

**Universidad de las Ciencias Informáticas
Facultad 4**



**Título: Diseño e Implementación de las capas de
Negocio y Acceso a Datos del
Módulo Observación y Clasificación
del Subsistema Observación, Clasificación y
Tratamiento del SIGEP**

Trabajo de Diploma para optar por el título de
Ingeniero en ciencias Informáticas

Autor(es): Hermes Miguel Velázquez Domínguez

Angel Luis Lecuona Rodriguez

Tutor(es): Ing. Luis Alberto Pimentel Gonzalez

Ciudad de la Habana, Cuba

Mayo, 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Hermes Miguel Velázquez Domínguez

Angel Luis Lecuona Rodríguez

Firma del Autor

Firma del Autor

Ing. Luis Alberto Pimentel González

Firma del Tutor

AGRADECIMIENTOS

Compartidos

A nuestro tutor por toda su ayuda en la realización de este trabajo. A la UCI por ser nuestra casa en estos 5 años. A Javier López del Castillo, Yanet Vega y en especial a Yadira Benavides por la valiosa ayuda brindada. A quienes nos ayudaron en estos 5 años.

A todos, sinceramente, muchas gracias.

De Angel Luis

A Hermes, mi compañero de tesis y amigo, por todas las noches juntos, por su calma y paciencia porque sin él esto sencillamente no fuera posible.

A mis amigos, que son muchos, los de antes y los de ahora, con los que he compartido los buenos y malos momentos.

A mi Aylin que ha sabido apoyarme, guiarme y quererme todo el tiempo que hemos estado juntos.

A Bera por llegar para quedarse y darme todo su apoyo en este duro tramo de la vida.

De Hermes

A Angel Luis, mi compañero de tesis, por permitirme hacer la tesis juntos y por permitirme ser su amigo.

A Yoel y Fernando, por ser mis mejores amigos y estar siempre presentes.

A todos mis amigos y mis amigas que me han apoyado siempre.

DEDICATORIA

De Angel

A mis abuelos (Meche y Ago) por su cariño incondicional, y por hacerme vivir momentos muy felices a su lado.

A mis tíos Belkis, Luis, Agusti, Dania e Isabel, que me han enseñado mucho en la vida, mostrándome con su ejemplo como ser mejor persona.

A mis primos Lady, Papito y la loquilla de Diane, que quiero como a mis hermanos y han llenado mi vida de felicidad.

A mi papá que siempre ha estado orgulloso de mí y ha confiado en lo que podía llegar a ser.

Especialmente a mi mamá, mi ejemplo, mi guía, por su preocupación y apoyo, por enseñarme las cosas bellas de este mundo y ser, sobre todas las cosas mi amiga.

De Hermes

A mis padres, por ser la razón de mí existir, por siempre darme su apoyo incondicional.

A mi hermana y mis dos bellos sobrinos, por ser tan importantes en mi vida.

A toda mi familia, por su preocupación y apoyo brindado en cada momento.

RESUMEN

A raíz de la grave situación existente en los penales de la República Bolivariana de Venezuela se decide crear un sistema que gestione toda la información penitenciaria, denominado Sistema de Gestión Penitenciaria (SIGEP). El SIGEP constituye la solución de software para la informatización de la gestión de los privados de libertad de la República Bolivariana de Venezuela. Una de las áreas a informatizar dentro del sistema, es la de Observación, Clasificación y Tratamiento, en la cual se encuentra el módulo Observación y Clasificación, que tiene como objetivo obtener la clasificación del penado, a través de una serie de entrevistas Psicológicas, Sociales, y de Criminología, obteniendo además una caracterización del mismo.

En la etapa inicial de la construcción del SIGEP se definieron los requisitos que satisfacen las funcionalidades del módulo, siendo necesario posteriormente diseñar e implementar las capas de Negocio y Acceso a Datos, permitiendo de esta forma registrar las entrevistas psicológicas, sociales y criminológicas del penado, así como la clasificación de este dentro del sistema penitenciario.

Se diseñó e implementó la solución del módulo cumpliendo con los requisitos, teniendo en cuenta el uso de los patrones de diseño y aplicando los lineamientos definidos en la arquitectura del sistema. Se logró informatizar los procesos relacionados con la Observación y Clasificación.

PALABRAS CLAVES

SIGEP, Observación, Clasificación, Entrevistas, Penado, Diseño, Implementación

TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	I
DEDICATORIA.....	II
RESUMEN.....	III
INTRODUCCIÓN.....	1
DESARROLLO.....	4
DESCRIPCIÓN MÁS DETALLADA DEL PROBLEMA.....	4
ARTEFACTOS DE ENTRADA AL DISEÑO.....	5
Breve descripción de los Casos de Uso del Sistema.....	6
INTEGRACIÓN CON LA ARQUITECTURA.....	14
Capa de Servicios de Negocio.....	15
Capa de Acceso a Datos.....	16
Ambiente de desarrollo.....	16
TECNOLOGÍAS USADAS.....	20
Programación orientada a aspectos.....	20
Programación mediante Eventos.....	21
PATRONES DE DISEÑO.....	21
FLUJOS DE TRABAJO.....	22
DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	25
Diagrama de Clases del Diseño.....	25
Caso de Uso Consultar Histórico Clasificaciones.....	29
Caso de Uso Gestionar información del Área Laboral.....	33
Modelo Físico de Datos.....	39
CONCLUSIONES.....	43
RECOMENDACIONES.....	44
BIBLIOGRAFÍA.....	45
GLOSARIO DE TÉRMINOS.....	46

INTRODUCCIÓN

Con la llegada a la presidencia de la República Bolivariana de Venezuela del Comandante Hugo Rafael Chávez Frías en el año 1998, se comenzó a desarrollar un profundo programa de transformaciones sociales, pero para ello era necesario una nueva constitución que sirviera de base legal para todas las actividades que se querían llevar a cabo, por lo que a finales del año 1999 se aprueba la Constitución de la República Bolivariana de Venezuela y comienza una ola de reformas encaminadas a resolver los principales problemas que se habían heredado de las políticas neoliberales llevadas a cabo por los anteriores gobiernos.

A principios del año 2005 se realizó un censo nacional de la situación judicial de la población penitenciaria. Para el censo se elaboró una encuesta que recogía información del interno sobre: datos personales, datos sobre el delito, datos sobre el proceso judicial y sobre su condición física. El censo permitió acercarse a la realidad en el momento en que fue aplicado, sin embargo, no resuelve la necesidad de gestionar la información de toda la actividad penitenciaria y además carece de un mecanismo de actualización.

Teniendo en cuenta la difícil situación penitenciaria por la cual atraviesa Venezuela, y partiendo de que una de las necesidades es la de contar con una herramienta que permita mantener actualizada la información relacionada con los internos y la gestión de los centros penitenciarios, se decide construir un sistema informático que contribuya a soportar la toma de decisiones estratégicas por parte de la Dirección General de Custodia y Rehabilitación del Recluso (DGCRR).

En este marco nace el proyecto de Humanización del Sistema Penitenciario, que además de integrar aspectos como la atención a la salud de los individuos y asesoría especializada, incluye un sistema informático para gestionar y automatizar los procesos penitenciarios conocido por las siglas SIGEP. El objetivo es realizar mejoras integrales de las condiciones de vida de los individuos en todos los centros de reclusión a lo largo y ancho del país.

El SIGEP constituye la solución de software para la informatización de la gestión de la información de los privados de libertad de la República Bolivariana de Venezuela. Está dividido en subsistemas¹ que constituyen las áreas a informatizar; uno de estos subsistemas es el de Observación, Clasificación y Tratamiento, dentro del cual se encuentra el módulo Observación y Clasificación, que tiene como

¹ Subsistema: Un subsistema se refiere a un conjunto de módulos que por razones de similitud o de perseguir objetivos comunes, son agrupados.

objetivo obtener la clasificación del penado, a través de una serie de entrevistas Psicológicas, Sociales, y de Criminología, obteniendo además una caracterización del mismo. Después de concluido el proceso de modelación de los procesos concernientes a dicho módulo y de igual forma realizada la definición del sistema se obtuvo como resultado los requisitos del módulo Observación y Clasificación del subsistema Observación, Clasificación y Tratamiento. De los requisitos obtenidos y la definición del sistema se concluye que es necesario definir un diseño, que permita llevar a cabo posteriormente la implementación de las capas de Negocio y Acceso a Datos del módulo, lo cual conduce al siguiente problema a resolver:

¿Cómo desarrollar el conjunto de funcionalidades necesarias, correspondientes a las capas de Negocio y Acceso de Datos, que satisfagan los requisitos del SIGEP permitiendo registrar las entrevistas Psicológicas, Sociales, y de Criminología del penado así como la clasificación de este dentro del Sistema Penitenciario?

Teniendo en cuenta el problema planteado se define como objeto de estudio: El proceso de desarrollo de las capas de Negocio y Acceso a Datos del módulo Observación y Clasificación del Subsistema Observación, Clasificación y Tratamiento del SIGEP.

Por lo que se especifica el siguiente campo de acción: Diseño e Implementación de las capas de Negocio y Acceso a Datos del módulo Observación y Clasificación del Subsistema Observación, Clasificación y Tratamiento del SIGEP.

Se plantea como objetivo general diseñar e implementar las capas de Negocio y Acceso a Datos del módulo Observación y Clasificación del subsistema Observación, Clasificación y Tratamiento, las funcionalidades correspondientes al registro de las entrevistas Psicológicas, Sociales, y de Criminología del penado así como la clasificación de este dentro del Sistema Penitenciario.

Se definen las siguientes tareas para lograr el objetivo general planteado:

- Diseñar la solución del problema y realizar los diagramas necesarios tomando como punto de partida el análisis realizado del módulo.
- Diseñar el modelo de datos, así como adicionar sus correspondientes tablas en la Base de Datos del SIGEP.
- Implementar la solución del problema, con las herramientas, tecnologías y convenciones definidas en la arquitectura.

En el proceso de desarrollo de la solución propuesta se utilizaron herramientas y tecnologías que se ajustan a las necesidades del desarrollo del proyecto. Se utilizó como lenguaje de programación Java bajo la especificación de Java Enterprise Edition (JEE) que ofrece potentes tecnologías para el desarrollo de aplicaciones empresariales sobre la Web. Como ambiente de desarrollo integrado se utilizó el IDE Eclipse 3.3 con un conjunto de plugins como: Spring IDE para la integración con Spring Framework, Web Tools Platform para el desarrollo de aplicaciones Web, Subclipse para la integración con el control de versiones e Hibernate Tools para la integración con el framework Hibernate. Como contenedor Web se utilizó Apache Tomcat 6.0 y como gestor de base de datos Oracle 10G. Para la modelación del sistema en sus etapas de desarrollo se utilizaron las herramientas de modelado Erwin y Visual Paradigm. El desarrollo se apoyó en un conjunto de frameworks como Spring, Hibernate, JUnit, entre otros, que simplifican y organizan potencialmente el proceso de elaboración del software.

DESARROLLO

DESCRIPCIÓN MÁS DETALLADA DEL PROBLEMA

Sistema Penitenciario:

Sistema Penitenciario: hace referencia al conjunto de normas, procedimientos y dependencias dispuestas por el Estado para la ejecución del régimen penitenciario entre los que se encuentran además los principios, programas, recursos humanos, dependencias e infraestructura que se encuentran relacionadas y destinadas a este régimen.

Se define sistema penitenciario también "al conjunto de normas legislativas o administrativas encaminadas a determinar los diferentes sistemas adoptados para que los penados cumplan sus penas. Se encamina a obtener la mayor eficacia en la custodia o en la readaptación social de los delincuentes. Esos regímenes son múltiples, varían a través de los tiempos; y van desde el aislamiento absoluto y de tratamiento rígido hasta el sistema de puerta abierta con libertad vigilada" (OSORIO).

En el caso de la República Bolivariana de Venezuela, tal sistema estaría constituido por la legislación vigente, los métodos que se emplearán para lograr su funcionamiento, las diferentes dependencias encargadas de su aplicación, los equipos de trabajo y la infraestructura carcelaria.

Observación y Clasificación:

Conjunto de acciones encaminadas a obtener información de interés del individuo acorde a una serie de criterios contemplados en la ley, a través de las entrevistas de tipo psicológica, social y criminológica, con el objetivo de darle la clasificación adecuada a cada individuo para su posterior tratamiento dentro de la institución penitenciaria.

Una vez que el individuo ingresa al Centro Penitenciario, comienza el período de Observación y Clasificación. Este tiene como objetivo realizar una evaluación integral del individuo en aras de clasificarlo y elaborar un plan de tratamiento que responda a sus necesidades particulares. Durante todo el periodo de Observación el individuo es entrevistado por distintos especialistas que pertenecen al Equipo Multidisciplinario del Centro Penitenciario. El objetivo de las entrevistas es obtener la

información preliminar necesaria para determinar la dirección del tratamiento e influir sobre la conducta del individuo.

Se recoge esta información y se pone a disposición del resto de los miembros del Equipo Multidisciplinario, salvando los detalles que quedan bajo el secreto profesional de cada área.

En los centros penitenciarios venezolanos existen grandes problemas en el proceso de Observación y Clasificación, en la mayoría de los casos el proceso se realiza manualmente, y cada especialista de manera individual elabora sus propios instrumentos para evaluar al individuo, esto unido a la falta de comunicación que existe entre ellos y que la información recogida de las entrevistas no es vista al mismo tiempo por los especialistas, provoca que exista un desajuste en el proceso de Observación lo que trae consigo deficiencias a la hora de clasificar correctamente al penado.

La falta de un sistema automatizado que controle la información referente al interno es un factor que tiende a deteriorar de forma directa la correcta ubicación de la población penal dentro de los centros y al mismo tiempo no permite una buena rehabilitación al no dar a los internos el correcto tratamiento.

La automatización de estos procesos resuelve en gran medida los problemas mencionados anteriormente, entre otras cosas permitirá que el trabajo del equipo de los especialistas de tratamiento sea más eficiente y por consiguiente las condiciones de vida y la rehabilitación de los internos serán mucho mejores.

ARTEFACTOS DE ENTRADA AL DISEÑO

Como resultado del proceso de captura de requisitos y definición del sistema efectuado en los meses de agosto y septiembre del año 2007 en Venezuela, por parte de los analistas del proyecto SIGEP con la colaboración activa de la parte venezolana implicada en el desarrollo del proyecto se obtuvieron los requerimientos funcionales y la definición de los casos de uso del sistema referentes al módulo de Observación y Clasificación del subsistema Observación, Clasificación y Tratamiento, los cuales son el principal punto de partida para el diseño e implementación del mismo.

Breve descripción de los Casos de Uso del Sistema.

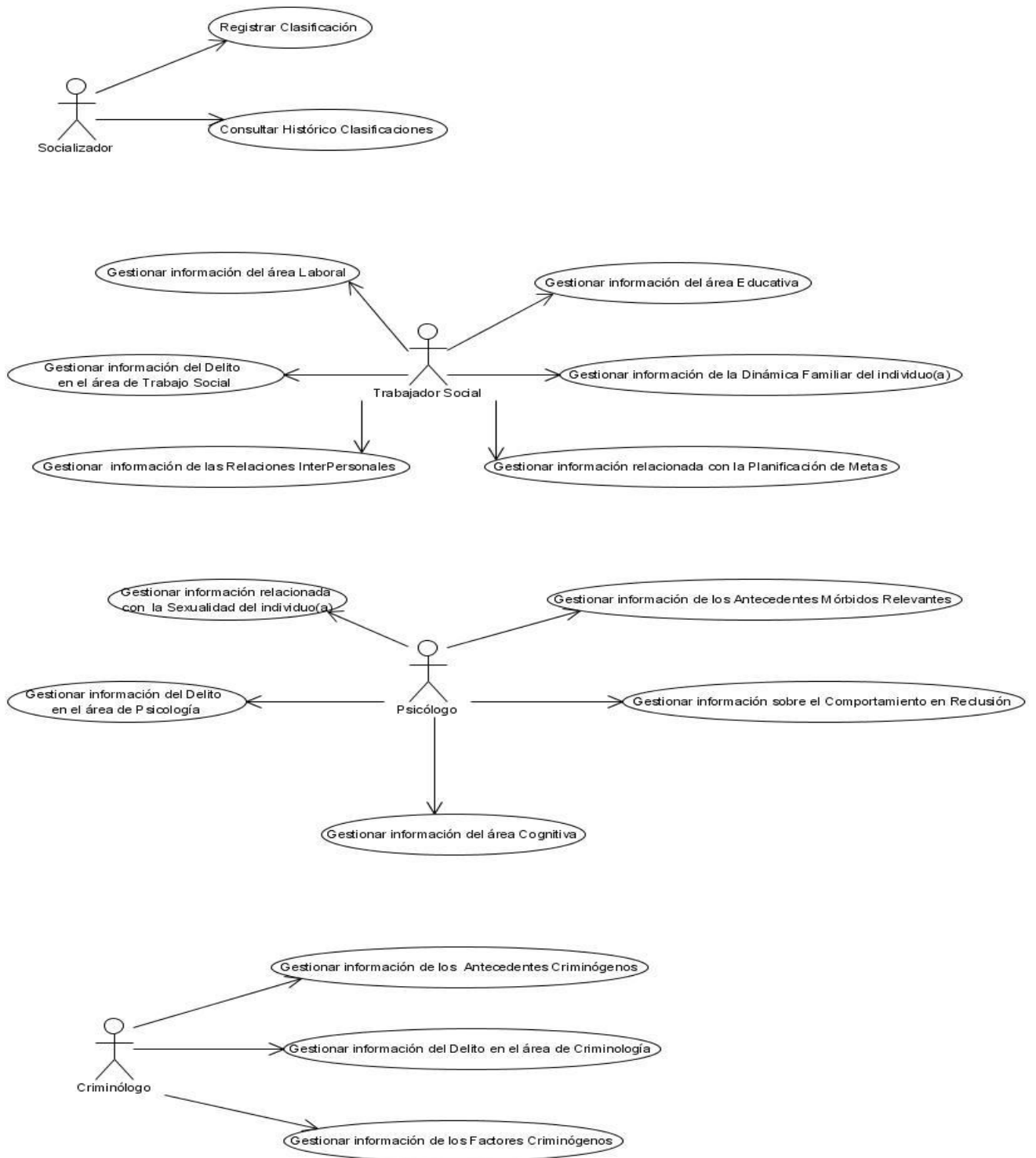


Figura 1 Diagramas de Casos de Uso del Sistema

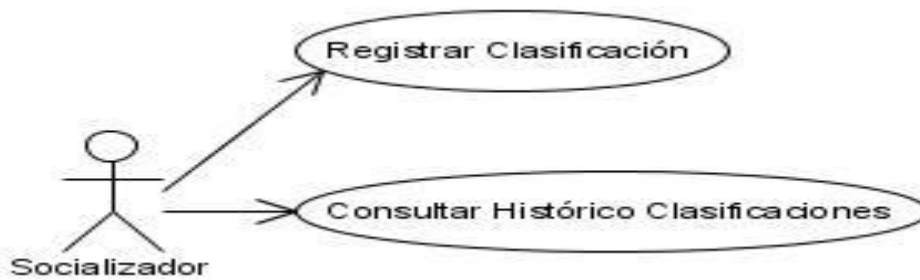


Figura 2 Diagrama de Caso de Uso de Clasificación

Descripción

Actor	Justificación
Socializador	Especialista Miembro del Equipo de Tratamiento que trabaja diariamente con el individuo. Se encarga de velar por su conducta y por el cumplimiento de su tratamiento dentro del Establecimiento Penitenciario.

Tabla 1. Actor Socializador

CU-1	Registrar Clasificación
Actor	Socializador
Breve Descripción	El Socializador solicita el registro de una clasificación de un individuo y se ejecuta el proceso de adicionar una clasificación.
Pre-Condiciones	El Socializador se ha autenticado ante el sistema. El Socializador ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha adicionado una clasificación al individuo.

Tabla 2. Resumen del Caso de Uso 1.

CU-2	Consultar Histórico Clasificaciones
Actor	Socializador
Breve Descripción	El Socializador solicita el histórico de clasificaciones, y se ejecuta el proceso de obtener el listado de las clasificaciones.
Pre-Condiciones	El Socializador se ha autenticado ante el sistema. El Socializador ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado el Histórico de Clasificaciones del individuo.

Tabla 3. Resumen del Caso de Uso 2.

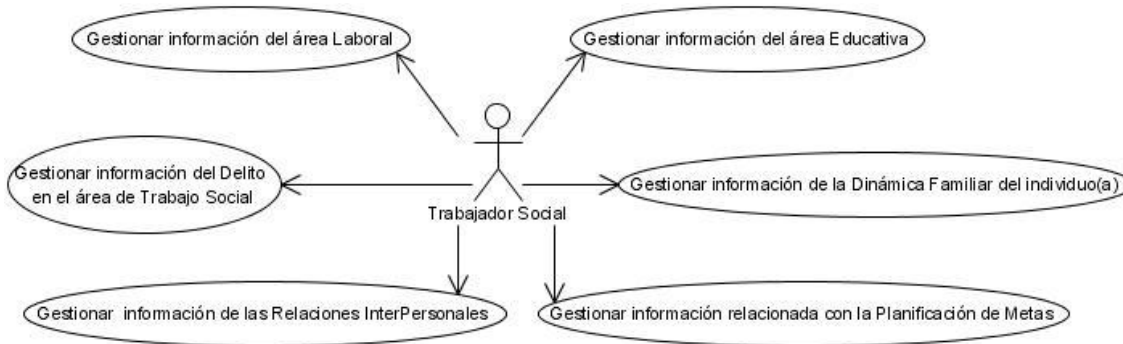


Figura 3 Diagrama de Casos de Uso del sistema de la Entrevista de Trabajo Social

Actor	Justificación
Trabajador Social	Especialista en Trabajo Social, miembro del Equipo de Tratamiento que se encarga de realizar entrevistas y evaluaciones en función de vincular al individuo a las distintas actividades de tratamiento.

Tabla 4. Actor Trabajador Social

CU-3	Gestionar información del Área Laboral
Actor	Trabajador Social
Breve Descripción	El Trabajador Social indica que desea gestionar los datos del área laboral de la Entrevista de Trabajo Social. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área laboral.
Pre-Condiciones	El Trabajador Social se ha autenticado ante el sistema. El Trabajador Social ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área laboral del individuo.

Tabla 5. Resumen del Caso de Uso 3.

CU-4	Gestionar información del Área Educativa
Actor	Trabajador Social
Breve Descripción	El Trabajador Social indica que desea gestionar los datos del área educativa de la Entrevista de Trabajo Social. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área educativa.
Pre-Condiciones	El Trabajador Social se ha autenticado ante el sistema. El Trabajador Social ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área educativa del individuo.

Tabla 6. Resumen del Caso de Uso 4.

CU-5	Gestionar información del Delito en el Área de Trabajo Social
Actor	Trabajador Social
Breve Descripción	El Trabajador Social indica que desea gestionar los datos del delito de la Entrevista de Trabajo Social. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área del delito.
Pre-Condiciones	El Trabajador Social se ha autenticado ante el sistema. El Trabajador Social ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área del delito de la Entrevista de Trabajo Social.

Tabla 7. Resumen del Caso de Uso 5.

CU-6	Gestionar información de la Dinámica Familiar del individuo
Actor	Trabajador Social
Breve Descripción	El Trabajador Social indica que desea gestionar los datos de dinámica familiar de la Entrevista de Trabajo Social. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área de dinámica familiar.
Pre-Condiciones	El Trabajador Social se ha autenticado ante el sistema. El Trabajador Social ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área de dinámica familiar de la Entrevista de Trabajo Social.

Tabla 8. Resumen del Caso de Uso 6.

CU-7	Gestionar información de las Relaciones Interpersonales
Actor	Trabajador Social
Breve Descripción	El Trabajador Social indica que desea gestionar los datos de las relaciones interpersonales de la Entrevista de Trabajo Social. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área de relaciones interpersonales.
Pre-Condiciones	El Trabajador Social se ha autenticado ante el sistema. El Trabajador Social ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área de relaciones interpersonales de la Entrevista de Trabajo Social.

Tabla 9. Resumen del Caso de Uso 7.

CU-8	Gestionar información relacionada con la Planificación de Metas
Actor	Trabajador Social
Breve Descripción	El Trabajador Social indica que desea gestionar los datos de la planificación de metas de la Entrevista de Trabajo Social. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área de planificación de metas.
Pre-Condiciones	El Trabajador Social se ha autenticado ante el sistema. El Trabajador Social ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área de planificación de metas de la Entrevista de Trabajo Social.

Tabla 10. Resumen del Caso de Uso 8.

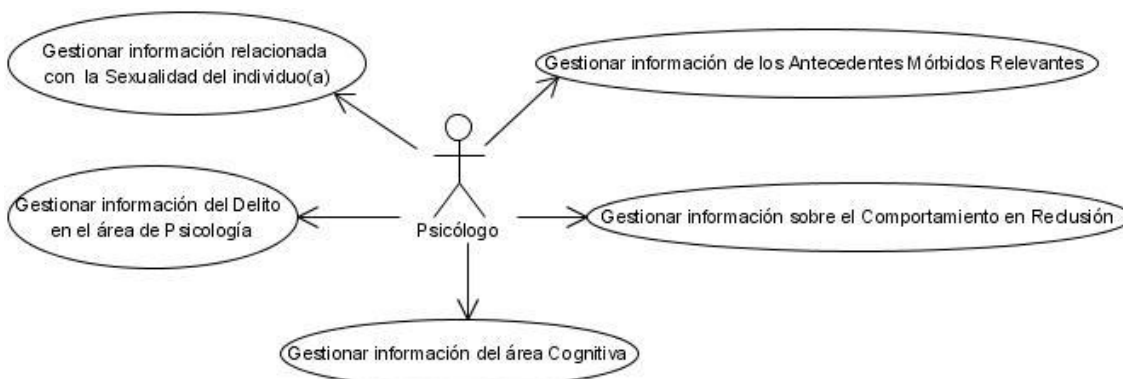


Figura 4 Diagrama de Casos de Uso del sistema de la Entrevista Psicológica

Actor	Justificación
Psicólogo	Especialista en Psicología miembro del Equipo de Tratamiento que se encarga de velar e influir en la conducta y evolución del individuo dentro del Establecimiento Penitenciario. Constantemente está evaluando al individuo mediante la aplicación de distintos instrumentos.

Tabla 11. Actor Psicólogo

CU-9	Gestionar información relacionada con la Sexualidad del individuo
Actor	Psicólogo
Breve Descripción	El Psicólogo indica que desea gestionar los datos de la planificación de metas de la Entrevista de Psicología. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área de planificación de metas.
Pre-Condiciones	El Psicólogo se ha autenticado ante el sistema. El Psicólogo ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la

	información del área de sexualidad de la Entrevista de Psicología.
--	--

Tabla 12. Resumen del Caso de Uso 9.

CU-10	Gestionar información de los Antecedentes Mórbido Relevantes
Actor	Psicólogo
Breve Descripción	El Psicólogo indica que desea gestionar los datos de los antecedentes mórbidos relevantes de la Entrevista de Psicología. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área de antecedentes mórbidos relevantes.
Pre-Condiciones	El Psicólogo se ha autenticado ante el sistema. El Psicólogo ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área de antecedentes mórbidos relevantes de la Entrevista de Psicología.

Tabla 13. Resumen del Caso de Uso 10.

CU-11	Gestionar información del Delito en el Área de Psicología
Actor	Psicólogo
Breve Descripción	El Psicólogo indica que desea gestionar los datos del delito de la Entrevista de Psicología. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área de los delitos.
Pre-Condiciones	El Psicólogo se ha autenticado ante el sistema. El Psicólogo ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área del delito de la Entrevista de Psicología.

Tabla 14. Resumen del Caso de Uso 11.

CU-12	Gestionar información sobre el Comportamiento en Reclusión
Actor	Psicólogo
Breve Descripción	El Psicólogo indica que desea gestionar los datos sobre el comportamiento en reclusión del individuo de la Entrevista de Psicología. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área del comportamiento en reclusión del individuo.
Pre-Condiciones	El Psicólogo se ha autenticado ante el sistema. El Psicólogo ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área del comportamiento en reclusión de la Entrevista de Psicología.

Tabla 15. Resumen del Caso de Uso 12.

CU-13	Gestionar información del área Cognitiva
Actor	Psicólogo
Breve Descripción	El Psicólogo indica que desea gestionar los datos del área cognitiva de la Entrevista de Psicología. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área cognitiva.
Pre-Condiciones	El Psicólogo se ha autenticado ante el sistema. El Psicólogo ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área cognitiva de la Entrevista de Psicología.

Tabla 16. Resumen del Caso de Uso 13.

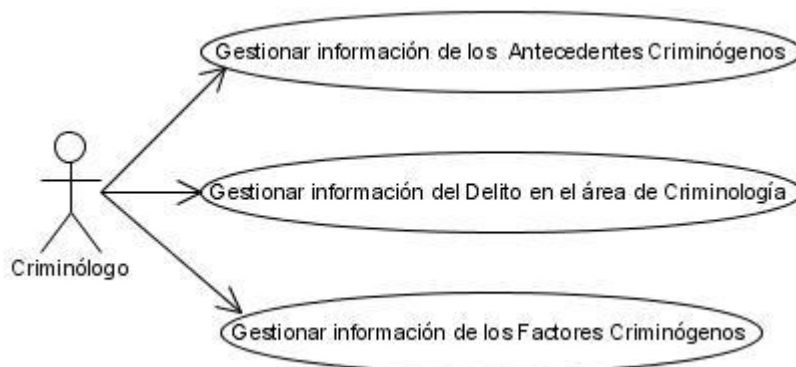


Figura 5 Diagrama de Casos de Uso de la Entrevista Criminológica

Actor	Justificación
Criminólogo	Especialista en Criminología miembro del Equipo de Tratamiento que se encarga de investigar sobre distintos factores que llevaron al individuo a cometer el delito. Mediante distintos instrumentos este especialista evalúa la evolución del individuo y le pone un tratamiento acorde a sus necesidades.

Tabla 17. Actor Criminólogo

CU-14	Gestionar información de los Antecedentes Criminógenos
Actor	Criminólogo
Breve Descripción	El Criminólogo indica que desea gestionar los datos de los antecedentes criminógenos de la Entrevista de Criminología. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área de antecedentes criminógenos.
Pre-Condiciones	El Criminólogo se ha autenticado ante el sistema. El Criminólogo ha accedido al Expediente de Tratamiento del Individuo.

Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área de antecedentes criminógenos de la Entrevista de Criminología.
------------------	---

Tabla 18. Resumen del Caso de Uso 14.

CU-15	Gestionar información del Delito en el Área de Criminología
Actor	Criminólogo
Breve Descripción	El Criminólogo indica que desea gestionar los datos del delito de la Entrevista de Criminología. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área delitos de la entrevista de criminología.
Pre-Condiciones	El Criminólogo se ha autenticado ante el sistema. El Criminólogo ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área de delito de la Entrevista de Criminología.

Tabla 19. Resumen del Caso de Uso 15.

CU-16	Gestionar información de los Factores Criminógenos
Actor	Criminólogo
Breve Descripción	El Criminólogo indica que desea gestionar los datos de los factores criminógenos de la Entrevista de Criminología. El sistema permite consultar, modificar, registrar y eliminar la información relacionada con el área de los factores criminógenos de la entrevista de criminología.
Pre-Condiciones	El Criminólogo se ha autenticado ante el sistema. El Criminólogo ha accedido al Expediente de Tratamiento del Individuo.
Post-Condiciones	Se ha consultado, modificado, registrado o eliminado la información del área de factores criminógenos de la Entrevista de Criminología.

Tabla 20. Resumen del Caso de Uso 16.

INTEGRACIÓN CON LA ARQUITECTURA

Se define la arquitectura de software como “la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución” (IEEE 2000).

La arquitectura definida para el desarrollo del SIGEP está basada en una arquitectura de tres capas lógicas fundamentales: Capa de Presentación, Capa de Lógica de Negocio, y Capa de Acceso a Datos (figura 6). Dichas capas están bien delimitadas una de la otra, una capa superior interactúa con la inferior mediante interfaces que definen las funcionalidades que la misma debe brindar.

En la arquitectura del SIGEP se define una fachada que representa la interacción entre las Capas de Presentación y Lógica de negocio, la utilización de esta fachada cumple con el patrón de diseño *Facade*.

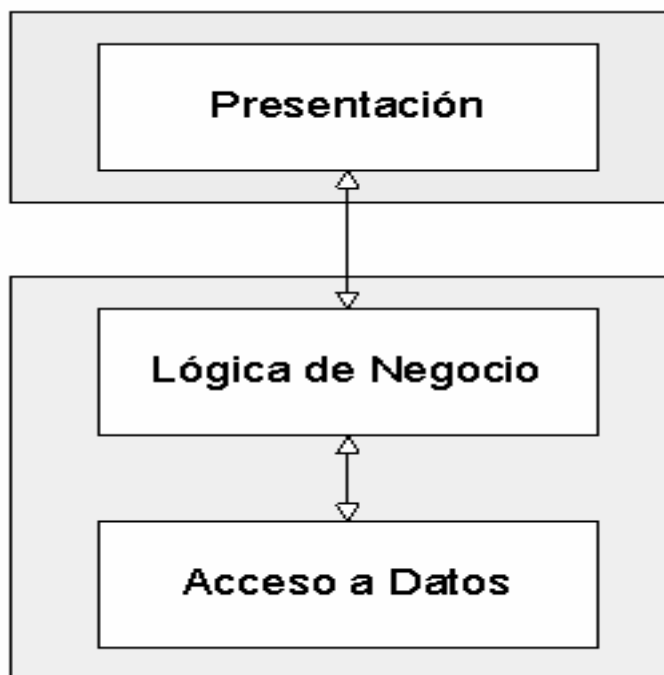


Figura 6 Representación básica de la arquitectura el SIGEP.

Capa de Servicios de Negocio

En esta capa radican los objetos de negocio. Los objetos de negocio separan los datos y la lógica de negocio usando un modelo de objetos.

Las funcionalidades fundamentales que presenta esta capa son:

- **Lógica de negocio específica de procesos de negocios:** A veces es más oportuno para los objetos de dominio contener lógica de negocio aplicable a muchos casos de uso específicos. Sin embargo, existen funcionalidades que son realizadas en la capa de servicios de negocios. Pero en esta definición de arquitectura los objetos de dominio no presentan ningún tipo de lógica de negocio, sino que esta responsabilidad recae sobre los objetos de negocio, permitiendo usar a los objetos de dominio como objetos de transferencia que se mueven entre las capas arquitectónicas de la aplicación.
- **Puntos de entrada muy bien definidos para las operaciones de negocio implementadas:** Los objetos de negocio brindan las interfaces usadas por la capa de presentación.
- **Control de transacciones:** Las políticas de transacciones de la aplicación serán planteadas sobre los objetos de negocio.
- **Ejecución de restricciones de seguridad:** Las restricciones de seguridad en esta capa se realizarán en los puntos de entrada a la capa media, es decir en los objetos de negocio.
- **Ejecución de la auditoría:** Sobre esta capa se llevará a cabo la auditoría sobre los métodos de negocio.

Por cada módulo se definirá una o más fachadas en caso de que se requiera que agrupen los métodos de negocio implementados en los manejadores o Managers que serán explicados más adelante. La fachada de un módulo se basa en el patrón *Facade* para permitir una clara división entre las capas arquitectónicas. Los *Managers* son las clases que se especializan en un conjunto de funcionalidades que representan el negocio sobre una o varias entidades. Los Managers son las únicas clases en la aplicación que tendrán lógica de negocio mientras que las fachadas se limitarán solamente a agrupar las funcionalidades para ser expuestas a capas superiores.

Capa de Acceso a Datos

La interacción del negocio con la capa de acceso a datos se realizará mediante el uso de interfaces. El término Objeto de Acceso a Datos o en inglés, *Data Access Object* (DAO) es ampliamente usado en el desarrollo de software.

Los DAO encapsulan el manejo de acceso a datos de los objetos de dominio, proveen la persistencia de los objetos transitorios y las actualizaciones de los objetos existentes en la base de datos. Las implementaciones de los DAO estarán disponibles para los objetos (típicamente para los objetos de negocio) haciendo uso de la inyección de dependencias con los objetos de negocio y las instancias de los DAO, configuradas en el contenedor de inversión de control de Spring Framework.

Las interfaces de los DAO contienen básicamente los siguientes tipos de métodos:

- **Métodos para descubrir:** Estos localizan los objetos almacenados para ser usados por la capa de servicios de negocio.
- **Métodos para persistir o salvar:** Estos hacen persistentes a los objetos transitorios.
- **Métodos para eliminar:** Estos eliminan a los objetos guardados en el medio de almacenamiento (generalmente una base de datos).
- **Métodos para conteos y otras funciones agregadas:** Estos devuelven resultados de operaciones, que es más eficiente implementarlas, usando funcionalidades de la base de datos, que iterar sobre los mismos objetos.

Vale aclarar que en este caso la interacción entre la capa de lógica de negocio y la capa de acceso a datos se realiza a través de una fachada, similar a lo que ocurre entre la capa de presentación y la capa de lógica de negocio, de este modo se puede apreciar una clara división entre estas dos capas y se agrupan las funcionalidades del acceso a datos que serán expuestas a la capa de lógica de negocio.

Ambiente de desarrollo

El ambiente de desarrollo (Development Environment) es imprescindible en la producción de software. Aquí se definen el conjunto de herramientas y tecnologías (frameworks), que intervienen en un proceso de desarrollo de software así como las versiones a usar y su integración.

Ambiente de desarrollo integrado

Un ambiente de desarrollo integrado (*Integrated Development Environment* (IDE)) es un programa compuesto por un conjunto de herramientas para un programador. El mismo puede dedicarse a un solo lenguaje de programación o a varios.

Herramientas utilizadas en nuestro ambiente de desarrollo:

Tecnología J2EE

La tecnología Java 2 Enterprise Edition (J2EE) proporciona una completa y potente plataforma orientada al desarrollo de aplicaciones corporativas distribuidas y a los servicios web. La misma integra un conjunto de APIs, *frameworks* y patrones de programación que permiten responder de una forma robusta y flexible a todas las demandas de este tipo de aplicaciones.

Eclipse

Eclipse es una plataforma de software de código abierto que típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE).

Aunque Eclipse es escrito en Java y su principal uso es como IDE para Java, este es un lenguaje neutral. El soporte para desarrollo en Java es proveído por un componente enchufado o plugin, pero además están disponibles plugins para otros lenguajes, como C/C++, Cobol, C#.

Plugins

Un plugin o plug-in, es una aplicación que interactúa con otra para agregarle una funcionalidad específica y es ejecutada por la aplicación principal. En el caso particular de Eclipse no son más que un conjunto de clases que permiten hacerlo más extensible y aumentar su potencial.

Web Tool Platform (WTP)

La plataforma de herramientas web de Eclipse o Eclipse Web Tool Platform (WTP) provee varias APIs para desarrollo de aplicaciones sobre la Web y J2EE. Éstas incluyen editores gráficos, de código fuente para una variedad de lenguajes, asistentes y aplicaciones incorporadas para simplificar el desarrollo de servicios web, además de herramientas y API para soportar el despliegue, ejecución y prueba de aplicaciones.

Spring IDE

Spring IDE es un plugin que sirve como interfaz de usuario gráfica para la configuración de los archivos usados por Spring Framework. Permite el completamiento de etiquetas, valores de atributos y elementos en estos archivos de configuración.

Presenta un editor gráfico con completamiento y validaciones para los archivos de definiciones del flujo web usado por Spring Web Flow.

Hibernate Tools

Hibernate Tools es un conjunto de herramientas para trabajar con el framework Hibernate.

Este plugin tiene las siguientes funcionalidades:

- Editor de mapeos.
- Consola.
- Ingeniería inversa.
- Asistentes.
- Tareas de Ant.

Subclipse

Subclipse es un plugin para Eclipse que adiciona integración para el control de versiones (Subversion, específicamente), permitiendo operaciones de sincronización, actualización, entre otras. Permite bloqueos a recursos para que otros usuarios no puedan modificarlo. Dispone de una vista de comparación entre el recurso local y remoto en caso que exista conflicto entre la versión del recurso local con el remoto.

Spring Framework

Spring Framework es un framework de aplicación de código abierto que ayuda a hacer el desarrollo en JEE mucho más fácil. Ayuda a estructurar aplicaciones completas en una manera consistente y productiva para crear arquitecturas coherentes.

Presenta un diseño que brinda una gran flexibilidad arquitectónica e interviene en todas las capas de una aplicación.

El mismo presenta varios módulos de los cuales los más importantes son:

- Contenedor de Inversión de Control.

- Framework para la Programación Orientada a Aspectos (AOP).
- Abstracción de acceso a datos.
- Simplificación de JDBC.
- Administración de transacciones.
- Framework Web MVC.
- Simplificación para trabajar con JNDI, JTA y otros APIs de JEE.
- Comunicación remota de peso ligero (*Lightweight remoting*).
- Soporte para Servicio de Mensajería de Java (*Java Message Service, JMS*).
- Soporte para *Java Management Extensión (JMX)*.
- Soporte para comprensivas estrategias de pruebas para desarrolladores de aplicación.

Hibernate

Hibernate es un potente framework de mapeo objeto/relacional y servicio de consultas para Java (y disponible también para .Net con el nombre de NHibernate). Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. Soporta diferentes entornos como: Oracle, DB2, MySQL, etcétera.

JUnit

JUnit es un framework que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. También es un medio de controlar las pruebas de regresión, necesarias cuando una parte del código ha sido modificado y se desea ver que el nuevo código cumple con los requerimientos anteriores y que no se ha alterado su funcionalidad después de la nueva modificación.

Ehcache

Ehcache es un framework de caché para Java. Provee almacenamiento en memoria y disco y distribuye las operaciones para clusters. Es muy usado en proyectos de código abierto (open source) como Hibernate y Spring.

Apache Tomcat

Apache Tomcat es un contenedor de Servlet usado en la implementación de referencia oficial para las tecnologías Java Servlet y JavaServer Pages. Es desarrollado en un ambiente participativo y abierto.

Se usa en numerosas aplicaciones web de gran escala y críticas en diversas industrias y organizaciones que se referencian en su sitio oficial.

Embarcadero ER/Studio

Embarcadero ER/Studio es una herramienta para el diseño de base de datos, que brinda productividad en su diseño, generación, y mantenimiento de aplicaciones. Desde un modelo lógico de los requerimientos de información, hasta el modelo físico perfeccionado para las características específicas de la base de datos diseñada. ER/Studio soporta principalmente bases de datos relacionales SQL y bases de datos que incluyen Oracle, Microsoft SQL Server, Sybase. El mismo modelo puede ser usado para generar múltiples bases de datos, o convertir una aplicación de una plataforma de base de datos a otra.

Oracle

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés *Relational Data Base Management System*), fabricado por *Oracle Corporation*. Se considera uno de los sistemas de bases de datos más completos. Es un sistema gestor de base de datos robusto, tiene muchas características que garantizan la seguridad e integridad de los datos; que las transacciones se ejecuten de forma correcta, sin causar inconsistencias; ayuda a administrar y almacenar grandes volúmenes de datos; estabilidad, escalabilidad y es multiplataforma.

Visual Paradigm Suite

Visual Paradigm Suite es un conjunto de herramientas CASE que permiten realizar el modelado dentro del proceso de desarrollo de software.

TECNOLOGÍAS USADAS

Programación orientada a aspectos

La Programación Orientada a Aspectos (POA) es un paradigma de programación relativamente reciente cuya intención es permitir una adecuada modularización de las aplicaciones y posibilitar una mejor separación de conceptos. Permite insertar funcionalidades sobre otras ya implementadas sin necesidad de modificar el código original.

Se utilizó la programación orientada a aspectos en la definición de las transacciones sobre los métodos de negocio. Esto significa que cada método de negocio ejecutado es envuelto dentro de un contexto transaccional.

Programación mediante Eventos

La programación dirigida por eventos es un paradigma de programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema o que ellos mismos provoquen.

Se utilizó la programación orientada a eventos en el módulo de Observación y Clasificación para intercambiar información con otros módulos de la aplicación.

En el proceso de registrar una nueva clasificación a un individuo, surge la necesidad, de que otros módulos de la aplicación actualicen su información en dependencia de la nueva clasificación registrada, es por eso que es preciso informar de esta operación. La programación mediante eventos permite publicar un evento que podrá ser capturado por los módulos que necesiten conocer esta operación.

PATRONES DE DISEÑO

Los desarrolladores orientados a objetos con experiencia (y otros desarrolladores de software) acumulan un repertorio tanto de principios generales como de soluciones basadas en aplicar ciertos estilos que les guían en la creación de software. Estos principios y estilos, si se codifican con un formato estructurado que describa el problema y la solución, se les denominan patrones.

De manera más simple se puede decir que un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos.

A continuación se mencionan algunos de los patrones de diseño que son utilizados en el diseño del módulo Observación y Clasificación.

Patrón *Facade*: El patrón de diseño *Facade* sirve para proveer de una interfaz unificada sencilla que haga de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas.

El patrón *Facade* es implementado a través de las fachadas que dividen las capas de la aplicación, esto resuelve el problema de que las capas no dependan directamente de las interfaces de la capa inferior, en este caso se conectan a la interfaz de la fachada que se encarga de exponer funcionalidades a la capa superior. En el caso de que se decida cambiar la implementación de las clases no es necesario que la capa superior se entere de tal situación, sencillamente la fachada no permite que esto ocurra, o sea, no existe una dependencia directa entre las capas.

Patrón DAO: es un patrón de diseño Core JEE, se encarga de manejar todo lo relacionado con la capa de Acceso a Datos de una aplicación. Se usa para aislar la aplicación de la tecnología de persistencia utilizada, en el caso de que se quiera cambiar o actualizar la tecnología de persistencia no es necesario que otras partes de la aplicación sean afectadas por dicho cambio.

Por cada entidad persistente del dominio se crea un objeto DAO, representado por una interface que expone las funcionalidades y una implementación de esa interface que está relacionada directamente con la tecnología de persistencia utilizada.

FLUJOS DE TRABAJO

Para realizar el diseño y la implementación de las capas de Negocio y Acceso a Datos del módulo Observación y Clasificación se definen un conjunto de actividades que garantizan que el proceso de desarrollo se ejecute de la mejor manera posible. Este conjunto de actividades es conocido como Flujo de Trabajo.

Flujo de Trabajo (Diseñador)

- Analizar artefactos de entrada al diseño (Descripción de Casos de Uso).
- Diseñar el modelo de datos.
- Identificar clases del dominio.
- Representar las relaciones entre las clases del dominio en un diagrama de clases.
- Definir las interfaces de las Fachadas, los Managers y los DAO.

- Representar las relaciones de las Fachadas, los Managers y los DAO en un diagrama de clases.
- Realizar diagramas de interacción de cada una de las funcionalidades.
- Declarar las interfaces de las Fachadas, los Managers y los DAO

Flujo de Trabajo (Implementador de Negocio).

- Declarar las entidades de dominio.
- Implementar las Fachadas y configurarlas en el xml de configuración de Spring.
- Implementar los Managers y configurarlos en el xml de configuración de Spring.
- Realizar pruebas unitarias a la implementación de los Managers.

Flujo de Trabajo (Implementador de Acceso a Datos).

- Crear ficheros de mapeo.
- Implementar los DAOs y configurarlos en el xml de configuración.
- Realizar pruebas unitarias a la implementación de los DAO.

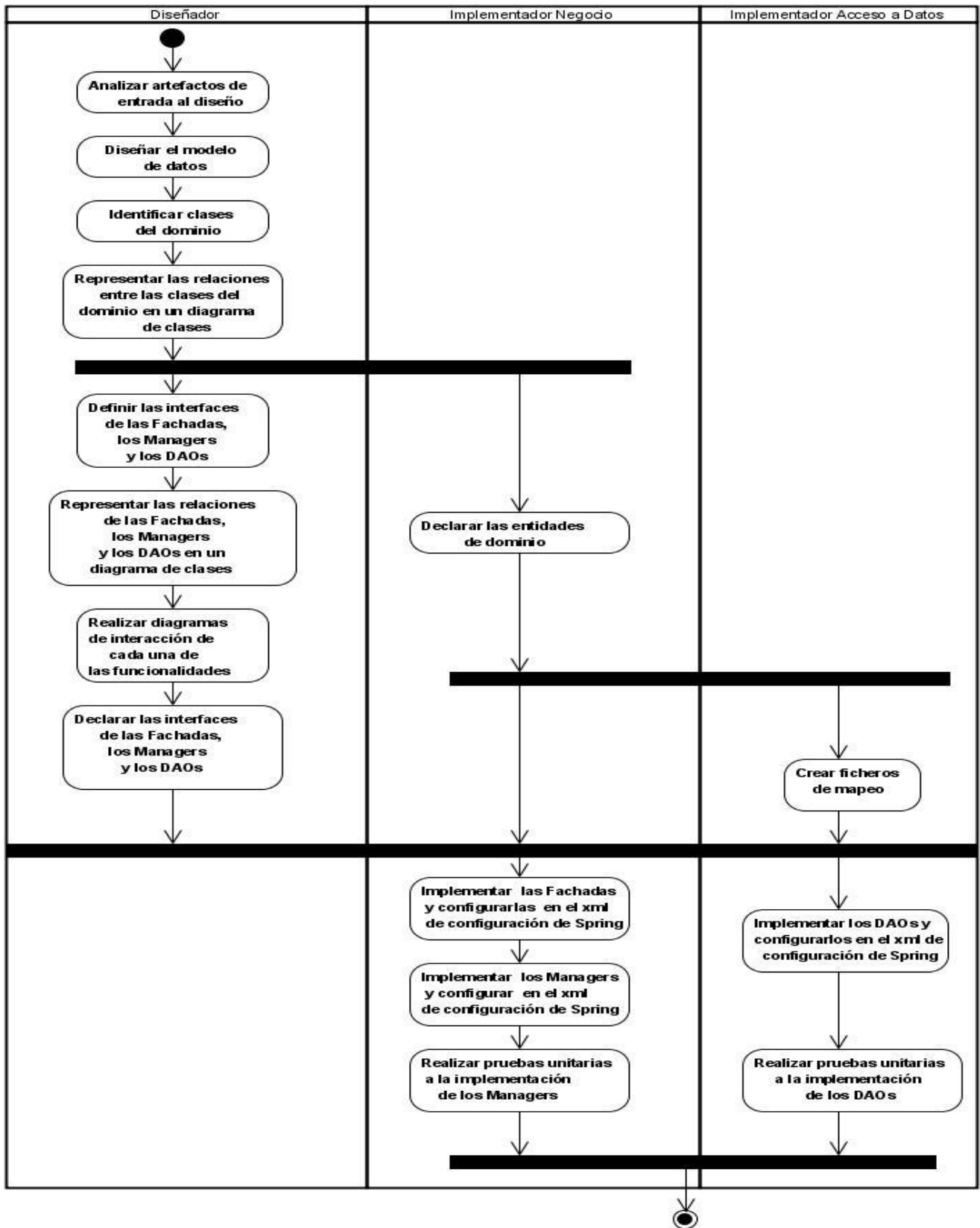


Figura 7 Diagrama de actividades de los flujos de trabajo

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

A la hora de diseñar un software uno de los principales artefactos a generar y utilizar son los diagramas de clases del sistema, en lo siguiente se van a reflejar los mismos agrupándolos por los requerimientos señalados anteriormente.

Diagrama de Clases del Diseño

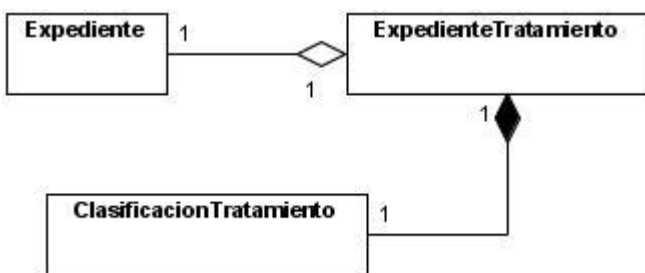


Figura 8 Diagrama de Clases del Dominio correspondiente a los Casos de Uso 1 y 2

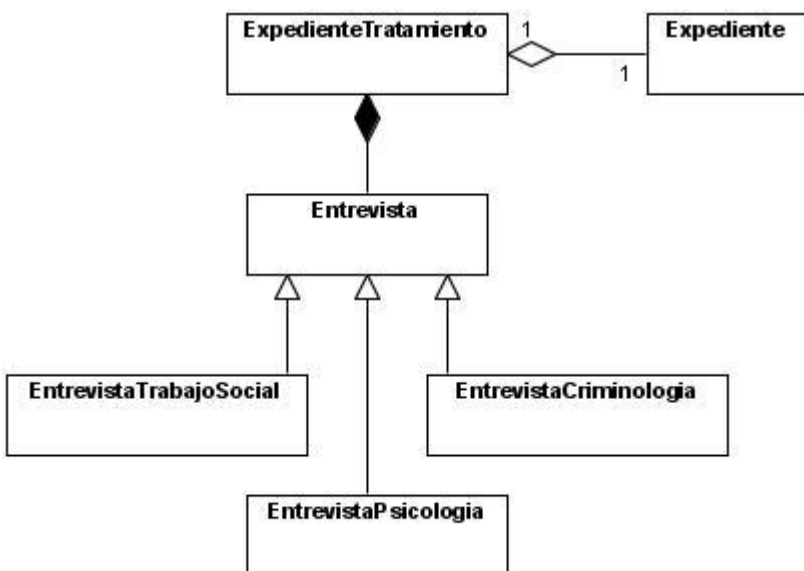


Figura 9 Diagrama de Clases de Dominio que agrupa los diferentes tipos de entrevistas.

El diagrama de la figura 9 representa una panorámica sobre cómo están relacionadas las entrevistas, como tal el diagrama por sí solo no representa ninguna funcionalidad, a continuación se muestran los diagramas correspondientes a cada entrevista en particular.

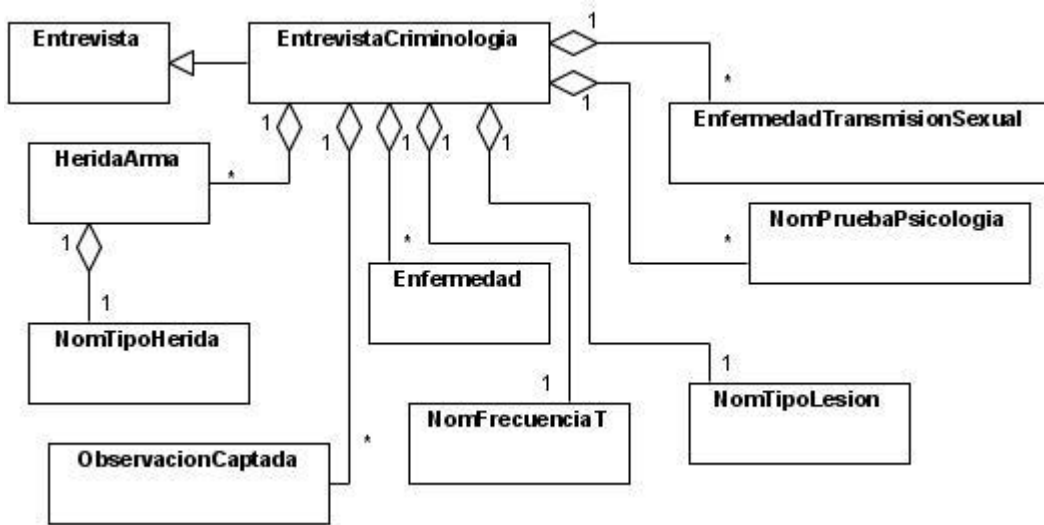


Figura 10 Diagrama de Clases del Dominio correspondiente a los Casos de Uso 14, 15 y 16

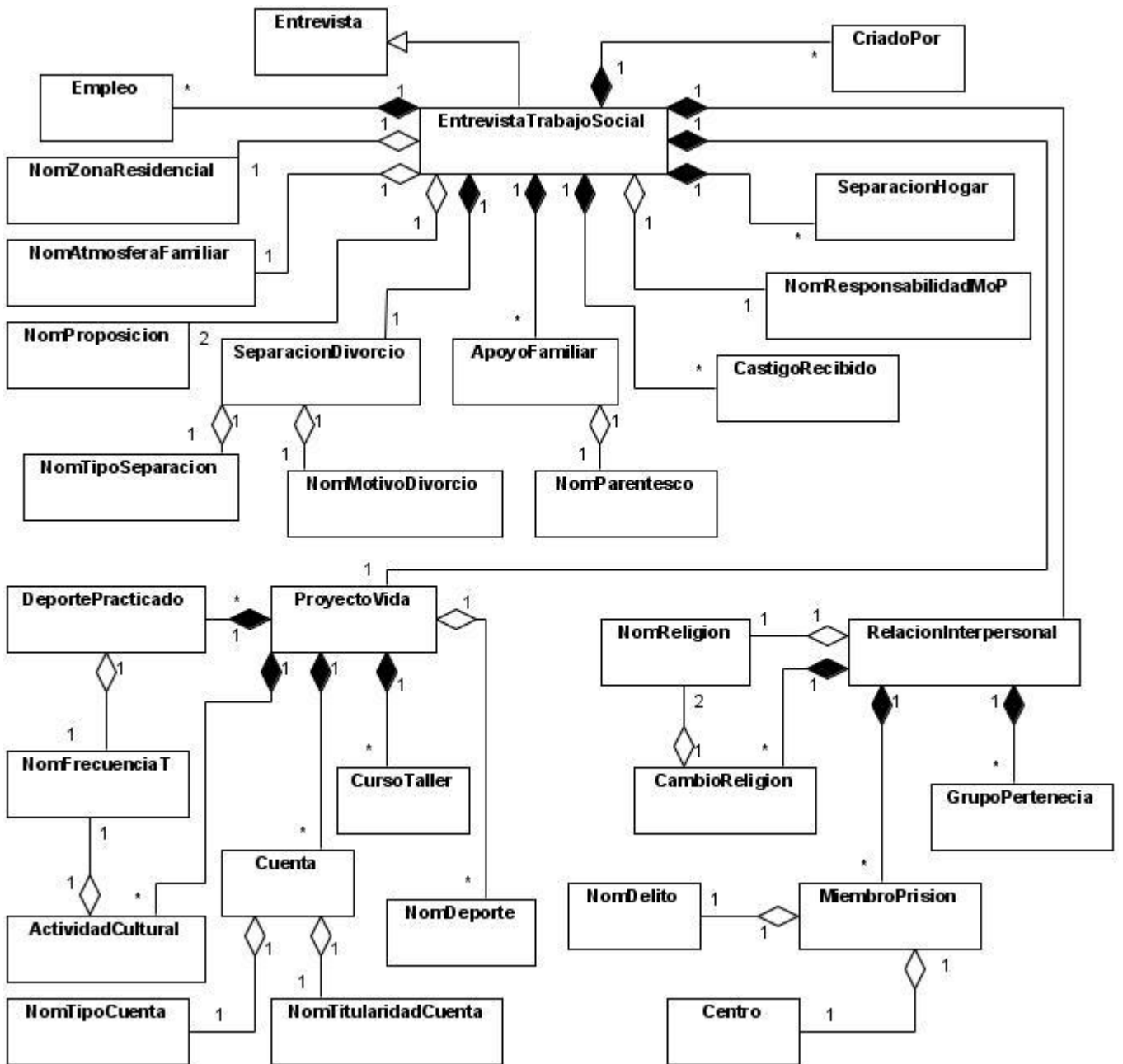


Figura 11 Diagrama de Clases del Dominio correspondiente a los Casos de Uso 3, 4, 5, 6, 7 y 8

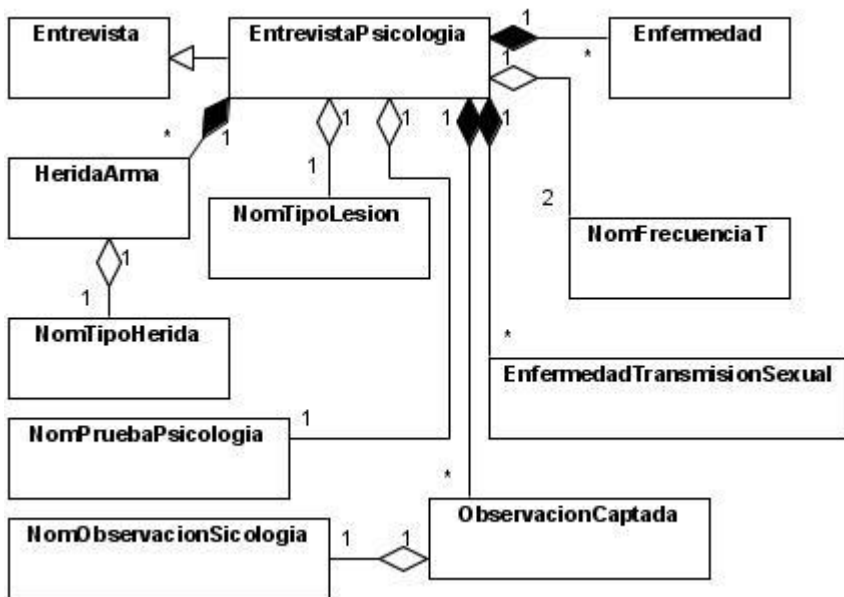


Figura 12 Diagrama de Clases del Dominio correspondiente a los Casos de Uso 9, 10, 11, 12 y 13

Como se ha podido observar en la descripción de los Casos de Uso del Sistema correspondientes al módulo Observación y Clasificación, la mayoría de estos describen funcionalidades similares; por ejemplo el proceso de registrar los resultados de las entrevistas, las cuales están divididas por especialidades en tres tipos fundamentales (Entrevista de Trabajo Social, Criminológica y Psicológica), recogiendo datos de cada una de ellas, aunque estos datos son muy específicos para cada entrevista lo que provoca que sean diferentes, es fácil notar que el proceso de registrarlos es muy semejante. Debido a esto se muestra a continuación los diagramas de clases y de secuencia de dos Casos de Uso, siendo estos los más representativos.

Los Casos de Uso del Sistema que se representarán son:

- Consultar Histórico Clasificaciones.
- Gestionar información del Área Laboral.

Caso de Uso Consultar Histórico Clasificaciones

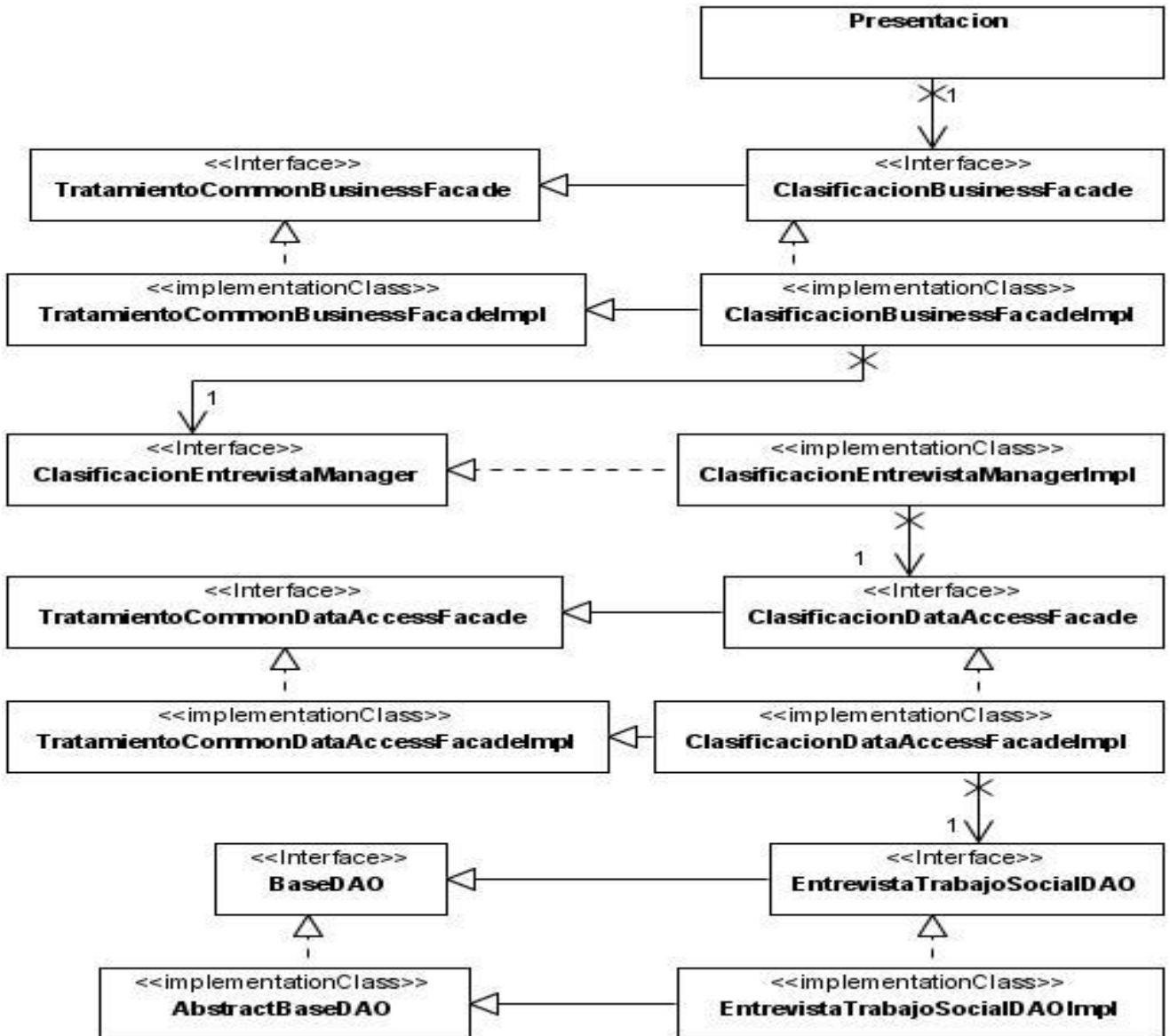


Figura 13 Diagrama de Clases involucradas en el proceso de Consultar el Histórico de Clasificaciones

Descripción de las Clases e Interfaces.

TratamientoCommonBusinessFacade: Es una interfaz que expone funcionalidades comunes para todos los módulos del subsistema de Observación, Clasificación y Tratamiento.

TratamientoCommonBusinessFacadeImpl: Esta clase implementa la interfaz **TratamientoCommonBusinessFacade**. En ella no se implementa ninguna lógica de negocio, simplemente se limita a delegar las responsabilidades a la clase **Manager** correspondiente.

ClasificacionBusinessFacade: Es una interfaz que contiene las funcionalidades particulares del módulo Observación y Clasificación, que sirven de fachada a la Capa de Presentación. Hereda de la interfaz **TratamientoCommonBusinessFacade**.

ClasificacionBusinessFacadeImpl: Esta clase implementa la interfaz **ClasificacionBusinessFacade**. Hereda de la clase **TratamientoCommonBusinessFacadeImpl**. En ella no se implementa ninguna lógica de negocio, simplemente se limita a delegar las responsabilidades a la clase **Manager** correspondiente.

ExpedienteTratamientoManager: Es una interfaz que contiene las funcionalidades relacionadas con el expediente de tratamiento de un penado.

ExpedienteTratamientoManagerImpl: Esta clase implementa la interfaz **ExpedienteTratamientoManager**. Se encarga de implementar las funcionalidades brindadas por la fachada como la de obtener la cantidad de clasificaciones de un penado y mostrar una lista con todas las clasificaciones que ha tenido.

TratamientoCommonDataAccessFacade: Es una interfaz que agrupa funcionalidades comunes relacionados con los métodos de acceso a datos del subsistema Observación, Clasificación y Tratamiento.

TratamientoCommonDataAccessFacadeImpl: Esta clase implementa la interfaz **TratamientoCommonDataAccessFacade**. En esta clase no se implementa ninguna lógica de acceso a datos, ella sólo se encarga de delegar esas responsabilidades a los DAOs.

ClasificacionDataAccessFacade: Es una interfaz que brinda los servicios relacionados con los métodos de acceso a datos del módulo Observación y Clasificación. Hereda de la interfaz TratamientoCommonDataAccessFacade.

ClasificacionDataAccessFacadeImpl: Esta clase implementa la interfaz ClasificacionDataAccessFacade. En esta clase no se implementa ninguna lógica de acceso a datos, ella sólo se encarga de delegar esas responsabilidades a los DAOs. Hereda de la clase TratamientoCommonDataAccessFacadeImpl.

ClasificacionTratamientoDAO: Es una interfaz que brinda las funcionalidades correspondientes al manejo del acceso a datos de la entidad ClasificacionTratamiento.

ClasificacionTratamientoDAOImpl: Esta clase implementa la interfaz ClasificacionTratamientoDAO. En ella se implementa el manejo del acceso a datos correspondiente a la entidad ClasificacionTratamiento.

Diagrama de Secuencia del Caso de Uso Consultar Histórico Clasificaciones.

El diagrama de secuencia del caso de uso Consultar Histórico de Clasificaciones define el orden en que se va a realizar un conjunto de llamadas, que siguen con la arquitectura definida aplicada a las especificaciones de nuestro módulo que brinda el resultado esperado, y entre qué entidades se van a realizar estas llamadas. Como resultado final se obtiene una lista con todas las clasificaciones del penado.

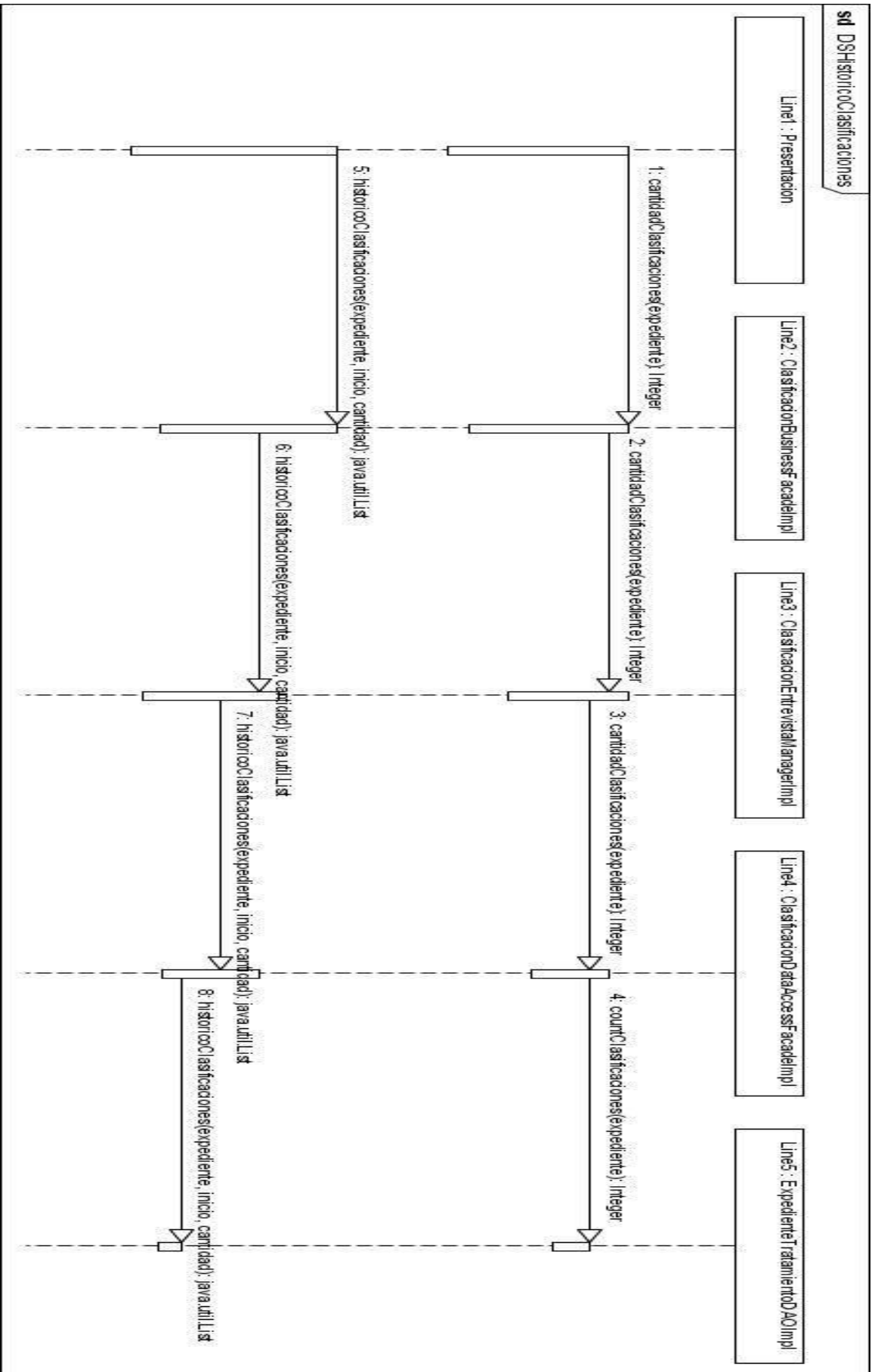


Figura 14 Diagrama de Secuencia de Histórico de Clasificaciones

Caso de Uso Gestionar información del Área Laboral.

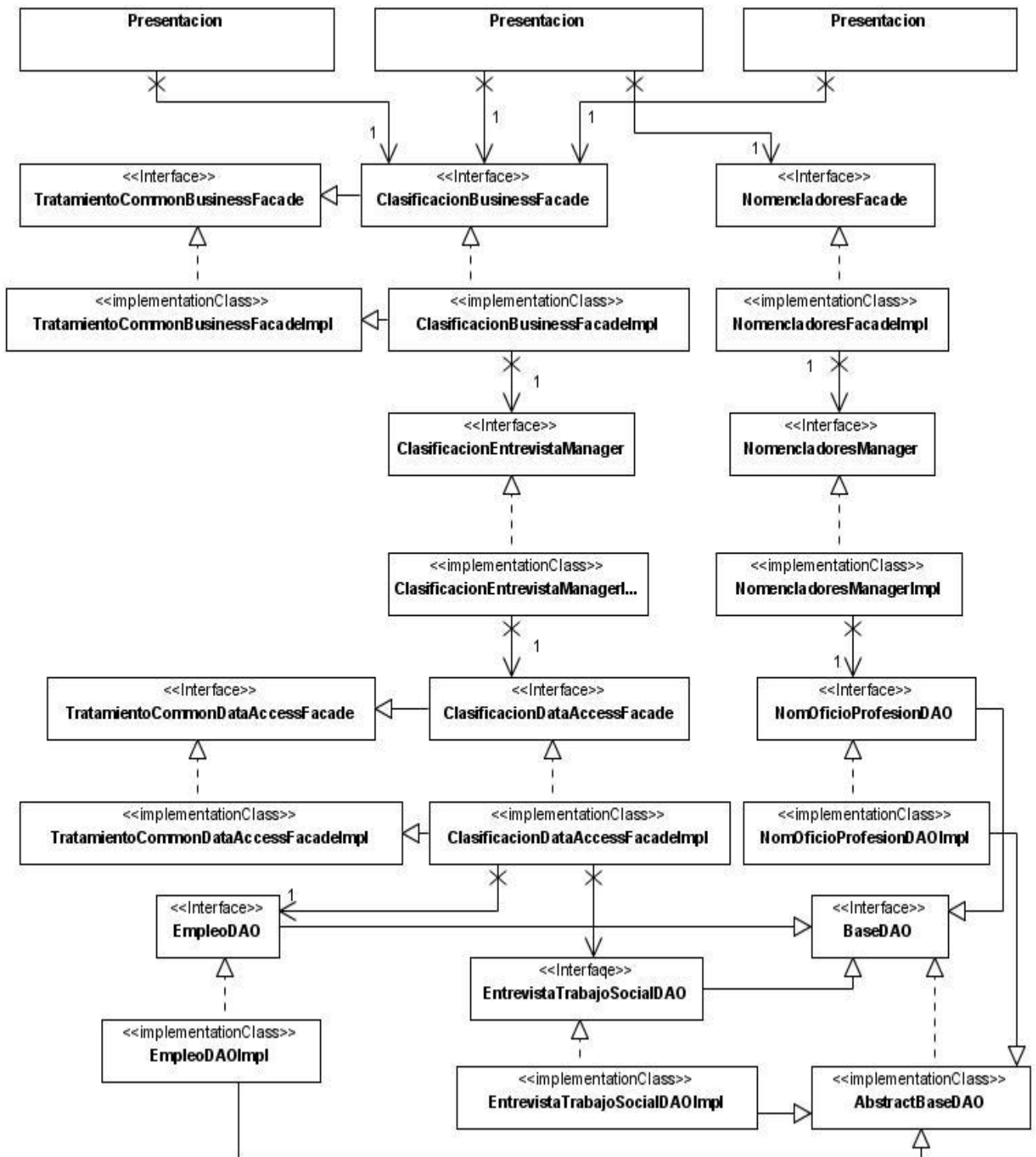


Figura 15 Diagrama de Clases involucradas en Registrar Área Laboral

Descripción de las Clases e Interfaces

NomencladoresFacade: Es una interfaz que contiene las funcionalidades relacionadas con los nomencladores del sistema.

NomencladoresFacadeImpl: Esta clase implementa la interfaz NomencladoresFacade. Se encarga de implementar las funcionalidades brindadas por la fachada como la de listar todos los tipos de oficios.

NomencladoresManager: Es una interfaz que contiene las funcionalidades relacionadas con los nomencladores del sistema.

NomencladoresManagerImpl: Esta clase implementa la interfaz NomencladoresManager. Se encarga de implementar las funcionalidades brindadas por la fachada como la de listar todos los Oficios Profesión.

NomOficioProfesionDAO: Es una interfaz que contiene las funcionalidades específicas relacionadas con el nomenclador NomOficioProfesion.

NomOficioProfesionDAOImpl: Esta clase implementa la interfaz NomOficioProfesionDAO. Se encarga de implementar las funcionalidades brindadas por la fachada como la de listar todos los Oficios o Profesiones.

EmpleoDAO: Es una interfaz que contiene las funcionalidades específicas relacionadas con la entidad Empleo.

EmpleoDAOImpl: Esta clase implementa la interfaz EmpleoDAO. Se encarga de implementar las funcionalidades brindadas por la fachada como la de obtener un empleo específico, o modificar un empleo.

EntrevistaTrabajoSocialDAO: Es una interfaz que contiene las funcionalidades específicas relacionadas con la entidad EntrevistaTrabajoSocial.

EntrevistaTrabajoSocialDAOImpl: Esta clase implementa la interfaz EntrevistaTrabajoSocialDAO. Se encarga de implementar las funcionalidades brindadas por la fachada como la de persistir una EntrevistaTrabajoSocial.

Diagrama de Secuencia del Caso de Uso Gestionar Información del Área Laboral

El proceso de gestionar la información del Área Laboral es un proceso que requiere un mayor trabajo en la capa de presentación, esto provoca que sean 3 instancias las que piden funcionalidades a la fachada, por este motivo se han dividido en 3 diagramas de secuencia para hacer más factible su entendimiento.

El diagrama de secuencia del Caso de Uso Gestionar Información del Área Laboral define el orden en que se va a realizar un conjunto de llamadas, que siguen con la arquitectura definida aplicada a las especificaciones de nuestro módulo que brinda el resultado esperado, y entre qué entidades se van a realizar estas llamadas. Como entrada se recibe el expediente del penado y la información relacionada con el área educativa que se está registrando, obteniendo como resultado final el estado en que quedó la aplicación.

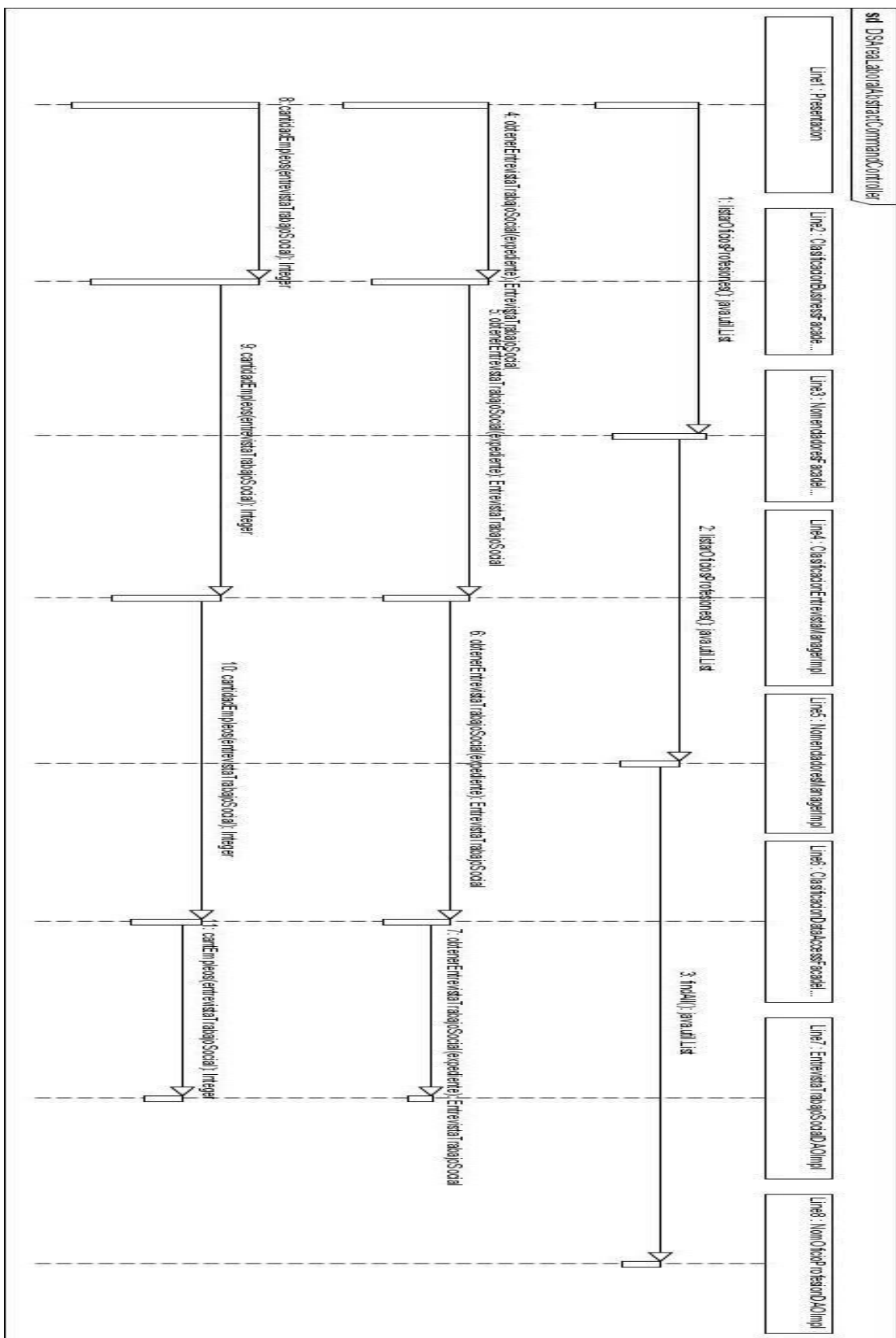


Figura 16 Diagrama de Secuencia de Registrar Información Área Laboral 1

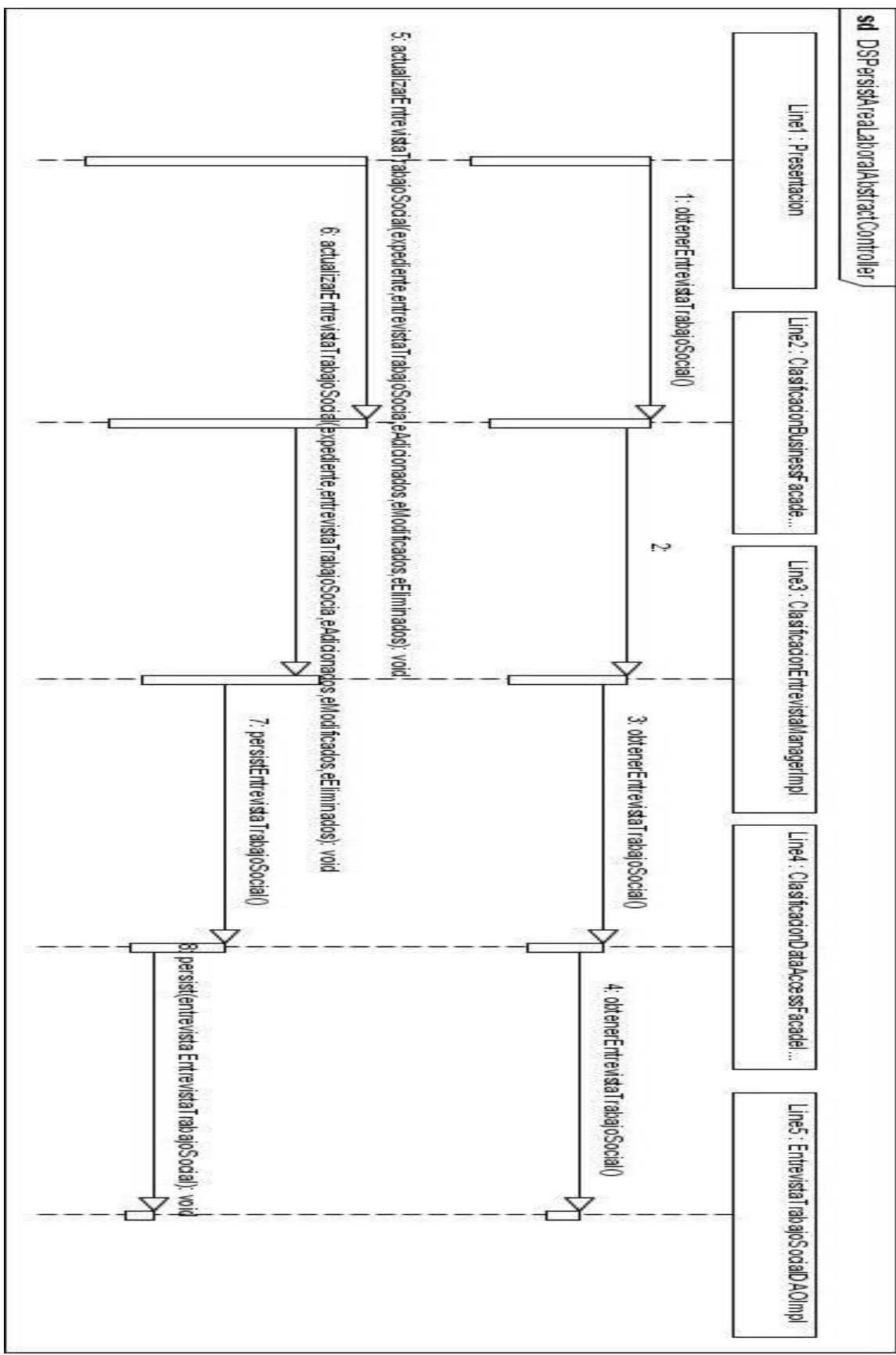


Figura 17 Diagrama de Secuencia de Registrar Información Área Laboral 2

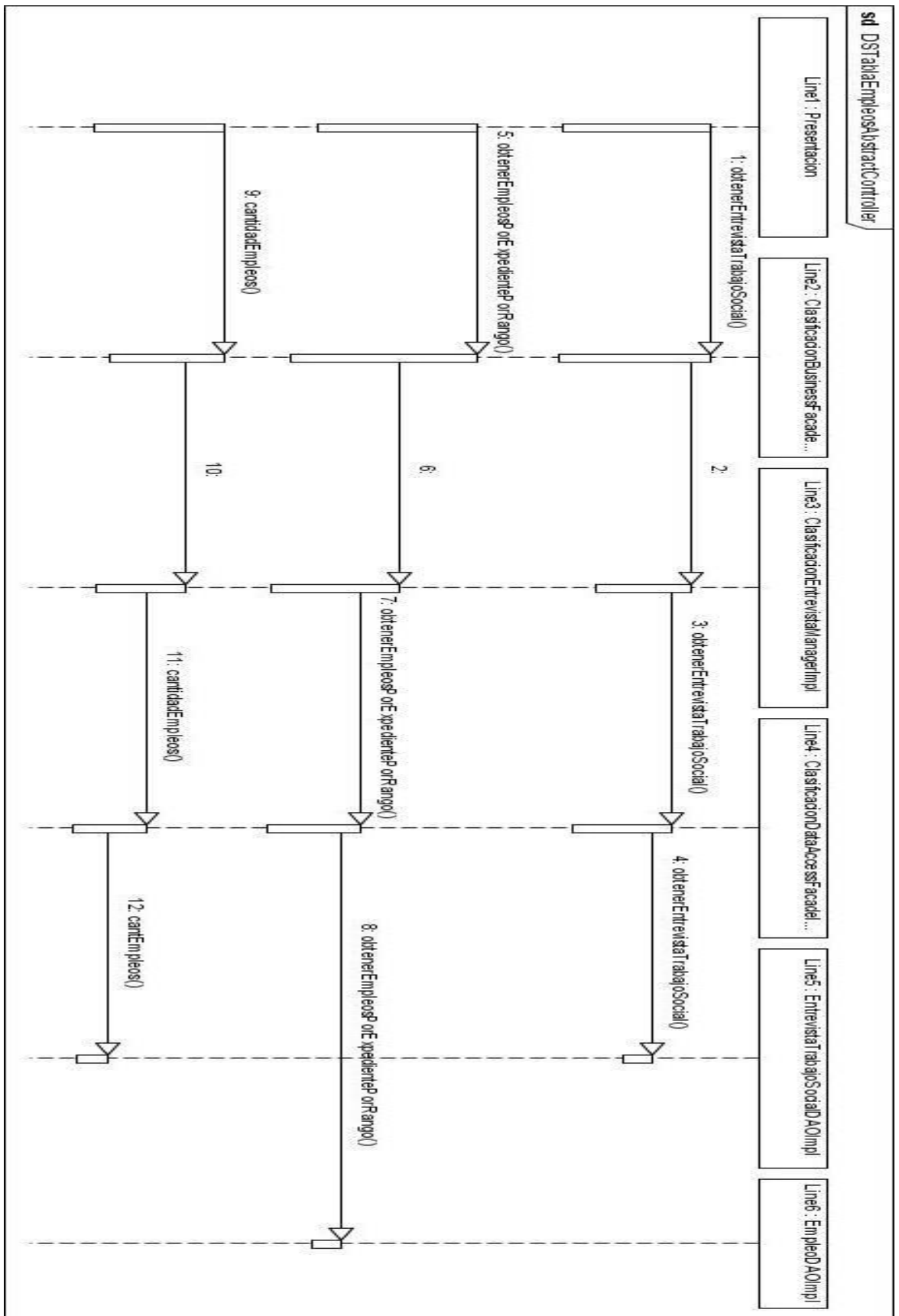


Figura 18 Diagrama de Secuencia de Registrar Información Área Laboral 3

Modelo Físico de Datos

Debido a la gran extensión del modelo lógico de datos diseñado para el módulo Observación y Clasificación se decide dividirlo en 4 sub-modelos que representan agrupaciones de funcionalidades similares.

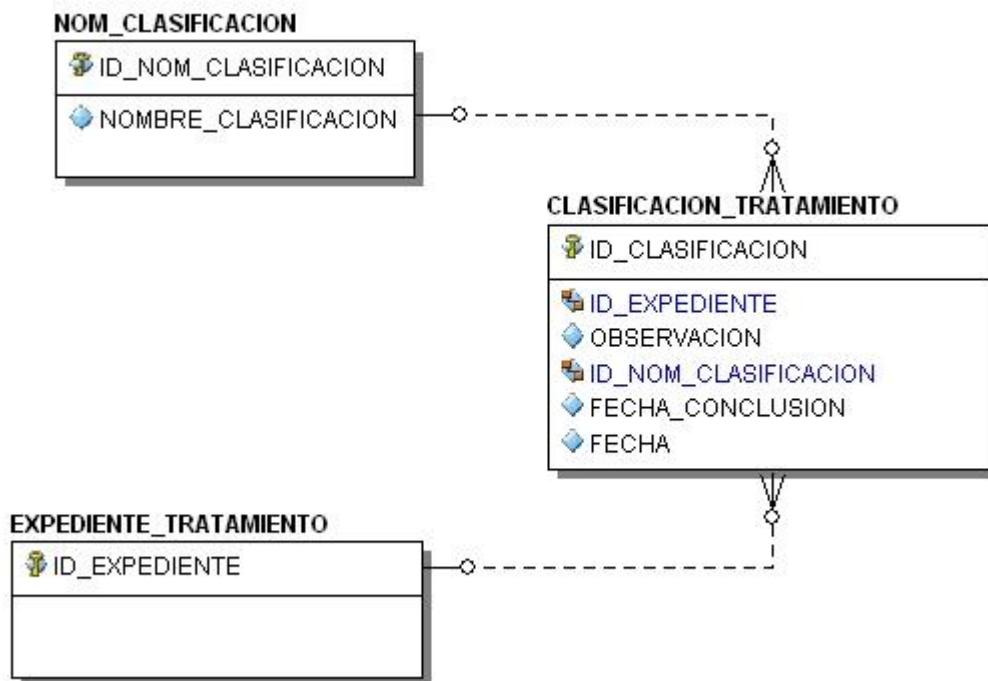


Figura 19 Sub-modelo Físico de Datos Clasificación

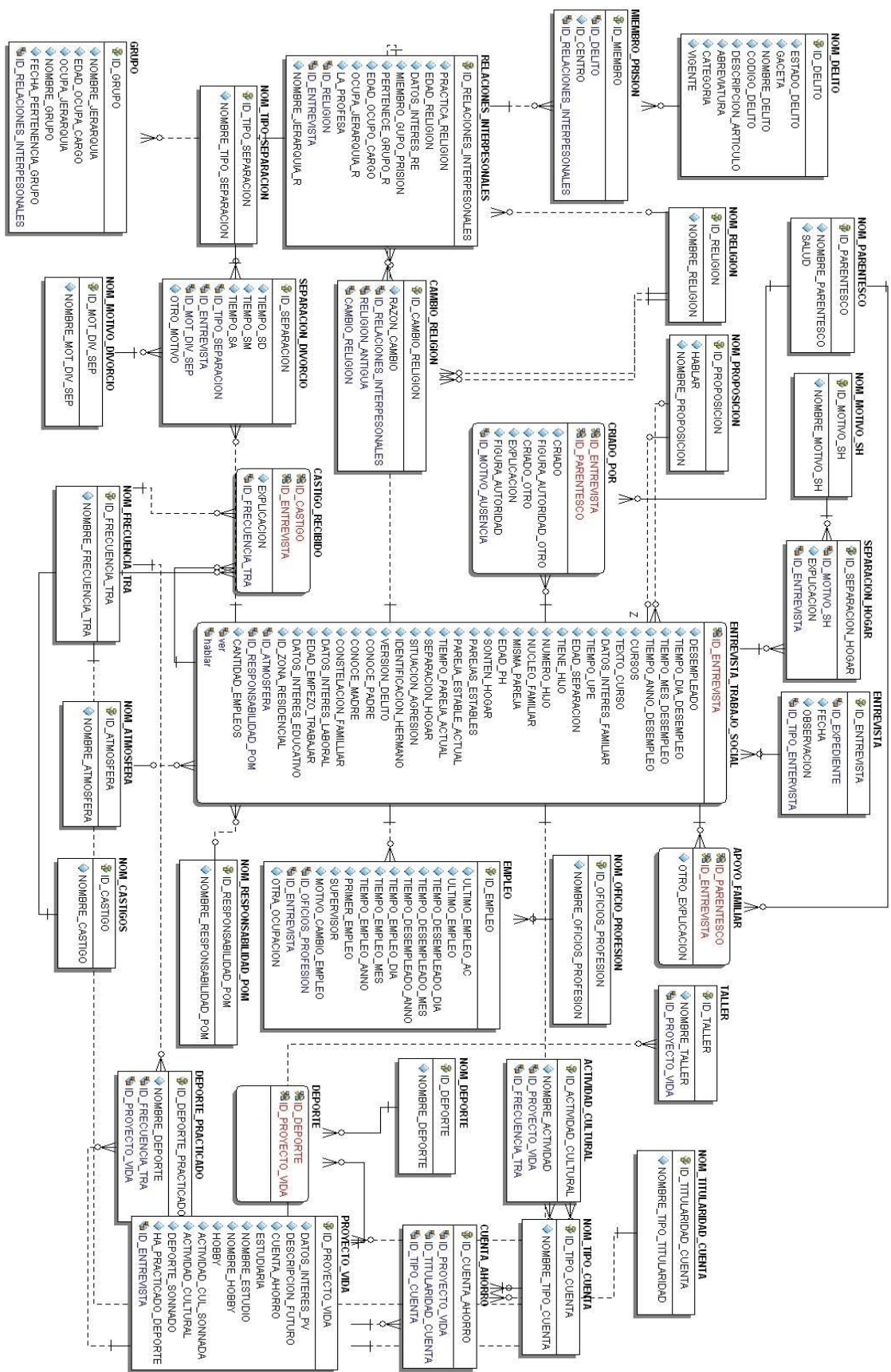


Figura 20 Sub-modelo Físico de Datos Entrevista Trabajo Social

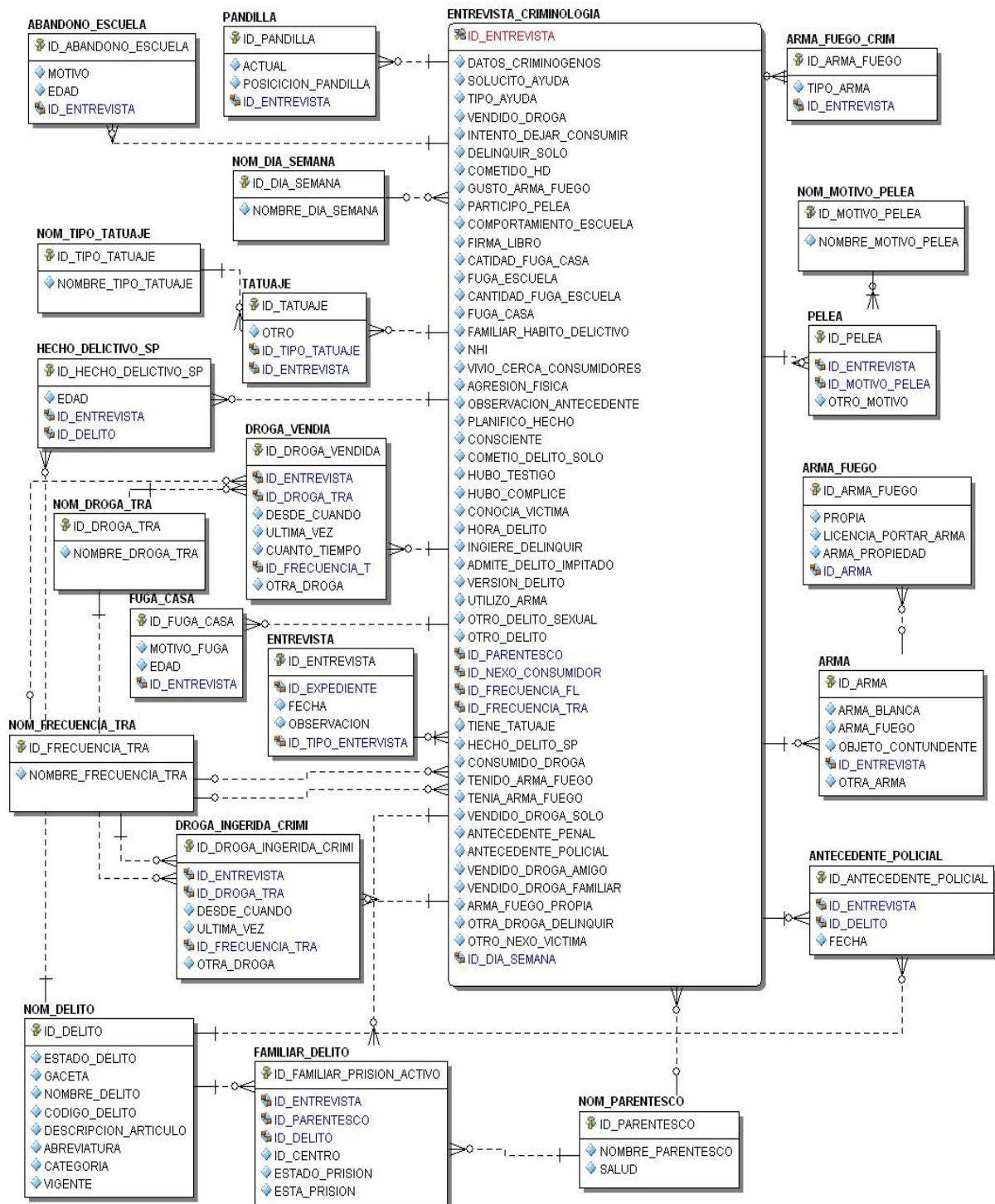


Figura 21 Sub-modelo Físico de Datos Entrevista Criminológica

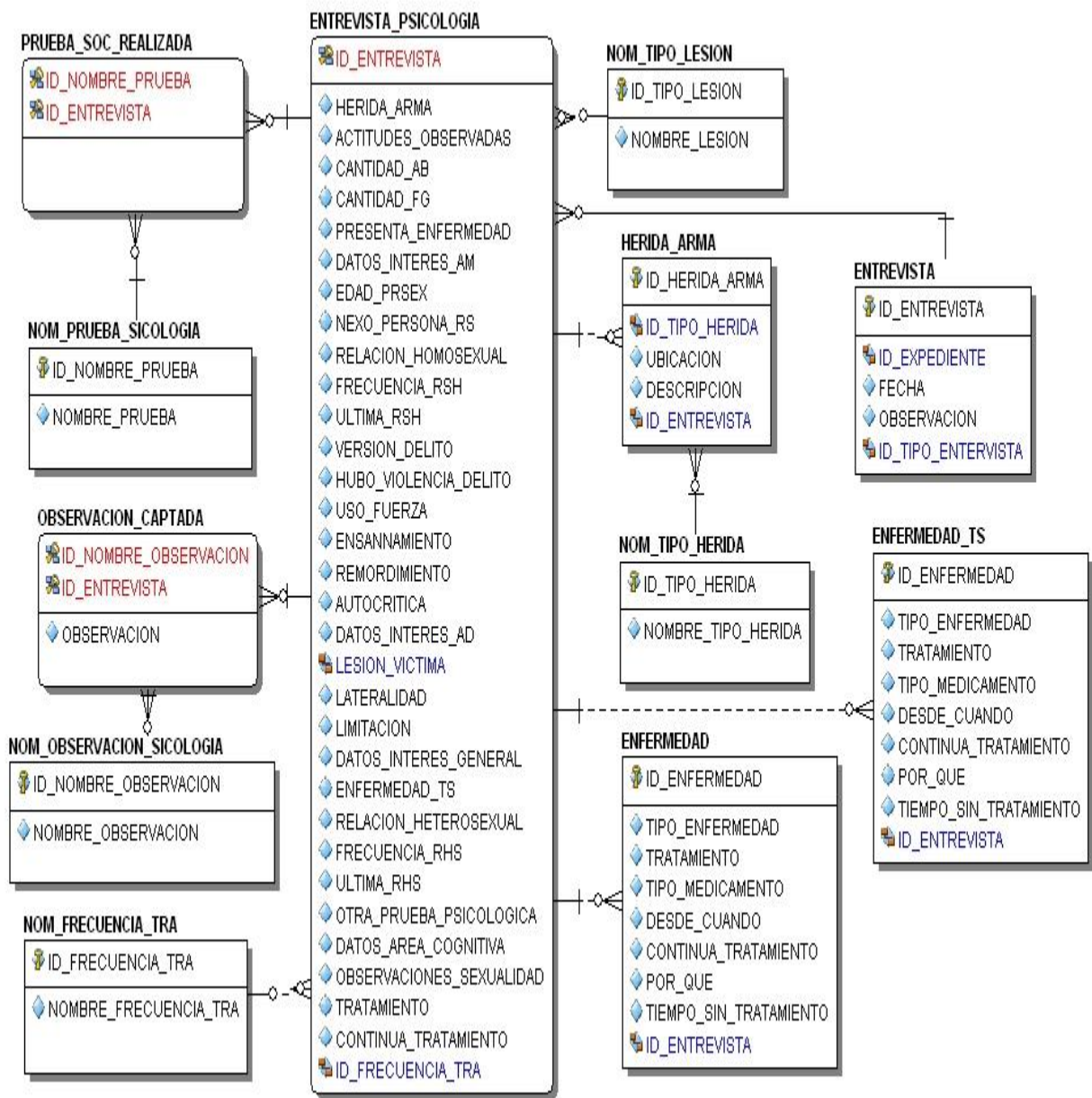


Figura 22 Sub-modelo Físico de Datos Entrevista Psicológica

CONCLUSIONES

Se concluye que se cumplió el objetivo propuesto al realizar el diseño de las capas de Negocio y Acceso a Datos del módulo Observación y Clasificación realizando los diagramas de clases y diagramas de interacción necesarios.

Se diseñó un modelo de datos acorde con las entidades involucradas en la solución del módulo.

Se implementó la solución del problema usando los artefactos brindados por el diseño con las herramientas, tecnologías y convenciones definidas en la arquitectura del proyecto.

Se logró cumplir con lo requerido por el cliente pues el módulo está probado y aceptado por el mismo y se encuentra funcionando en un penal piloto en Venezuela.

RECOMENDACIONES

Se recomienda hacer pruebas del módulo bajo condiciones extremas donde la base de datos contenga una mayor cantidad de registros de los penados. Esto no ha sido posible probarlo en las condiciones actuales de la base de datos, debido a que está en fase de prueba no contiene la cantidad de registros necesarios como para medir el rendimiento en casos extremos.

BIBLIOGRAFÍA

- BAUER, C. and G. KING. *Hibernate in Action*. 2005. p. 1932394-15-X
- CRAIG WALLS, R. B. *Spring in Action Second Edition*. 2008. p. 1-933988-13-4
- IEEE *Recommended Practice for Architectural Description of Software-Intensive Systems*. IEEE Std 1471-2000
- JOHNSON, R. *Professional Java Development with the Spring Framework*. 2005. p. 0764574833
- KUCHANA, P. *Software Architecture Design Patterns in Java*. Auerbach, 2004. p.
- LARMAN, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos.*, Prentice-Hall Hispanoamerica, 1999. p. 8425
- ORIZONDO, A. C. A. *Proyecto Técnico de Asesoría Especializada, Colaboración Médica Odontológica, Comunicación Institucional y Solución Tecnológica para apoyar la modernización del Sistema Penitenciario de la República Bolivariana de Venezuela*. Venezuela, 2006. p.
- OSORIO, M. *Diccionario de Ciencias Jurídicas, Políticas y Sociales*. p.
- PIMENTEL, L. A. and J. E. MARTÍNEZ. *Documento de Arquitectura de Software*. Cuba, 2008. p.
- PIMENTEL, L. A. and I. P. RIVERO. *ArBaWeb: ARQUITECTURA BASE SOBRE LA WEB*. La Habana, Universidad de las Ciencias Informáticas, 2007. p.

GLOSARIO DE TÉRMINOS

API: Especificación de una librería o utilidad que documenta su interfaz y permite su uso sin conocimiento de su interior.

Aplicación informática: Es un tipo de programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo.

Atributos: Son las características o propiedades que definen a un objeto.

Caché: Es un conjunto de datos duplicados de otros originales, con la propiedad de que los datos originales son costosos de acceder, normalmente en tiempo, respecto a la copia en el caché. Cuando se accede por primera vez a un dato, se hace una copia en el caché; los accesos siguientes se realizan a dicha copia, haciendo que el tiempo de acceso medio al dato sea menor.

Caso de uso: Fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.

Clase: Es una abstracción sobre un conjunto de objetos que tienen en común estructura y comportamiento.

Código abierto: Relativo al software para el cual el código fuente está disponible en forma gratuita.

Código o código fuente: Conjunto de instrucciones que componen un programa, escrito en cualquier lenguaje.

Dato: Información en un formato que pueda ser procesado por un ordenador.

Entidad de Dominio: Es una clase que representa conceptos reales del negocio.

Framework: Conjunto de APIs y herramientas destinadas a la construcción de un determinado tipo de aplicaciones de manera generalista.

Herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador): Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Interface: Es un conjunto de constantes y métodos a los que no se les da implementación, también se puede ver como una clase que solo cuenta con métodos abstractos.

Método: Son operaciones que pueden modificar el estado de un objeto o simplemente obtener datos sobre el mismo.

Módulo: Encapsula un conjunto de funciones que debe realizar el sistema, las cuales son agrupadas por tener características muy similares y se definen en la etapa de diseño.

Objeto: Es el resultado de la instanciación de una clase.

Objetos persistentes: Son los objetos que van a contener la información persistente de la aplicación, en este caso los objetos persistentes serán los objetos de dominio.

Patrón: Son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Programación: Se define como el proceso de creación de un programa de computadora, mediante la aplicación de procedimientos lógicos.

Prueba de regresión: Son pruebas que se hacen seguidamente de modificaciones a programas previamente probados para asegurar que no hayan sido introducidos nuevos errores, como resultado de los cambios realizados.

Servlet: Es una clase Java que ofrece funciones suplementarias al servidor.

Software: Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

Subsistema: Un subsistema se refiere a un conjunto de módulos que por razones de similitud o de perseguir objetivos comunes, son agrupados.

