado	56. Dejar Cuestionario Preparado
	ASIGNAR_SUPERVISOR	57. Asignar Supervisor a Cuestionario
	BORRAR_CUESTIONARIO	58. Eliminar Cuestionario

Se añade a la clave de identificación el NIF del empleado que realiza el cuestionario para así tener contabilizados y ordenados los cuestionarios por empleado.

Para la clase Cuestionario se definen por restricciones que el empleado que lo realiza es está relacionado con la evaluación de conocimiento a la que pertenece y que el número de respuestas no es superior al de preguntas.

Las Derivaciones de esta clase son las siguientes:

- *Puntuacion* es la suma del atributo *PuntuacionObtenida* de la clase *Respuesta_Empleado* dividido por el atributo *ValorTotal* de la clase

Plantilla_Cuestionario. Representa la puntuación que obtiene un empleado por las respuestas que da al cuestionario.

- *NRespuestas* es la cantidad de instancias de la clase *Respuesta_Empleado* relacionadas con este cuestionario.

- *PorcentajeObtenido* es el porcentaje de puntos que obtenidos en el cuestionario con respecto a los puntos totales de la plantilla de cuestionario. Se calcula cuando el atributo *Estado* vale "V" de "Validado".

El atributo *Estado* puede tomar valores "E" de "En Edición", "P" de "Preparado", "R" de "En respuesta", "F" de "Finalizado" y "V" de Validado. El valor de este atributo varía con la ejecución de servicios de gestión y realización de cuestionarios.

La transacción *CREAR_CUESTIONARIO* ejecuta el evento de creación de la clase *Cuestionario* y el evento de creación de la clase *PesoCuestionario* para asignarlo a una evaluación de conocimiento. Este servicio da valor al atributo *FechaCreacion* y asigna al atributo *Estado* el valor "E".

El servicio *cambiar_estado* asigna el valor "P" al atributo *Estado* si se supera la precondición de que haya empleado supervisor relacionado con el cuestionario.

La transacción *ASIGNAR_SUPERVISOR* elimina el supervisor asignado al cuestionario y lo relaciona con el empleado supervisor de entrada. Para eso, se deben pasar las precondiciones de que el empleado supervisor esté vigente y que el cuestionario no esté ya validado.

La transacción *BORRAR_CUESTIONARIO* tiene una precondición que obliga a que el cuestionario no esté asignado a evaluaciones de conocimiento en estado distinto de "En Edición". Si se cumple con la precondición, el cuestionario se elimina del sistema.

En la tabla se detalla la clase *PesoCuestionario*.

Tabla 132. Correspondencias entre clase *PesoCuestionario* y Gestión de Cuestionarios

CLASE	PesoCuestionario	
ATRIBUTOS	<ul style="list-style-type: none"> - NIF - id_Cuestionario - id_Conocimiento - id_Valoracion - Peso - PesoPorcentaje 	<ul style="list-style-type: none"> - Atributo identificador de <i>PesoCuestionario</i> definido en la clase <i>Empleado</i>; tipo de dato String de 9 posiciones. - Atributo identificador de <i>PesoCuestionario</i> definido en la clase <i>Cuestionario</i>; tipo de dato Autonumérico. - Atributo identificador de <i>PesoCuestionario</i> definido en la clase <i>Conocimiento</i>; tipo de dato Autonumérico. - Atributo identificador de <i>PesoCuestionario</i> definido en la clase <i>Valoracion</i>; tipo de dato Autonumérico. - Atributo variable; tipo de dato Real. - Atributo derivado a partir del atributo <i>Peso</i> y del atributo <i>PorcentajeObtenido</i> de la clase <i>Cuestionario</i>; tipo de dato Real.
SERVICIOS Y SUS CORRESPONDENCIAS	create_instance	53. Asignar Cuestionario a Evaluación de Conocimiento
	delete_instance	54. Desasignar Cuestionario de Evaluación de Conocimiento
	edit_instance	55. Editar Cuestionario

Esta clase no tiene atributos de identificación propios porque sólo existe si se establece la relación entre un cuestionario y una evaluación de conocimiento.

Como Restricción de la clase se define que el atributo *Peso* sea mayor o igual que cero y se diseña como Derivación que el atributo *PesoPorcentaje* sea el resultado del atributo *PorcentajeObtenido* de la clase *Cuestionario* por el *Peso*.

Para ejecutar cualquiera de los servicios de esta clase, la evaluación debe estar en estado de "Edición". Este requisito se define como precondition. Para el servicio *create_instance* se definen más preconditiones que obligan a que el conocimiento del cuestionario sea el mismo que el de la evaluación del conocimiento, que sólo se asocian cuestionarios a evaluaciones conocimientos que no se dividen en subconocimientos y a que el empleado que realiza el cuestionario sea el mismo que el de la evaluación de conocimiento.

Los agentes que pueden realizar acciones de Gestión de Cuestionarios son *AdminitradorValoracion* y *ResponsableAdministrador* de la administración a la que pertenecen los empleados a evaluar.

Los servicios son accesibles desde la IIU *CuestionarioAdministrador* y *PIU CuestionarioAdministrador*. Además, los servicios de creación y asignación de cuestionarios están accesibles también desde la IIU *Valoracion_Conocimiento*, *PIU Valoracion_Conocimiento*, *MDIU Valoracion* y *MDIU Valoracion_ConocimientoCo*.

Realizar Evaluación

El modelado de los Casos de Uso referentes a la **Realización de la Evaluación** se realiza mediante las clases *Cuestionario*, *Respuesta_Empleado*, *Pregunta_Plantilla*, *Respuesta*, *Valoracion_Conocimiento* y *PesoCuestionario*. Las relaciones entre ellas se pueden ver en la Figura 43. Las Respuestas a cada Preguntas de los Cuestionarios se representan en el modelo con la clase *Respuesta_Empleado*. Las instancias de esta clase se relacionan obligatoriamente con una instancia de la clase *Cuestionario*, otra de la clase *Pregunta_Plantilla* y otra de la clase *Respuesta*.

Por otra parte, los Cuestionarios se relacionan con las Evaluaciones de Conocimiento a través de la clase *PesoCuestionario*. A su vez la clase *Valoracion_Conocimiento* se debe relacionar con una *Valoracion*.

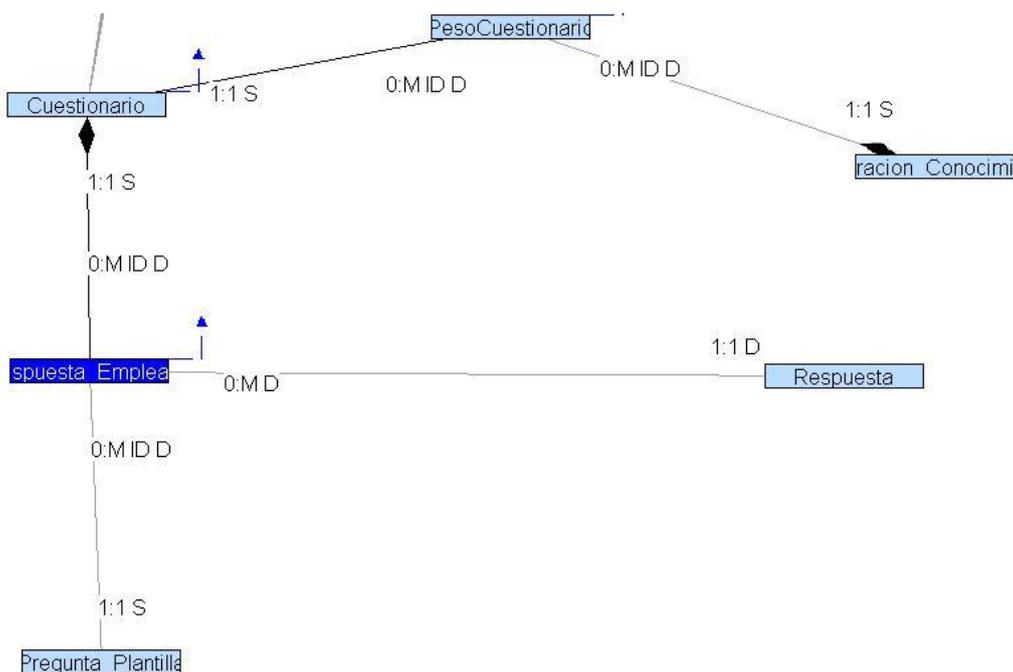


Figura 43. Modelo de objetos para la Realización de la Evaluación.

En la siguiente tabla se especifican los servicios de la clase Cuestionario que participan en la Realización de evaluaciones.

Tabla 133. Correspondencias entre clase Cuestionario y Realización de Evaluaciones

CLASE	Cuestionario	
SERVICIOS Y SUS CORRESPONDENCIAS	empezar_cuestionario	59. Responder Cuestionario
	empezar_cuestionario	60. Comenzar Cuestionario
	terminar_cuestionario	62. Terminar Cuestionario

El servicio *empezar_cuestionario* es el diseño de los casos de uso 59.-Responder Cuestionario. y 60.-Comenzar Cuestionario.. Se define una Navegación Condicional a la *SIU empezar_cuestionario* para llegar a la Unidad de Interacción que tiene las acciones de respuesta, *MDIU Plantilla_CuestionarioAcc*. Este servicio también asigna el valor “R” de “En respuesta” al Estado del cuestionario. Para poder ejecutarlo se debe cumplir con la Precondición de que el estado sea “Preparado” o “En Respuesta”.

El servicio *terminar_cuestionario* cambia el estado del cuestionario a “Finalizado”, para que pueda ser validado por el empleado supervisor. Tiene como Precondiciones que el estado sea “En Respuesta” y que el número de respuestas sea igual al de preguntas de la plantilla de cuestionario.

En la siguiente tabla se detalla la clase Respuesta_Empleado.

Tabla 134. Correspondencias entre clase Respuesta_Empleado y Realización de Evaluaciones

CLASE	Respuesta_Empleado	
ATRIBUTOS	- id_RespuestaEmpleado	- Atributo identificador de Respuesta_Empleado; tipo de dato Autonumérico.
	- id_Cuestionario	- Atributo identificador de Respuesta_Empleado definido en Cuestionario; tipo de dato Autonumérico.
	- id_Pregunta	- Atributo identificador de Respuesta_Empleado definido en la clase Pregunta; tipo de dato Autonumérico.
	- NIF	- Atributo identificador de Respuesta_Empleado definido en la clase Empleado; tipo de dato String de 9 posiciones.
	- PuntuacionObtenida	- Atributo derivado obtenido a partir del

		atributo <i>Puntuacion</i> de la clase <i>Respuesta</i> y del atributo <i>Valor</i> de la clase <i>Pregunta_Plantilla</i> ; tipo de dato Real.
	- Correccion	- Atributo variable; tipo de dato Texto.
SERVICIOS Y SUS CORRESPONDENCIAS	responder	61. Responder Pregunta de Cuestionario
	elige_respuesta	61. Responder Pregunta de Cuestionario
	borrar_respuesta	61. Responder Pregunta de Cuestionario

La clase *Repuesta_Empleado* tiene además de su identificador propio, los atributos de identificación de las clases *Cuestionario* y *Pregunta_Plantilla*, ya que una vez que se crea la instancia su relación con estas dos clases es estática.

En esta clase, se define como derivación la puntuación obtenida por el empleado con su respuesta. Se hace mediante el atributo *PuntuacionObtenida*, que es el resultado de multiplicar el atributo *Puntuacion* de la clase *Respuesta* y el atributo *Valor* de la clase *Pregunta_Plantilla*.

El caso de uso 61.- Responder Pregunta de Cuestionario. se diseña con 3 servicios; el servicio *responder* se puede lanzar cuando aún no hay respuesta a una pregunta de cuestionario, el servicio *elige_respuesta* se diseña para poder modificar la opción de respuesta elegida y el servicio *borrar_respuesta* se diseña para dar la posibilidad de eliminar la opción de respuesta seleccionada.

Para ejecutar estos 3 servicios se definen Precondiciones para que el cuestionario esté en estado de "En Respuesta" y que el empleado que contesta a la pregunta sea en realidad el que está asignado al cuestionario.

El agente que puede realizar estas acciones es el Empleado. Los servicios de la clase *Cuestionario* son accesibles desde la *IIU CuestionarioEmpleado* y *PIU CuestionarioEmpleado* Por su parte, los servicios de *Respuesta_Empleado* son accesibles desde la *SIU empezar_cuestionario*, que define una Navegación Condicional que permite llegar a la *MDIU Plantilla_cuestionarioAcc*.

Esta Unidad de Interacción está formada por la *IIU Plantilla_cuestionarioAcc* y la *MDIU Pregunta_PlantillaPIU*, que a su vez se compone de una población de preguntas de plantillas, una población de sus opciones de respuesta y la respuesta seleccionada por el empleado. Para mostrar la respuesta elegida por el empleado a una pregunta de cuestionario se emplea una Condición de Filtrado. En la Figura 44 se muestra la definición de esta MDIU y el mecanismo

definido para que se muestre sólo la instancia de la clase *Respuesta_Empleado* que corresponde.

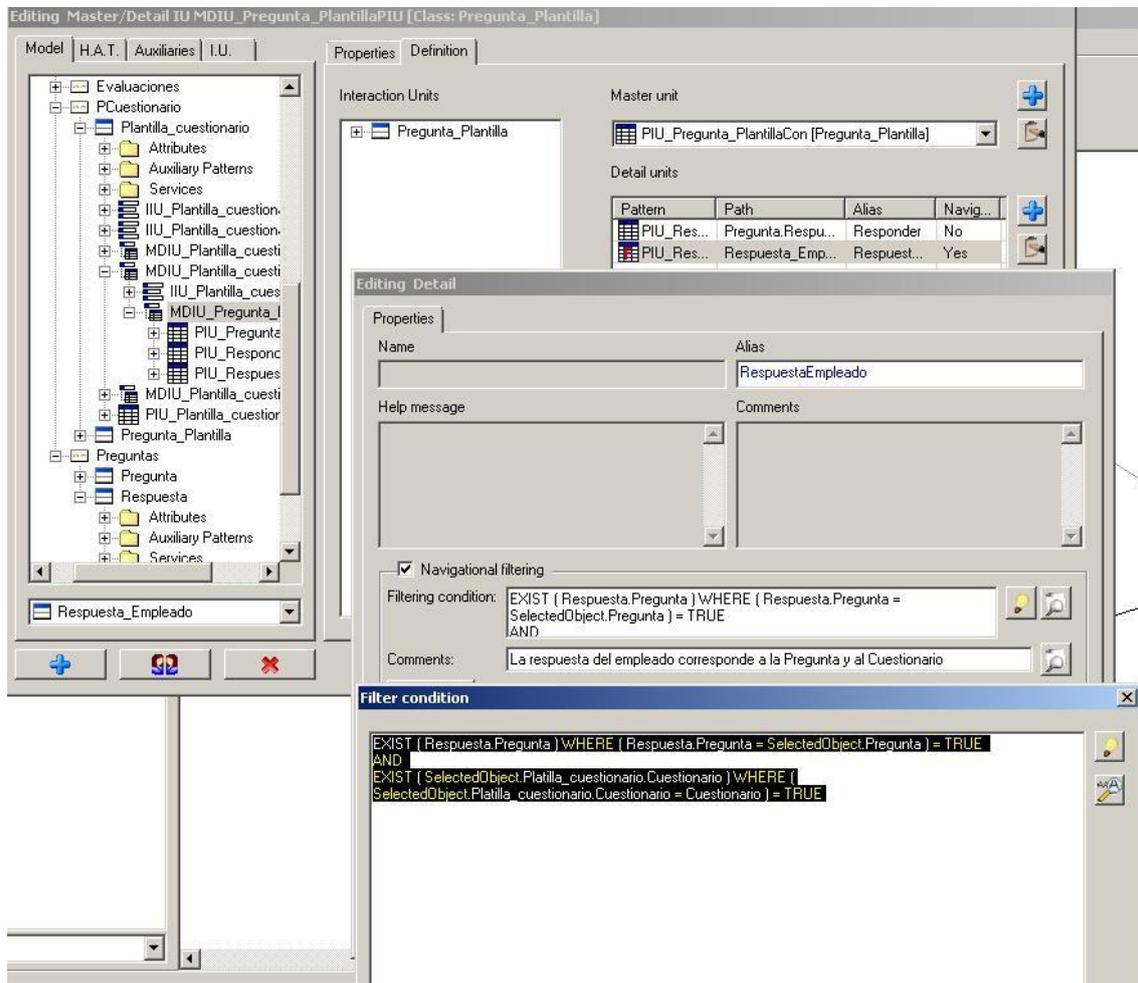


Figura 44. Definición de la MDIU *Pregunta_Plantilla*PIU.

Por si parte, los casos de uso referentes a la Consulta de detalles de Cuestionarios, Evaluación y Evaluaciones de Conocimientos no se modelan mediante servicios en alguna de las clases diseñadas. Sólo tienen representación en el modelo de Presentación. En la siguiente tabla se presenta la correspondencia entre el Caso de Uso y las instancias del Modelo de Presentación que los representan.

Tabla 135. Correspondencias entre Instancias del modelo de presentación y Realización de Evaluaciones

CASO DE USO	INSTANCIAS DEL MODELO DE PRESENTACIÓN
63. Consultar Detalles Cuestionario	MDIU Plantilla_CuestionarioCon: Maestro -> IIU Plantilla_CuestionarioAcc Detalle -> MDIU Pregunta_PlantillaPIUCon: Maestro -> PIU Pregunta_PlantillaCon Detalle -> PIU Respuesta_EmpleadoCon Detalle -> PIU RespuestaCon
64. Consultar Detalles Evaluación	MDIU Valoracion: Maestro -> IIU Valoracion Detalle -> MDIU Valoracion_Conocimiento
65. Consultar Detalles Evaluación de Conocimiento	MDIU Valoracion_ConocimientoCo: Maestro -> PIU Valoracion_Conocimiento Detalle -> PIU PesoCuestionario

Los atributos de Valoracion_Conocimiento y de Cuestionario pueden ser consultados por el AdministradorValoracion, el ResponsableAdministrador de la administración correspondiente, el Empleado que los realiza y el EmpleadoSupervisor que los valida. Además, los tres primeros agentes pueden consultar también los atributos de la clase Valoracion.

Validar Cuestionario

El modelado de los Casos de Uso referentes a la **Validación de Cuestionarios** se realiza mediante las clases Cuestionario y Respuesta_Empleado. En la siguiente tabla se muestran los servicios de la clase Cuestionario implicados en la Validación de Cuestionarios.

Tabla 136. Correspondencias entre clase Cuestionario y Validación de Cuestionarios

CLASE	Cuestionario
SERVICIOS Y SUS	validar_cuestionario: 66. Aprobar Cuestionario
CORRESPONDENCIAS	corregir_cuestionario: 68. Solicitar corrección Cuestionario

El servicio *validar_cuestionario* cambia el estado del cuestionario a “Validado” y asigna valor a las fechas de corrección y de validación. El cambio de estado a “Validado” hace que el atributo *PorcentajeObtenido* obtenga su valor definitivo y se consiga la puntuación de las evaluaciones de conocimiento y las evaluaciones globales.

El servicio *corregir_cuestionario* cambia el estado del cuestionario a “En Respuesta” y asigna valor a las fechas de corrección. Estos dos servicios tienen que cumplir la Precondición de que el estado del cuestionario sea “Finalizado”.

En la siguiente tabla se muestran los servicios de la clase *Respuesta_Empleado* implicados en la Validación de Cuestionarios.

Tabla 137. Correspondencias entre clase *Respuesta_Empleado* y Validación de Cuestionarios

CLASE	<i>Respuesta_Empleado</i>	
SERVICIOS Y SUS CORRESPONDENCIAS	motivo_correccion	67. Comentar Respuesta de Empleado

El servicio *motivo_correccion* permite añadir al empleado supervisor una explicación por la que se sugiere la corrección de una respuesta, siempre que se cumpla la validación de que el cuestionario esté en estado “Finalizado”.

Los 3 servicios de la validación de cuestionario son accesibles desde la *SIU comenzar_revisión*, para el que se ha definido una Navegación Condicional que lleva a la *MDIU Plantilla_cuestionarioAcc*, presentada en la Figura 44.

En el siguiente capítulo se presenta el Espacio de la Solución, donde se cuenta cómo se obtiene el código mediante las herramientas Olivanova, las acciones que hay que realizar para conseguir la aplicación final y se presentan las transformaciones del modelo conceptual al modelo de implementación, la aplicación final.

5. ESPACIO DE LA SOLUCIÓN

En el capítulo 3 se han presentado los requisitos funcionales de los Sistemas de Información de este Proyecto Final de Carrera. En términos de de OO-Method, la Especificación de Requisitos se corresponde con el Modelo Independiente de Computación (CIM), mientras que el Esquema Conceptual desarrollado es el Modelo Independiente de Plataforma (PIM). La transformación entre uno y otro modelo se realizado manualmente, mediante un proceso de análisis.

En una ingeniería de desarrollo de aplicaciones tradicional, partiendo del esquema conceptual ya definido, se crearía manualmente el Modelo Específico de Plataforma (PSM) para posteriormente conseguir de forma también manual el Modelo de Implementación (IM), o aplicación final. En el desarrollo dirigido por modelos, esta transformación se debe realizar de la manera más automática posible. En este capítulo se presenta cómo se realiza esta transformación.

5.1. Transformación de PIM a PSM en OO-Method

OO-Method sugiere que esta transformación se haga de forma sistemática y automática. El proceso por el que se traduce el esquema conceptual a una aplicación se llama Compilación de modelos conceptuales. El esquema conceptual supone un nivel de abstracción superior al código fuente, de igual forma que éste es también un nivel superior al código máquina. De ahí que se utilice el término compilación para nombrar a este proceso.

Los lenguajes de programación convencionales no incorporan detalles de la arquitectura hardware, como la organización física de la memoria o el uso de registros. Es el compilador el que añade estos requisitos cuando se genera el lenguaje máquina. De forma análoga, el compilador de modelos debe añadir ciertas características de la plataforma software destino al código fuente generado, las cuales no se especifican en el esquema conceptual, tales como mecanismos de control de la ejecución o estrategias de accesos a datos.

Los esquemas conceptuales son totalmente independientes de la plataforma software destino. Esto significa que un mismo esquema conceptual puede compilarse para distintas plataformas. La transformación a la aplicación final se basa en la representación software del conjunto de elementos conceptuales. Por tanto, caracterizar un Compilador de Modelos consiste en determinar las correspondencias entre los elementos conceptuales y sus representaciones software de forma precisa, teniendo en cuenta la arquitectura en la que se van a ejecutar [19].

La compilación de un modelo conceptual en una aplicación debe ser un proceso determinista que pueda aplicarse de forma sistemática [32]. Para ello, se deben cumplir los siguientes requisitos:

- La manera en que el esquema conceptual se representa en un entorno de desarrollo cualquiera debe estar determinada, considerando los aspectos estáticos, dinámicos y de presentación.
- Se debe definir la arquitectura de la aplicación resultante.
- Se debe establecer una estrategia de ejecución que garantice la equivalencia funcional entre la especificación y su implementación.

Estrategia de ejecución de aplicaciones

En el proceso de compilación de modelos conceptuales se debe establecer cómo las aplicaciones representarán los elementos definidos en el modelo conceptual. Es decir, se debe asegurar que la semántica de los elementos conceptuales se respeta, para que la aplicación resultante del proceso de transformación (en el espacio de la solución) haga realmente lo diseñado en el esquema (en el espacio del problema).

OO-Method propone definir los elementos de modelado, en lugar de los entornos en que las aplicaciones pueden ejecutarse. El resultado es el *Modelo de Ejecución Abstracto*, que sigue centrando la atención en qué representa y no en cómo se representa. Además, tiene las ventajas de que puede emplearse independientemente de una plataforma destino específica y asegura la equivalencia funcional entre el esquema conceptual y su correspondiente aplicación software obtenida como resultado del proceso.

Arquitectura de las aplicaciones

La estrategia de ejecución de aplicaciones fija un conjunto de mecanismos que especifican lo que debe implementar una aplicación, pero no especifican el cómo. La arquitectura de la aplicación es la que debe proporcionar medios que indican la manera en que se implementa una aplicación. Por tanto, la definición de una u otra arquitectura es el primer escenario diferencial en el proceso de compilación de modelos conceptuales.

La arquitectura de una aplicación debe tener los mecanismos adecuados para representar cada elemento del modelo conceptual, como las clases, los servicios o las relaciones. Estos mecanismos deben interactuar en tiempo real para proporcionar una ejecución de la aplicación equivalente a lo definido en el modelo conceptual. Además de esto, también deben proveer de mecanismos auxiliares como el control de errores, los protocolos de comunicación o el acceso a datos.

OO-Method aprovecha que hay un consenso amplio en el mundo del desarrollo de software para la construcción de aplicaciones en tres capas para definir arquitecturas software

basadas en este esquema. La arquitectura de tres capas consiste en diferenciar una aplicación en los siguientes tres niveles:

1. Capa de presentación, que contiene los componentes para crear la interfaz de usuario.
2. Capa de negocio, en la que están los servicios que implementan la funcionalidad de una aplicación.
3. Capa de persistencia, que tiene los mecanismos para asegurar el almacenamiento de los datos tratados por la aplicación.

La Figura 45 presenta gráficamente la arquitectura de tres capas y la comunicación entre cada una de ellas.

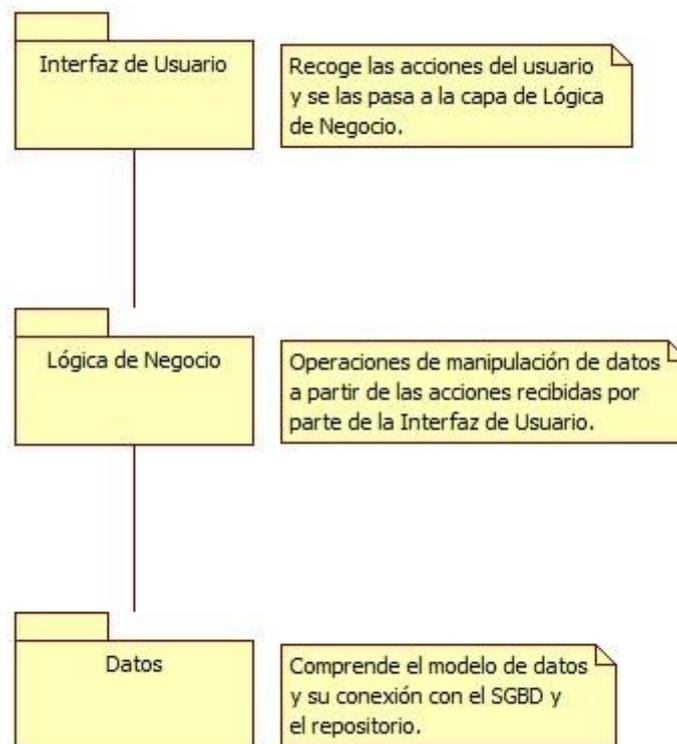


Figura 45. Esquema de la arquitectura de tres capas [33]

La arquitectura de aplicación seleccionada para el proceso de compilación de modelos conceptuales debe proporcionar los mecanismos adecuados tanto para adaptar la funcionalidad de cada uno de los 3 niveles como para asegurar la comunicación entre ellos. Además, la generación de cada uno de los niveles puede hacerse de forma independiente del resto.

Estrategia de transformación

El tercer aspecto que completa el proceso, es la estrategia de transformación, entendida como la forma en que se pasa del Modelo Conceptual al Modelo de Aplicación. Se basa en la definición de:

1. Correspondencias, que establecen relaciones entre los elementos del Modelo Conceptual y el Modelo de Aplicación.
2. Transformaciones, que determinan cómo se obtiene la aplicación a partir de cada elemento del Modelo de Aplicación [32].

El objetivo de las correspondencias es definir cómo los elementos del Modelo Conceptual se pueden crear en el Modelo de Aplicación. Es posible que un elemento conceptual tenga varias posibles traducciones a elementos software, o viceversa. Por tanto, en la definición de las correspondencias, se debe elegir la mejor alternativa posible.

Las transformaciones definen cómo obtener un fragmento de código a partir de un elemento del Modelo de Aplicación. Cada elemento del Modelo de Aplicación debe tener una transformación asociada.

Para definir completamente la estrategia de transformación, pueden ser necesarias varias iteraciones de definición de correspondencias y transformaciones.

Por último, se debe tener en cuenta que debe existir un equilibrio entre la Arquitectura de la Aplicación y las transformaciones, puesto que cuanto mejor definida y más completa sea la primera, más sencillas serán las transformaciones.

OLIVANOVA Transformation Engines

OLIVANOVA Transformation Engines son las implementaciones de diferentes procesos de compilación OO-Method. Cada uno de ellos, implementa un repositorio del Modelo Conceptual, un repositorio del Modelo de Aplicación, una colección de correspondencias y una colección de mapeos [32].

En la Figura 46 se presenta el esquema de un *Transformation Engine*.

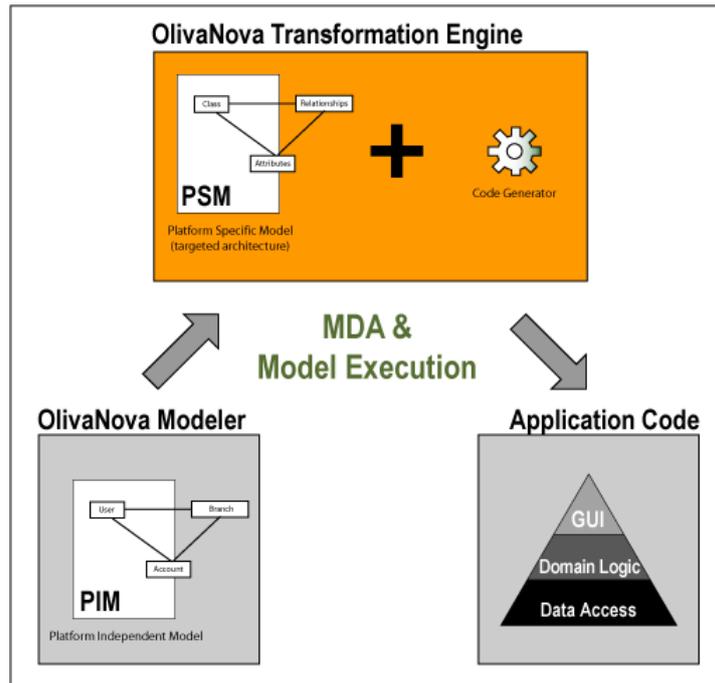


Figura 46. Esquema OLIVANOVA Transformation Engine [34].

El esquema conceptual se carga a partir de un fichero XML generado por *OLIVANOVA Modeler*. Las correspondencias son operaciones que leen datos del repositorio del modelo conceptual y lo asignan al repositorio del modelo de aplicación, junto con los valores correspondientes de las propiedades. El repositorio del modelo de aplicación contiene elementos asociados a cada elemento conceptual y a información necesaria para la estrategia de ejecución de aplicación para la plataforma software.

Las transformaciones leen datos del repositorio del modelo de aplicación y de las propiedades de cada elemento del modelo para generar y ensamblar los fragmentos de código que harán la aplicación final.

5.2 Obtención de código con el sistema Olivanova STAR

Hasta este punto, se ha empleado la herramienta Olivanova Modeler para construir el Modelo Conceptual (PIM) de los sistemas de información que se diseñan en este Proyecto Final de Carrera. La transformación del modelo a código ejecutable se hace de forma automática mediante las Olivanova Transformation Engines. La herramienta Olivanova STAR es la encargada de recibir las peticiones de transformación a código fuente.

El proceso para utilizar el sistema STAR comienza con la validación del esquema conceptual. El validador detecta defectos [35] en el modelo, presentando un informe con los

constructores conceptuales asociados a cada defecto. Una vez corregidos los defectos y con el modelo validado, se genera un fichero XML con la información del modelo conceptual diseñado. Este fichero XML es la entrada del sistema STAR. En la Figura 47, se muestran las propiedades generales de una petición de generación de código. Además de esta información, se deben configurar los perfiles transformación, que contienen la información para que los Olivanova Transformation Engines generen el código fuente.

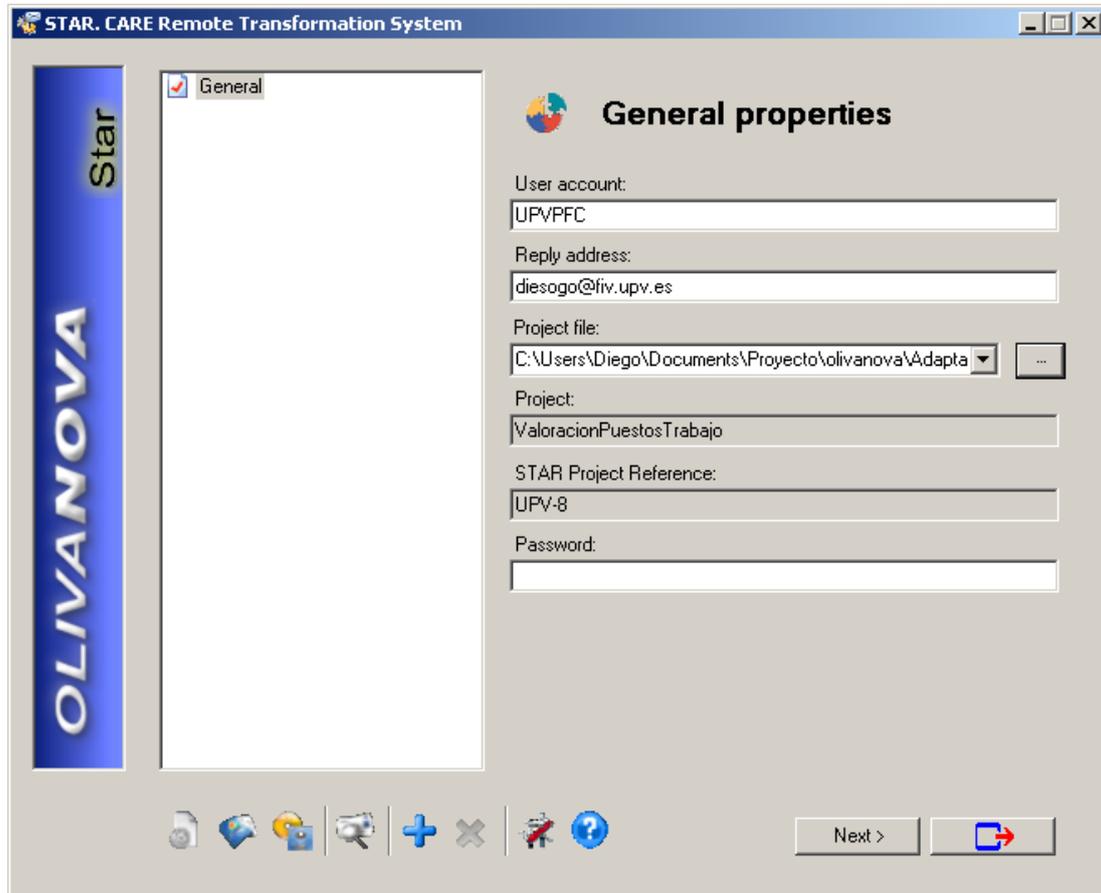


Figura 47. Propiedades generales del Sistema Star.

Los servicios de transformación ofrecidos, clasificados por la capa que implementan son los siguientes [34]:

- A. Para la capa de lógica de negocio:
 - 1. C# .NET 2.0
 - 2. JAVA / EJB2
- B. Para la capa de presentación:
 - 1. ASP .NET 2.0

2. C# / .NET 2.0
3. JSF (Java Server Faces)

Adicionalmente, se puede solicitar también la generación automática de documentación del modelo y el cálculo del tamaño del programa resultante en puntos de función. En la Figura 48, se pueden ver los perfiles seleccionables en el sistema STAR.

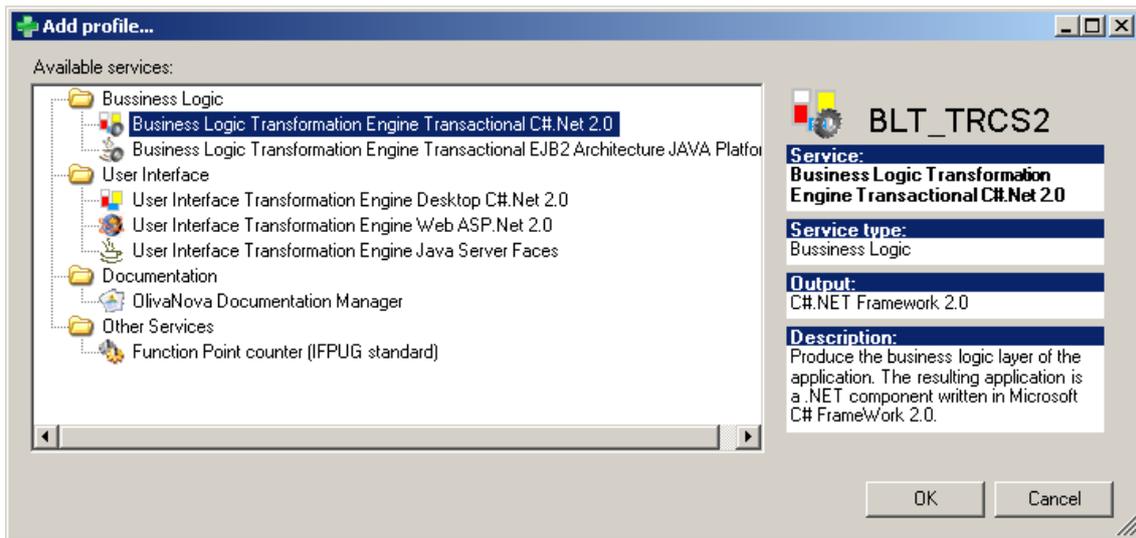


Figura 48. Perfiles disponibles en Olivanova STAR Client.

Para el caso concreto de este PFC, se ha elegido C# .NET 2.0 para la capa de negocio y Desktop C# .NET 2.0 para la capa de presentación. Estos perfiles se deben configurar para solicitar la transformación. En el perfil correspondiente a la capa de negocio, es necesario especificar el nombre de la aplicación, la Base de Datos que se va a utilizar para la capa de persistencia, el tipo de conexión a ella y el nombre de la conexión. El resto de las propiedades están especificadas por defecto por el sistema STAR y pueden ser modificables por el usuario, por ejemplo la versión del transformation engine utilizado para la generación del código. Para la capa de presentación, se debe especificar el nombre de la aplicación que presenta la interfaz y el nombre de la aplicación que implementa la capa de negocio.

La Figura 49 muestra los valores de las opciones de transformación seleccionados para la capa de negocio. Los datos personalizados son el nombre de la aplicación *VPTrabajoSrv*, el del NameSpace *VPTrabajoSrvNS*, el tipo de base de datos *SQLServer 2000* y el tipo de conexión a la base de datos *ODBC* y la cadena de conexión *DSN=VPTDB*. El resto de opciones son las especificadas por defecto.

Las opciones de transformación de la capa de presentación son la vista seleccionada para la compilación del modelo, *V_Desktop*, el nombre de la aplicación cliente *VPTrabajoClient* y su NameSpace *VPTrabajoClientNS* y el nombre de la aplicación servidor a la que se conecta

VPTrabajoSrv y VPTrabajoSrvNS. El resto de propiedades son las que se especifican por defecto. La Figura 50 muestra las propiedades de esta transformación.

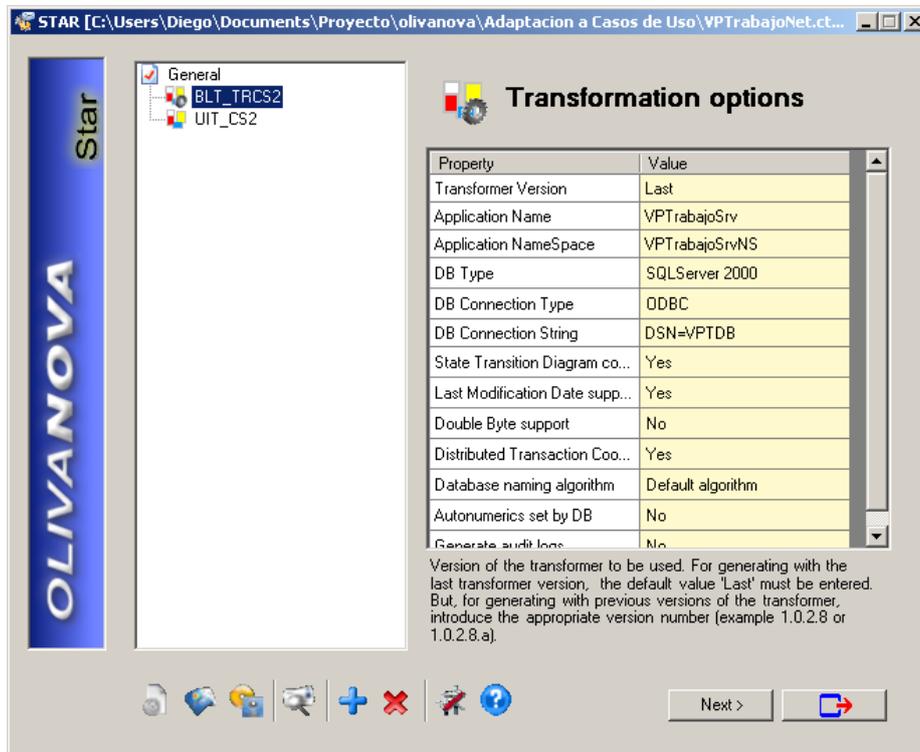


Figura 49. Opciones de transformación de C# .NET 2.0.

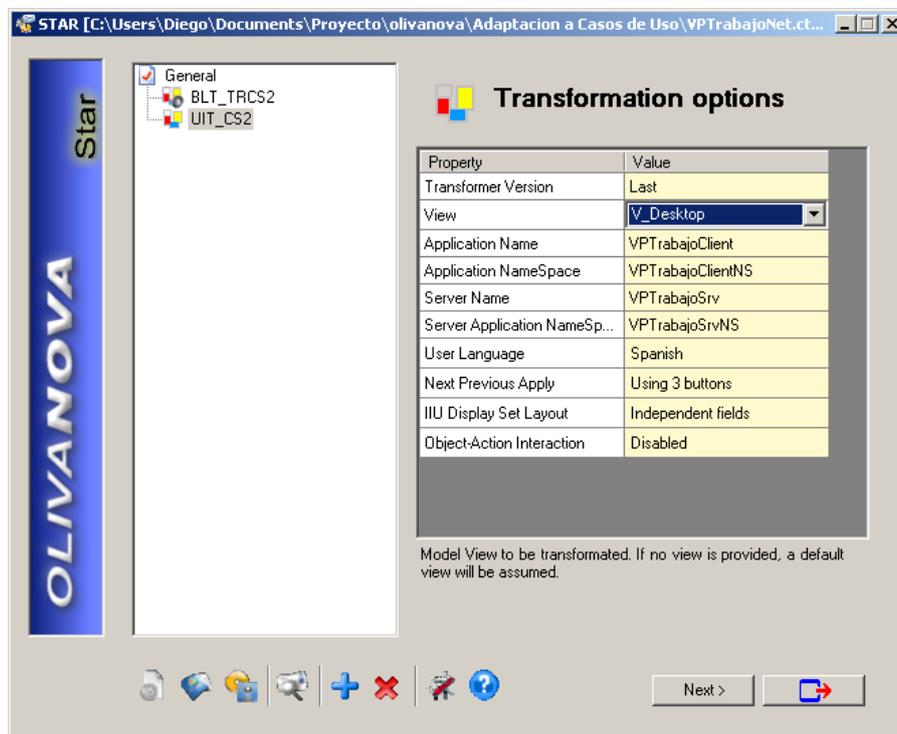


Figura 50. Opciones de transformación de Desktop C# .NET 2.0.

5.3. Preparación de Base de datos

Una vez configurados todos los parámetros necesarios para la compilación del modelo OO-Method, esta configuración junto con el modelo definido se envían a al compilador de modelos Olivanova mediante dos archivos en formato XML. Con esta información el compilador genera el código final de la aplicación en la plataforma seleccionada y lo envía en dos archivos comprimidos, uno por cada perfil especificado (Lógica de Negocio e Interfaz de Usuario).

En el paquete que tiene la implementación de la capa de negocio, se incluye archivos del código fuente estructurado en directorios y el archivo de proyecto de Microsoft Visual C#, que es *VPTrabajoClient.csproj*. Este paquete también contiene los scripts necesarios para implementar la capa de persistencia.

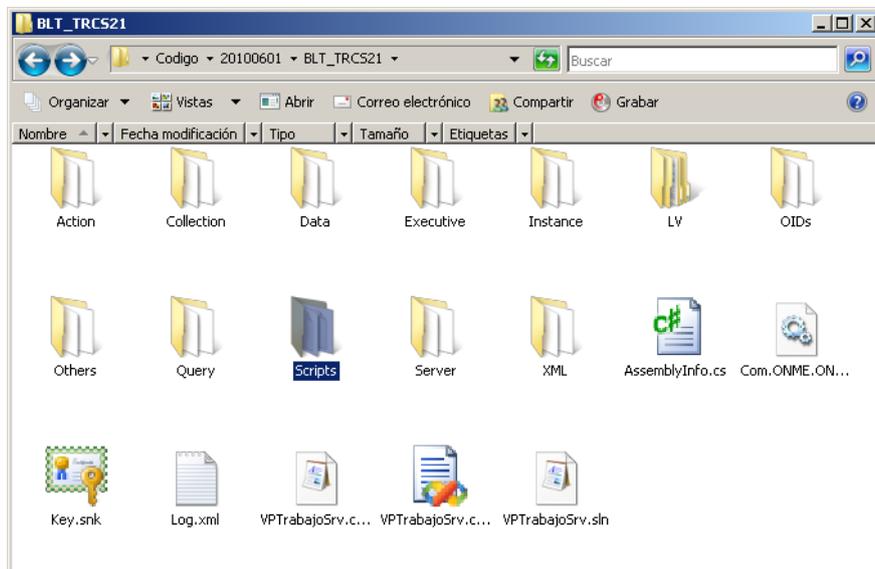


Figura 51. Contenido del paquete de C# .NET 2.0.

El Sistema Gestor de Bases de Datos que se utiliza, tal como se indicó en la configuración del perfil es SQL Server 2000. Por tanto, los Scripts que ha generado el Transformation Engine correspondiente son instrucciones SQL que se deben ejecutar sobre SQL Server.

El primer paso es crear la Base de Datos que se va a utilizar. En este caso concreto se llama "VPTDB". Con la base de datos creada, se pueden ejecutar los scripts desde el SQL Server Management Studio. Con la opción *Consulta de motor de la base de datos*, se copia el contenido del script en el área de comandos, se selecciona la base de datos sobre la que se van a ejecutar y se ejecutan los scripts. Para que la base de datos esté completa, se deben ejecutar los 5 scripts de creación en el siguiente orden: "CreateTable.sql" para crear las tablas,

“PrimaryKeys.sql” para las claves primarias, “ForeignKeys.sql” para las claves ajenas, “Indexes.sql” para los índices y “CreateAgents.sql” para los agentes.

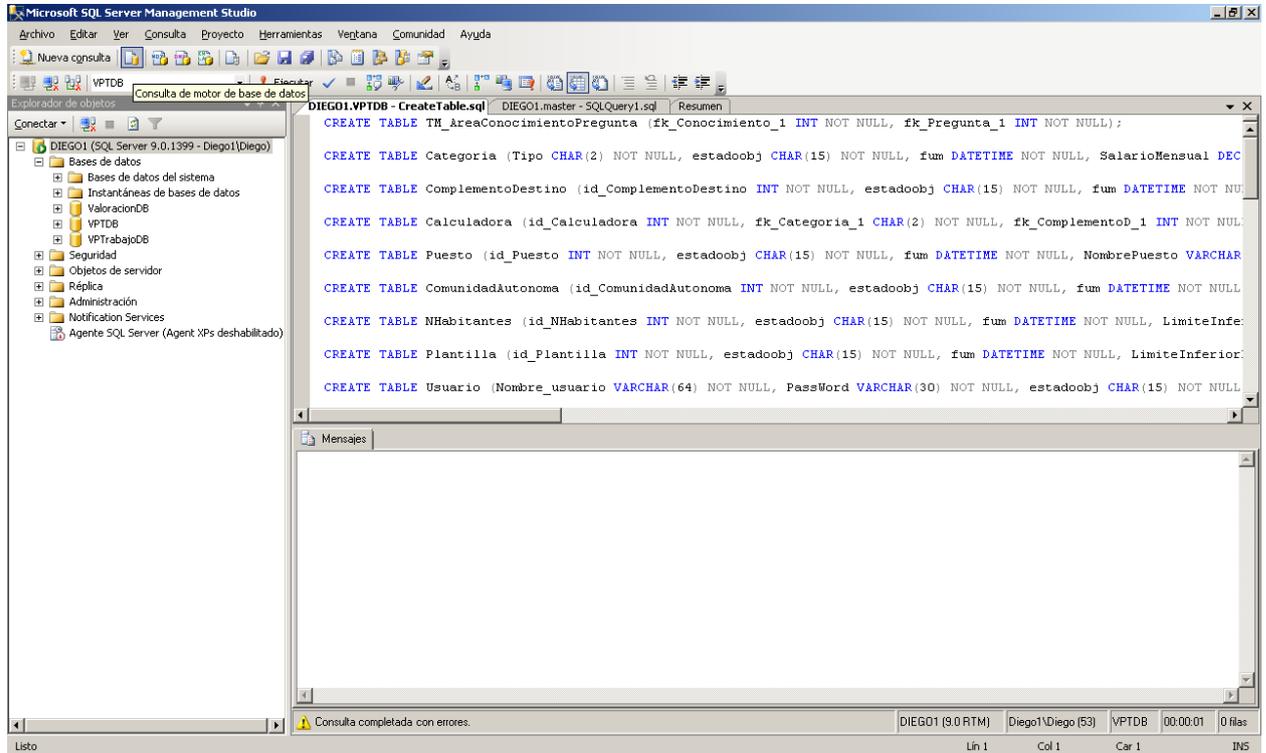


Figura 52. Ejecución de script de creación de tablas

Configuración ODBC

Una vez creada la base de datos, se debe configurar la conexión entre ésta y la aplicación final. Esto se consigue mediante ODBC. ODBC son las siglas de Open DataBase Connectivity, que es un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation. El objetivo de ODBC es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos (DBMS por sus siglas en inglés) almacene los datos [32].

Para configurar un ODBC, se debe acceder al Administrador de orígenes de datos ODBC de Windows. Se llega a él a través de *Inicio -> Panel de Control -> Herramientas Administrativas -> Orígenes de datos ODBC*. En este administrador se debe seleccionar la pestaña de DSN de Usuario y la opción *Agregar*. Dentro de esta opción, se especifica el nombre del ODBC, del servidor de bases de datos y de la base de datos predeterminada. Para el caso de este PFC, el nombre del ODBC es “VPTDB”, como se especificó en el perfil seleccionado para la capa de persistencia (Figura 49), y la base de datos predeterminada es la recién creada “VPT”.

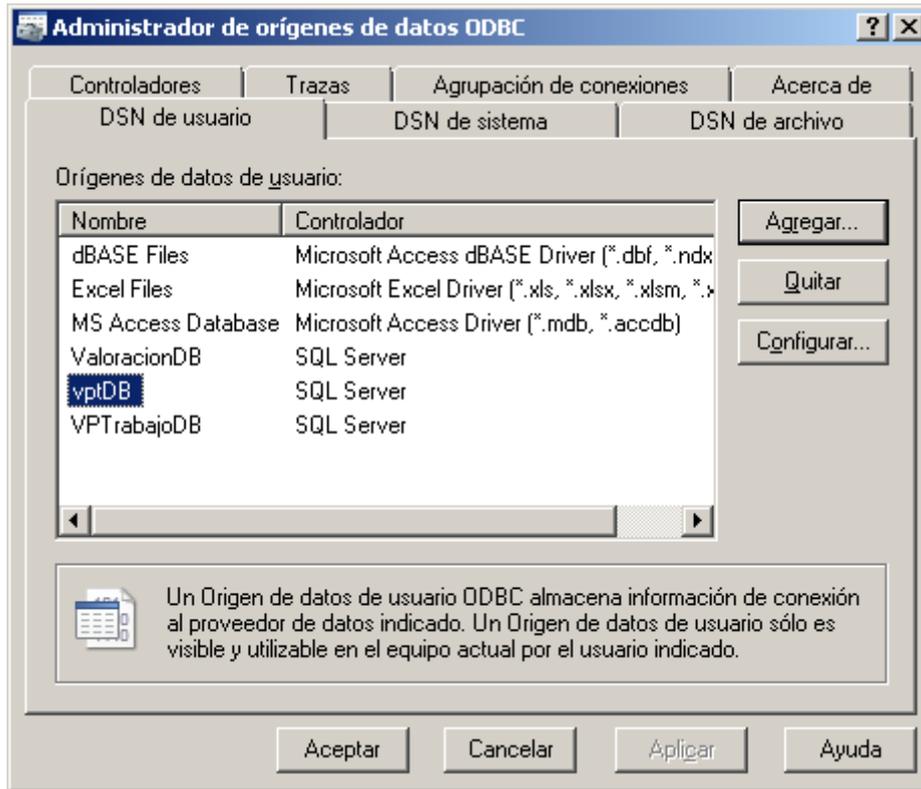


Figura 53. Administrador de orígenes ODBC con vptDB.

5.4. Compilación del código y puesta en marcha

Con la base de datos ya preparada y la conexión ODBC configurada, es el momento de centrarse en la compilación del código. En términos de la arquitectura de tres capas, ya se tiene la capa de persistencia y el mecanismo de comunicación entre ésta y el nivel de la lógica de negocio. Por tanto, falta por conseguir el software final de las capas de negocio y de presentación, así como la intercomunicación entre ellas.

El proceso de compilación del código recibido se realiza con Microsoft Visual Studio 2005. A continuación se detallan los pasos llevados a cabo para lograr el funcionamiento de la aplicación:

- 1.- Se abre el proyecto que contiene el código de la lógica de negocio. Para ello, se ejecuta la opción *Archivo -> Abrir -> Proyecto o solución* y se selecciona el elemento "VPTrabajoSrv.csproj". Después de este paso, se ha creado en el Explorador de soluciones una solución llamada "VPT" que contiene al proyecto seleccionado.

2.- En el menú contextual de la solución, se ejecuta la acción *Agregar -> Proyecto existente* y se selecciona el elemento “VPTrabajoClient.csproj”, que es el proyecto de la capa de presentación. Esto añade a la solución, el proyecto “VPTrabajoClient”.

3.- Se deben configurar las dependencias entre ambos proyectos y el orden de generación. En el menú contextual de la solución, se selecciona la acción *Dependencias del Proyecto* y se marca que el “VPTrabajoClient” depende de “VPTrabajoSrv”, lo que indica que la compilación del primero depende de la del segundo. El orden de generación será, por tanto, la compilación de “VPTrabajoSrv” y después la de “VPTrabajoClient”.

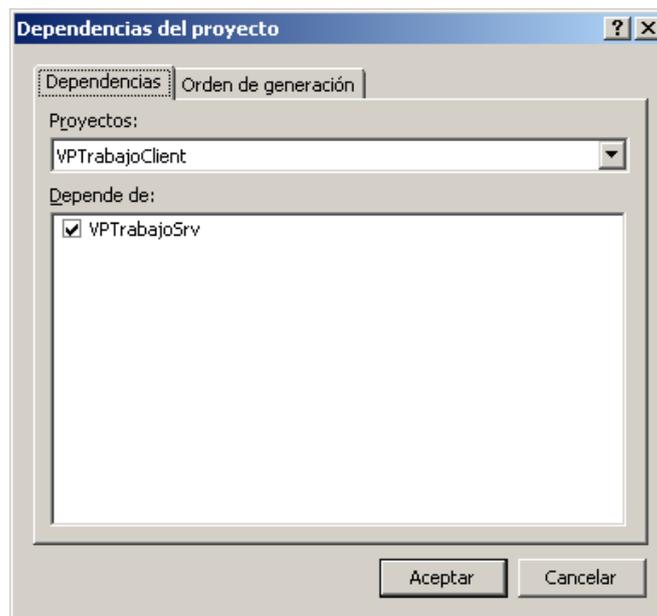


Figura 54. Dependencias del proyecto.

4.- Antes de lanzar la compilación, se debe agregar el proyecto de la capa de negocio como referencia del proyecto de la capa de presentación y establecer este último como proyecto de inicio. En el menú contextual de “VPTrabajoClient” se selecciona la acción *Agregar Referencia*; dentro de este menú se selecciona la pestaña proyectos y se marca “VPTrabajoSrv”. Por último, en el mismo menú contextual, se selecciona la opción *Establecer como proyecto de inicio*. Al concluir estos pasos, el *Explorador de soluciones* queda como en la Figura 56.

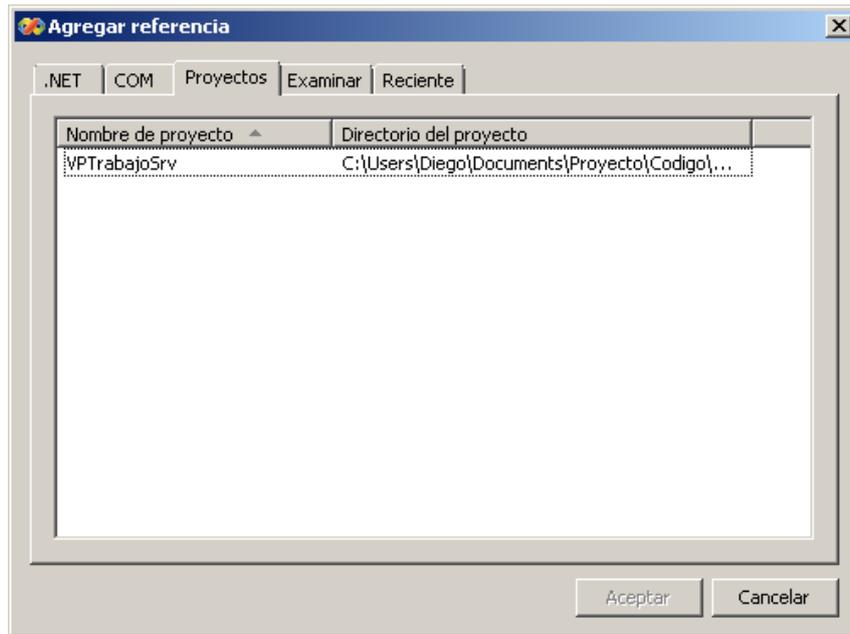


Figura 55. Dependencias del proyecto.

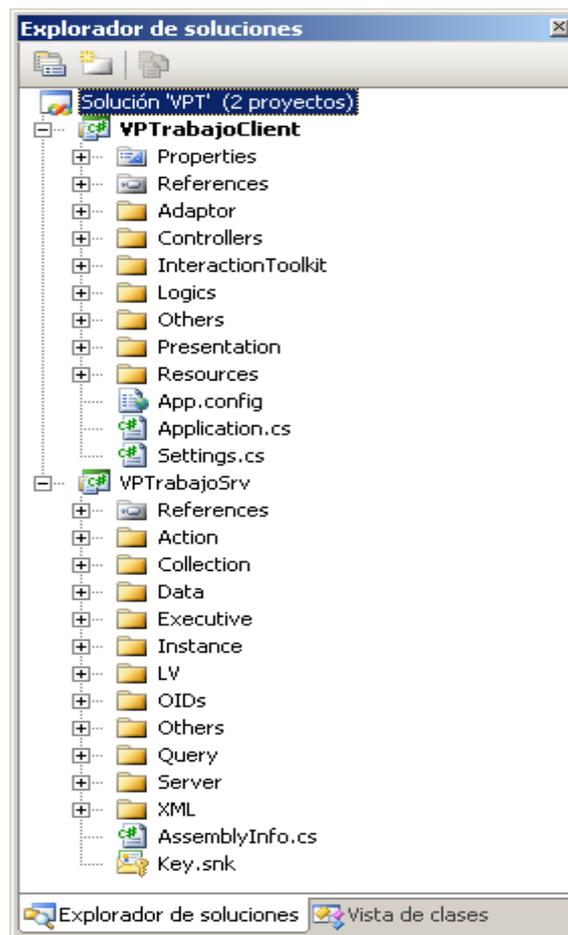


Figura 56. Solución VPT.

5.- En el momento de la compilación, se debe configurar la plataforma destino y el resultado final, que puede ser “Release” para obtener el código final o “Debug” para corregir errores. Para ello se selecciona la acción *Generar -> Administrador de Configuración*, donde marcamos el proyecto como “Release” y a ejecutar en “Any Platform. Hecho esto, se ejecuta *Generar -> Generar solución*.”

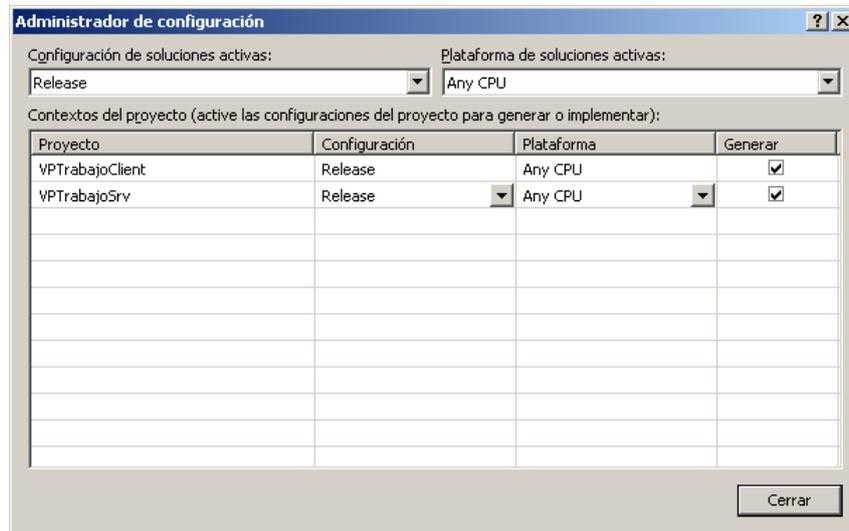


Figura 57. Configuración de la compilación para plataforma .Net.

Si no ha habido ningún error de compilación ni de generación de la aplicación, se obtiene como resultado un archivo ejecutable con la aplicación en el directorio *bin\Release* del paquete que contiene la capa de presentación con el nombre *VPTrabajoClient.exe*. En el directorio *bin\Release* del paquete con la lógica de negocio se ha creado una librería *VPTrabajoSrv.dll*. Además se ha creado un componente COM+ con la aplicación correspondiente a la lógica de negocio.

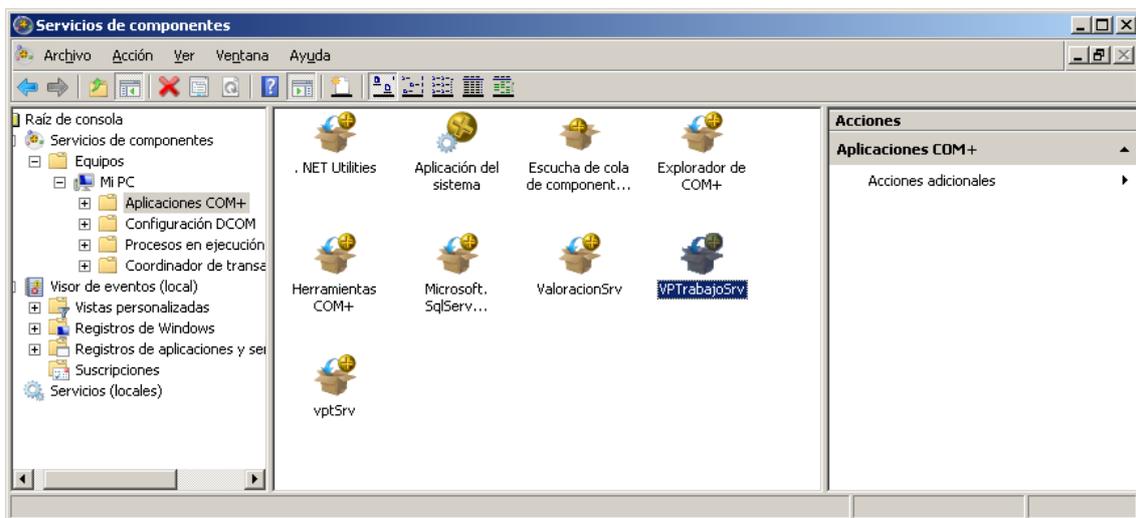


Figura 58. Explorador de componentes COM+ con “VPTrabajoSrv”.

5.5. El modelo de Implementación

Una vez generada la aplicación, se ejecuta el archivo *VPTrabajoClient.exe*. Para entrar en la aplicación, es necesario registrarse como uno de los usuarios agentes.



Figura 59. Ventana de identificación de la aplicación.

En el modelo de presentación del esquema conceptual se diseña el siguiente Árbol de Jerarquía de Acciones (HAT, por sus siglas en inglés) para la vista *V_Desktop* que se solicitó en el sistema STAR para la compilación del modelo (Figura 50). La especificación de este HAT se muestra en la Figura 60 y el resultado de su compilación se puede ver en las Figura 61 y Figura 62.

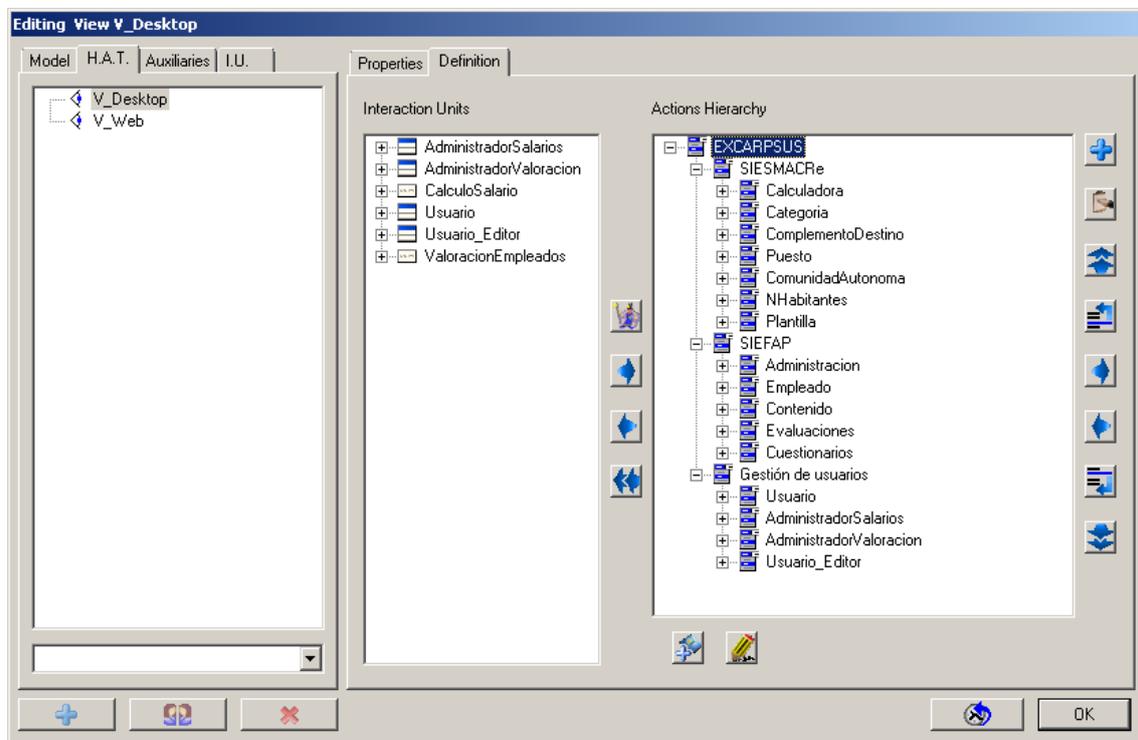


Figura 60. Árbol de Jerarquía de Acciones (HAT).

Cada agente tiene una vista distinta del menú de jerarquía de acciones según la visibilidad que tiene sobre las clases y sus atributos y servicios. El menú de jerarquía de acciones del modelo de Implementación para el agente *AdministradorSalarios* es el siguiente:

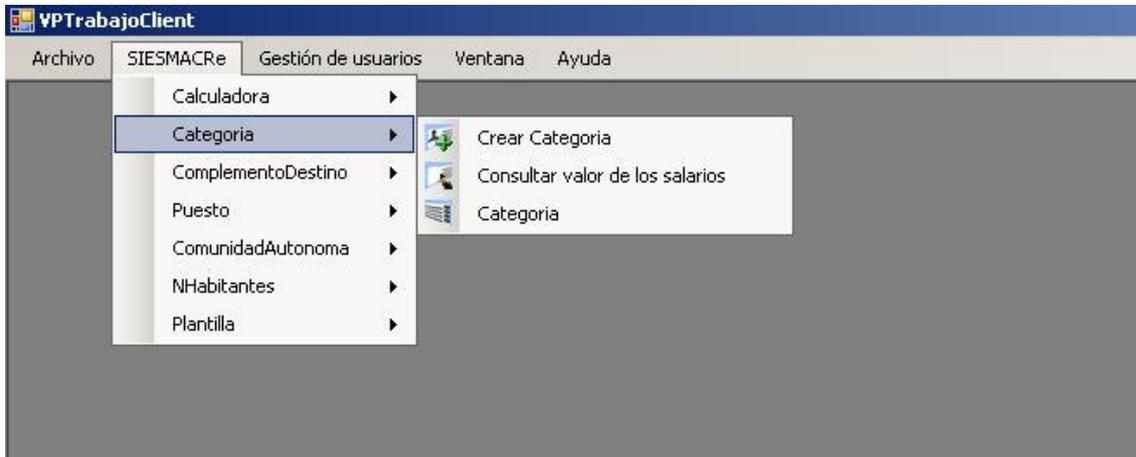


Figura 61. Menú inicial para el agente *AdministradorSalarios*.

La Figura 62 muestra el menú de jerarquía de acciones del modelo de Implementación para el agente *AdministradorValoracion*.

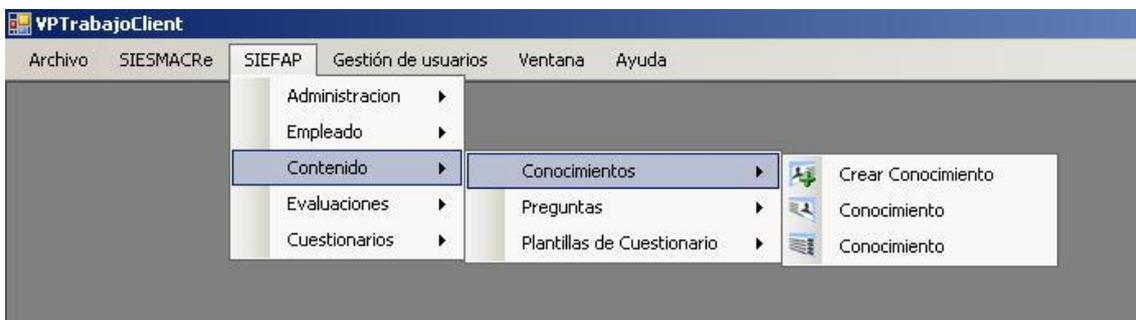


Figura 62. Menú inicial para el agente *AdministradorValoracion*.

Para demostrar que la transformación ha creado una aplicación utilizable y equivalente al diseño del esquema conceptua, en este PFC se explican algunas acciones de la lógica de negocio de la aplicación, que muestran el funcionamiento de todos los tipos de elementos conceptuales empleados. Son las siguientes:

A. Con el agente *AdministradorValoracion*

1. Se da de alta un Empleado y se muestra su información.
2. Se realiza la navegación de este Empleado a su Departamento.
3. Desde el Departamento se navega a población de sus empleados.

B. Con el agente *Usuario_Editor*

1. Se crea una Pregunta y se asocia a un Conocimiento.
2. Se muestra la pregunta recién creada, junto con sus respuestas.

C. Con el agente *ResponsableAdministracion*

1. Se crea una Evaluación global para el Empleado creado.
2. Se crea una Evaluación de Conocimiento para esta Evaluación global.
3. Se asocia un Cuestionario a esta evaluación de conocimiento y se cambia a estado "Preparado".

D. Con el agente *Empleado*

1. Se responde al Cuestionario de la Evaluación de conocimiento.

E. Con el agente *EmpleadoSupervisor*

1. Se valida el Cuestionario "Finalizado" por el empleado y se cambia a estado "Validado".

F. Con el agente *Empleado*

1. Se consultan los detalles de la evaluación, la evaluación del conocimiento y el cuestionario.

A.1. Alta de Empleado

El alta de empleado se realiza a través de la Unidad de Interacción de Servicio (SIU, por sus siglas en inglés) *create_instance* de la clase *Empleado*. La Figura 63 muestra la especificación de este servicio en la clase *Empleado*. A partir de esta especificación, el compilador de modelos genera una aplicación totalmente funcional.

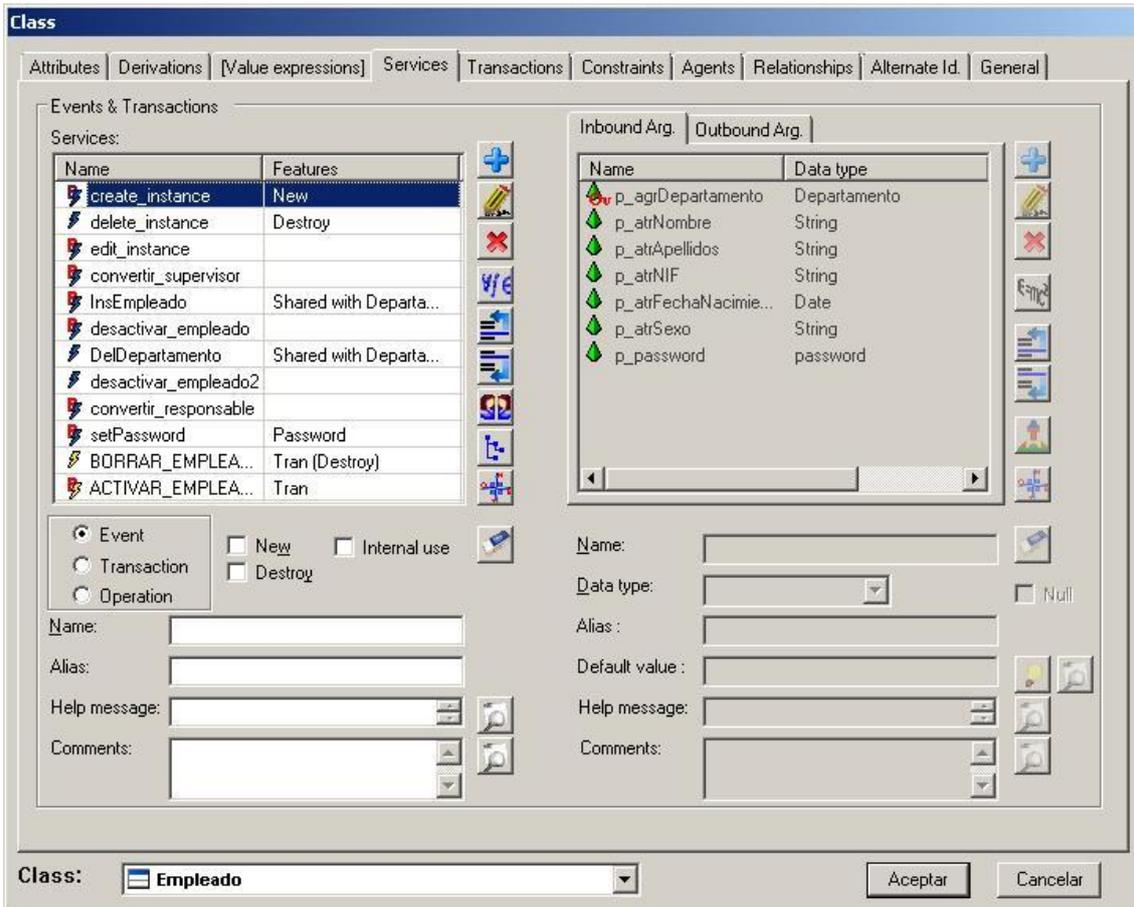


Figura 63. Argumentos de entrada del servicio *create_instance* de la clase *Empleado*.

La Figura 64 muestra la entrada de este servicio. Los datos obligatorios para su ejecución se muestran sobre fondo amarillo, mientras que los que no son obligatorios se muestran sobre fondo blanco.

Para el argumento departamento se diseña una Carga de Población (Population Preload en inglés) en el esquema conceptual, que se representa en la aplicación como un desplegable.



Figura 64. Entrada del servicio de creación de empleado.

La Figura 65 es la salida de la ejecución del servicio. La especificación de este mensaje en el modelo conceptual se puede visualizar en la Figura 66. De esta manera se puede observar que el nombre del empleado, los apellidos y el DNI se obtienen de la base de datos en tiempo de ejecución del servicio.

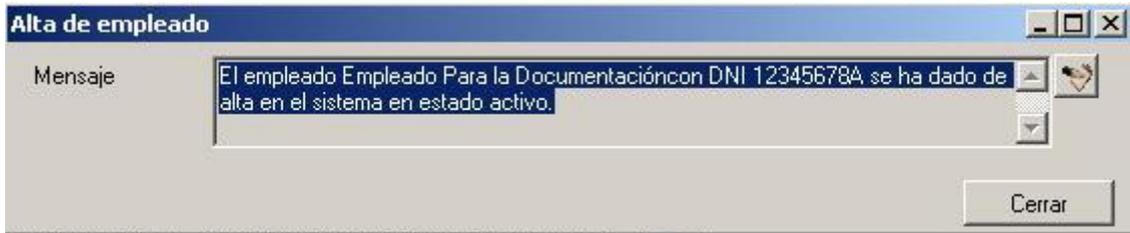


Figura 65. Salida del servicio de creación de empleado.

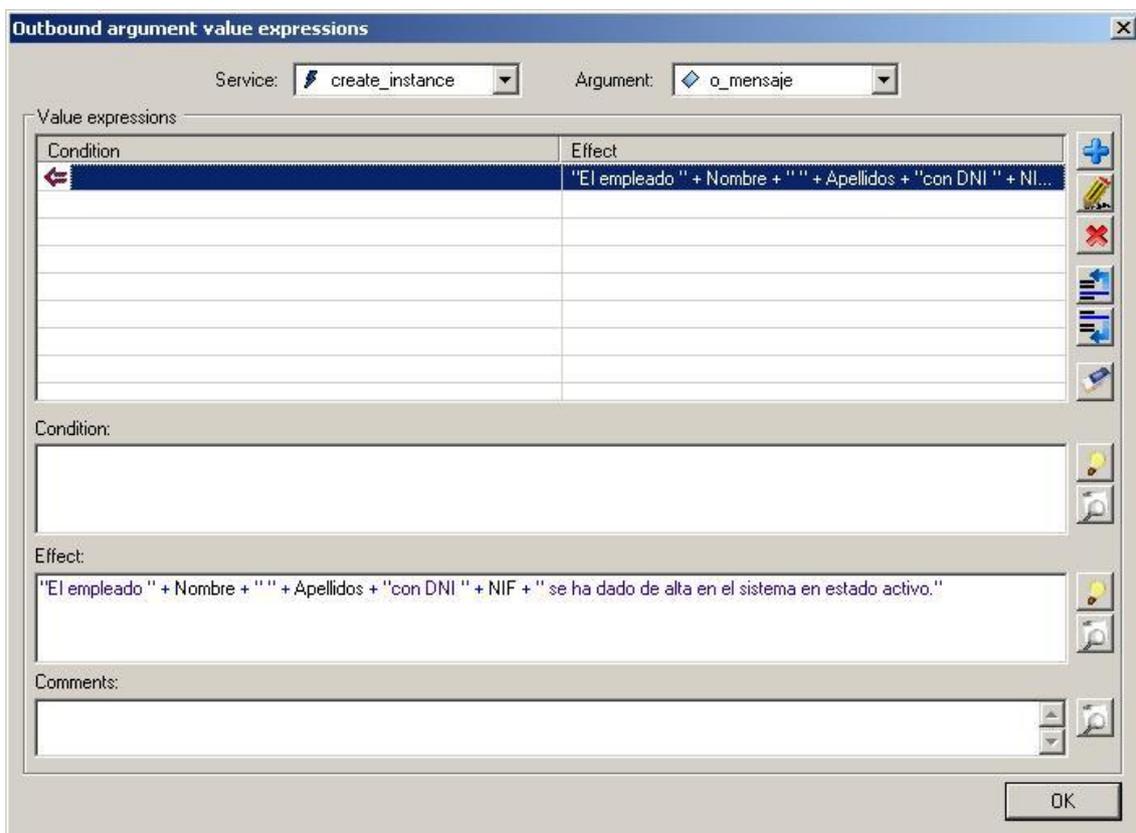


Figura 66. Argumento de salida del servicio *create_instance* de la clase *Empleado*.

El empleado recién creado se muestra en la Unidad de Interacción de Instacia (IIU, por sus siglas en inglés) *Empleado*. La *IIU Empleado* está formada por el Conjunto de atributos mostrados (Display Set - DS, por sus siglas en inglés) *Empleado*, el Patrón de acciones (Action Pattern - AP por sus siglas en inglés) *Empleado* y el Patrón de navegación (Navigation Pattern - NP por sus siglas en inglés) *Empleado*. Las Figura 67, Figura 68 y Figura 69 muestran la definición de los patrones del *IIU Empleado*. El DS_Empleado muestra atributos de la clase *Empleado* y atributos de las clases que tiene relacionadas. El patrón de acciones *Empleado*

muestra unidades de interacción de servicio (SIUs) para los servicios definidos en la clase Empleado. El patrón de navegación muestra las navegaciones a las clases relacionadas a la clase Empleado: Departamento, Valoracion y Cuestionario.

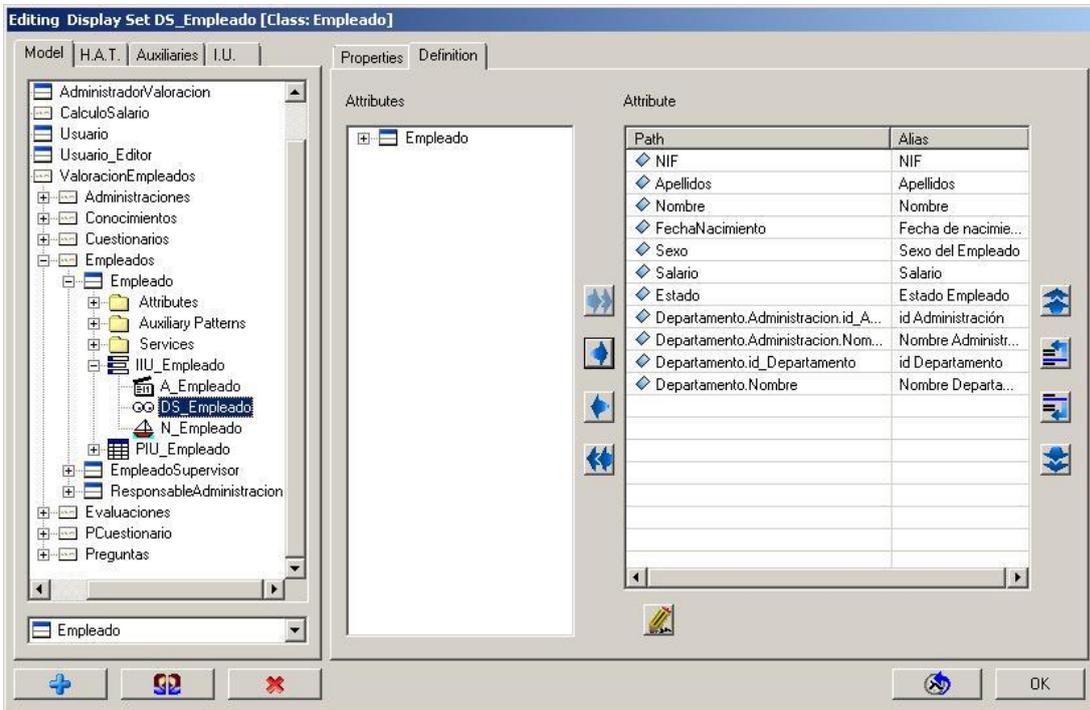


Figura 67. Display Set Empleado.

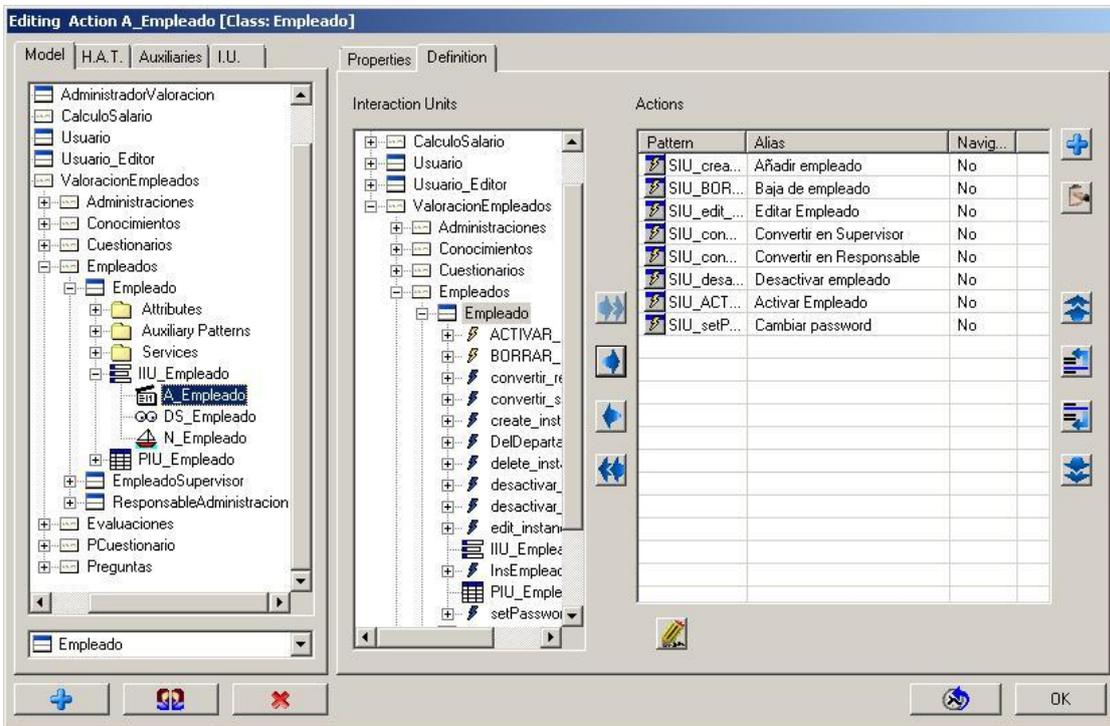


Figura 68. Action Pattern Empleado.

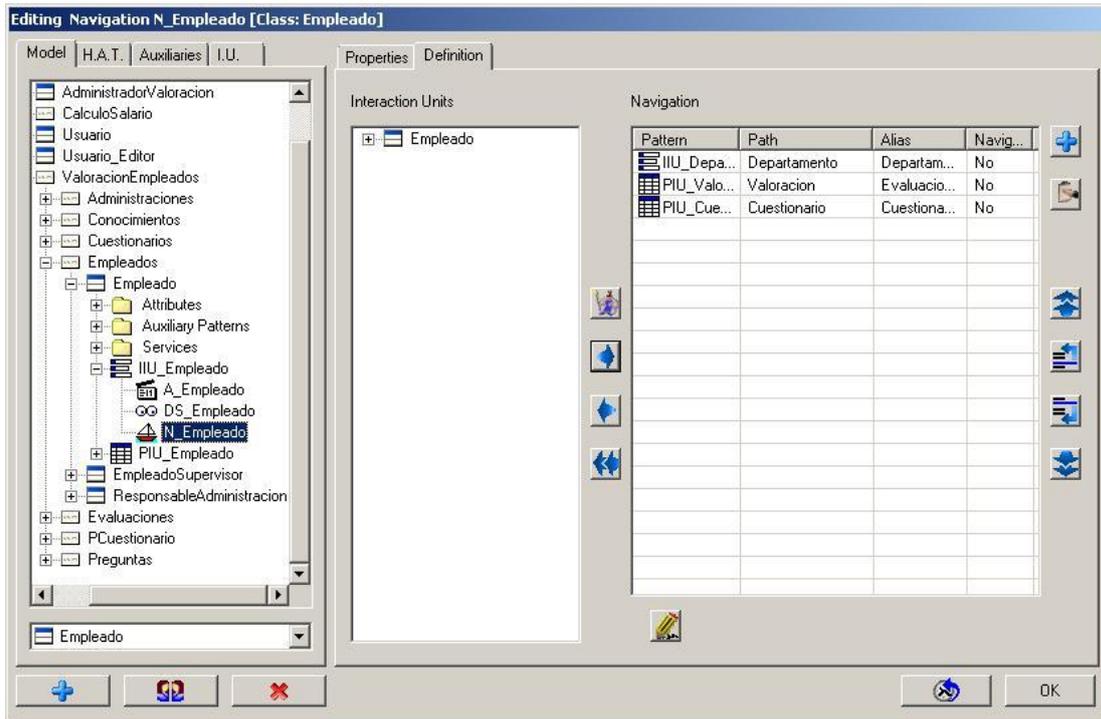


Figura 69. Navigational Pattern Empleado.

El resultado de la transformación de de la *IIU Empleado* del esquema conceptual al modelo de aplicación se muestra en la Figura 70. En la parte central, se muestran los atributos del empleado, mientras que en la parte superior está el atributo identificador. La columna derecha se reserva para los servicios disponibles mediante el patrón de acciones definido y la parte inferior muestra las opciones de navegación a otras Unidades de Interacción.

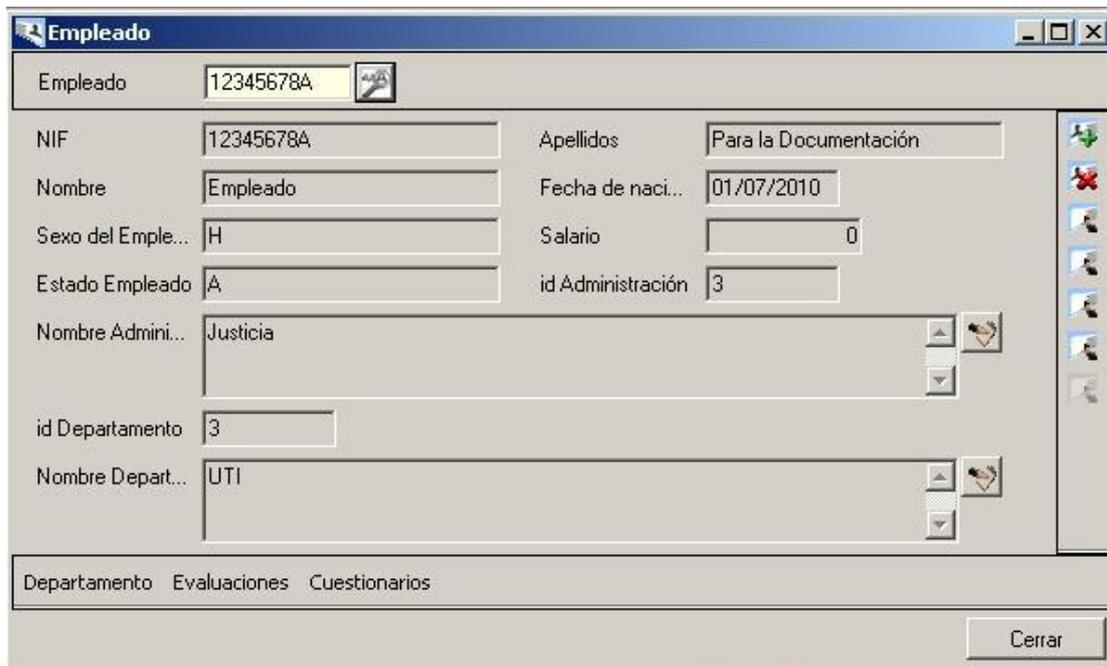
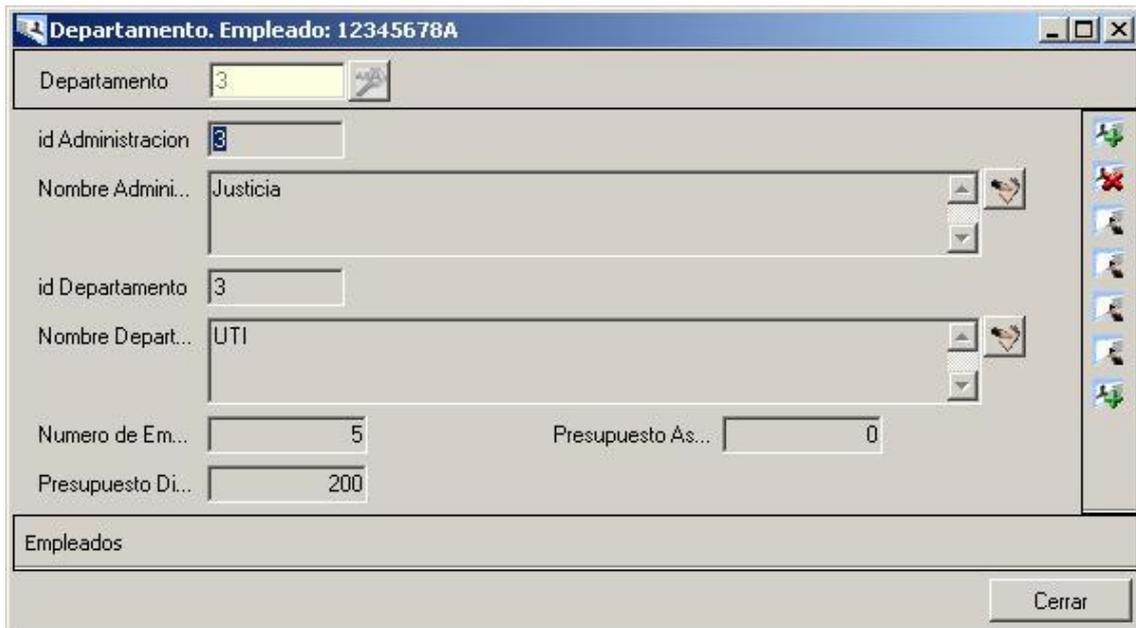


Figura 70. IIU Empleado.

A.2. Departamento del empleado

Al igual que en la definición de la *IIU Empleado*, para la *IIU Departamento* se ha especificado un conjunto de visualización, un patrón de acciones y un patrón de navegaciones en la vista de presentación del esquema conceptual del sistema. A partir de esta especificación se ha generado automáticamente la *IIU Departamento*. La Figura 71 muestra la *IIU Departamento*, a la que se llega desde la navegación del empleado de la *IIU Empleado*. Se puede observar en el título de la ventana que se muestra el identificador del empleado desde el que se ha llegado a esta IIU.



The screenshot shows a window titled "Departamento. Empleado: 12345678A". The window contains several input fields and buttons. The fields are: "Departamento" (value: 3), "id Administracion" (value: 3), "Nombre Admini..." (value: Justicia), "id Departamento" (value: 3), "Nombre Depart..." (value: UTI), "Numero de Em..." (value: 5), "Presupuesto As..." (value: 0), and "Presupuesto Di..." (value: 200). There is a "Cerrar" button at the bottom right. The window also has a vertical toolbar on the right side with several icons.

Figura 71. IIU Departamento.

A.3. Empleados del departamento

La Figura 72 es la Unidad de Interacción de Población (PIU, por sus siglas en inglés) *Empleado*, a la que se llega desde la navegación del departamento de la *IIU Departamento*. Se puede observar en el título de la ventana que se muestra el identificador del departamento desde el que se ha llegado a esta PIU.

En la cabecera de la *PIU Empleado* se pueden observar seis carpetas que permiten filtrar la información a presentar mediante el uso de filtros definidos en la PIU. En la primera carpeta se observa que también aparece un desplegable con criterios de ordenación. Para obtener esta unidad de interacción, ha sido necesario especificar en el esquema conceptual los conjuntos de visualización, los filtros, los criterios de ordenación de las instancias desplegadas, las acciones y las navegaciones. La Figura 73 muestra la definición del PIU en el esquema conceptual, con el *Display Set Empleado*, seis Filtros y diez Criterios de ordenación.

NIF	Apellidos	Nombre	Fecha de nacimiento	Sexo	Salario	Estado	id	Nombre Administración	id	Nombre Departamento
12345678A	Para la Documentación	Empleado	01/07/2010	H		0 A	3	Justicia	3	UTI
2	Cruz Cruz	Jorge	25/07/1982	H	200	A	3	Justicia	3	UTI
5	Ruiz Ruiz	Laura	01/02/1970	M		0 I	3	Justicia	3	UTI
6	Andrade	Juana	24/07/1988	M		0 A	3	Justicia	3	UTI
E4	Cuatro	Empleado	01/04/2010	V		0 A	3	Justicia	3	UTI
E6	Seis	Empleado	05/05/2010	V		0 A	3	Justicia	3	UTI

Figura 72. PIU Empleado.

Figura 73. Definición de *PIU Empleado* en el esquema conceptual.

A modo de ejemplo, la Figura 74 presenta la definición del filtro de empleado por *Nombre*. En esta figura se puede observar que la variable de filtro es un String que será ingresado por el usuario del sistema en tiempo de ejecución para restringir las instancias mostradas en la *PIU Empleado* de acuerdo al nombre de los empleados que estén activos.

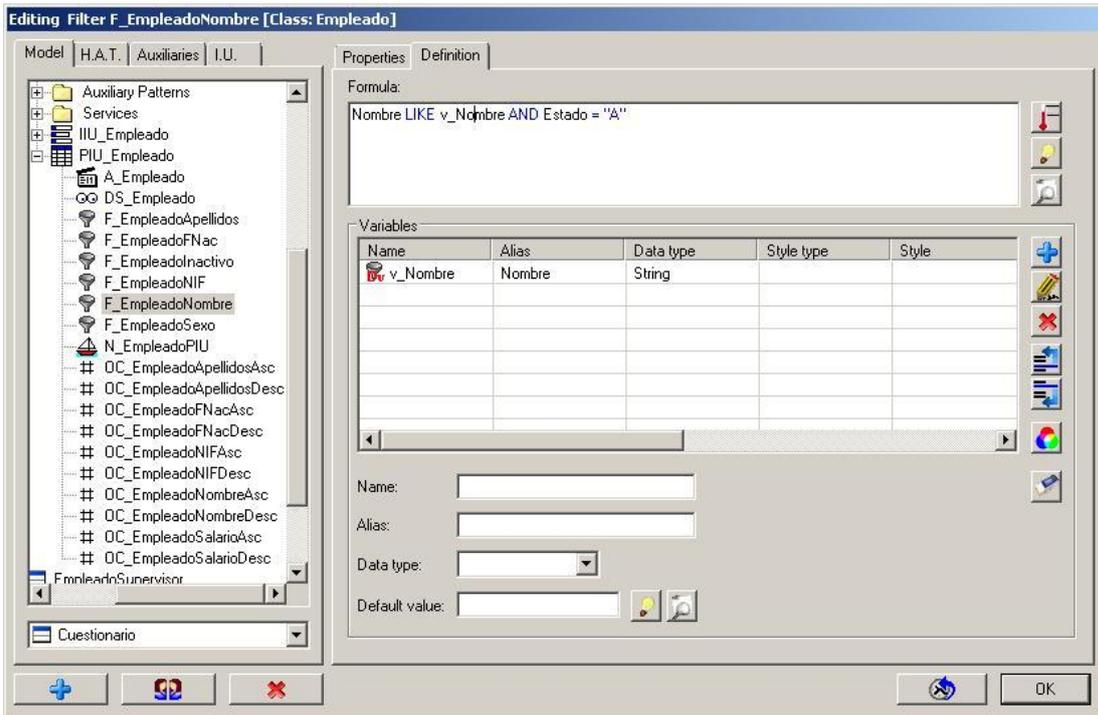


Figura 74. Definición del Filtro *Nombre* de la clase *Empleado* en el esquema conceptual.

La Figura 75 presenta la especificación de un criterio de ordenación, concretamente del *OC EmpleadoFNacAsc*, que permite ordenar a los empleados por fecha de nacimiento de forma ascendente.

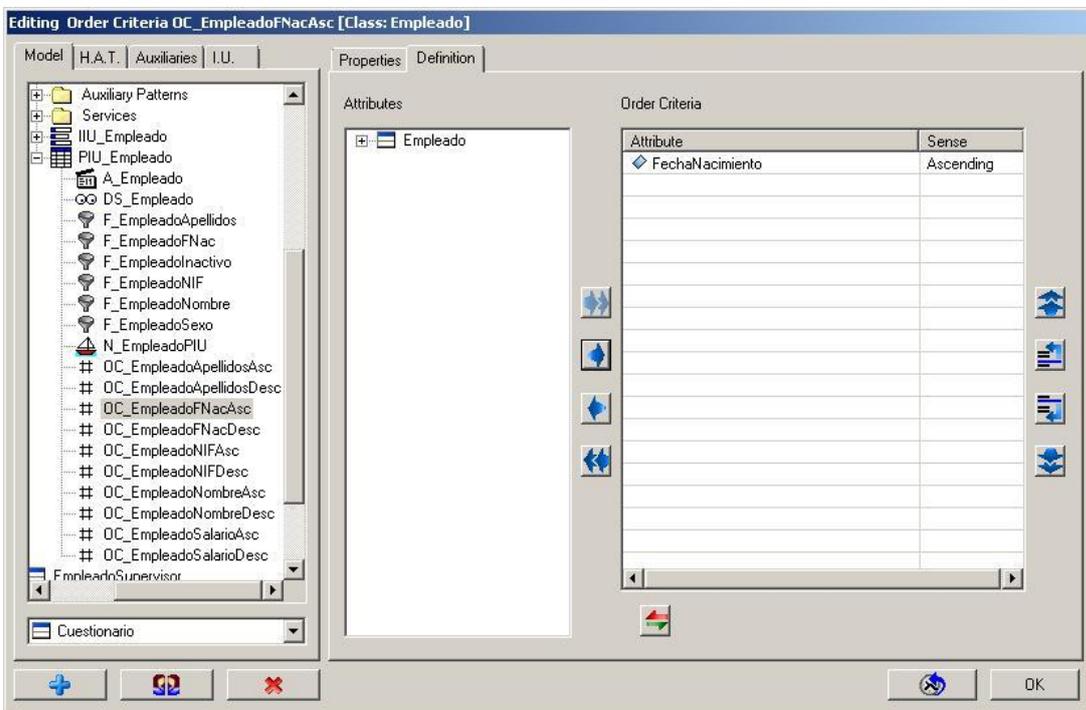


Figura 75. Definición de *OC EmpleadoFNacAsc* de la clase *Empleado* en el esquema conceptual.

B.1. Creación de Pregunta

Con el agente Usuario_Editor se crea una pregunta a través de la SIU *CREA_PREGUNTA* de la clase Pregunta. En esta SIU es obligatorio rellenar los argumentos de la pregunta y de al menos una respuesta, aunque se pueden informar hasta 5.

El servicio *CREA_PREGUNTA* se diseña en el modelo conceptual con argumentos que pueden tomar valor nulo, como muestra la Figura 76.

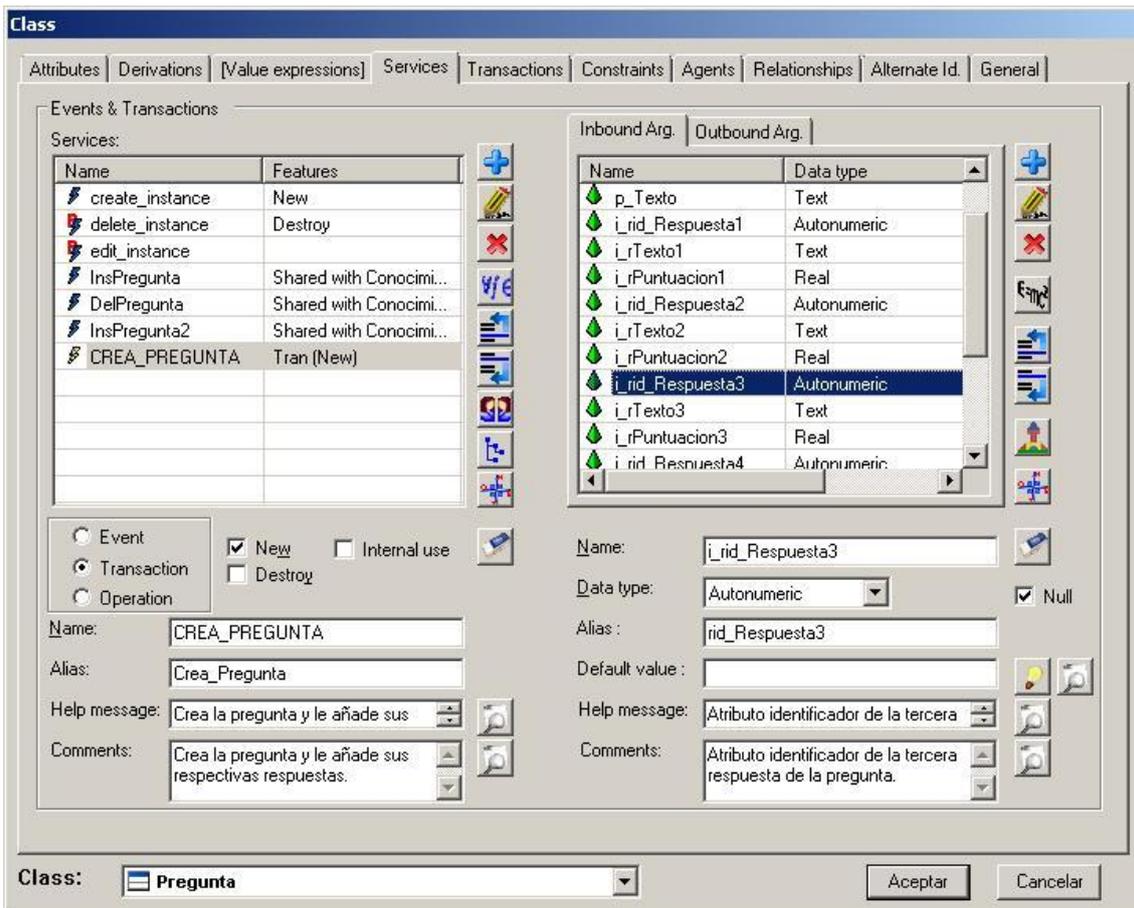


Figura 76. Atributo de entrada *i_rid_Respuesta3* del servicio *CREA_PREGUNTA* con posible valor nulo.

Se definen además reglas del tipo Evento-Condición-Acción (Event-Condition-Action en inglés) que obliga a que si se rellena el texto de una Respuesta, se debe rellenar también su puntuación, como se observa en la Figura 77 mediante la clausula *SetMandatory* especificada utilizando el lenguaje formal OASIS.

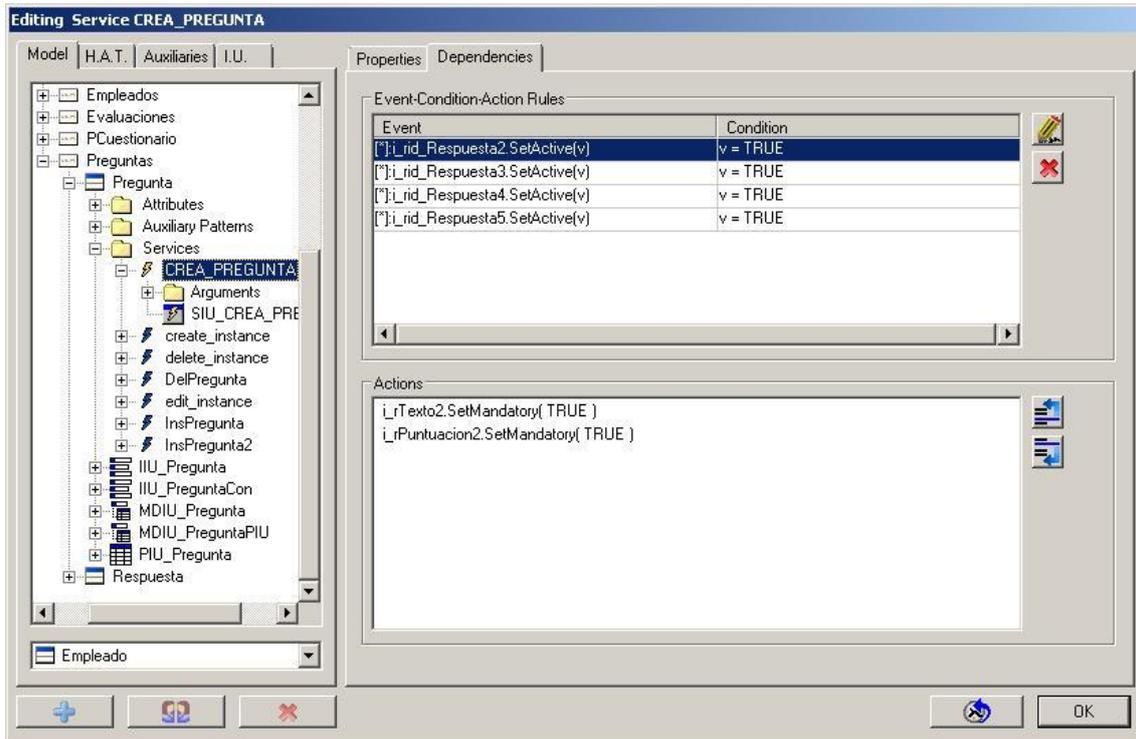


Figura 77. Event-Condition-Action de SIU CREA_PREGUNTA.

La Figura 78 muestra el funcionamiento erróneo del alta de pregunta cuando se informa el texto de una respuesta, pero no la puntuación asociada.

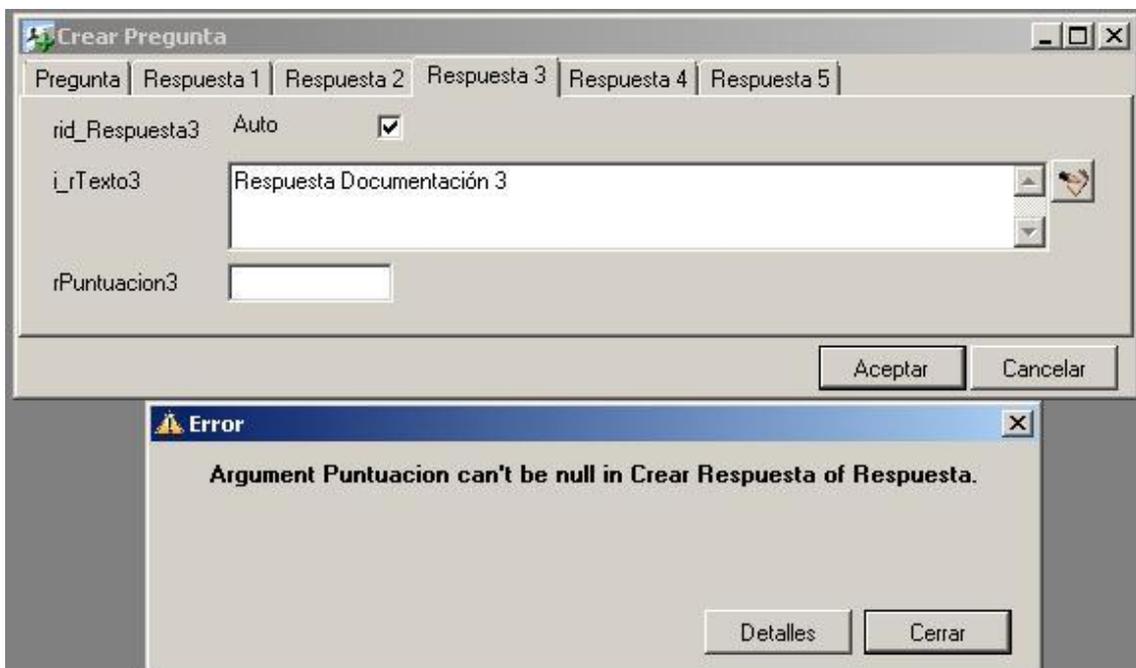


Figura 78. Entrada de SIU CREA_PREGUNTA.

Para esta SIU también se diseña una Navegación Condicional (Conditional Navigation) para poder añadir más respuestas si la ejecución del servicio es correcta. La Figura 79 muestra su especificación en el modelo conceptual y la Figura 80 su representación en la aplicación final que ha sido generada por el compilador de modelos.

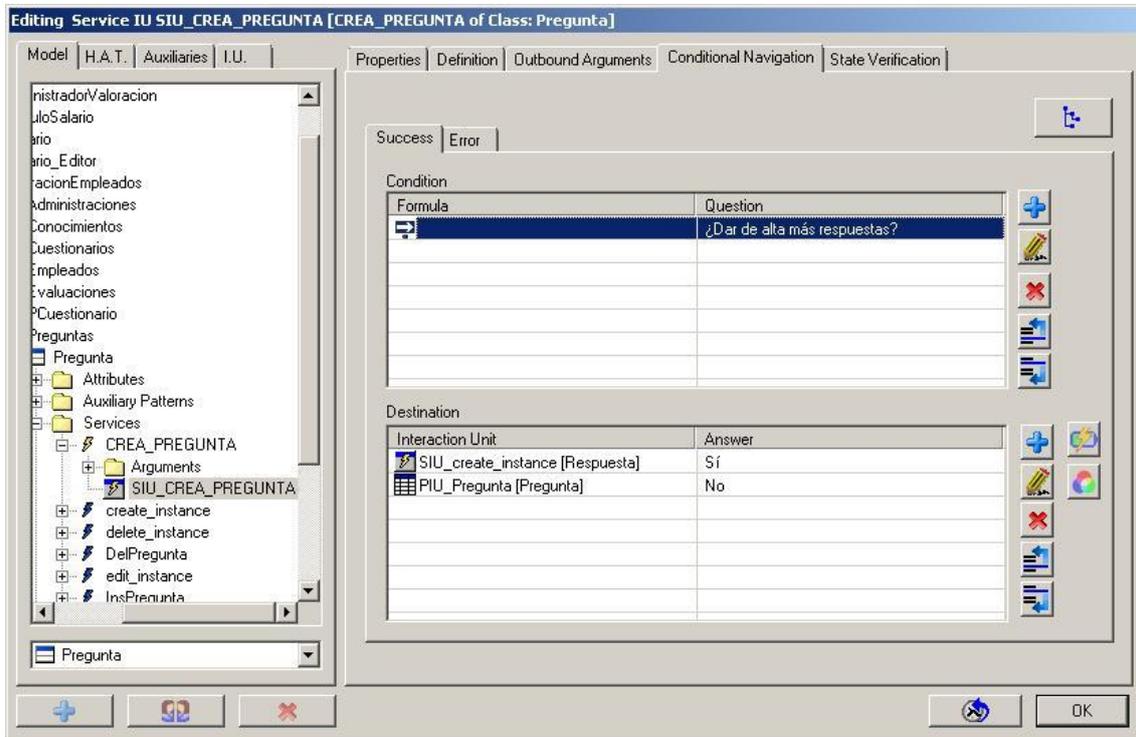


Figura 79. Especificación de la navegación condicional de la SIU CREA_PREGUNTA.

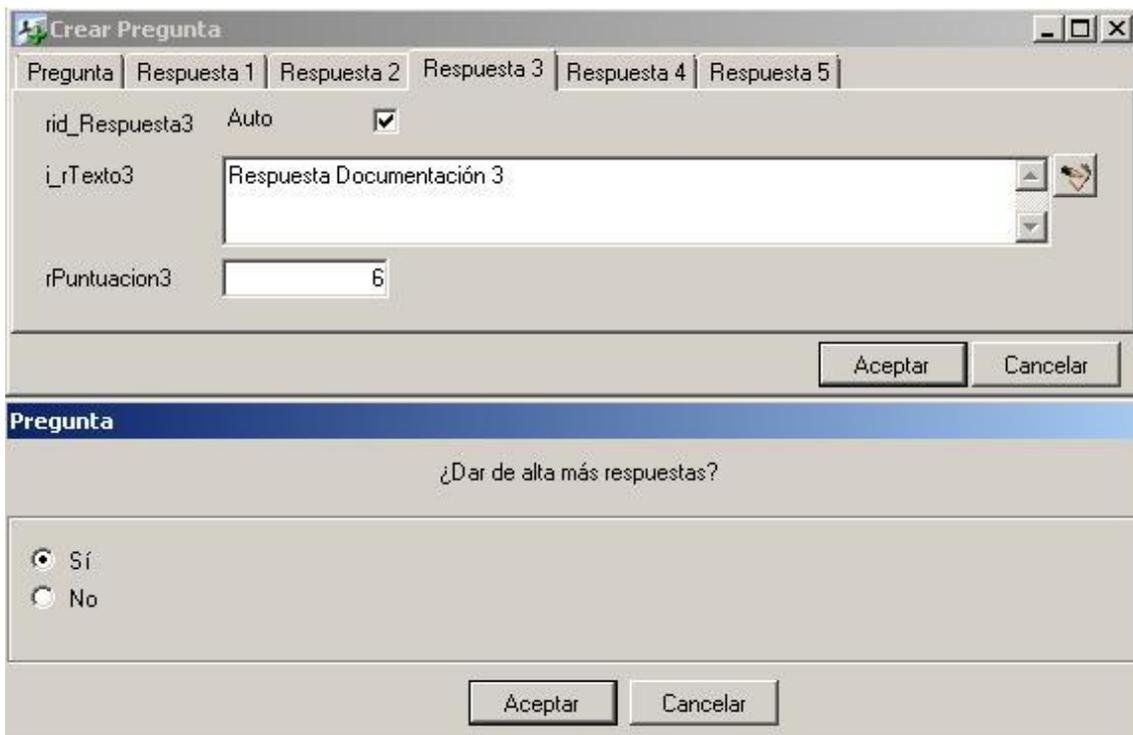


Figura 80. Representación de la *Conditional Navigation* de la *SIU CREA_PREGUNTA*.

La Figura 81 muestra la entrada de la *SIU create_instance* de la clase *Respuesta*, a la que se llega navegando de la ejecución del servicio anterior.

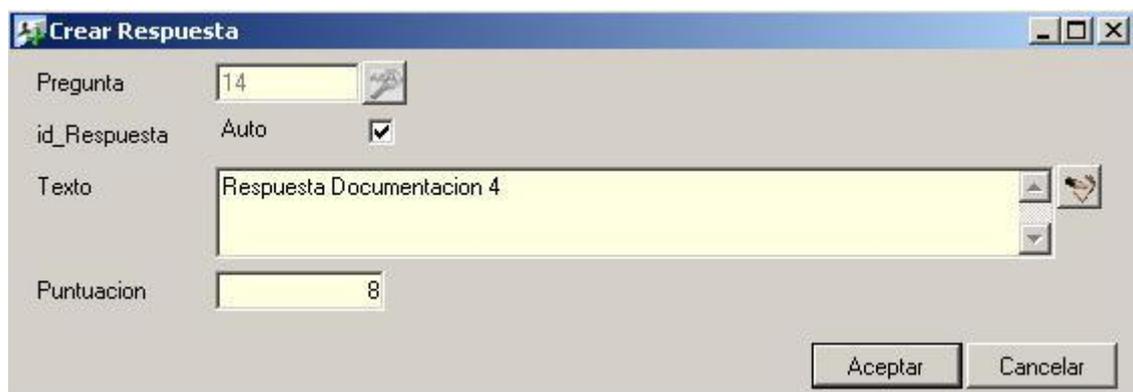


Figura 81. Entrada al servicio de creación de respuesta.

B.2. Detalles de Pregunta

Para los detalles de una pregunta se diseña la Unidad de Interacción Maestro-Detalle (MDIU, por sus siglas en inglés) *Pregunta*. La unidad de interacción maestra es la *IIU Pregunta* y la de detalle es la *PIU Respuesta*. La Figura 82 muestra la especificación de la MDIU en el esquema conceptual.

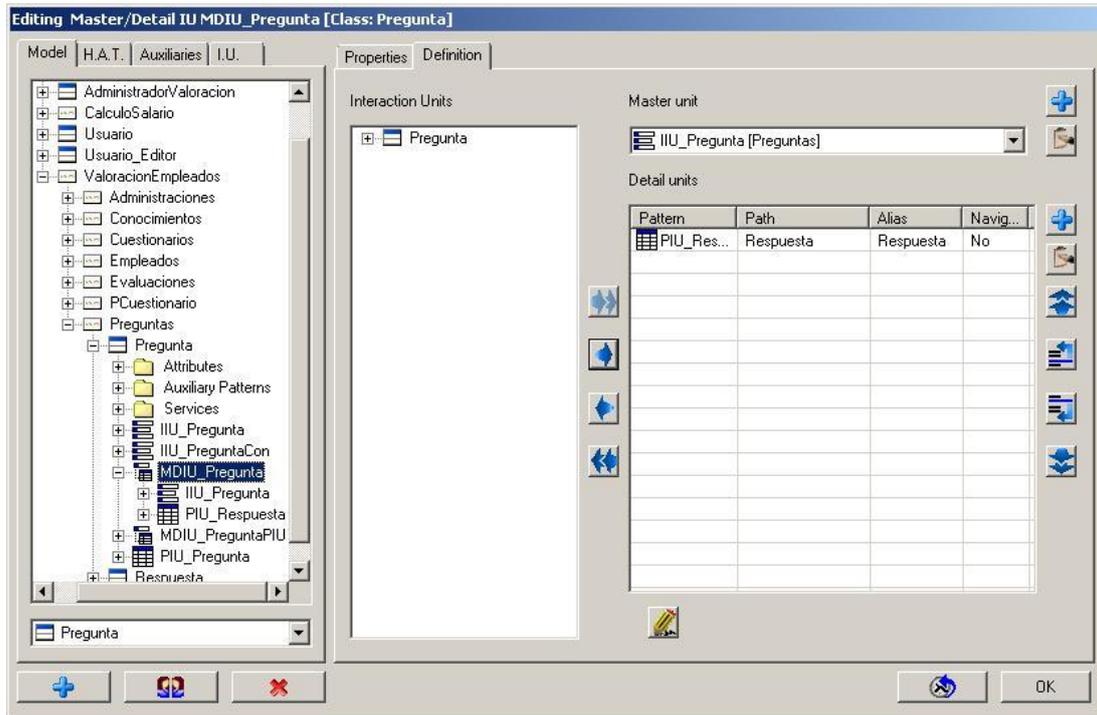


Figura 82. Especificación de MDIU Pregunta.

La Figura 83 muestra cómo se ha generado la pregunta especificada anteriormente con sus respuestas asociadas.

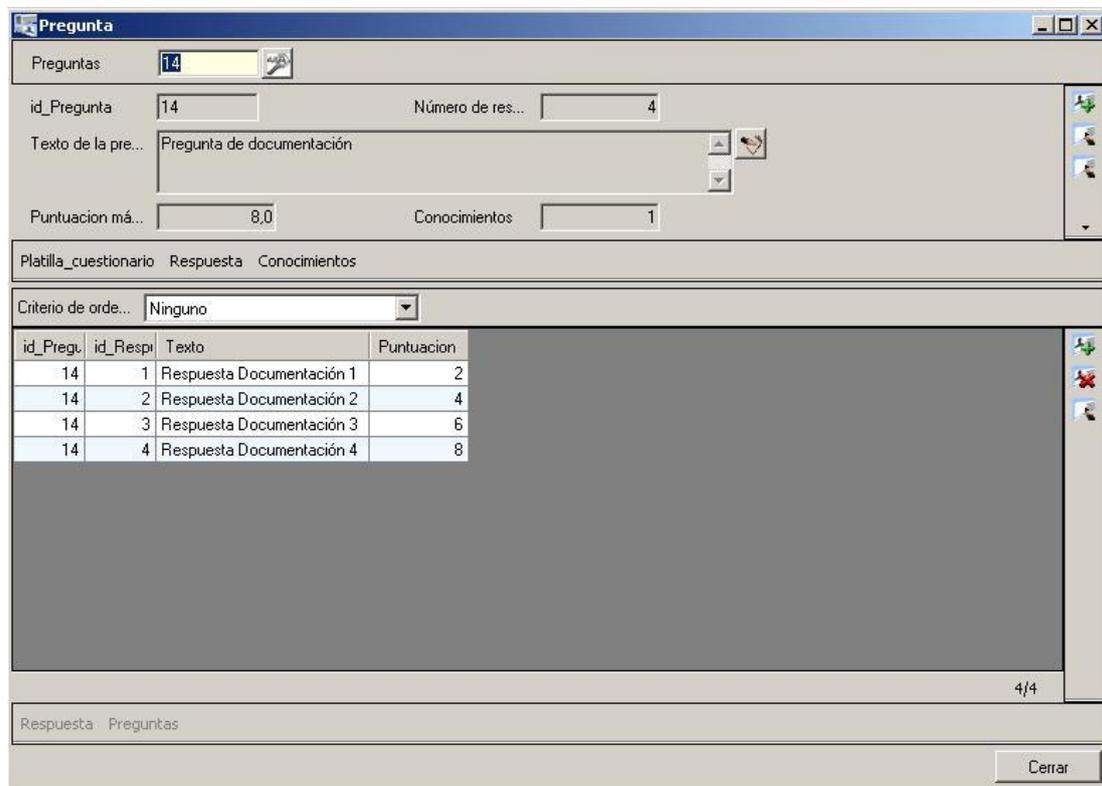


Figura 83. Representación de la MDIU Pregunta.

C.1. Creación de Evaluación

La Figura 84 muestra la entrada del servicio de creación de evaluación, ejecutada en este caso con el agente *ResponsableAdministracion* de la Administración Justicia, a la que pertenece el empleado que se ha creado.

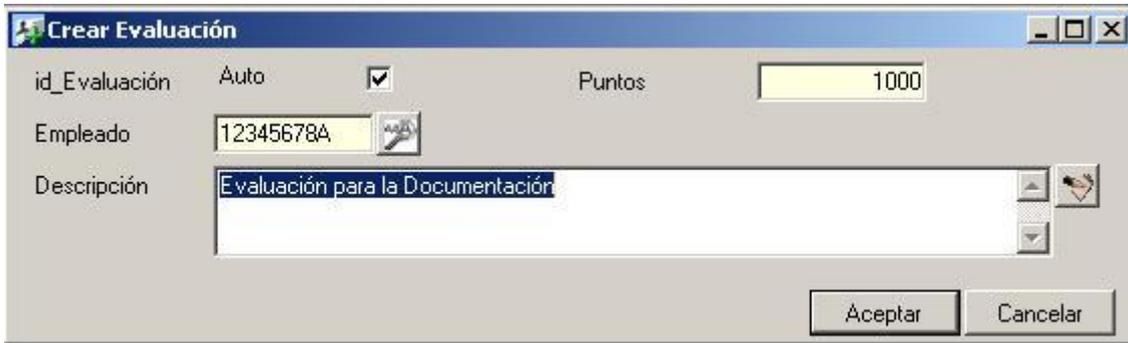


Figura 84. Entrada del servicio de creación de una evaluación.

La Figura 85 muestra la salida de la ejecución de este servicio. Al igual que los otros servicios mostrados anteriormente, este servicio ha sido generado por el compilador de modelos a partir de la especificación de una unidad de interacción de servicio para crear una evaluación en el esquema conceptual, con sus respectivos argumentos de entrada y salida.

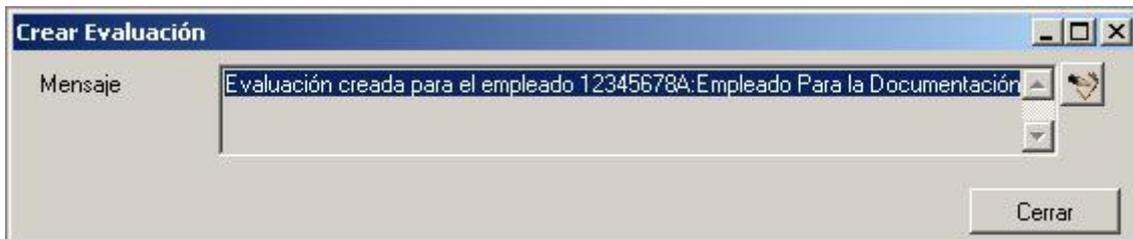


Figura 85. Salida del servicio de creación de una evaluación.

C.2. Creación de Evaluación de Conocimiento

El servicio de creación de evaluación de conocimiento se ejecuta con el agente *ResponsableAdministracion*. La diferencia entre este agente y el *AdministradorValoracion* es que el primero tiene visibilidad sobre las instancias de la clase de la *Administracion* a la que pertenece, mientras que el segundo tiene visibilidad sobre todas las instancias. La Figura 86 muestra la población de conocimientos para el agente *ResponsableAdministracion* y la Figura 87 muestra los conocimientos accesibles para el *AdministradorValoracion*. En estas figuras se puede apreciar claramente la diferencia antes mencionada, ya que en la primera figura sólo se ven conocimientos de la Administración de Justicia y en la segunda se ven también conocimientos de la Conselleria de Sanidad, Hacienda y otras Administraciones de prueba.

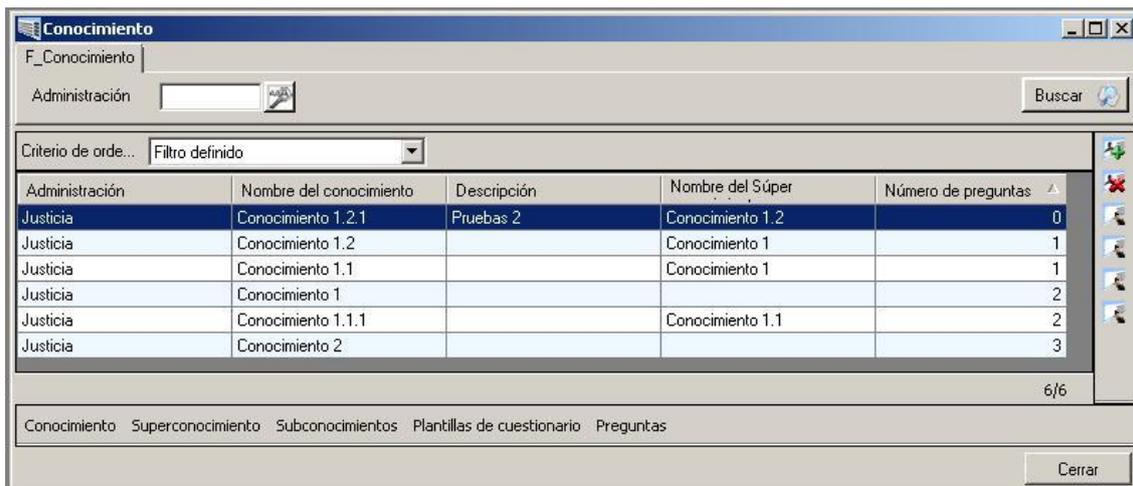


Figura 86. Visibilidad de *PIU Conocimientos* para *ResponsableAdministracion*.

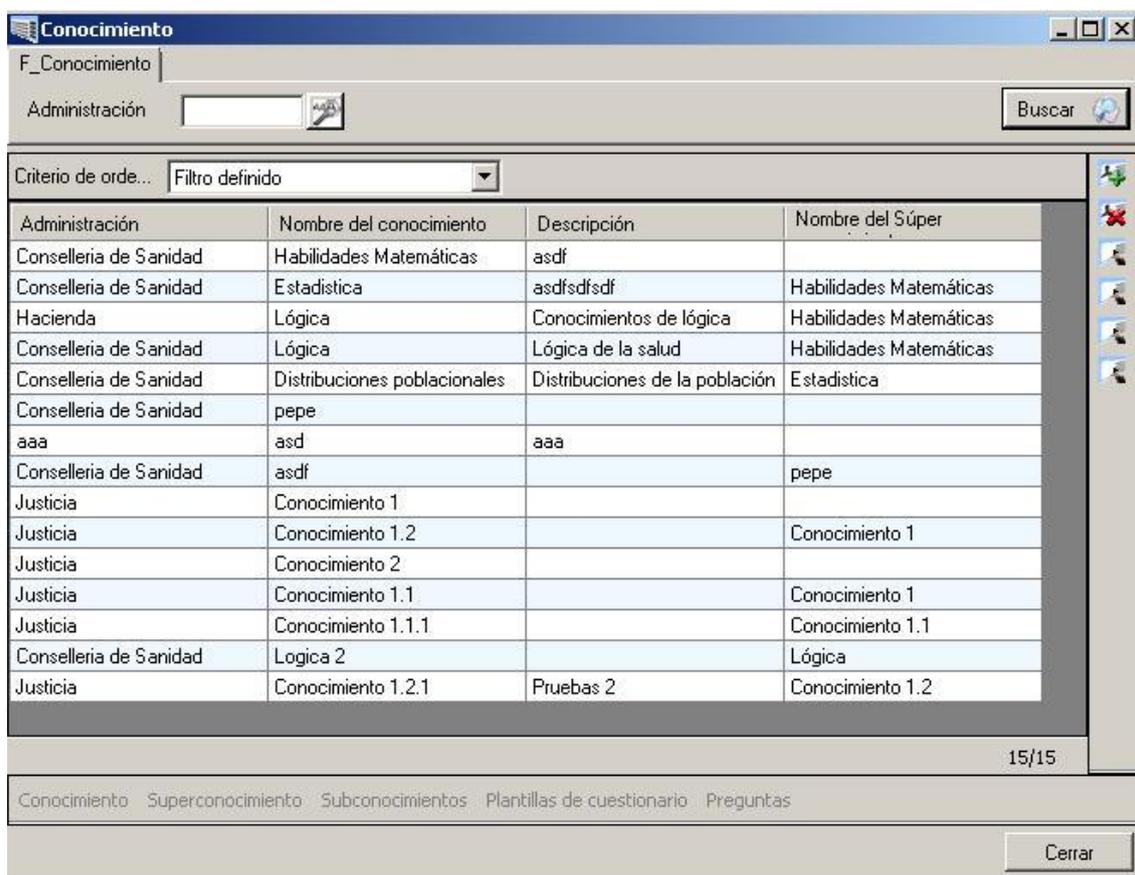


Figura 87. Visibilidad de *PIU Conocimientos* para *AdministradorValoracion*.

Para restringir la visibilidad de la información desplegada en la unidad de interacción de población de conocimiento para los distintos agentes, es necesario especificar una fórmula en la clase conocimiento. La fórmula que define la visibilidad sobre conocimientos del agente *ResponsableAdministracion* se muestra en la Figura 88. En esta fórmula se puede apreciar que

el *ResponsableAdministracion* puede ver los conocimientos de tengan como administración la misma a la cual el pertenece dicho agente.

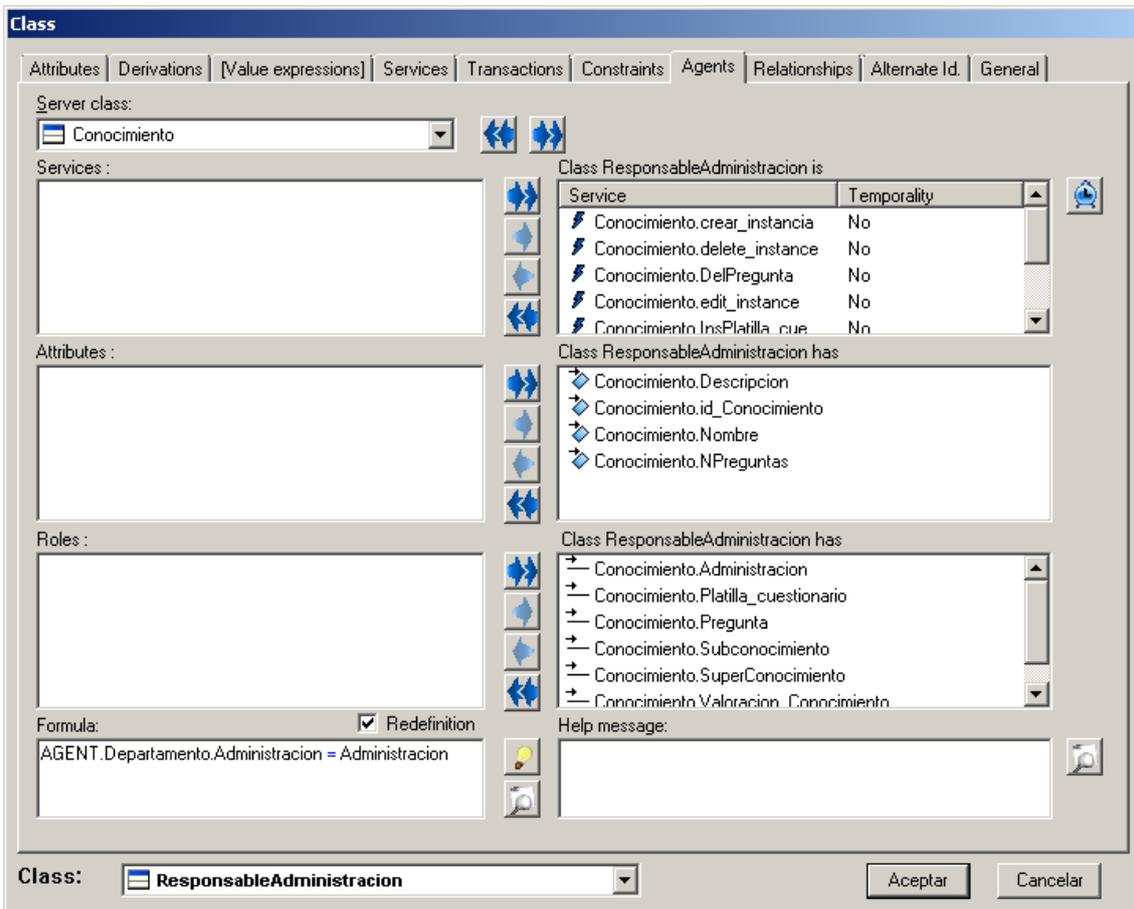


Figura 88. Definición de la visibilidad del agente *ResponsableAdministracion* sobre la clase *Conocimiento*.

La Figura 89 muestra la entrada del servicio de creación de evaluación de conocimiento, para el que se elige el "Conocimiento 12". La Figura 90 es la salida de este servicio.

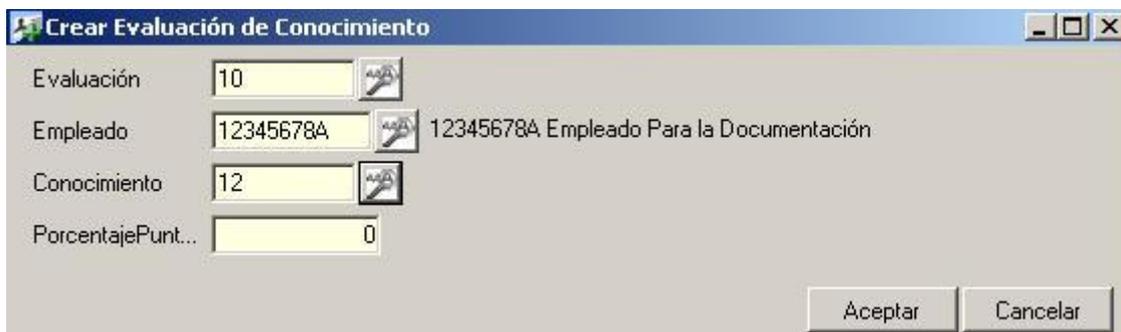


Figura 89. Entrada de servicio de creación de evaluación de conocimiento.

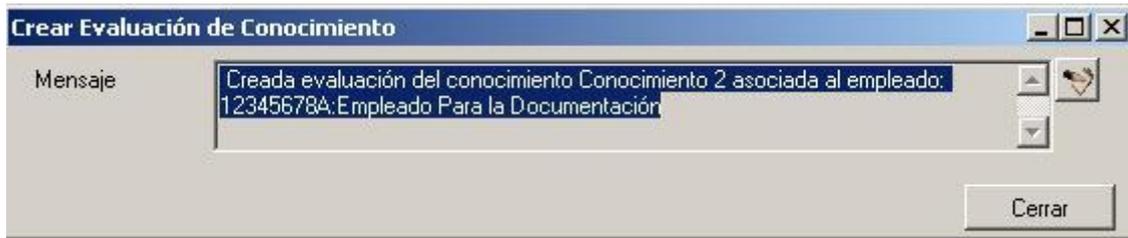


Figura 90. Salida de servicio de creación de evaluación de conocimiento.

En la Figura 91 se puede observar la *MDIU Valoración*, formada por una instancia de evaluación y la población de evaluaciones de conocimiento relacionadas. El atributo puntos de la evaluación de conocimiento es derivado como resultado del 100% del atributo puntos totales de la evaluación y se corresponde con el atributo derivado *PuntosConocimiento* definido en la clase *Valoracion_Conocimiento*, como se muestra en la Figura 92.

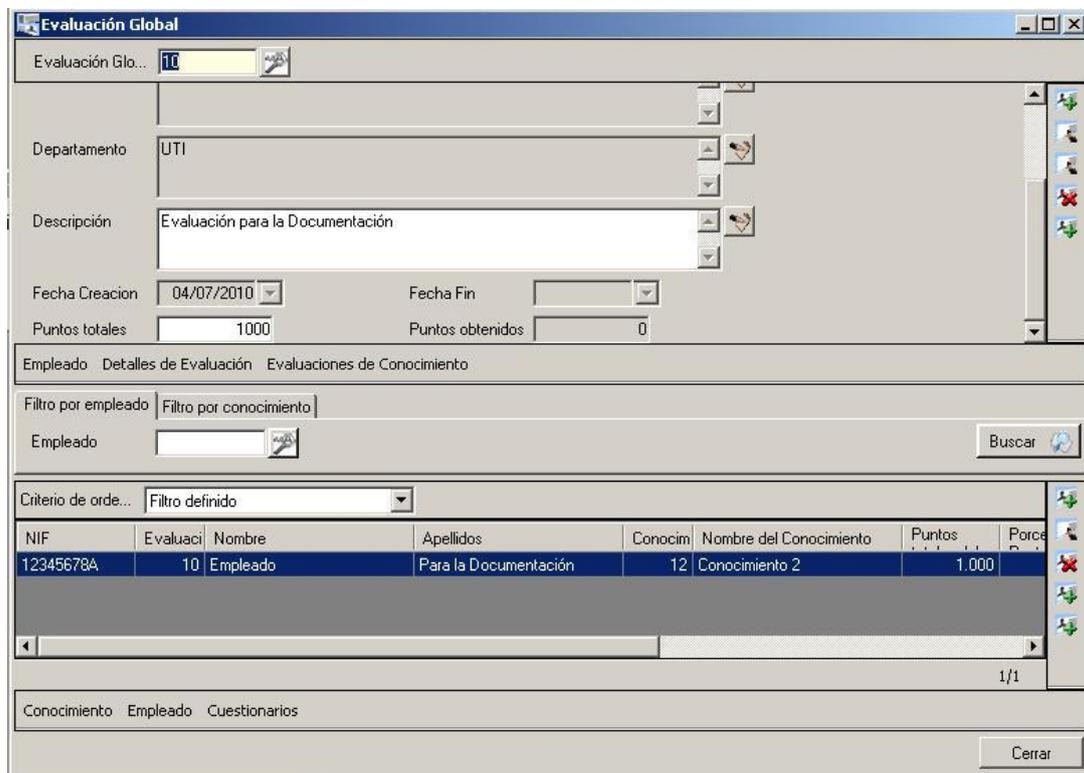


Figura 91. *MDIU Valoracion.*

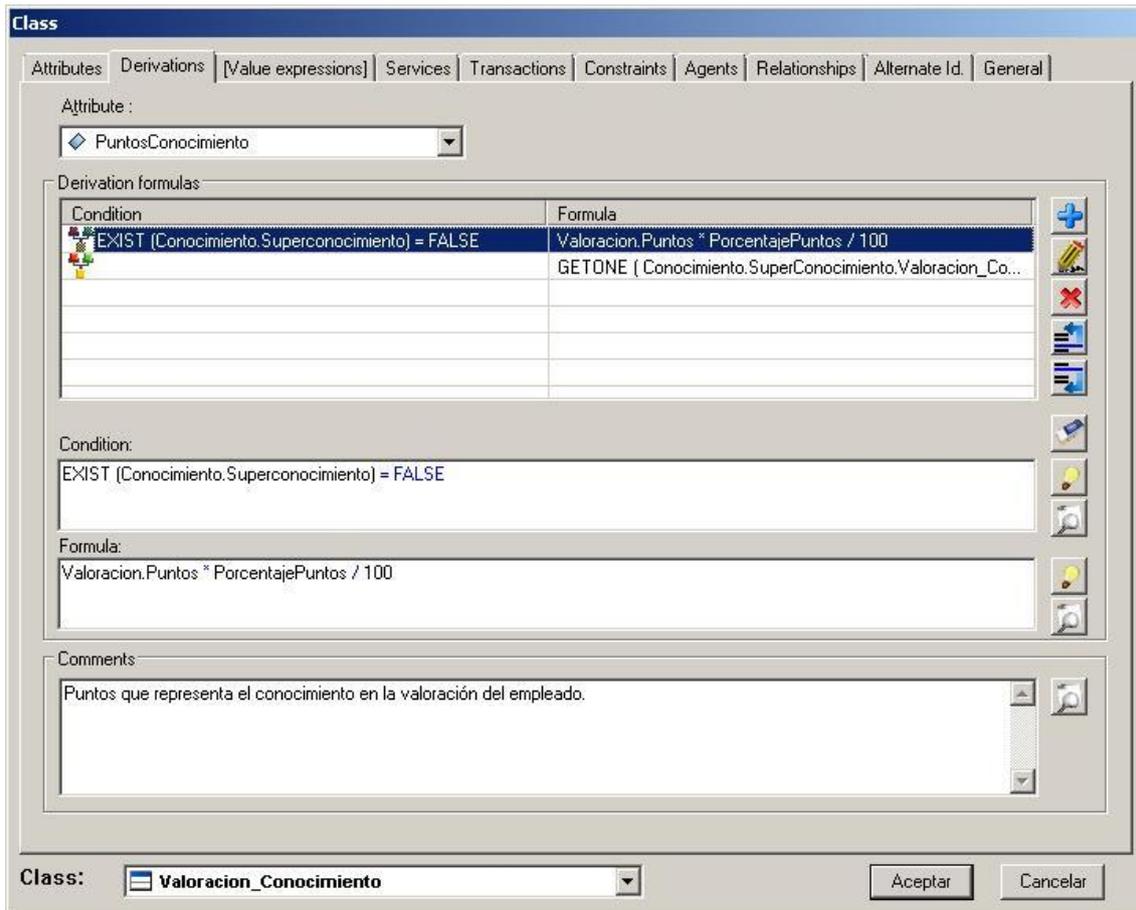


Figura 92. Derivación del atributo *PuntosConocimiento* de la clase *Valoracion_Conocimiento*.

En la Figura 93 se observa la representación de una restricción de integridad que no se cumple, que corresponde a que el porcentaje esté entre cero y 100.

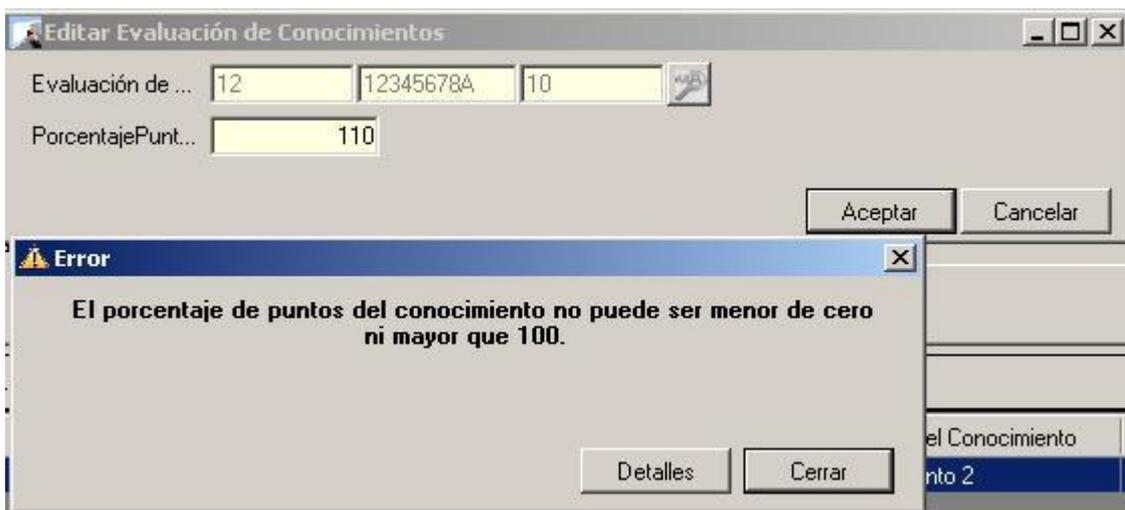


Figura 93. Restricción incumplida al ejecutar servicio de edición de evaluación de conocimiento.

Esta restricción está especificada en el esquema conceptual en la clase *Valoracion_Conocimiento*, como se observa en la Figura 94.

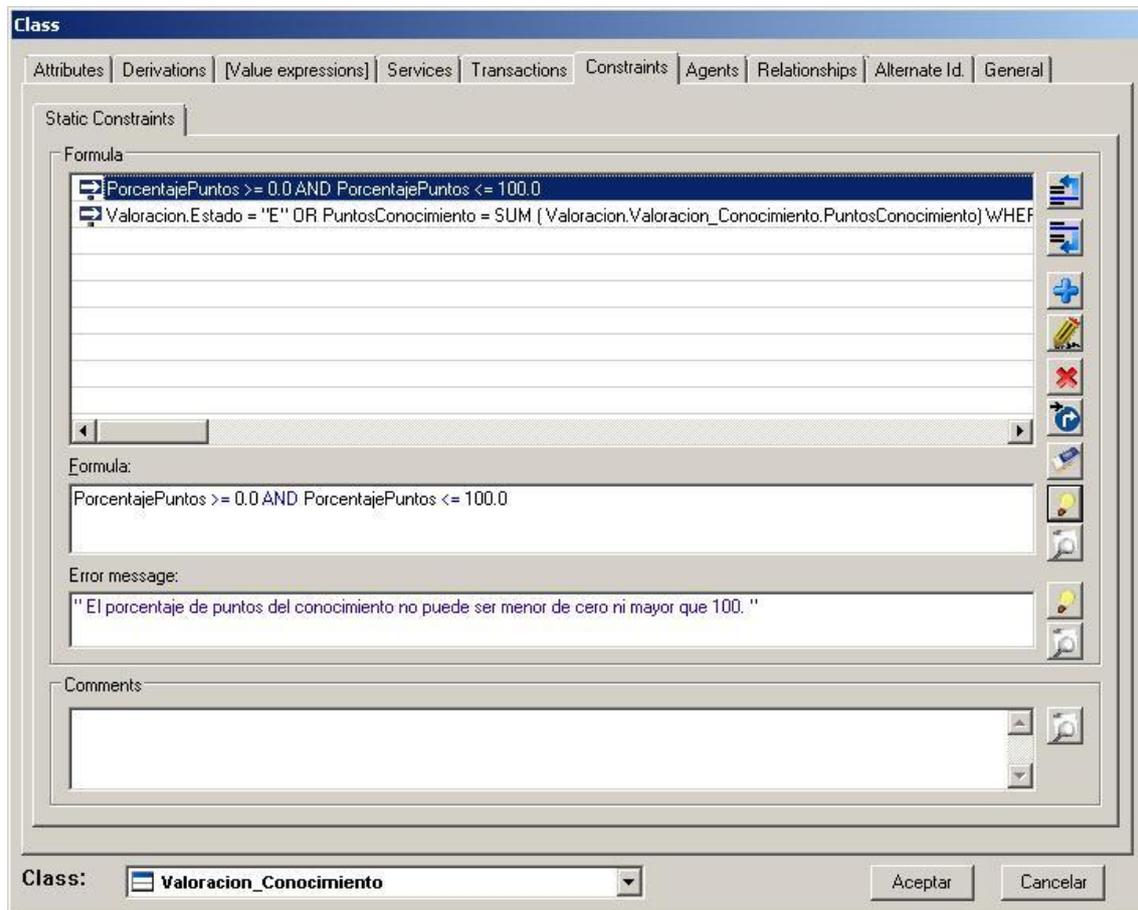


Figura 94. Restricción definida en la clase *Valoracion_Conocimiento*.

C.2. Creación de Cuestionario

Se crea un cuestionario para la evaluación de conocimiento. En la Figura 95 se muestra la entrada del servicio de creación de cuestionario. En este servicio se informan la plantilla de cuestionario, el empleado, el empleado supervisor y la evaluación de conocimiento con las que se va a asociar.

Figura 95. Entrada de servicio de creación de cuestionario.

La Figura 96 representa la *IIU Cuestionario*. Se puede observar, entre otros atributos, que la fecha de creación y el estado del cuestionario toman el valor por defecto que se define en el esquema conceptual.

Figura 96. IIU Cuestionario.

D.1. Realización de Cuestionario

La Figura 97 muestra la visibilidad que tiene el empleado *12345678A* sobre los cuestionarios. Sólo tiene un cuestionario asignado y, por tanto, es el único al que tiene acceso. El estado del cuestionario es "Preparado", por lo que se puede contestar. Se puede observar también la entrada del servicio *empezar_cuestionario* de la clase *Cuestionario*.

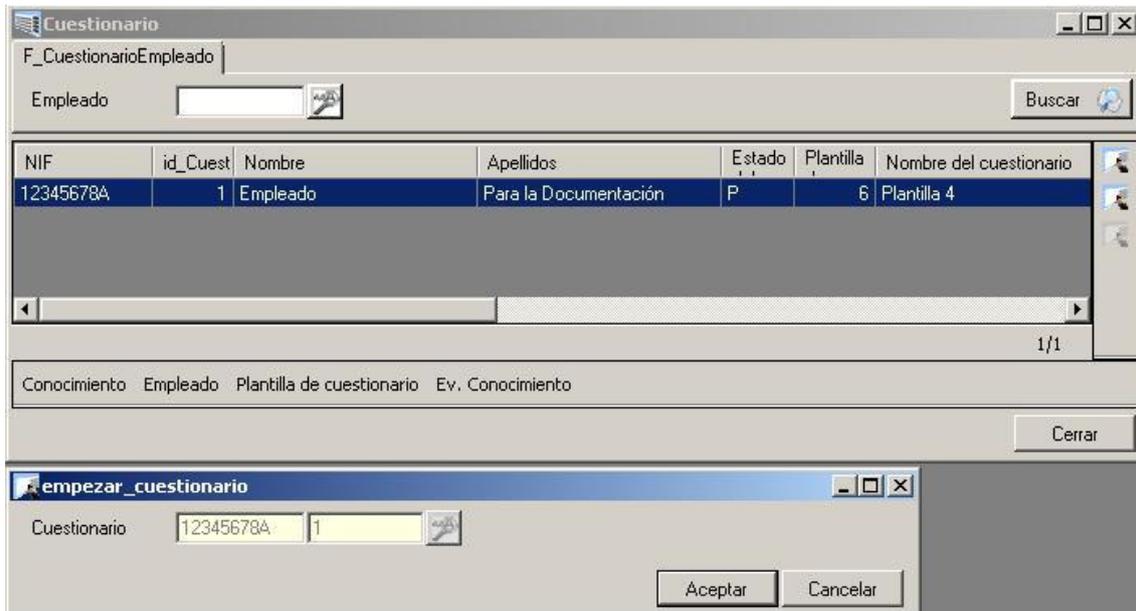


Figura 97. PIU *Cuestionario* para el empleado 12345678A.

La *SIU* *empezar_cuestionario* tiene una navegación condicional que lleva a la *MDIU* *Plantilla_cuestionarioAcc* mostrada en la Figura 44. La Figura 98 muestra la transformación de esta unidad de interacción en la aplicación final.

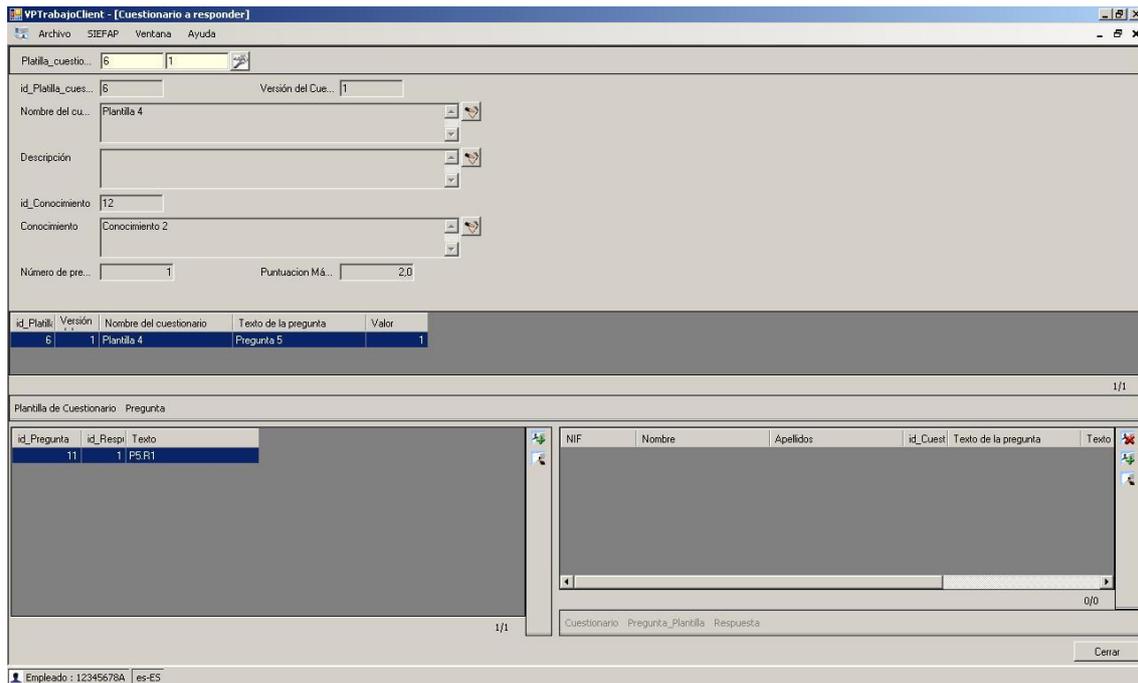


Figura 98. MDIU Plantilla_cuestionarioAcc.

En la Figura 99 se puede observar la entrada del servicio *responder* de la clase *Respuesta_Empleado* y, en la Figura 100, se observa la respuesta dada por el empleado, filtrada con la fórmula definida en el *MDIU Plantilla_cuestionarioAcc*.

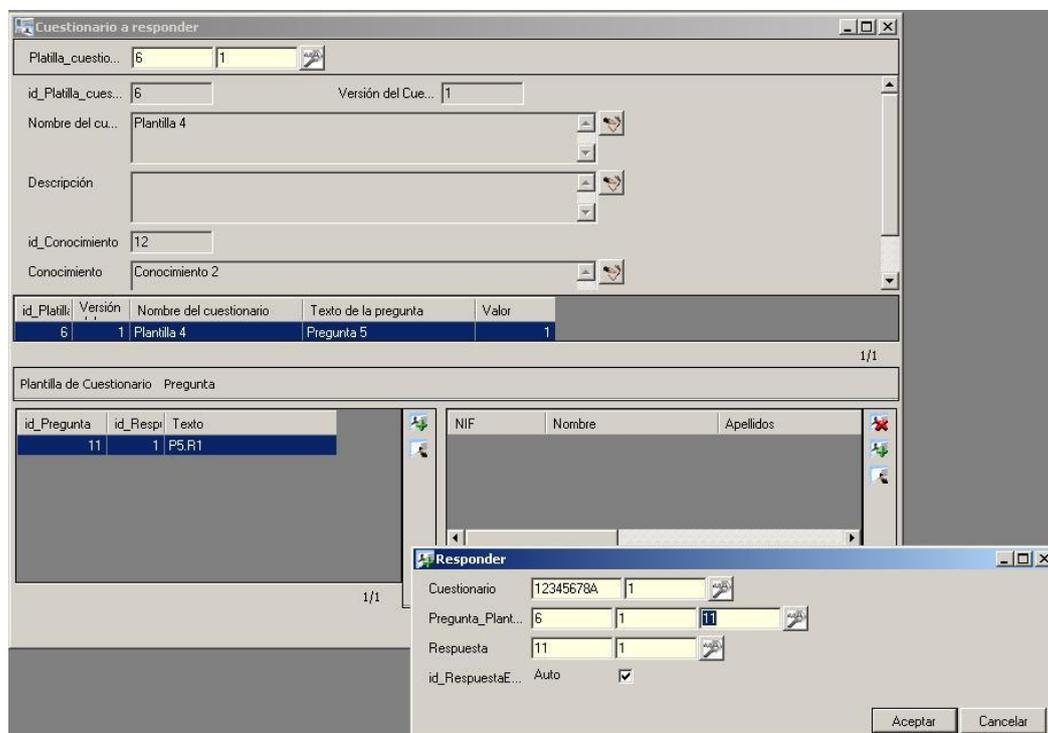


Figura 99. Entrada del servicio responder.

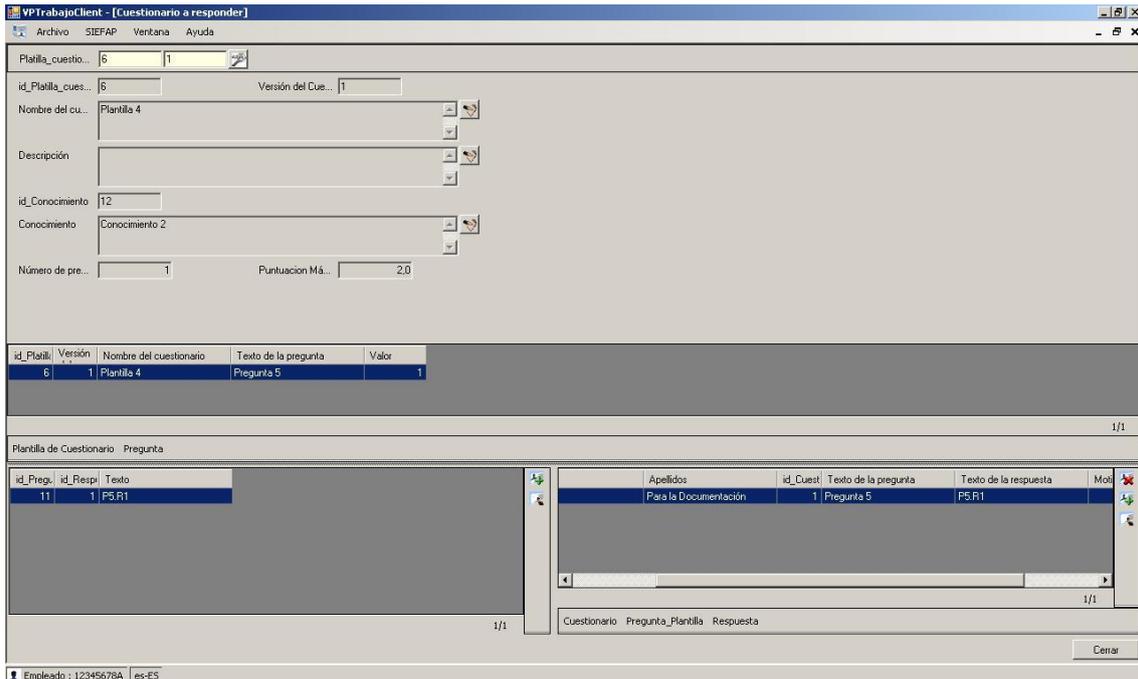


Figura 100. MDIU *Plantilla_cuestionarioAcc* con respuesta seleccionada.

E.10. Validación de Cuestionario

Con el agente *EmpleadoSupervisor*, se ejecuta el servicio *validar_cuestionario* de la clase *Cuestionario*. La especificación de este servicio en el modelo funcional se muestra en la Figura 101.



Figura 101. Modelado funcional del servicio *validar_cuestionario*.

El cuestionario está en estado “Finalizado”, como exige su precondition y pasa a estado “Validado”. La Figura 102 muestra la población de cuestionarios sobre los que el supervisor tiene visibilidad, además de la entrada del servicio de validación de cuestionario.

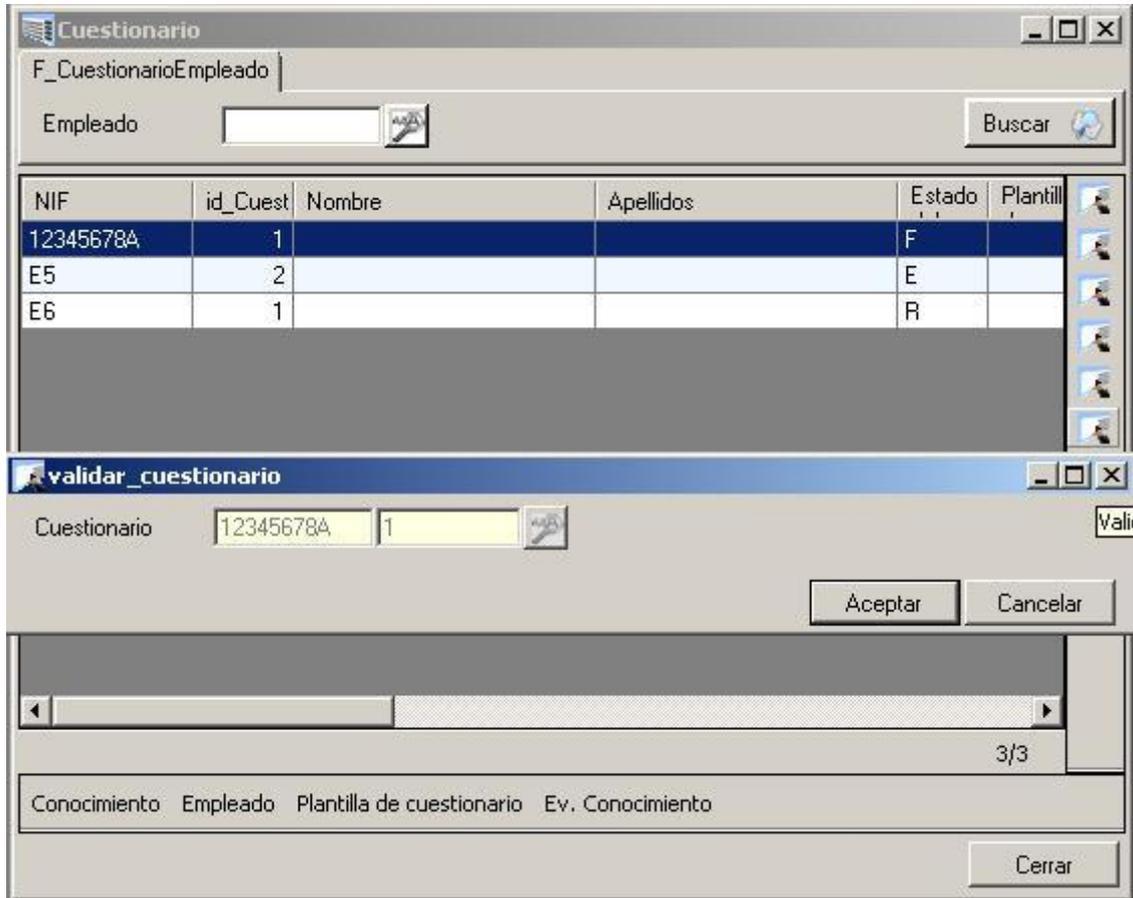


Figura 102. Ejecución de servicio de validación de cuestionario.

F.11. Detalles de Evaluación, Evaluación de Conocimiento y Cuestionario

Por último, se muestra nuevamente la transformación de la *MDIU Valoracion* en la Figura 103, la *MDIU Valoracion_ConocimientoCo* en la Figura 104 y la *IUU Cuestionario* en la Figura 105. En la primera figura, se puede observar la evaluación global y la evaluación de conocimiento relacionada. Al estar el cuestionario validado, los valores de puntos para ambas toman el valor correspondiente que define su Derivación.

Aplicación práctica de MDA con Olivanova para una aplicación de evaluación de recursos humanos

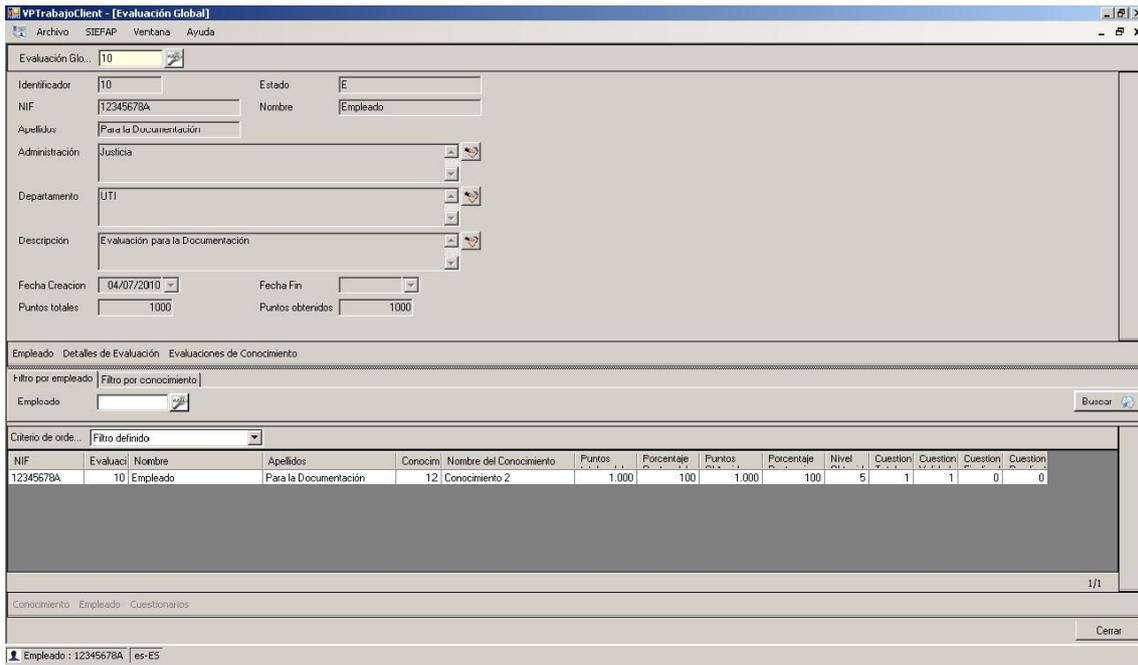


Figura 103. MDIU Valoracion.

En la siguiente figura, se puede observar la evaluación de conocimiento y el cuestionario asociado a ella.

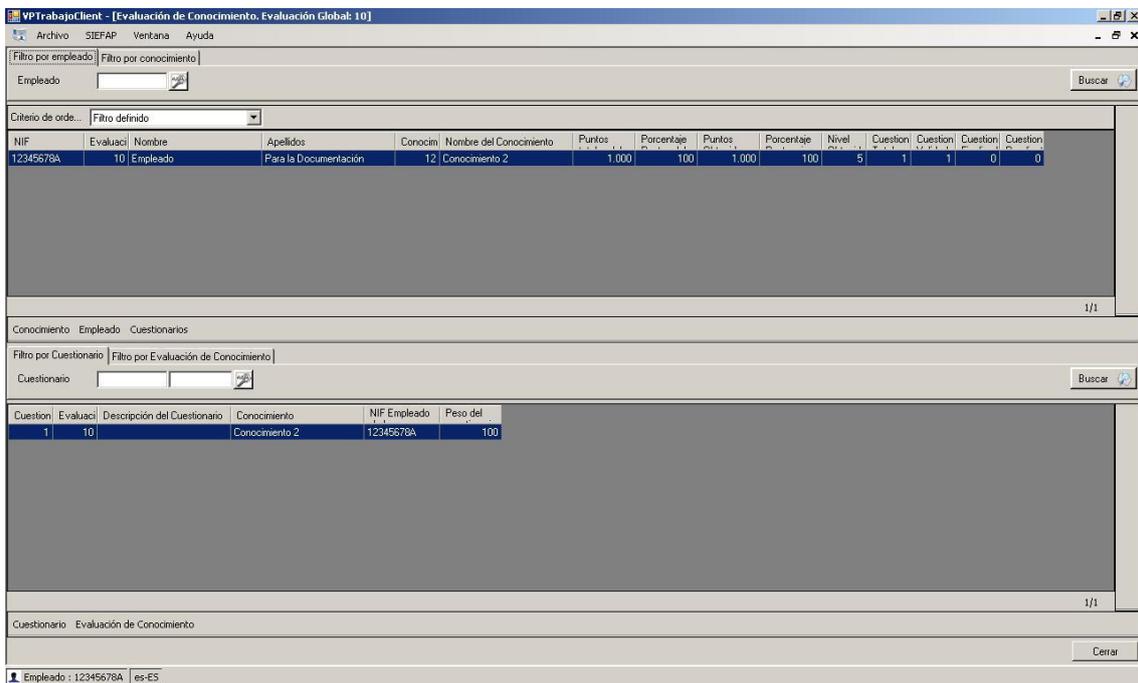


Figura 104. MDIU Valoracion_ConocimientoCo.

La siguiente figura muestra la información del cuestionario. A diferencia de cuando se crea el cuestionario, ahora se muestra el valor de las fechas de comienzo, corrección y fin, el

valor de la puntuación obtenida y del número de respuestas. Se observa que el estado del cuestionario es ahora “Validado”, que está representado con la letra V.

The screenshot shows a window titled "Cuestionario. PesoCuestionario: 12345678A 1 12 12345678A 10". The form contains the following fields:

Cuestionario	12345678A	1	
NIF	12345678A	id_Cuestionario	1
Nombre	Empleado	Apellidos	Para la Documentación
Estado del cue...	V	Plantilla de cue...	6
Nombre del cu...	Plantilla 4		
Descripción			
id Conocimiento	12		
Conocimiento	Conocimiento 2		
Número de pre...	1	Número de res...	1
Puntuación Má...	2,0	Puntuacion	2,0
Fecha Comienzo	04/07/2010	FechaCorreccion	04/07/2010
FechaFin	04/07/2010		

At the bottom of the form, there is a breadcrumb trail: "Conocimiento Empleado Plantilla de cuestionario Ev. Conocimiento". A "Cerrar" button is located at the bottom right.

Figura 105. *IIU Cuestionario.*

Finalmente, la Figura 106 muestra la especificación de una Precondición para el servicio *terminar_cuestionario* que impide que se lance su ejecución si el estado del cuestionario no es “En Respuesta”.

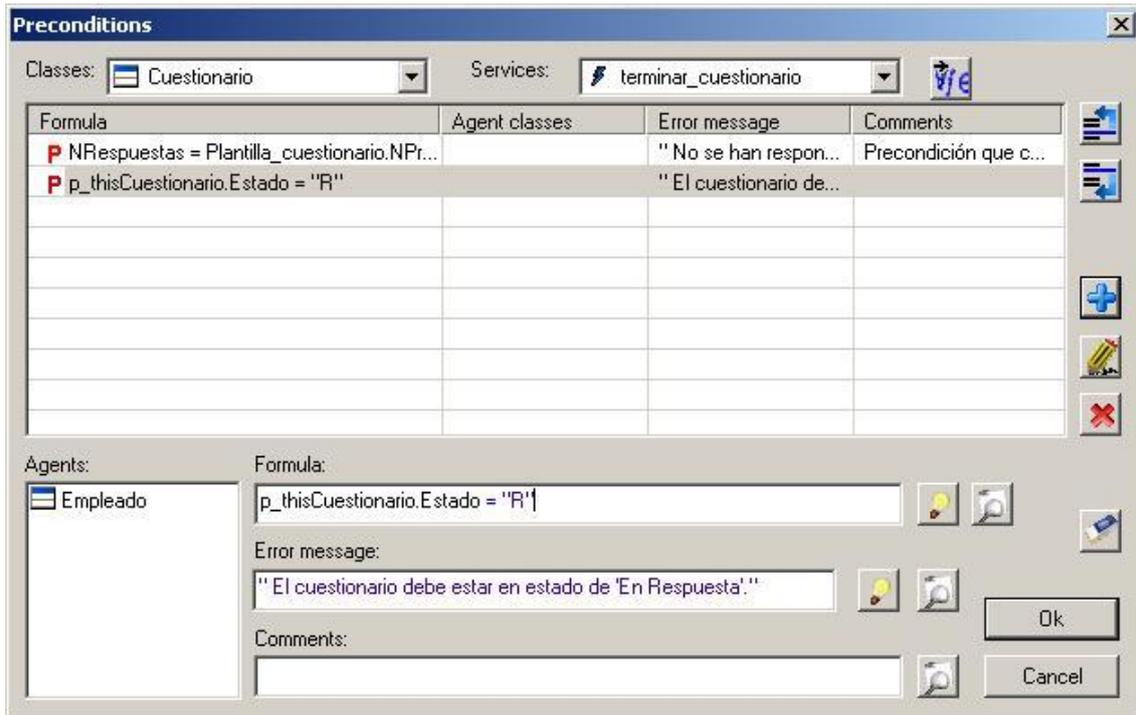


Figura 106. Precondiciones para el servicio *terminar_cuestionario* de la clase *Cuestionario*.

En la Figura 107 se ve cómo se representa el incumplimiento de esta precondición.

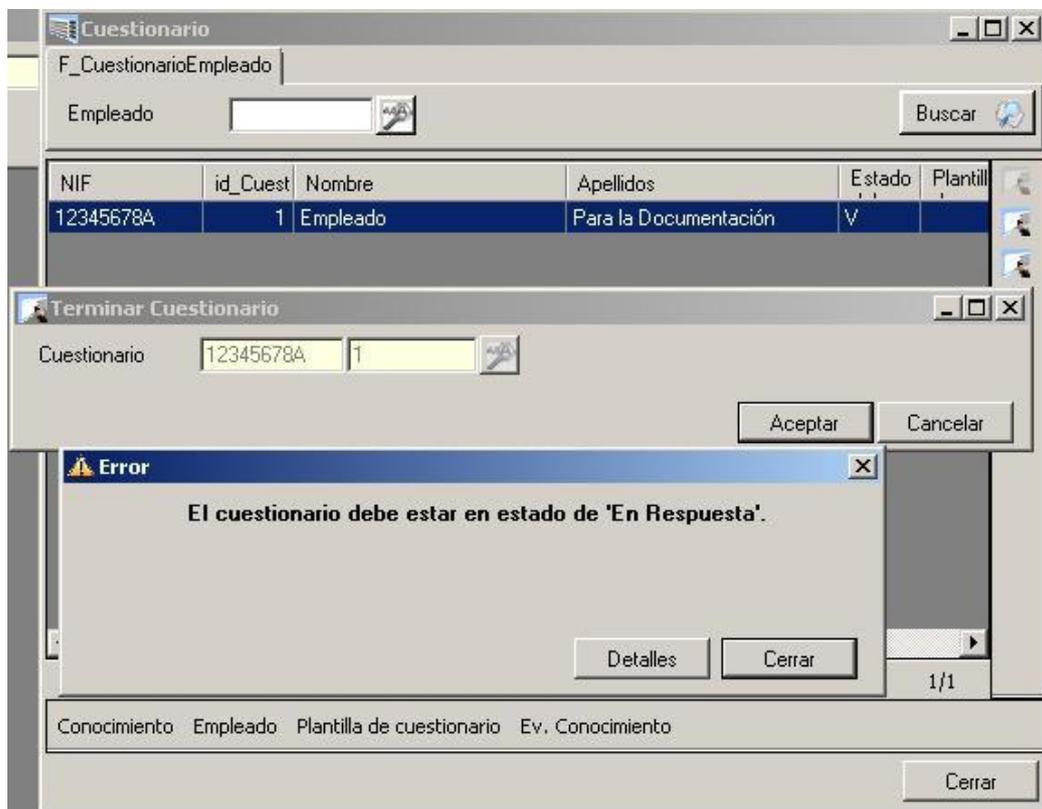


Figura 107. Precondición incumplida para SIU *terminar_cuestionario*.

Una vez explicado cómo se llega al Espacio de la Solución mediante las herramientas Olivenova y la correspondencia entre elementos del Espacio del Problema y del Espacio de la Solución, en el siguiente capítulo de este documento nos centraremos en presentar las conclusiones obtenidas en este Proyecto Final de Carrera.

6. CONCLUSIONES

En este capítulo de conclusiones se analizan aquellos aspectos relevantes identificados durante la realización de este Proyecto Final de Carrera y que están ligados a la tecnología MDD utilizada para la realización del mismo: OO-Method y las herramientas Olivanova. Luego, de acuerdo a la experiencia adquirida, destacamos situaciones especiales que hemos identificado en el uso de las herramientas industriales que implementan la tecnología OO-Method. A su vez, planteamos posibles mejoras que podrían ser integradas a estas herramientas y al proceso de compilación en versiones futuras.

Además, se analizan los objetivos planteados al comienzo de este trabajo y se contrastan con los resultados obtenidos. Además, se presentan posibles ampliaciones del trabajo realizado. Estas extensiones van dirigidas tanto al producto software construido como las posibilidades de implantación de en entornos de producción.

6.1 OO-Method y Olivanova

Este proyecto ha partido de la base teórica de que es posible generar código 100% operativo a partir de modelos conceptuales. Para una aplicación práctica de esta teoría, se ha utilizado la herramienta Olivanova Model Execution, basada en la metodología OO-Method.

Con esta herramienta se consigue el objetivo de crear código de manera automática a partir de un modelo conceptual. Las ventajas más importantes de emplear esta tecnología de desarrollo, como se presuponía en la teoría son estas:

- Método de construcción de software más centrado en la funcionalidad del sistema, lo que permite entender mejor el espacio del problema dejando a un lado elementos técnicos del espacio de la solución. Esto permite acortar los tiempos de desarrollo y añade facilidad de entendimiento entre el analista y el usuario o promotor de la aplicación.
- Facilidad de detección y resolución de incidencias. A lo largo de este PFC se han generado distintos prototipos de la aplicación final. En la fase de pruebas de éstos, la detección de incidencias ha consistido en encontrar los defectos de diseño en el esquema conceptual y solucionarlos en el mismo esquema. Por tanto, este trabajo se realiza completamente sobre el Espacio del Problema. El hecho de dejar de lado aspectos técnicos es de gran ayuda para resolver las deficiencias del diseño.

- La creación de un solo esquema conceptual sirve para obtener distintos productos software finales con igual funcionamiento lógico. El esquema conceptual es un modelo independiente de plataforma y para el modelo de implementación en este PFC se ha elegido la tecnología .NET. Sin embargo, también se ha trabajado sobre tecnología Java.

Sin embargo, es importante destacar que el código generado es difícil de entender por personas que no sean expertas en como el proceso de compilación genera el código fuente final. En el proceso teórico de generación de código a partir de modelos, esto no implica un mayor problema; en particular, a lo largo del desarrollo de este PFC no hemos tenido la necesidad de manipular directamente el código generado, pues todas las modificaciones se han realizado en el nivel del modelo conceptual.

Para implantar el producto software son necesarios unos requisitos específicos en el sistema destino. Estos requisitos comprenden el tipo y versiones de servidor de aplicaciones, el Sistema Gestor de Base de Datos (SGBD), el tipo de conexión entre aplicación y base de datos y librerías necesarias para la ejecución correcta. Estos requisitos se definen en función de la Transformation Engine seleccionada. En este PFC se han cumplido estos requisitos y el resultado es una aplicación generada 100% funcional a partir de las especificaciones del esquema conceptual.

Sin embargo, en organizaciones reales, donde existen entornos de producción ya implantados, los servidores de aplicaciones, de bases de datos y entornos de trabajo están definidos con anterioridad. Esto puede provocar que existan requisitos especiales asociados a la generación del producto software final y su ejecución que no puedan ser especificados mediante las posibilidades de configuración de la Transformation Engine, y por lo tanto, será necesario adaptar fragmentos del código final generado. Así, el no contar con un buen conocimiento del código generado puede presentar un inconveniente importante. Este inconveniente se daría sólo en el caso de que las circunstancias obligasen a trabajar a nivel de código generado; pues si se puede trabajar a nivel de modelo conceptual, las modificaciones se realizan de forma correcta y automática sobre la aplicación final. En este punto, asumimos la eventual aparición de funcionalidades que no puedan ser modeladas y deban ser implementadas manualmente.

6.2 Situaciones detectadas y posibles mejoras

Cabe destacar también que este trabajo se ha realizado con una tecnología de desarrollo de última generación y, por tanto, bastante reciente. A pesar de que esta tecnología es totalmente estable, es posible encontrar algún defecto a nivel de validación del modelo que permita que la transformación del modelo genere código fuente con algún fallo.

Estos defectos que no se detectan en la validación, bien pueden impedir la compilación, bien pueden causar algún fallo inesperado en tiempo de ejecución, o bien puede afectar a la funcionalidad de la aplicación.

Veamos un ejemplo de cada uno de estos defectos:

- En el modelo funcional, al diseñar una evaluación por situación de un atributo, se definen para un mismo atributo dos posibles valores finales, dependiendo del valor que tiene antes de ejecutarse un servicio. Además, se define una condición para una de estas evaluaciones, como se ve en la Figura 108.

Attribute	Event		Effect	Condition	Current v...
Estado	cambiar_estado	=	"F"	Estado = "A"	"A"
Estado	cambiar_estado	=	"A"		"E"

Figura 108. Defecto de modelado en evaluación de atributos por situación

En el código generado, estas dos evaluaciones se definen como dos instrucciones “if”, la segunda de las cuales tiene una parte “else” que no toma valor, como se observa en el siguiente fragmento de código:

```

try
{
    // Current Value: "A"
    // Condition: Estado = "A"
    if (Instance.EstadoAttr == new ONString(@"A") &&
Instance.EstadoAttr == new ONString(@"A"))
    // Effect: "F"
    lEstadoAttr = new ONString(@"F");
    // Current Value: "E"
    if (Instance.EstadoAttr == new ONString(@"E"))
    else
    // Effect: "A"
    lEstadoAttr = new ONString(@"A");
}
    
```

Esto impedía la compilación de la aplicación, pero se soluciona fácilmente eliminando la parte “else” a nivel de código generado.

- En una de las primeras versiones de la aplicación, se diseñó en el esquema conceptual que los mensajes de salida de eventos de destrucción de instancia tomaran valor del argumento de entrada de tipo objeto que se iba a eliminar. En el momento de ejecución, la aplicación producía una excepción no controlada porque el objeto de donde quiere recuperar atributos ya no existe. La solución a este problema, fue tomar el valor de la salida de los atributos del objeto de la instancia de la clase directamente. Un ejemplo de esto me ocurrió en el servicio de borrado de la clase *Categoria*; el error aparecía cuando el argumento de salida tomaba el valor *p_thisCategoria.Tipo*,

mientras que la solución fue diseñar el mensaje de salida de forma que toma el valor *Tipo*.

- Un tercer tipo de defecto encontrado en este PFC, está asociado a las restricciones de integridad de clases que contienen atributos de otras clases relacionadas. En este caso, la restricción definida se comprueba en la creación y destrucción del objeto de clase relacionada, pero no en el resto de servicios que cambia valor de atributos de la clase relacionada. De este modo, es posible que una restricción se incumpla sin que ello provoque un fallo en la aplicación. Para solucionar este problema, se han creado precondiciones en los servicios que modifican valores de atributos que aparecen en la restricción.

A lo largo del periodo de trabajo de este PFC se ha comprobado que la tecnología Olivanova suele ser actualizada permanentemente para corregir los defectos identificados y añadir nueva expresividad de modelado al modelo conceptual.

Algunas de estas nuevas funcionalidades que han facilitado el modelado de la aplicación final han sido:

- La incorporación de claves alternativas para una clase, lo que permite implementar reglas de unicidad.
- La definición de instrucciones If-else en la definición de transacciones.
- La definición en el modelo conceptual del servicio de modificación de contraseña que se crea de forma automática para los agentes del modelo.

Estos tres elementos conceptuales no existían al comienzo de este PFC, pero dada su expresividad, se han utilizado para reemplazar otros elementos funcionalmente equivalentes pero más complejos.

Con la experiencia adquirida con herramientas Olivanova en este PFC, se ha identificado la carencia de dos elementos que, bajo nuestro criterio, pueden enriquecer a expresividad del modelado conceptual. Uno de estos elementos es la posibilidad de redefinir la visibilidad de agentes en clases jerárquicas, ya que la clase especializada puede redefinir la fórmula que limita su visibilidad, pero no puede eliminar de su interfaz atributos o servicios heredados de la definición de la interfaz de la clase padre.

Otro elemento que puede resultar interesante para mejorar el modelado conceptual en una herramienta industrial que implemente OO-Method es la posibilidad de acceder a la población de elementos de una clase en los servicios de creación de instancias. Así se podrían definir precondiciones para el objeto que se va a crear comparándolos con los objetos ya existentes. Este elemento sería especialmente útil para comprobar que el intervalo de una instancia que se va a crear no se solapa con el intervalo de instancias ya existentes. Actualmente esta funcionalidad se puede expresar mediante transacciones globales, pero el

modelado sería más simple si se pudiese hacer desde los propios servicios de creación de las clases.

En resumen, es totalmente posible generar código 100% operativo a partir de modelos conceptuales con la tecnología Olivanova, con las ventajas que eso conlleva. Además, es una tecnología que está en constante evolución, añadiendo nueva funcionalidad al modelado y subsanando de forma constante los pocos inconvenientes de versiones antiguas que puedan aparecer.

Como en otras tecnologías de desarrollo de aplicaciones, es necesario adquirir experiencia para conocer cómo se pueden modelar los requisitos de un sistema o cómo elegir los elementos conceptuales más adecuados para ciertos requisitos. Los inconvenientes encontrados a lo largo del PFC los he ido solventando modificando el esquema conceptual, obteniendo en la aplicación final una representación más precisa de los requisitos del sistema conforme aprendía cómo se realizaban transformaciones de los elementos de modelado que he utilizado en el esquema conceptual.

En el plano personal, considero que el haber trabajado con OO-Method me ha permitido conseguir una mejor capacidad de análisis y especificación de requisitos de sistemas de información que me ayudan en mi vida laboral. Además he aprendido a emplear esta tecnología de desarrollo de aplicaciones de última generación, que puede ser útil a la hora de buscar nuevas oportunidades de carrera profesional.

6.3 Cumplimiento de objetivos y ampliaciones del PFC

El sistema Excarpsus es un reto muy ambicioso y complejo de implantación de nuevos sistemas de valoración de puestos trabajo en la Administración Pública. El alcance de este proyecto ha sido la Estimación de Salarios y la Evaluación de empleados, obteniendo dos sistemas de información para su aplicación.

Los objetivos que se marcaron en relación al sistema Excarpsus se han cumplido en la aplicación final, ya que se permite definir unos criterios claros y precisos para la asignación salarial y realizar evaluaciones a los empleados de las administraciones públicas para conocer si los cumplen. Haciendo un debido uso de la información que proporciona la aplicación generada, se puede evaluar si se cumplen los criterios de equilibrio interno y de equidad externa en la política de atribuciones salariales.

La aplicación construida también tiene una pequeña funcionalidad dirigida a todos los ciudadanos para permitir el conocimiento de estadísticas salariales en la Administración Pública; de esta manera se cumple también con el objetivo dirigido a la ciudadanía.

Además de la funcionalidad de la aplicación creada en este PFC, se puede construir un sistema de información para el Cálculo automático de retribuciones a partir de las evaluaciones que se realizan a los empleados, con lo que se completarían todos los puntos que trata el modelo Excarsus. Por tanto, se podría ampliar el trabajo realizado hasta ahora en este sentido.

En el punto en que se encuentra el trabajo realizado en este PFC, se puede plantear nuevos proyectos, que serían los siguientes:

- La implantación de los sistemas creados en un sistema de real producción de una Administración Pública, y el asesoramiento en la correcta utilización de los mismos hasta su mantenimiento.
- La implantación de los sistemas creados en un entorno multiusuario, donde aparezcan todos los agentes que se han creado en el diseño. La arquitectura de la aplicación generada permite el entorno multiusuario, al estar separada la aplicación de la capa de negocio y la de presentación y tener definidos 7 agentes con distinta funcionalidad. Por tanto, eso sería un reto de comunicación entre servidores de aplicación centralizados y de acceso al servidor vía cliente o navegador.

También se podría adaptar la aplicación al sector privado realizando pequeñas modificaciones. Los criterios de evaluación de empleados deben ser más parecidos entre distintas Administraciones Públicas que entre dos empresas privadas, pero los sistemas de información generados se podrían adaptar a los requerimientos de cada empresa privada en particular. Gracias al enfoque de desarrollo dirigido por modelos utilizado en este PFC, realizar estas adaptaciones tiene un impacto menor en relación a un entorno de desarrollo tradicional, ya que, los cambios se definen en el modelo de la aplicación y el código se genera automáticamente. Esto facilita la reutilización de los aspectos funcionales establecidos en el modelo, y reduce el costo asociado a aspectos de implementación que deben ser considerados al realizar cambios directamente en el código de la aplicación.

En definitiva, en este PFC se ha cumplido el objetivo de crear una herramienta de evaluación de recursos humanos para la Administración Pública que sirva de soporte para el modelo Excarsus. Sin embargo, consideramos que este trabajo no es el punto final en el desarrollo de un sistema adecuado para la gestión de recursos humanos, por el contrario, lo vemos como un sólido punto de partida para futuros proyectos dirigidos en esta línea. Es precisamente en esta dirección, donde las tecnologías de desarrollo dirigido por modelos se posicionan por sobre esquemas tradicionales de desarrollo, donde los artefactos de modelado no sólo apoyan el proceso de desarrollo, reduciendo los tiempos en la generación del software final y los errores de la programación manual. Los modelos también son un artefacto esencial en la transmisión del conocimiento, que apoyan la reutilización de la experiencia de proyectos previos y permiten una mejor discusión de las ideas y soluciones propuestas.

El uso de OO-Method ha facilitado en gran medida la tarea de crear un sistema de información a medida para un proyecto de gestión de recursos humanos que no tenía soporte informático. El alcance de este trabajo ha resultado ser muy superior al que se planteaba en entornos de desarrollo tradicional de aplicaciones. El producto final de este proyecto es un sistema de información perfectamente utilizable, que no presenta errores en el espacio de la solución.

Por este motivo, se considera perfectamente abordable realizar las ampliaciones expuestas mediante la construcción y adaptación de modelos conceptuales al esquema ya realizado. Este esquema es también útil como documentación de la funcionalidad de la aplicación final y para el entendimiento de la aplicación por parte del usuario del sistema. Con un usuario más involucrado en el uso del producto desarrollado en este proyecto, se potencia la demanda de nueva funcionalidad y nuevos desarrollos dirigidos por modelos.

En resumen, mediante una tecnología de desarrollo dirigido por modelos como son los sistemas Olivanova, se ha conseguido un producto software perfectamente funcional para la gestión de recursos humanos en Administraciones Públicas. Además, la tecnología empleada permite la reutilización de partes de este producto para distintos entornos y favorece la ampliación del alcance de los sistemas de información desarrollados a través de nuevos proyecto y tomando como base lo ya realizado en este Proyecto Final de Carrera.

REFERENCIAS

1. Wikipedia: Estado del bienestar. http://es.wikipedia.org/wiki/Estado_del_bienestar
2. Wikilearning: Administración de remuneración e incentivos
[http://www.wikilearning.com/curso_gratis/administracion de remuneracion e incentivos/15709](http://www.wikilearning.com/curso_gratis/administracion_de_remuneracion_e_incentivos/15709)
3. Sitio web de empresa ICESA. <http://www.icsarrhh.com>
4. Sitio web de empresa Deloitte. <http://www.deloitte.com/>
5. Documento de trabajo interno “©Excarpsus, Un modelo de organización, clasificación y valoración de puestos de trabajo en la administración Local”. Sepó Górriz, Joaquín – 2005
6. Sitio web www.salarios.net
7. Constitución Española de 1978. Título I. Capítulo II. Sección segunda. Artículo 35.
8. Wikipedia: Salario mínimo. http://es.wikipedia.org/wiki/Salario_mínimo
9. Convenio colectivo: <http://www.ugt.es/fta/consultasbasicas/02queesunconvenio.htm>
10. Mitecnologico: concepto de encuesta salarial
<http://www.mitecnologico.com/Main/EncuestaSalarialConcepto>
11. Observatorio salarial de la empresa ICESA:
<http://www.icsarrhh.com/index.php?op=ObservatorioSalarial>
12. Sitio web www.tusalario.es
13. Instituto Nacional de Estadística: Encuesta de Estructura Salarial.
<http://www.ine.es/jaxi/menu.do?type=pcaxis&path=/t22/p133&file=inebase&N=&L=>
14. Boletín Oficial del Estado número 89. Viernes, 13 de abril de 2007.
15. Ley 9/1987, de 12 de junio, de Órganos de Representación, Determinación de las Condiciones de Trabajo y Participación del Personal al Servicio de las Administraciones Públicas. Boletín Oficial del Estado número 144 de 17/06/1987.
16. Documento de trabajo interno del año 2006.
17. Pons Villagordo, Cristina; Terricabras Ibáñez, Ester: Valoració de llocs de treball amb el mètode de puntuació de factors. Implantació de l'aplicació ISOS en empreses. Proyecto Final de Carrera. Universidad Politécnica de Cataluña (2006).
http://bibliotecna.upc.es/pfc/mostrardades_PFC.asp?id=31027
18. Sepó, J., Martínez, P.: Propuesta de aplicación de Excarpsus en L'ajuntament de L'Elia. 2006

19. Pastor, O., Molina, J.C.: Model-Driven Architecture in Practice: A Software Production Environment Based on Conceptual Modeling. Springer, New York (2007)
20. Giachetti, G.: Integración de UML y Lenguajes de Modelado Específicos de Dominio mediante la Generación Automática de Perfiles UML. Tesis de Máster. Departamento de Sistemas Informáticos y Computación Unviersidad Politécnica de Valencia, Valencia (2008).
21. OMG: Object Management Group Web site, <http://www.omg.org/>
22. Marín, B.: Un Procedimiento de Medición de Tamaño Funcional para Modelos Conceptuales en entornos MDA. Tesis de Máster. Departamento de Sistemas Informáticos y Computación Unviersidad Politécnica de Valencia, Valencia (2008).
23. OMG: MDA Web site, <http://www.omg.org/mda/>
24. Pastor, O., Hayes, F., Bear, S.: OASIS: An Object-Oriented Specification Language. In: Proceedings of International Conference on Advanced Information Systems Engineering (CAiSE), pp. 348–363 (1992)
25. OO-Method: RETO Tool Web Site, <http://reto.dsic.upv.es/reto/>
26. CARE-Technologies: Web site, <http://www.care-t.com/>
27. Olivanova Modeler User Guide. (Noviembre 2009), disponible en <http://www.care-t.com/>
28. OMG, "UML 2.1.2 Superstructure Specification", 2007.
29. O'Really, "UML 2.0 in a nutshell", 2005.
30. Sparx Systems, "UML Tutorial", (http://www.sparxsystems.com/resources/tutorial/use_case_model.html), 04-07-2009.
31. Wikipedia, "Definición de Caso de Uso", (http://es.wikipedia.org/wiki/Caso_de_uso), 04-07-2009.
32. Jordi Fons Cors : Aplicación práctica de MDA con OLIVANOVA para el desarrollo de una aplicación de grupos de investigación. Proyecto Final de Carrera. Universidad Politécnica de Valencia (2007).
33. Esquema de arquitectura de 3 capas:
<http://pfc.alcuetas.com/wp-content/uploads/2009/08/ArquitecturadelSistema.jpg>
34. OLIVANOVA Transformation Engine. OLIVANOVA STAR Client User Guide en <http://www.care-t.com/technology>
35. IEEE 2009. IEEE 1044 Standard Classification for Software Anomalies.

Anexo I. Inventario del mapeo entre casos de uso y elementos conceptuales.

En este anexo se muestran 6 tablas con el mapeo entre los casos de uso de los sistemas de información SIEFAP y SIESMACRe y los elementos conceptuales empleados en el diseño.

La **¡Error! No se encuentra el origen de la referencia.** muestra los casos de uso del sistema SIESMACRe, la clase que se crea para su diseño, las relaciones con otras clases, los atributos derivados que se ven afectados en su valor cuando ocurre el caso de uso y las restricciones diseñadas para la clase. Si aparece el nombre de una clase entre corchetes, indica que el atributo al que precede pertenece a la clase entre corchetes y no a la del caso de uso.

Tabla 138. Casos de uso de SIESMACRe y clases, relaciones, atributos derivados y restricciones que participan en su diseño.

Casos de Uso	Clases	Relaciones	Atributos derivados	Restricciones
1. Alta de Categoría	Categoría	-	SalarioAnual	Salario mensual > 0
2. Baja de Categoría	Categoría	-	-	-
3. Modificación de salario de Categoría	Categoría	-	SalarioAnual	Salario mensual > 0
4. Actualización de salario de Categoría	Categoría	-	SalarioAnual	-
5. Consulta de salario de Categoría	Categoría	-	SalarioAnual	-
6. Alta de Complemento de Destino	ComplementoDestino	-	ValorAnual	ValorMensual > 0

7. Baja de Complemento de Destino	ComplementoDestino	-	-	-
8. Modificación de valor económico	ComplementoDestino	-	ValorAnual	ValorMensual > 0
9. Actualización de valor económico	ComplementoDestino	-	ValorAnual	-
10. Consulta de valor económico	ComplementoDestino	-	ValorAnual	-
11. Alta de Puesto de Trabajo	Puesto	-	-	Percentiles y media > 0 Orden de percentiles
12. Baja de Puesto de Trabajo	Puesto	-	-	-
13. Modificación de nombre de Puesto de Trabajo	Puesto	-	-	-
14. Modificación de la media del salario del Puesto de Trabajo	Puesto	-	-	Media > 0
15. Modificación de los percentiles del salario del Puesto de Trabajo	Puesto	-	-	Orden de percentiles
16. Actualización de la media del salario	Puesto	-	-	-

17. Actualización de los percentiles del salario	Puesto	-	-	-
18. Consulta de la media del salario	Puesto	-	-	-
19. Consulta de los percentiles del salario	Puesto	-	-	-
20. Alta de Comunidad Autónoma	ComunidadAutonoma	-	-	-
21. Baja de Comunidad Autónoma	ComunidadAutonoma	-	-	-
22. Modificación de nombre de Comunidad Autónoma	ComunidadAutonoma	-	-	-
23. Modificación del factor de Comunidad Autónoma	ComunidadAutonoma	-	-	-
24. Consulta del factor de Comunidad Autónoma	ComunidadAutonoma	-	-	-
25. Alta de intervalo de Número de Habitantes	Nhabitantes	-	-	Límite inferior > 0 Límite inferior <= Límite superior

26. Baja de intervalor de Número de Habitantes	Nhabitantes	-	-	-
27. Modificación de intervalo de Número de Habitantes	Nhabitantes	-	-	Límite inferior > 0 Límite inferior <= Límite superior
28. Modificación del factor de intervalo de Número de Habitantes	Nhabitantes	-	-	-
29. Consulta del factor de intervalo del Número de Habitantes	Nhabitantes	-	-	-
30. Alta de intervalo de Plantilla	Plantilla	-	-	Límite inferior > 0 Límite inferior <= Límite superior
31. Baja de intervalo de Plantilla	Plantilla	-	-	-
32. Modificación de intervalo de Plantilla	Plantilla	-	-	Límite inferior > 0 Límite inferior <= Límite superior
33. Modificación del factor de intervalo de Plantilla	Plantilla	-	-	-
34. Consulta del factor de intervalo de Plantilla	Plantilla	-	-	-

35. Obtención de informe	Calculadora	-	-	-
36. Cálculo del salario	Calculadora	Categoría ComplementoDestino	SalarioMensual SalarioAnual ValorCDMensual ValorCDAnual ValorCEMensual ValorCEAnual PagaExtra IngresosAnuales IngresosMensuales	-
37. Obtención de estadísticas	Calculadora	Puesto ComunidadAutonoma NHabitantes Plantilla	Media Percentil10 Percentil25 Percentil50 Percentil75 Percentil90	-

La Tabla 139 muestra los casos de uso del sistema SIESMACRe y el servicio que se crea para su diseño. Del servicio se muestran también los argumentos de entrada, los argumentos de salida y las precondiciones que se deben cumplir para que se ejecute. Si aparece el nombre de una clase entre corchetes, indica que el atributo al que precede pertenece a la clase entre corchetes y no a la del caso de uso.

Tabla 139. Casos de uso de SIESMACRe y servicios con sus entradas, salidas y precondiciones que participan en su diseño.

Casos de Uso	Servicios	Precondiciones	Entradas	Salidas
1. Alta de Categoría	crear_categoria	-	Tipo, Salario Mensual	Mensaje
2. Baja de Categoría	borrar_categoria	-	Categoría	Mensaje
3. Modificación de salario de Categoría	modificar_salario	-	Categoría, Salario Mensual	-
4. Actualización de salario de Categoría	actualizar_salario	- a_porcentaje > 0 - a_cambio = "A" or "R"	Categoría, porcentaje, cambio	-
5. Consulta de salario de Categoría	consultar_salarios	-	Categoría	Salario mensual y anual
6. Alta de Complemento de Destino	crear_complementoDestino	-	id_ComplementoDestino, ValorMensual	Mensaje
7. Baja de Complemento de Destino	borrar_complementoDestino	-	Complemento de Destino	Mensaje
8. Modificación de valor económico	modificar_valor	-	Complemento de Destino, Valor mensual	-

9. Actualización de valor económico	actualizar_valor	- a_porcentaje > 0 - a_cambio = "A" or "R"	Complemento de Destino, porcentaje, cambio	-
10. Consulta de valor económico	consultar_valores	-	Complemento de Destino	Valor mensual y anual
11. Alta de Puesto de Trabajo	crear_puesto	-	Nombre del puesto, media y percentiles	Mensaje
12. Baja de Puesto de Trabajo	borrar_puesto	-	Puesto de Trabajo	Mensaje
13. Modificación de nombre de Puesto de Trabajo	modificar_nombre	-	Puesto de Trabajo, nombre	Mensaje
14. Modificación de la media del salario del Puesto de Trabajo	modificar_media	-	Puesto de Trabajo, media	-
15. Modificación de los percentiles del salario del Puesto de Trabajo	AUMENTAR_PERCENTILES DISMINUIR_PERCENTILES	-	Puesto de Trabajo y percentiles	-
16. Actualización de la media del salario	actualizar_media	- a_porcentaje > 0 - a_cambio = "A" or "R"	Puesto, porcentaje, cambio	-
17. Actualización de los percentiles del salario	actualizar_percentiles	- a_porcentaje > 0 - a_cambio = "A" or "R"	Puesto, porcentaje, cambio	-
18. Consulta de la media del salario	consultar_media	-	Puesto	Media

19. Consulta de los percentiles del salario	consultar_percentiles	-	Puesto	percentil10, percentil25, percentil50, percentil75 y percentil90
20. Alta de Comunidad Autónoma	ALTA_COMUNIDADFACTOR	-	Nombre de la Comunidad, factor	Mensaje
21. Baja de Comunidad Autónoma	borrar_comunidad	-	Comunidad	Mensaje
22. Modificación de nombre de Comunidad Autónoma	modificar_nombre	-	Comunidad, nombre	Mensaje
23. Modificación del factor de Comunidad Autónoma	modificar_factorCCAA	-	Comunidad, factor	-
24. Consulta del factor de Comunidad Autónoma	consultar_factorCCAA	-	Comunidad	FactorCCAA
25. Alta de intervalo de Número de Habitantes	ALTA_HABITANTESFACTOR	-	Límite inferior, límite superior, factor	Mensaje
26. Baja de intervalo de Número de Habitantes	borrar_Nhabitantes	-	Intervalo habitantes	Mensaje

27. Modificación de intervalo de Número de Habitantes	MODIFICAR_LIMITESNH	Límite inferior entrada <= Límite superior de entrada	Intervalo, límite inferior, límite superior	Mensaje
28. Modificación del factor de intervalo de Número de Habitantes	modificar_factorNH	-	Intervalo, factor	-
29. Consulta del factor de intervalo del Número de Habitantes	consultar_factorNH	-	Intervalo	factorNH
30. Alta de intervalo de Plantilla	ALTA_PLANTILLAFACTOR	-	Límite inferior, límite superior, factor	Mensaje
31. Baja de intervalo de Plantilla	borrar_plantilla	-	Intervalo plantilla	Mensaje
32. Modificación de intervalo de Plantilla	MODIFICAR_LIMITESPLANT	Límite inferior entrada <= Límite superior de entrada	Intervalo, límite inferior, límite superior	Mensaje
33. Modificación del factor de intervalo de Plantilla	modificar_factorPlant	-	Intervalo, factor	-
34. Consulta del factor de intervalo de Plantilla	consultar_factorPlant	-	Intervalo	factorPlantilla
35. Obtención de informe	crear_calculadora	-	- Categoría - Complemento de destino - Complemento específico - Puesto de Trabajo - Comunidad Autónoma	- Valor anual y mensual de la Categoría - Valor anual y mensual del Complemento de destino - Valor anual y mensual del

			<ul style="list-style-type: none"> - Intervalo de número de habitantes - Intervalo de Plantilla de empleados 	<p>Complemento específico</p> <ul style="list-style-type: none"> - Valor anual de la Paga Extra - Ingresos totales mensuales y anuales
				<ul style="list-style-type: none"> - Media del puesto de trabajo - Percentiles 10, 25, 50, 75 y 90 del puesto de trabajo - Posición respecto a media y percentiles
36. Cálculo del salario	crear_calculadora	-	<ul style="list-style-type: none"> - Categoría - Complemento de destino - Complemento específico 	-
37. Obtención de estadísticas	crear_calculadora	-	<ul style="list-style-type: none"> - Puesto de trabajo - Comunidad Autónoma - Número de habitantes - Plantilla de empleados 	-

La Tabla 140 muestra los casos de uso del sistema SIESMACRe, los agentes que los pueden lanzar, las instancias del modelo de presentación que participan en el diseño del caso de uso, y si la Unidad de Interacción de Servicio (SIU, por sus siglas en inglés) se puede ejecutar directamente desde el Árbol de Jerarquía de Acciones (HAT, por sus siglas en inglés), el nodo en que se encuentra.

Tabla 140. Casos de uso de SIESMACRe, agentes que los pueden lanzar y elementos del modelo de presentación que participan en su diseño.

Casos de Uso	Presentación	Agente	HAT
1. Alta de Categoría	IIU Categoría, PIU Categoría, SIU crear_categoria	AdministradorSalarios	SIESMACRe.Categoría
2. Baja de Categoría	IIU Categoría, PIU Categoría, SIU borrar_categoria	AdministradorSalarios	-
3. Modificación de salario de Categoría	IIU Categoría, PIU Categoría, SIU modificar_salario	AdministradorSalarios	-
4. Actualización de salario de Categoría	IIU Categoría, PIU Categoría, SIU actualizar_salario	AdministradorSalarios	-
5. Consulta de salario de Categoría	IIU Categoría, PIU Categoría, SIU consultar_salarios	AdministradorSalarios	SIESMACRe.Categoría
6. Alta de Complemento de Destino	IIU ComplementoDestino, PIU ComplementoDestino, SIU crear_complementoDestino	AdministradorSalarios	SIESMACRe.ComplementoDestino
7. Baja de Complemento de Destino	IIU ComplementoDestino, PIU ComplementoDestino, SIU borrar_complementoDestino	AdministradorSalarios	-
8. Modificación de valor económico	IIU ComplementoDestino, PIU ComplementoDestino, SIU modificar_valor	AdministradorSalarios	-

9. Actualización de valor económico	IIU ComplementoDestino, PIU ComplementoDestino, SIU actualizar_valor	AdministradorSalarios	-
10. Consulta de valor económico	IIU ComplementoDestino, PIU ComplementoDestino, SIU consultar_valores	AdministradorSalarios	SIESMACRe.ComplementoDestino
11. Alta de Puesto de Trabajo	IIU Puesto, PIU Puesto, SIU crear_puesto	AdministradorSalarios	SIESMACRe.Puesto
12. Baja de Puesto de Trabajo	IIU Puesto, PIU Puesto, SIU borrar_puesto	AdministradorSalarios	-
13. Modificación de nombre de Puesto de Trabajo	IIU Puesto, PIU Puesto, SIU modificar_nombre	AdministradorSalarios	-
14. Modificación de la media del salario del Puesto de Trabajo	IIU Puesto, PIU Puesto, SIU modificar_media	AdministradorSalarios	-
15. Modificación de los percentiles del salario del Puesto de Trabajo	IIU Puesto, PIU Puesto, SIU AUMENTAR_PERCENTILES, SIU DISMINUIR_PERCENTILES	AdministradorSalarios	-
16. Actualización de la media del salario	IIU Puesto, PIU Puesto, SIU actualizar_media	AdministradorSalarios	-
17. Actualización de los percentiles del salario	IIU Puesto, PIU Puesto, SIU actualizar_percentiles	AdministradorSalarios	-
18. Consulta de la media del salario	IIU Puesto, PIU Puesto, SIU consultar_media	AdministradorSalarios	SIESMACRe.Puesto
19. Consulta de los percentiles del salario	IIU Puesto, PIU Puesto, SIU consultar_percentiles	AdministradorSalarios	SIESMACRe.Puesto

20. Alta de Comunidad Autónoma	IIU ComunidadAutonoma, PIU ComunidadAutonoma, SIU ALTA_COMUNIDADFACTOR	AdministradorSalarios	SIESMACRe.ComunidadAutonoma
21. Baja de Comunidad Autónoma	IIU ComunidadAutonoma, PIU ComunidadAutonoma, SIU borrar_comunidad	AdministradorSalarios	-
22. Modificación de nombre de Comunidad Autónoma	IIU ComunidadAutonoma, PIU ComunidadAutonoma, SIU modificar_nombre	AdministradorSalarios	-
23. Modificación del factor de Comunidad Autónoma	IIU ComunidadAutonoma, PIU ComunidadAutonoma, SIU modificar_factorCCAA	AdministradorSalarios	-
24. Consulta del factor de Comunidad Autónoma	IIU ComunidadAutonoma, PIU ComunidadAutonoma, SIU consultar_factorCCAA	AdministradorSalarios	SIESMACRe.ComunidadAutonoma
25. Alta de intervalo de Número de Habitantes	IIU NHabitantes, PIU Nhabitantes, SIU ALTA_HABITANTESFACTOR	AdministradorSalarios	SIESMACRe.Nhabitantes
26. Baja de intervalo de Número de Habitantes	IIU NHabitantes, PIU Nhabitantes, SIU borrar_Nhabitantes	AdministradorSalarios	-
27. Modificación de intervalo de Número de Habitantes	IIU NHabitantes, PIU Nhabitantes, SIU MODIFICAR_LIMITESNH	AdministradorSalarios	-

28. Modificación del factor de intervalo de Número de Habitantes	IIU NHabitantes, PIU NHabitantes, SIU modificar_factorNH	AdministradorSalarios	-
29. Consulta del factor de intervalo del Número de Habitantes	IIU NHabitantes, PIU NHabitantes, SIU consultar_factosNH	AdministradorSalarios	SIESMACRe.Nhabitantes
30. Alta de intervalo de Plantilla	IIU Plantilla, PIU Plantilla, SIU ALTA_PLANTILLAFACTOR	AdministradorSalarios	SIESMACRe.Plantilla
31. Baja de intervalo de Plantilla	IIU Plantilla, PIU Plantilla, SIU borrar_plantilla	AdministradorSalarios	-
32. Modificación de intervalo de Plantilla	IIU Plantilla, PIU Plantilla, SIU MODIFICAR_LIMITESPLANT	AdministradorSalarios	-
33. Modificación del factor de intervalo de Plantilla	IIU Plantilla, PIU Plantilla, SIU modificar_factorPlant	AdministradorSalarios	-
34. Consulta del factor de intervalo de Plantilla	IIU Plantilla, PIU Plantilla, SIU consultar_factorPlant	AdministradorSalarios	SIESMACRe.Plantilla
35. Obtención de informe	IIU Calculadora, PIU Calculadora, SIU crear_calculadora	Usuario_Anonimo AdministradorSalarios	SIESMACRe.Calculadora

<p>36. Cálculo del salario</p>	<p>IIU Calculadora, PIU Calculadora, SIU crear_calculadora</p>	<p>-</p>	<p>SIESMACRe.Calculadora</p>
<p>37. Obtención de estadísticas</p>	<p>IIU Calculadora, PIU Calculadora, SIU crear_calculadora</p>	<p>-</p>	<p>SIESMACRe.Calculadora</p>

La Tabla 141 muestra los casos de uso del sistema SIEFAP, la clase que se crea para su diseño, las relaciones con otras clases, los atributos derivados que se ven afectados en su valor cuando ocurre el caso de uso y las restricciones diseñadas para la clase. Si aparece el nombre de una clase entre corchetes, indica que el atributo al que precede pertenece a la clase entre corchetes y no a la del caso de uso.

Tabla 141. Casos de uso de SIEFAP y clases, relaciones, atributos derivados y restricciones que participan en su diseño.

Casos de Uso	Clases	Relaciones	Atributos derivados	Restricciones
<p>1. Alta de Empleado</p>	<p>Empleado</p>	<p>Departamento</p>	<p>Valor por defecto de Estado</p>	<p>-</p>
<p>2. Activar Empleado</p>	<p>Empleado</p>	<p>Departamento</p>	<p>-</p>	<p>-</p>

3. Desactivar Empleado	Empleado	-	Estado	Estado del responsable = "A"
4. Baja de Empleado	Empleado	Valoracion, Valoracion_conocimiento, Cuestionario	-	-
5. Modificación de detalles de Empleado	Empleado	-	-	Salario >= 0
6. Convertir Empleado en Supervisor	Empleado	Empleado_supervisor	Valor por defecto de fecha	-
6. Convertir Empleado en Supervisor	Empleado_supervisor	-	-	-
7. Abandonar Rol de Supervisor	Empleado_supervisor	-	-	-
8. Convertir Empleado en Responsable	Empleado	ResponsableAdministración	-	Estado = "A"
9. Abandonar Rol de Responsable	ResponsableAdministración	-	-	-
10. Crear usuario	Usuario	-	-	-
11. Borrar Usuario	Usuario	-	-	-
12. Convertir Usuario en Administrador (SIEFAP)	Usuario	AdministradorValoracion	-	-
12. Convertir Usuario en Administrador (SIESMACRE)	Usuario	AdministradorSalarios	-	-

13. Abandonar Rol de Administrador (SIEFAP)	AdministradorValoracion	Usuario	-	-
13. Abandonar Rol de Administrador (SIESMACRe)	AdministradorSalarios	Usuario	-	-
14. Convertir Usuario en Editor	Usuario	Usuario_Editor	-	-
15. Abandonar Rol de Editor	Usuario_Editor	Usuario	-	-
16. Alta de Departamento	Departamento	Administracion	-	-
17. Baja de Departamento	Departamento	-	-	-
18. Modificación del nombre de Departamento	Departamento	-	-	-
19. Asignar Presupuesto a Departamento	Departamento	-	-	PresupuestoAsignado >= 0
20. Consulta de Presupuesto de Departamento	Departamento	Empleado	PresupuestoDispuesto	-
21. Consulta de Empleados de Departamento	Departamento	Empleado	Nempleados	-
22. Alta de Administración	Administracion	-	-	-
23. Baja de Administración	Administracion	Departamento -> Empleado	Nempleados	-
24. Modificación del nombre de Administración	Administracion	-	-	-
25. Asignar Presupuesto a Administración	Administracion	-	-	PresupuestoAsignado >= 0

26. Consulta de Presupuesto de la Admistración	Administracion	Departamento	PresupuestoDispuesto	-
27. Consulta de empleados de la Administración	Administracion	Departamento	Nempleados	-
28. Crear Conocimiento	Conocimiento	Administracion	-	-
29. Modificar Conocimiento	Conocimiento	SuperConocimiento	-	-
30. Eliminar Conocimiento	Conocimiento	-	-	-
31. Crear Subconocimiento	Conocimiento	SuperConocimiento, Administracion	-	-
32. Crear Pregunta	Pregunta	Conocimiento, Respuesta	Estado, PuntuacionMaxima	[Respuesta] Puntuacion >= 0
33. Modificar Pregunta	Pregunta	Respuesta	-	[Respuesta] Puntuacion >= 0
34. Crear Respuesta	Respuesta	Pregunta	[Pregunta] Nrespuestas y PuntuacionMaxima	-
35. Modificar Respuesta	Respuesta	Pregunta	[Pregunta] Nrespuestas y PuntuacionMaxima	Puntuacion >= 0

36. Eliminar Respuesta	Respuesta	Pregunta	[Pregunta] Nrespuestas y PuntuacionMaxima	-
37. Asignar Pregunta a Conocimiento	Pregunta, Conocimiento	Conocimiento	[Pregunta] Nconocimientos	-
38. Desasignar Pregunta de Conocimiento.	Conocimiento	Pregunta	[Pregunta] Nconocimientos	-
39. Crear Plantilla de Cuestionario	Plantilla_Cuestionario	Conocimiento	Estado	-
40. Modificar Plantilla de Cuestionario	Plantilla_Cuestionario	Pregunta_Plantilla	-	-
41. Asignar Pregunta a Plantilla de Cuestionario	Pregunta_Plantilla	Plantilla_Cuestionario, Pregunta	- [Plantilla de Cuestionario] Npreguntas - [Plantilla de Cuestionario] PuntuacionMaxima	Valor >= 0
42. Desasignar Pregunta de Plantilla de Cuestionario	Pregunta_Plantilla	-	- [Plantilla de Cuestionario] Npreguntas - [Plantilla de Cuestionario] PuntuacionMaxima	-
43. Cambiar valor de Pregunta	Pregunta_Plantilla	-	- [Plantilla de Cuestionario] PuntuacionMaxima	Valor >= 0

44. Asignar Plantilla de Cuestionario a Conocimiento	Plantilla_Cuestionario	Conocimiento	-	-
45. Crear Evaluación	Valoracion	Empleado	Valor por defecto de fecha de inicio Valor por defecto de estado	- Puntos > Suma de puntos de valoraciones de conocimiento
46. Crear Evaluación de Conocimiento	Valoracion_Conocimiento	Empleado, Valoracion, Conocimiento	PuntosConocimiento	- $0 \leq \text{PorcentajePuntos} \leq 100$ - Suma de PuntosConocimiento $\leq [\text{Valoracion}] \text{ Puntos}$
47. Editar Evaluación	Valoracion	-	-	-
48. Cambiar estado	Valoracion	-	-	- Puntos = Suma de puntos de valoraciones de Superconocimientos
49. Editar Evaluación de Conocimiento	Valoracion_Conocimiento	Valoracion	PuntosConocimiento	- $0 \leq \text{PorcentajePuntos} \leq 100$ - $[\text{Valoración}] \text{ Puntos} \geq \text{Suma de puntos de valoraciones de Conocimiento}$
50. Eliminar Evaluación	Valoracion	Valoracion_Conocimiento	-	-
51. Eliminar Evaluación de Conocimiento	Valoracion_Conocimiento	-	-	-

52. Crear Cuestionario	Cuestionario	Empleado, EmpleadoSupervisor, PlantillaCuestionario, PesoCuestionario	Valor por defecto de estado y de FechaCreacion	- Puntuación >= 0 - [PesoCuestionario] Peso > 0 - [Cuestionario] Empleado = [Valoración] Empleado
53. Asignar Cuestionario a Evaluación de Conocimiento	PesoCuestionario	Cuestionario, ValoracionConocimiento	[Valoracion_Conocimiento] NCuestionariosTotales	Peso >= 0
54. Desasignar Cuestionario de Evaluación de Conocimiento	PesoCuestionario	Cuestionario, ValoracionConocimiento	[Valoracion_Conocimiento] NCuestionariosTotales	-
55. Editar Cuestionario (Cambiar peso)	PesoCuestionario	-	[Valoracion_Conocimiento] PorcentajePuntuacionObtenida	Peso >= 0
56. Dejar Cuestionario Preparado	Cuestionario	-	Estado = "P"	-
57. Asignar Supervisor a Cuestionario	Cuestionario, EmpleadoSupervisor	-	-	-
58. Eliminar Cuestionario	Cuestionario	-	[Valoracion_Conocimiento] NCuestionariosTotales	-
59. Responder Cuestionario	Cuestionario			
60. Comenzar Cuestionario	Cuestionario	-	Estado = "R" FechaComienzo	-

61.1. Responder Pregunta de Cuestionario	Respuesta_Empleado	Cuestionario, Pregunta_Plantilla, Respuesta	PuntuacionObtenida	Respuestas de Cuestionario <= Número de preguntas
61.2. Responder Pregunta de Cuestionario	Respuesta_Empleado	Cuestionario, Pregunta_Plantilla, Respuesta	PuntuacionObtenida	Respuestas de Cuestionario <= Número de preguntas
61.3. Responder Pregunta de Cuestionario	Respuesta_Empleado	-	PuntuacionObtenida	-
62. Terminar Cuestionario	Cuestionario	-	Estado = "F"	-
63. Consultar Detalles Cuestionario	Cuestionario	Empleado, EmpleadoSupervisor, PlantillaCuestionario, PesoCuestionario	-	-
64. Consultar Detalles de Evaluación	Valoracion	Empleado	-	-

65. Consultar Detalles de Evaluación del Conocimiento	Valoracion_Conocimiento	Valoracion, Empleado	-	-
66. Validar Cuestionario	Cuestionario	EmpleadoSupervisor	- PorcentajeObtenido - Estado = "V" - Fecha de corrección - Fecha de validación	-
67. Comentar Respuesta de Empleado	Respuesta_Empleado	Cuestionario	- Estado = "R" - Fecha de corrección	-
68. Solicitar Corrección Cuestionario	Cuestionario	EmpleadoSupervisor	-	-

La Tabla 142 muestra los casos de uso del sistema SIEFAP y el servicio que se crea para su diseño. Del servicio se muestran también los argumentos de entrada, los argumentos de salida y las precondiciones que se deben cumplir para que se ejecute. Si aparece el nombre de una clase entre corchetes, indica que el atributo al que precede pertenece a la clase entre corchetes y no a la del caso de uso.

Tabla 142. Casos de uso de SIEFAP y servicios con sus entradas, salidas y precondiciones que participan en su diseño.

Casos de Uso	Servicios	Precondiciones	Entradas	Salidas
--------------	-----------	----------------	----------	---------

1. Alta de Empleado	create_instance	-	NIF, Nombre, Apellidos, Fecha de nacimiento, Sexo	Mensaje
2. Activar Empleado	ACTIVAR_EMPLEADO	Estado <> "A"	Empleado, Departamento	Mensaje
3. Desactivar Empleado	desactivar_empleado	Estado <> "I"	Empleado	Mensaje
4. Baja de Empleado	BORRAR_EMPLEADO	-	Empleado	Mensaje
5. Modificación de detalles de Empleado	edit_instance	Estado = "A"	Empleado, Nombre, Apellidos, Fecha de nacimiento, Salario, Sexo	Mensaje
6. Convertir Empleado en Supervisor	convertir_supervisor	Estado = "A"	Empleado	Mensaje
6. Convertir Empleado en Supervisor	reconvertir_supervisor	Estado = "A" FechaFin <> NULL	Empleado_supervisor	Mensaje
7. Abandonar Rol de Supervisor	abandonar_supervisor	Estado = "A" FechaFin = NULL	Empleado_supervisor	Mensaje
8. Convertir Empleado en Responsable	convertir_responsable	Estado = "A"	Empleado	Mensaje
9. Abandonar Rol de Responsable	abandonar_responsable	-	ResponsableAdministracion	Mensaje
10. Crear usuario	crear_usuario	-	Nombre_usuario, password	Mensaje
11. Borrar Usuario	borrar_usuario	-	Usuario	Mensaje

12. Convertir Usuario en Administrador (SIEFAP)	convertir_administrador	-	Usuario	Mensaje
12. Convertir Usuario en Administrador (SIESMACRE)	convertir_AdministradorSalarios	-	Usuario	Mensaje
13. Abandonar Rol de Administrador (SIEFAP)	abandonar_administrador	-	AdministradorValoracion	Mensaje
13. Abandonar Rol de Administrador (SIESMACRe)	abandonar_AdmSalarios	-	AdministradorValoracion	Mensaje
14. Convertir Usuario en Editor	convertir_editor	-	Usuario	Mensaje
15. Abandonar Rol de Editor	abandonar_editor	-	Usuario_Editor	Mensaje
16. Alta de Departamento	create_instance	-	Administracion, Nombre	Mensaje
17. Baja de Departamento	delete_instance	Número de empleados activos = 0	Departamento	Mensaje
18. Modificación del nombre de Departamento	modificar_nombre	-	Departamento, nombre	Mensaje
19. Asignar Presupuesto a Departamento	asignar_presupuesto	-	Departamento, presupuesto	Mensaje
20. Consulta de Presupuesto de Departamento	consulta_presupuesto	-	Departamento	PresupuestoAsignado, PresupuestoDispuesto
21. Consulta de Empleados de Departamento	consultar_empleados	-	Departamento	Número de empleados
22. Alta de Administración	create_instance	-	Nombre	Mensaje
23. Baja de Administración	BORRAR_ADMINISTRACION	-	Administacion	Mensaje, Número de empleados

24. Modificación del nombre de Administración	modificar_nombre	-	Administracion, Nombre	Mensaje
25. Asignar Presupuesto a Administración	asignar_presupuesto	-	Administracion, Presupuesto	Mensaje
26. Consulta de Presupuesto de la Administración	consulta_presupuesto	-	Administracion	PresupuestoAsignado, PresupuestoDispuesto
27. Consulta de empleados de la Administración	consultar_empleados	-	Administracion	Número de empleados
28. Crear Conocimiento	crear_instancia	-	Nombre, Descripción, Administracion	Mensaje
29. Modificar Conocimiento	edit_instance	-	Conocimiento, Nombre, Descripción	Mensaje
30. Eliminar Conocimiento	delete_instance	- No tiene Preguntas asociadas - No tiene Plantilla de Cuestionario - No tiene Subconocimientos - No tiene Valoraciones de Conocimiento	Conocimiento	Mensaje
31. Crear Subconocimiento	crear_instancia	- La administración del SuperConocimiento debe ser la misma que la de entrada	SuperConocimiento, Nombre, Descripción, Administracion	Mensaje
32. Crear Pregunta	CREA_PREGUNTA	-	- Texto, Conocimiento, - Texto y puntuación de	Mensaje

			Respuestas	
33. Modificar Pregunta	edit_instance	Estado = "N"	Pregunta, texto	-
34. Crear Respuesta	create_instance	[Pregunta] Estado = "N"	Texto, puntuación, pregunta	-
35. Modificar Respuesta	edit_instance	[Pregunta] Estado = "N"	Pregunta, Respuesta, texto, puntuación	-
36. Eliminar Respuesta	delete_instance	[Pregunta] Estado = "N" La Respuesta no debe ser la última asociada a la Pregunta	Respuesta	-
37. Asignar Pregunta a Conocimiento	InsPregunta	-	Pregunta, conocimiento	Mensaje
38. Desasignar Pregunta de Conocimiento.	DelPregunta	- No se pueden desasignar Preguntas de Conocimiento que estén siendo utilizadas en Plantillas de Cuestionario de ese Conocimiento. - No se puede desasignar Preguntas que sólo están relacionadas con un Conocimiento	Conocimiento, Pregunta	-
39. Crear Plantilla de Cuestionario	create_instance	-	Conocimiento, Nombre, Descripción	Mensaje

40. Modificar Plantilla de Cuestionario	edit_instance	- [Plantilla de Cuestionario] Estado = "N"	Plantilla_Cuestionario, Nombre, Descripcion	-
41. Asignar Pregunta a Plantilla de Cuestionario	create_instance	- [Plantilla de Cuestionario] Estado = "N" - Conocimiento de Plantilla_Cuestionario = Conocimiento de Pregunta	Plantilla_Cuestionario, Pregunta, Valor	-
42. Desasignar Pregunta de Plantilla de Cuestionario	delete_instance	- [Plantilla de Cuestionario] Estado = "N"	Pregunta_Plantilla	-
43. Cambiar valor de Pregunta	edit_instance	- [Plantilla de Cuestionario] Estado = "N"	Pregunta_Plantilla, Valor	-
44. Asignar Plantilla de Cuestionario a Conocimiento	InsPlatilla_cue	La Plantilla de Cuestionario no debe tener Preguntas	Plantilla_Cuestionario, Conocimiento	-
45. Crear Evaluación	create_instance	-	Empleado, descripción, puntos	Mensaje
46. Crear Evaluación de Conocimiento	create_instance	- Si Valoración_Conocimiento es de Subconocimiento, el Superconocimiento debe estar asociado - Estado Valoracion = "E" - Empleado.Departamento.Administracion = Conocimiento.Administrac	Valoracion, Empleado, Conocimiento, Porcentaje de puntos	Mensaje

		ion - Empleado = Valoracion.Empleado		
47. Editar Evaluación	edit_instance	- Estado Valoración = "E"	Valoracion, Puntos, Descripción	-
48. Cambiar estado	cambiar_estado	- Si Estado = "E", Puntos = Suma de puntos valoraciones de conocimiento - Estado = "E" o "A"	Valoracion	-
49. Editar Evaluación de Conocimiento	edit_instance	Estado Valoración = "E"	Valoracion_Conocimiento, Porcentaje de puntos	-
50. Eliminar Evaluación	BORRAR_EVALUACION	Estado = "E"	Valoracion	Mensaje
51. Eliminar Evaluación de Conocimiento	BORRAR_VCONOCIMIENTO	- No puede haber valoraciones de subconocimiento - Estado de la valoración = "E"	Valoracion_Conocimiento	Mensaje
52. Crear Cuestionario	CREAR_CUESTIONARIO	EmpleadoSupervisor.Fech aFin = NULL	PlantillaCuestionario, Valoracion_Conocimiento, Empleado, EmpleadoSupervisor, Peso	-

53. Asignar Cuestionario a Evaluación de Conocimiento	create_instance	<ul style="list-style-type: none"> - Ev. de conocimiento debe ser de Conocimiento sin Subconocimientos. - Estado de la Evaluación = "E" - El Conocimiento del Cuestionario y Ev. de Conocimiento debe ser el mismo - El Empleado del Cuestionario y Ev. de Conocimiento debe ser el mismo 	Cuestionario, Valoracion_Conocimiento, Peso	-
54. Desasignar Cuestionario de Evaluación de Conocimiento	delete_instance	Estado de la Evaluación = "E"	Valoracion_Conocimiento, Cuestionario	-
55. Editar Cuestionario (Cambiar peso)	edit_instance	- Estado de la Evaluación = "E"	PesoCuestionario, Peso	Mensaje
56. Dejar Cuestionario Preparado	cambiar_estado	<ul style="list-style-type: none"> - Debe existir el Empleado Supervisor - Estado del cuestionario = "E" 	Cuestionario,	Mensaje
57. Asignar Supervisor a Cuestionario	ASIGNAR_SUPERVISOR	<ul style="list-style-type: none"> - [Cuestionario] Estado <> "V" - Empleado Supervisor.FechaFin = NULL 	EmpleadoSupervisor, Cuestionario	Mensaje

58. Eliminar Cuestionario	BORRAR_CUESTIONARIO	- Las valoraciones en que participa el Cuestionario deben estar en estado de Edición	Cuestionario	Mensaje
59. Responder Cuestionario				
60. Comenzar Cuestionario	empezar_cuestionario	- Estado del cuestionario = "P" o "R"	Cuestionario	-
61.1. Responder Pregunta de Cuestionario	responder	- Estado del cuestionario = "R"	Cuestionario, Pregunta_Plantilla, Respuesta	-
61.2. Responder Pregunta de Cuestionario	elige_respuesta	- Estado del cuestionario = "R"	Respuesta_Empleado, Respuesta	-
61.3. Responder Pregunta de Cuestionario	borrar_respuesta	- Estado del cuestionario = "R"	Respuesta_Empleado	-
62. Terminar Cuestionario	terminar_cuestionario	- Número de Respuestas = Número de Preguntas - Estado del cuestionario = "R"	Cuestionario	Mensaje
63. Consultar Detalles Cuestionario	-	-	Cuestionario, Valoracion_Conocimiento	- id, Version y Nombre de Plantilla de Cuestionario - NIF, Nombre y Apellidos de Empleado - NIF, Nombre y Apellidos de

				Supervisor - Estado, peso y Fechas de Comienzo, Fin, Corrección y validación de Cuestionario
				- Número de preguntas y puntuación máxima de Plantilla de Cuestionario - Número de respuestas, Porcentaje obtenido y Puntuación del Cuestionario
64. Consultar Detalles de Evaluación	-	-	Valoración	- NIF, Nombre y Apellidos del Empleado - Estado, Fecha de Creación, Fecha de Fin, Puntos totales y Puntos obtenidos de la Valoración

<p>65. Consultar Detalles de Evaluación del Conocimiento</p>	<p>-</p>	<p>-</p>	<p>Valoracion_Conocimiento</p>	<p>- NIF, Nombre y Apellidos del Empleado - id y Puntos de Valoración - id, Nombre y Descripción de Conocimiento - Porcentaje de Puntuación, PuntosConocimiento, Nivel y Puntos Obtenidos de Valoracion_Conocimiento - Cuestionarios totales, pendientes, finalizados y validados de Valoracion_Conocimiento</p>
<p>66. Validar Cuestionario</p>	<p>validar_cuestionario</p>	<p>- Estado de Cuestionario = "F"</p>	<p>Cuestionario</p>	<p>Mensaje</p>
<p>67. Comentar Respuesta de Empleado</p>	<p>motivo_correccion</p>	<p>- Estado de Cuestionario = "F"</p>	<p>Respuesta_Empleado, corrección</p>	<p>-</p>
<p>68. Solicitar Corrección Cuestionario</p>	<p>corregir_cuestionario</p>	<p>- Estado de Cuestionario = "F"</p>	<p>Cuestionario</p>	<p>Mensaje</p>

La Tabla 143 muestra los casos de uso del sistema SIEFAP, los agentes que los pueden lanzar, las instancias del modelo de presentación que participan en el diseño del caso de uso, y si la Unidad de Interacción de Servicio (SIU, por sus siglas en inglés) se puede ejecutar directamente desde el Árbol de Jerarquía de Acciones (HAT, por sus siglas en inglés), el nodo en que se encuentra.

Tabla 143. Casos de uso de SIEFAP, agentes que los pueden lanzar y elementos del modelo de presentación que participan en su diseño.

Casos de Uso	Presentación	Agente	HAT
1. Alta de Empleado	IIU Empleado, PIU Empleado, IIU Departamento, PIU Departamento, SIU create_instance	ResponsableAdministración, AdministradorValoracion	SIEFAP.Empleado
2. Activar Empleado	IIU Empleado, PIU Empleado, SIU ACTIVAR_EMPLEADO	ResponsableAdministración, AdministradorValoracion	-
3. Desactivar Empleado	IIU Empleado, PIU Empleado, IIU desactivar_empleado	ResponsableAdministración, AdministradorValoracion	-
4. Baja de Empleado	IIU Empleado, PIU Empleado, SIU BORRAR_EMPLEADO	ResponsableAdministración, AdministradorValoracion	-
5. Modificación de detalles de Empleado	IIU Empleado, PIU Empleado, IIU edit_instance	ResponsableAdministración, AdministradorValoracion	-
6. Convertir Empleado en Supervisor	IIU Empleado, PIU Empleado, IIU convertir_supervisor	ResponsableAdministración, AdministradorValoracion	-
6. Convertir Empleado en Supervisor	IIU EmpleadoSupervisor, PIU EmpleadoSupervisor, SIU reconvertir_supervisor	ResponsableAdministración, AdministradorValoracion	-
7. Abandonar Rol de Supervisor	IIU EmpleadoSupervisor, PIU EmpleadoSupervisor, SIU abandonar_supervisor	ResponsableAdministración, AdministradorValoracion	-

8. Convertir Empleado en Responsable	IIU Empleado, PIU Empleado, SIU convertir_responsable	ResponsableAdministración, AdministradorValoracion	-
9. Abandonar Rol de Responsable	IIU ResponsableAdministracion, PIU ResponsableAdministracion, SIU abandonar_responsabe	ResponsableAdministración, AdministradorValoracion	-
10. Crear usuario	IIU Usuario, PIU Usuario, SIU crear_usuario	AdministradorValoracion, AdministradorSalarios	Gestión de usuarios.Usuario
11. Borrar Usuario	IIU Usuario, PIU Usuario, IIU Usuario_Editor, PIU Usuario_Editor, IIU Usuario_Administrador, PIU AdministradorValoracion, IIU AdministradorSalarios, PIU AdministradorSalarios, SIU borrar_usuario	AdministradorValoracion, AdministradorSalarios, Usuario	-
12. Convertir Usuario en Administrador (SIEFAP)	IIU Usuario, PIU Usuario, SIU convertir_administrador	AdministradorValoracion	-
12. Convertir Usuario en Administrador (SIESMACRE)	IIU Usuario, PIU Usuario, SIU convertir_AdministradorSalarios	AdministradorValoracion	-
13. Abandonar Rol de Administrador (SIEFAP)	IIU Usuario_Administrador, PIU AdministradorValoracion, SIU abandonar_administrador	AdministradorValoracion	Gestión de usuarios.AdministracionValoracion
13. Abandonar Rol de Administrador (SIESMACRe)	IIU AdministradorSalarios, PIU AdministradorSalarios, SIU abandonar_AdmSalarios	AdministradorValoracion	Gestión de usuarios.AdministracionSalarios
14. Convertir Usuario en Editor	IIU Usuario, PIU Usuario, SIU convertir_editor	AdministradorValoracion	-
15. Abandonar Rol de Editor	IIU Usuario_Editor, PIU Usuario_Editor, SIU abandonar_editor	AdministradorValoracion	Gestión de usuarios.Usuario_Editor

16. Alta de Departamento	IIU Departamento, PIU Departamento, SIU create_instance	ResponsableAdministración, AdministradorValoracion	SIEFAP.Administracion.Departamento
17. Baja de Departamento	IIU Departamento, PIU Departamento, SIU delete_instance	ResponsableAdministración, AdministradorValoracion	-
18. Modificación del nombre de Departamento	IIU Departamento, PIU Departamento, SIU modificar_nombre	ResponsableAdministración, AdministradorValoracion	-
19. Asignar Presupuesto a Departamento	IIU Departamento, PIU Departamento, SIU asignar_presupuesto	ResponsableAdministración, AdministradorValoracion	-
20. Consulta de Presupuesto de Departamento	IIU Departamento, PIU Departamento, SIU consulta_presupuesto	ResponsableAdministración, AdministradorValoracion	SIEFAP.Administracion.Departamento
21. Consulta de Empleados de Departamento	IIU Departamento, PIU Departamento, SIU consultar_empleados	ResponsableAdministración, AdministradorValoracion	SIEFAP.Administracion.Departamento
22. Alta de Administración	IIU Administracion, PIU Administracion, SIU create_instance	AdministradorValoracion	SIEFAP.Administracion
23. Baja de Administración	IIU Administracion, PIU Administracion, SIU BORRAR_ADMINISTRACION	AdministradorValoracion	-
24. Modificación del nombre de Administración	IIU Administracion, PIU Administracion, SIU modificar_nombre	ResponsableAdministración, AdministradorValoracion	-
25. Asignar Presupuesto a Administración	IIU Administracion, PIU Administracion, SIU asignar_presupuesto	ResponsableAdministración, AdministradorValoracion	-
26. Consulta de Presupuesto de la Administración	IIU Administracion, PIU Administracion, SIU	ResponsableAdministración, AdministradorValoracion	SIEFAP.Administracion

	consulta_presupuesto		
27. Consulta de empleados de la Administración	IIU Administracion, PIU Administracion, SIU consultar_empleados	ResponsableAdministración, AdministradorValoracion	SIEFAP.Administracion
28. Crear Conocimiento	IIU Conocimiento, PIU Conocimiento, SIU crear_instancia	ResponsableAdministración, AdministradorValoracion, Usuario Editor	SIEFAP.Contenido.Conocimientos
29. Modificar Conocimiento	IIU Conocimiento, PIU Conocimiento, SIU edit_instance	ResponsableAdministración, AdministradorValoracion, Usuario Editor	-
30. Eliminar Conocimiento	IIU Conocimiento, PIU Conocimiento, SIU delete_instance	ResponsableAdministración, AdministradorValoracion, Usuario Editor	-
31. Crear Subconocimiento	IIU Conocimiento, PIU Conocimiento, SIU crear_instancia	ResponsableAdministración, AdministradorValoracion, Usuario Editor	SIEFAP.Contenido.Conocimientos
32. Crear Pregunta	IIU Pregunta, PIU Pregunta, MDIU Pregunta, MDIU PreguntaPIU, SIU CREA_PREGUNTA	Usuario_Editor	SIEFAP.Contenido.Preguntas
33. Modificar Pregunta	IIU Pregunta, PIU Pregunta, MDIU Pregunta, MDIU PreguntaPIU, MDIU Pregunta	Usuario_Editor	-
34. Crear Respuesta	SIU CREA_PREGUNTA [Pregunta], IIU Respuesta, PIU Respuesta, MDIU Pregunta, MDIU PreguntaPIU, SIU create_instance,	Usuario_Editor	-

35. Modificar Respuesta	IU Respuesta, PIU Respuesta, MDIU Pregunta, MDIU PreguntaPIU, SIU edit_instance	Usuario_Editor	-
36. Eliminar Respuesta	IU Respuesta, PIU Respuesta, MDIU Pregunta, MDIU PreguntaPIU, SIU delete_instance	Usuario_Editor	-
37. Asignar Pregunta a Conocimiento	IIU Pregunta, PIU Pregunta, MDIU Pregunta, MDIU PreguntaPIU, IIU Conocimiento, PIU Conocimiento, SIU InsPregunta	Usuario_Editor, AdministradorValoracion, ResponsableAdministracion	-
38. Desasignar Pregunta de Conocimiento.	IIU Pregunta, PIU Pregunta, MDIU Pregunta, MDIU PreguntaPIU, IIU Conocimiento, PIU Conocimiento, SIU DelPregunta	Usuario_Editor, AdministradorValoracion, ResponsableAdministracion	-
39. Crear Plantilla de Cuestionario	IIU Plantilla_Cuestionario, PIU Plantilla_Cuestionario, MDIU Plantilla de cuestionario, SIU create_instance	Usuario_Editor	-
40. Modificar Plantilla de Cuestionario	IIU Plantilla_Cuestionario, PIU Plantilla_Cuestionario, MDIU Plantilla de cuestionario, SIU edit_instance	Usuario_Editor	-
41. Asignar Pregunta a Plantilla de Cuestionario	SIU create_instance [Plantilla_Cuestionario] IIU Plantilla_Cuestionario, PIU Plantilla_Cuestionario, MDIU Plantilla de cuestionario, SIU create_instance	Usuario_Editor	-

42. Desasignar Pregunta de Plantilla de Cuestionario	IIU Plantilla_Cuestionario, PIU Plantilla_Cuestionario, MDIU Plantilla de cuestionario, SIU delete_instance	Usuario_Editor	-
43. Cambiar valor de Pregunta	IIU Plantilla_Cuestionario, PIU Plantilla_Cuestionario, MDIU Plantilla de cuestionario, SIU edit_instance	Usuario_Editor	-
44. Asignar Plantilla de Cuestionario a Conocimiento	IIU Plantilla_Cuestionario, PIU Plantilla_Cuestionario, MDIU Plantilla de cuestionario, SIU InsPlatilla_Cue	Usuario_Editor, AdministradorValoracion, ResponsableAdministracion	-
45. Crear Evaluación	IIU Valoracion, PIU Valoracion, MDIU Valoracion, SIU create_instance	ResponsableAdministración, AdministradorValoracion	SIEFAP.Evaluaciones
46. Crear Evaluación de Conocimiento	IIU Valoracion, PIU Valoracion, MDIU Valoracion, IIU Valoracion_Conocimiento, PIU Valoracion_Conocimiento, MDIU Valoracion_ConocimientoCo, SIU create_instance	ResponsableAdministración, AdministradorValoracion	SIEFAP.Evaluaciones.Ev. Conocimiento
47. Editar Evaluación	IIU Valoracion, PIU Valoracion, MDIU Valoracion, SIU edit_instance	ResponsableAdministración, AdministradorValoracion	-
48. Cambiar estado	IIU Valoracion, PIU Valoracion, MDIU Valoracion, SIU cambiar_estado	ResponsableAdministración, AdministradorValoracion	-

49. Editar Evaluación de Conocimiento	IUU Valoracion_Conocimiento, PIU Valoracion_Conocimiento, MDIU Valoracion_ConocimientoCo, SIU edit_instance	ResponsableAdministración, AdministradorValoracion	-
50. Eliminar Evaluación	IUU Valoracion, PIU Valoracion, MDIU Valoracion, SIU BORRAR_EVALUACION	ResponsableAdministración, AdministradorValoracion	-
51. Eliminar Evaluación de Conocimiento	IUU Valoracion_Conocimiento, PIU Valoracion_Conocimiento, MDIU Valoracion_ConocimientoCo, SIU BORRAR_VCONOCIMIENTO	ResponsableAdministración, AdministradorValoracion	-
52. Crear Cuestionario	IUU CuestionarioAdministrador, PIU CuestionarioAdministrador, IIU Valoracion_Conocimiento, PIU Valoracion_Conocimiento, MDIU Valoracion, MDIU Valoracion_ConocimientoCo, SIU CREAR_CUESTIONARIO	ResponsableAdministración, AdministradorValoracion	SIEFAP.Cuestionarios.Gestión
53. Asignar Cuestionario a Evaluación de Conocimiento	IUU CuestionarioAdministrador, PIU CuestionarioAdministrador, IIU Valoracion_Conocimiento, PIU Valoracion_Conocimiento, MDIU Valoracion, MDIU Valoracion_ConocimientoCo, PIU PesoCuestionario, SIU create_instance	ResponsableAdministración, AdministradorValoracion	-
54. Desasignar Cuestionario de Evaluación de Conocimiento	IUU CuestionarioAdministrador, PIU CuestionarioAdministrador, PIU PesoCuestionario, MDIU	ResponsableAdministración, AdministradorValoracion	-

	Valoracion_ConocimientoCo, SIU delete_instance		
55. Editar Cuestionario (Cambiar peso)	IIU CuestionarioAdministrador, PIU CuestionarioAdministrador, PIU PesoCuestionario, MDIU Valoracion_ConocimientoCo, SIU edit_instance	ResponsableAdministración, AdministradorValoracion	-
56. Dejar Cuestionario Preparado	IIU CuestionarioAdministrador, PIU CuestionarioAdministrador, SIU cambiar_estado	ResponsableAdministración, AdministradorValoracion	-
57. Asignar Supervisor a Cuestionario	IIU CuestionarioAdministrador, PIU CuestionarioAdministrador, SIU ASIGNAR_SUPERVISOR	ResponsableAdministración, AdministradorValoracion	-
58. Eliminar Cuestionario	IIU CuestionarioAdministrador, PIU CuestionarioAdministrador, SIU BORRAR_CUESTIONARIO	ResponsableAdministración, AdministradorValoracion	-
59. Responder Cuestionario	SIU empezar_cuestionario, SIU responder, SIU elige_respuesta, SIU borrar_respuesta, SIU terminar_cuestioario	Empleado	
60. Comenzar Cuestionario	IIU CuestionarioEmpleado, PIU CuestionarioEmpleado, SIU empezar_cuestionario	Empleado	-
61.1. Responder Pregunta de Cuestionario	PIU Responder, MDIU Pregunta_PlantillaPIU, IIU Respuesta_Empleado, PIU Respuesta_Empleado, SIU empezar_cuestionario, SIU responder	Empleado	-

61.2. Responder Pregunta de Cuestionario	PIU Responder, MDIU Pregunta_PlantillaPIU, IIU Respuesta_Empleado, PIU Respuesta_Empleado, SIU empezar_cuestionario, SIU elige_respuesta	Empleado	-
61.3. Responder Pregunta de Cuestionario	IIU Respuesta_Empleado, PIU Respuesta_Empleado, MDIU Pregunta_PlantillaPIU, SIU empezar_cuestionario, SIU borrar_respuesta	Empleado	-
62. Terminar Cuestionario	IIU CuestionarioEmpleado, PIU CuestionarioEmpleado, SIU terminar_cuestionario	Empleado	-
63. Consultar Detalles Cuestionario	- MDIU Plantilla_CuestionarioCon: Master -> IIU Plantilla_CuestionarioAcc Detail -> MDIU Pregunta_PlantillaPIUCon: Master -> PIU Pregunta_PlantillaCon Detail -> PIU Respuesta_EmpleadoCon Detail -> PIU RespuestaCon	Empleado EmpleadoSupervisor AdministradorValoracion ResponsableAdministración	SIEFAP.Cuestionarios.Listado de Cuestionarios

64. Consultar Detalles de Evaluación	- MDIU Valoracion: Master -> IIU Valoracion Detail -> MDIU Valoracion_Conocimiento	Empleado AdministradorValoracion ResponsableAdministración	SIEFAP.Evaluaciones.Detalles Evaluaciones
65. Consultar Detalles de Evaluación del Conocimiento	- MDIU Valoracion_ConocimientoCo: Master -> PIU Valoracion_Conocimiento Detail -> PIU PesoCuestionario	Empleado EmpleadoSupervisor AdministradorValoracion ResponsableAdministración Empleado EmpleadoSupervisor AdministradorValoracion ResponsableAdministración	SIEFAP.Evaluaciones.Ev. Conocimiento.Evaluaciones de Conocimiento
66. Validar Cuestionario	IIU CuestionarioEmpleado, PIU CuestionarioEmpleado, SIU validar_cuestionario	EmpleadoSupervisor	-
67. Comentar Respuesta de Empleado	SIU comenzar_revision -> MDIU Plantilla_CuestionarioAcc -> MDIU Pregunta_PlantillaPIU -> PIU Respuesta_Empleado -> SIU motivo_correccion	EmpleadoSupervisor	-

68. Solicitar Corrección Cuestionario	IIU CuestionarioEmpleado, PIU CuestionarioEmpleado, SIU corregir_cuestionario	EmpleadoSupervisor	-
--	---	--------------------	---