**UNIVERSIDAD TECNICA PARTICULAR DE LOJA**



**PROCESOS DE INGENIERÍA DE SOFTWARE [B]**

**SISTEMA PARA EL MANEJO DE PROPUESTAS DE PROYECTOS DE FIN DE TITULACIÓN**

**DOCENTE:**

ING. DANIEL GUAMÁN

**PROFESIONAL EN FORMACION:**

CARLOS VINICIO CARAGUAY CARAGUAY

**LOJA – ECUADOR**

**2015**

**INDICE**

1. **OBJETIVOS**
2. **PROBLEMA**
3. **SOLUCION**
4. **CONCLUSIONES**

**OBJETIVOS**

**GENERAL:**

Implementar un sistema que permita el registro y manejo de propuestas de fin de titulación de la carrera de Ingeniería en Sistemas de la Universidad Técnica Particular de Loja.

**OBJETIVOS DETALLADOS:**

* Registrar propuestas de proyecto por parte de estudiantes o docentes.
* Revisión y calificación de propuestas de proyecto por parte de director como aprobado o reprobado.
* Visualización de reportes de propuestas de proyectos.

**PROBLEMA**

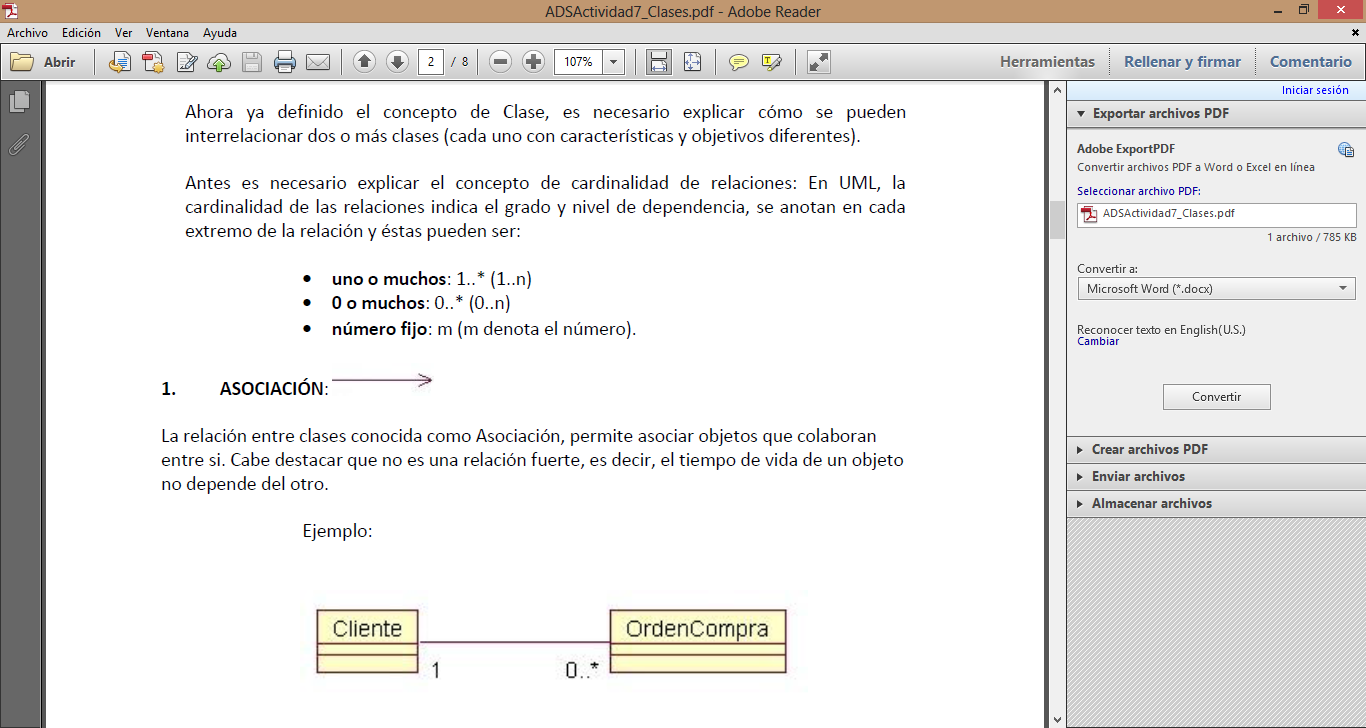
La organización y manejo de ideas o propuestas de proyectos que se emiten por parte de estudiantes o docentes de la carrera de Ingeniería en Sistemas de la Universidad Técnica Particular de Loja se ha venido llevando de una forma muy tradicional, la cual no permite un adecuado estudio de dichas ideas, es por eso que se ha visto la necesidad de automatizar dichos procesos con el fin de satisfacer necesidades de estudiantes, docentes y directivos que son los involucrados.

**SOLUCIÓN**

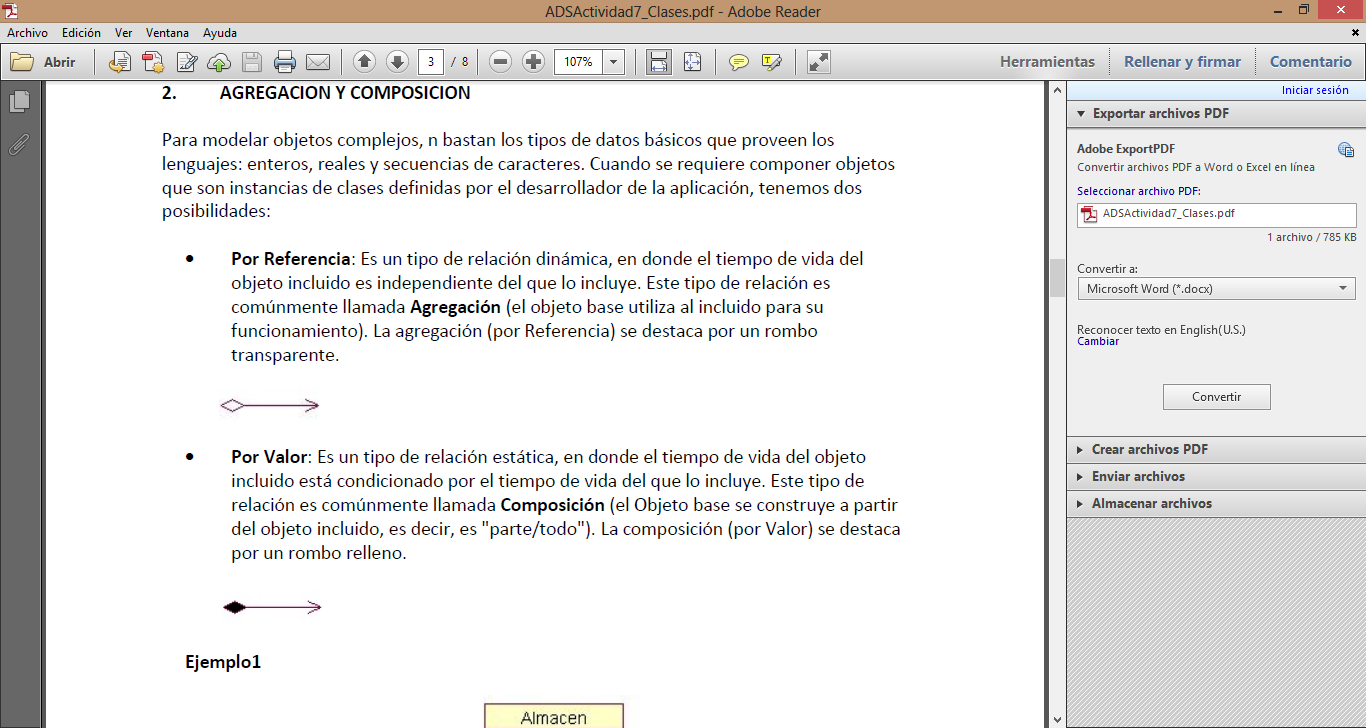
Dado el panorama que se tiene actualmente, se ha creído conveniente implementar un sistema que agilice los diferentes procesos, razón por la cual se procederá a automatizar todos los procesos que se llevan a cabo desde que un estudiante o docente plantean una propuesta de proyecto hasta que el mismo es calificado como aprobado o reprobado por parte del director de la titulación y finalmente se convierte en proyecto de fin de titulación.

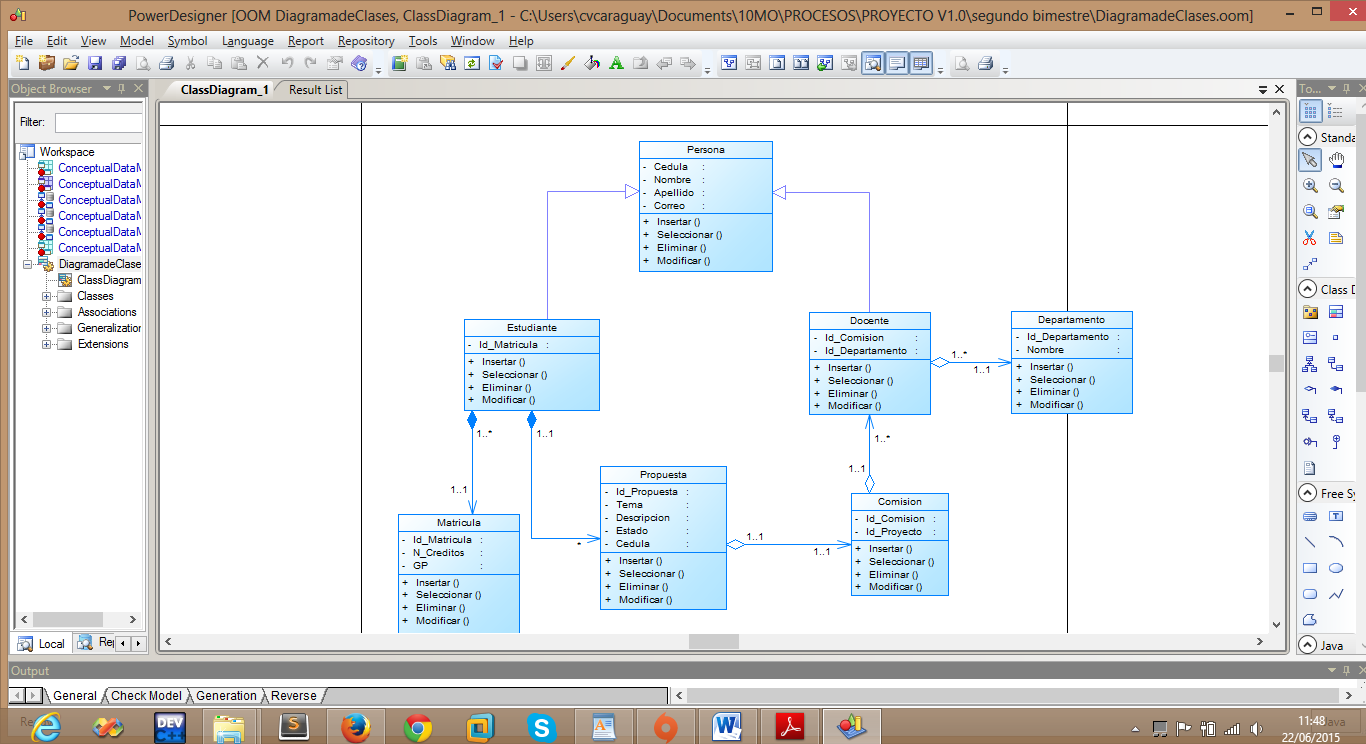
**DIAGRAMA DE CLASES**

**Asociación:** Permite asociar objetos que colaboran entre sí.

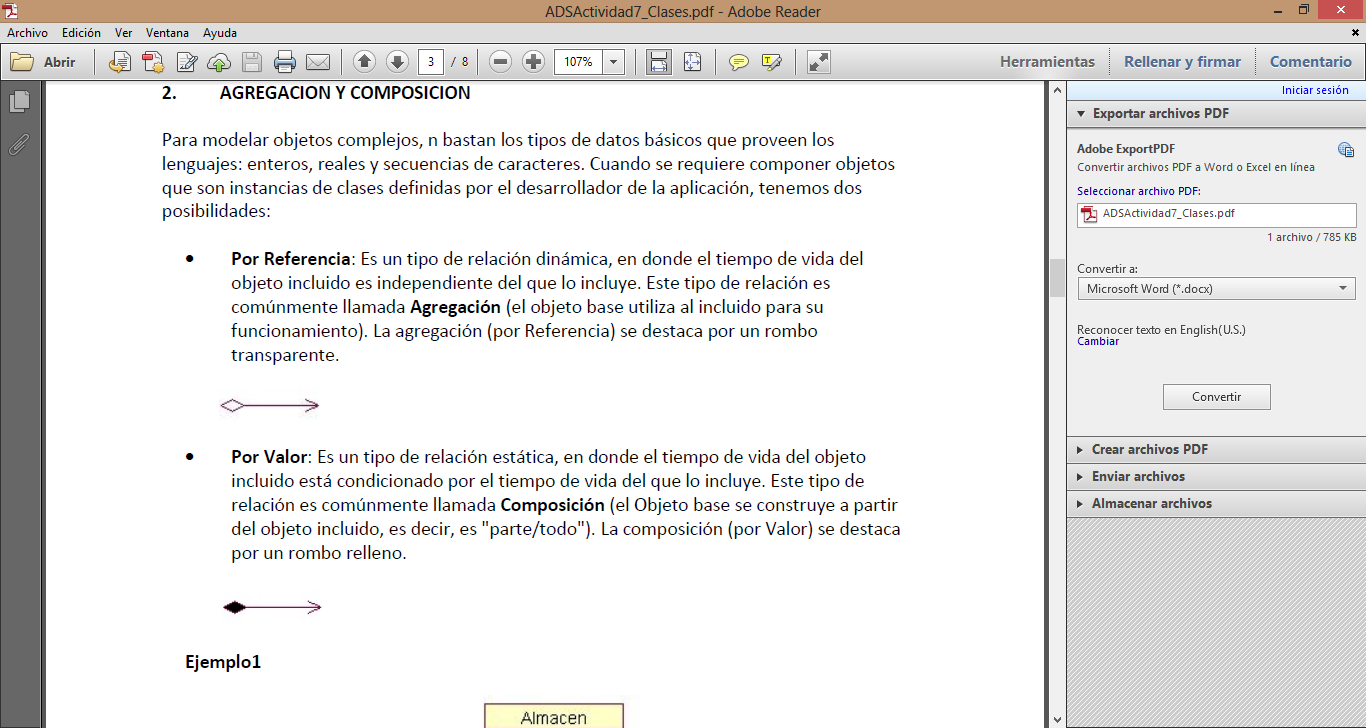


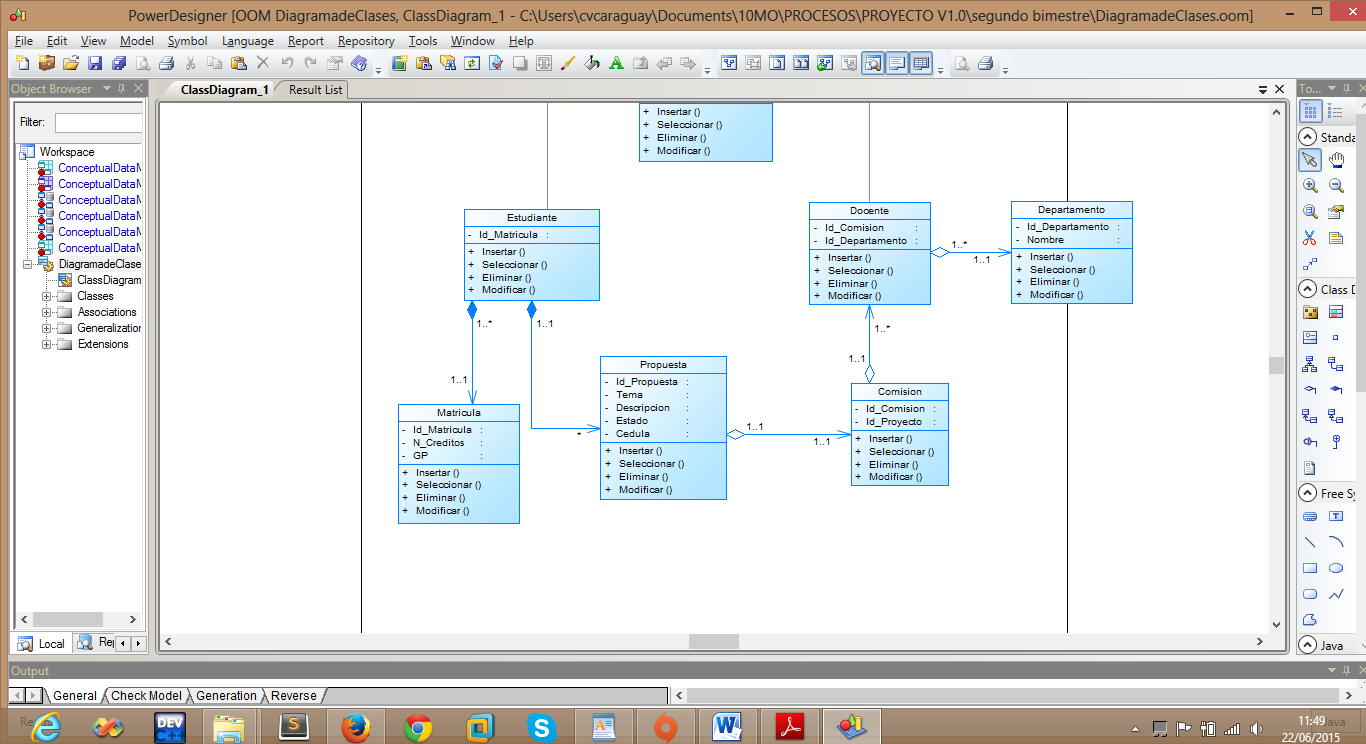
**Agregación:** Tipo de relación dinámica en donde el tiempo de vida del objeto incluido es independiente del que lo incluye.



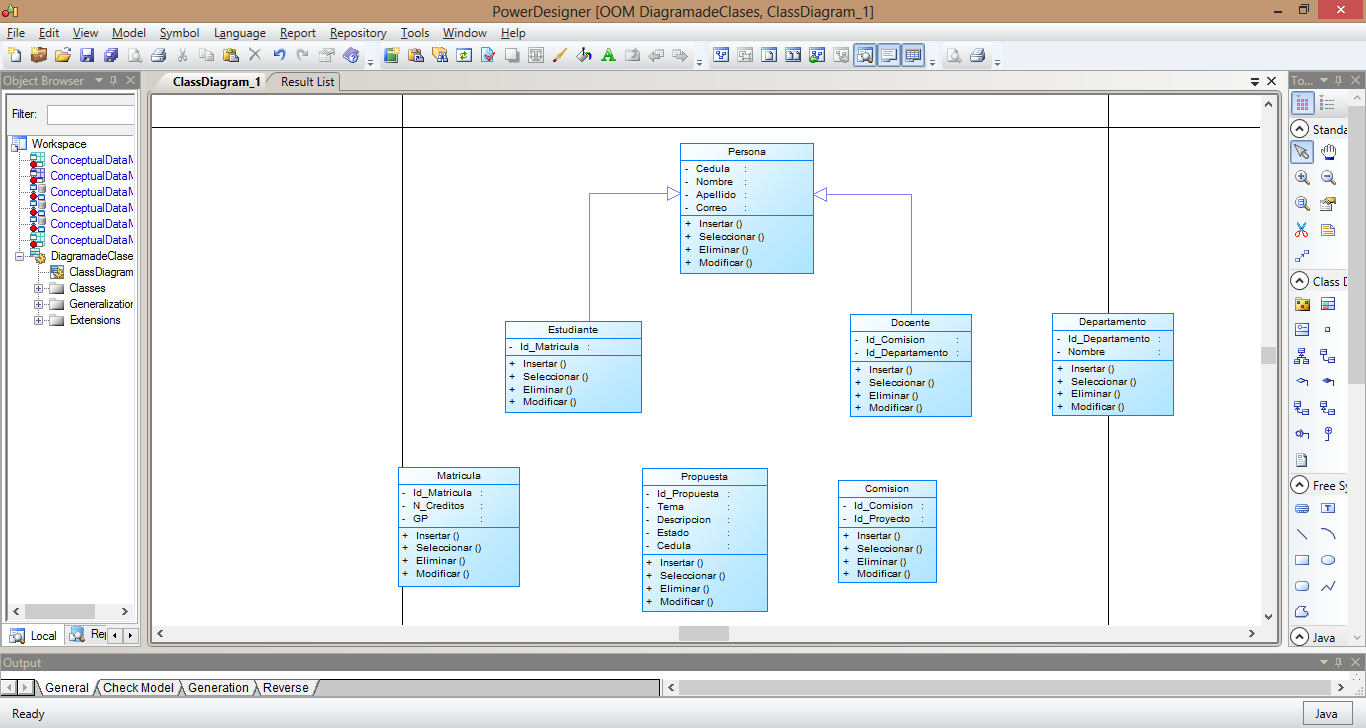


**Composición:** Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del que lo incluye.

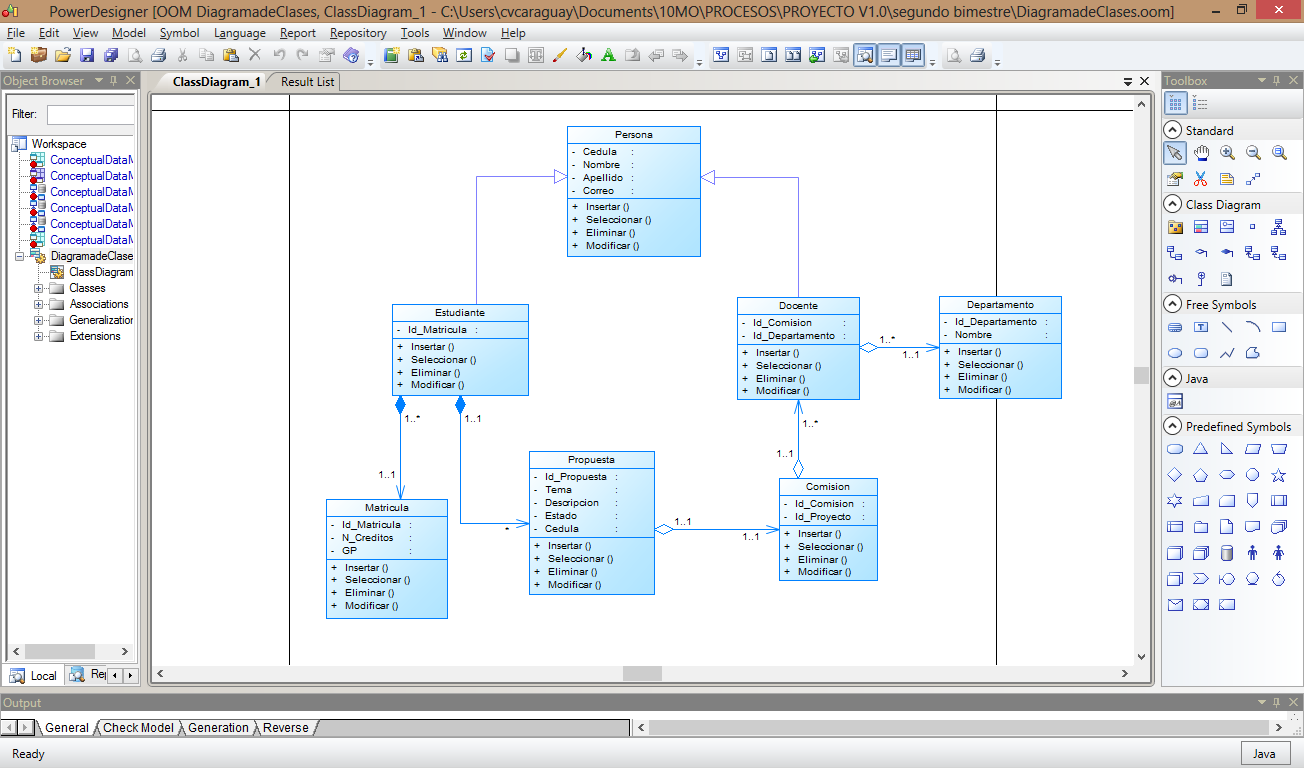




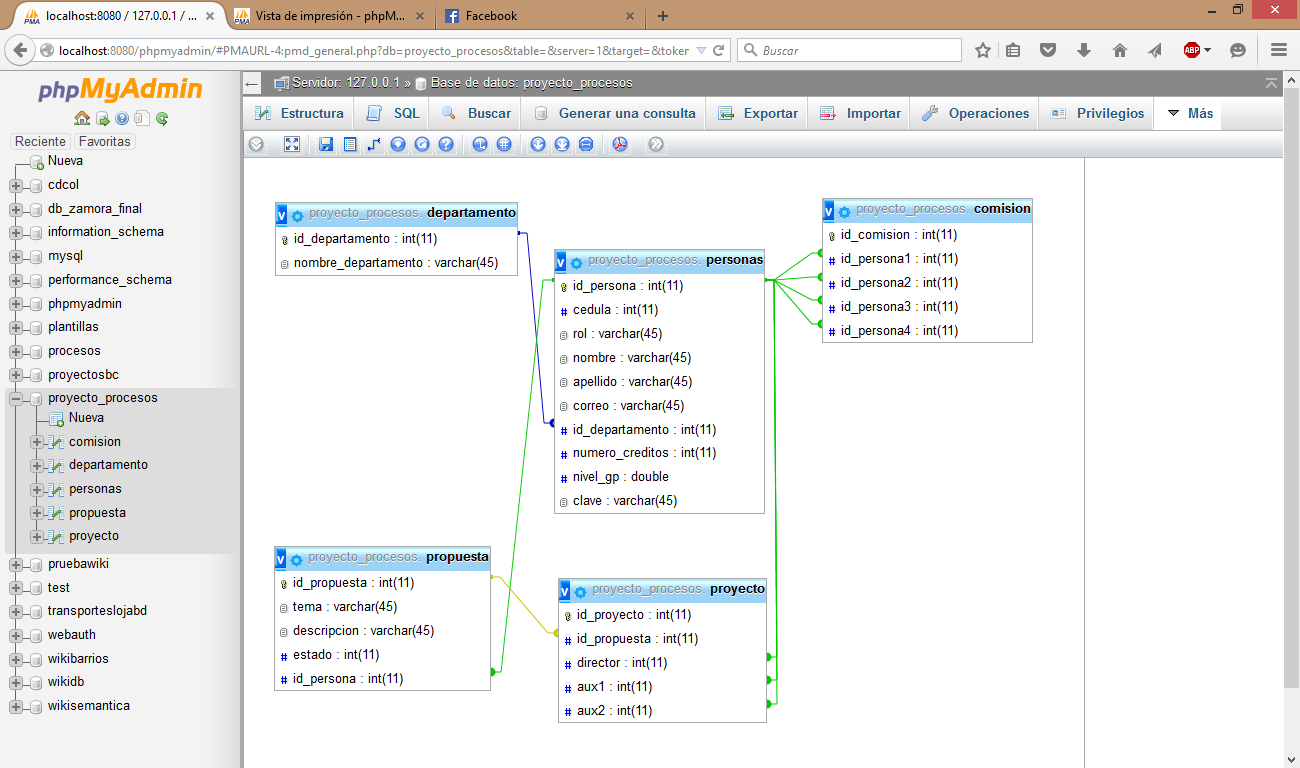
**Generalización:** Indica que una subclase hereda los métodos y atributos especificados por una superclase, por ende la Subclase además de poseer sus propios métodos y atributos, poseerá las características y atributos visibles de la superclase (public y protected)



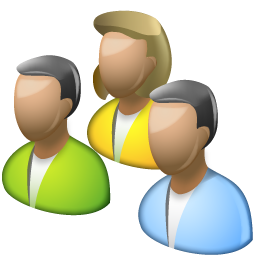
FINAL



**MODELO ENTIDAD - RELACIÓN:**

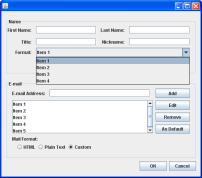


**ARQUITECTURA DE LA APLICACIÓN:**

****

Login

CAPA DE PRESENTACIÓN



GUI

CAPA DE NEGOCIO

JPA

Controladores

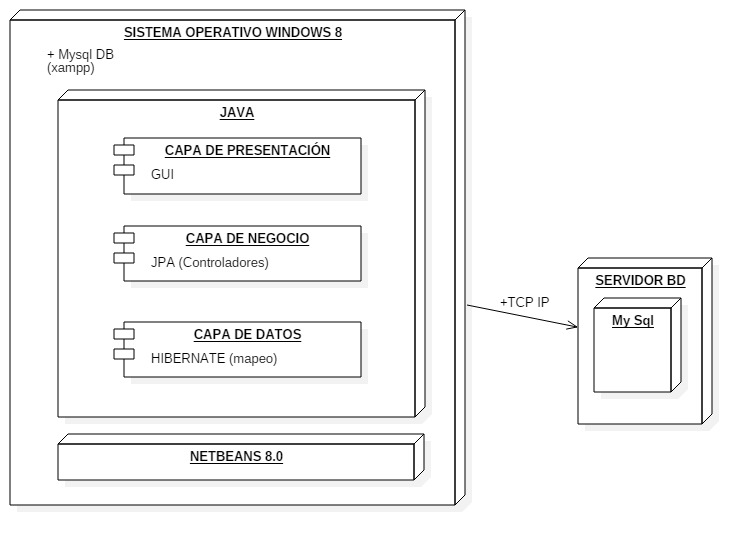
CAPA DE DATOS

HIBERNATE

Link de conexión JDBC



**DIAGRAMA DE ALTO NIVEL**



**EXPLICACIÓN DE LA CONSTRUCCIÓN DE LA APLICACIÓN:**

La aplicación realizada es una aplicación de escritorio que tiene como funcionalidades permitir: el registro de propuestas para proyectos de Fin de Titulación de la Carrera de Ing. En Sistemas de la UTPL. Además permite calificar dichas propuestas y esta una vez que la apruebe se convierte en proyecto de fin de titulación.

Se utilizó una **arquitectura 3 capas:**

Se enfoca en todo sistema en general que gestiona datos tendrá una base de datos para guardar esos datos y una interfaz de usuario que será con la que interactúan los usuarios. Además, una parte del sistema se encargará de procesar los datos y gestionar lo que se hace con ellos, razón por la cual se utilizó para el desarrollo del presente sistema.

La arquitectura en tres capas lo que hace es dividir el sistema en tres partes diferenciadas, de tal forma que cada capa solo se comunique con la inferior. Esas tres capas se denominan:

**Presentación:** Aquí se crea la interfaz de usuario que debe ser amigable y la función que tiene es pasar las acciones que realice el usuario a la capa de negocio.

**Negocio:** Se gestiona la lógica de la aplicación, es donde se decide qué hacer con los datos. Por ejemplo en el caso del login en este capa se llegaría a una conclusión si un usuario está registrado o no y si este es de tipo estudiante, docente o director, para según eso mostrar las ventanas al usuario.

**Datos:** que es donde se guardan los datos, es donde se gestiona lo relativo con la base de datos, creación, borrado, etc.

Cuando cada capa se comunique solo con la inmediatamente inferior, se consigue que si hay que realizar un cambio no se tope todo el código del sistema, sino solamente la capa involucrada.

Para el desarrollo de la aplicación se utilizó:

**JAVA:** Como lenguaje de programación java nos permitió realizar un aplicación rápida, segura y confiable, existen muchas aplicaciones y sitios web que han sido desarrollados en este lenguaje por lo que se creyó conveniente utilizar este lenguaje de programación.

**NET BEANS:** Se utilizó como entorno de desarrollo dado que es libre, y que principalmente se encuentra hecho para el lenguaje de programación JAVA. Es un producto libre y gratuito sin restricciones de uso. Nos permite desarrollar aplicaciones a partir de un conjunto de componentes de software llamados módulos, dichos módulos contienen clases de java las mismas que interactúan con las APIS de NetBeans. Soporta el desarrollo de todo tipo de aplicaciones (escritorio, web, aplicaciones móviles, etc.)

**JPA:** Java Persistence API, esta API es un framework del lenguaje de programación Java que maneja datos relacionales en aplicaciones.

El principal motivo por el que se utilizó esta API es porque:

* Incrementa la respuesta al cambio de las aplicaciones
* Reducción de código de acceso de datos

**HIBERNATE:** Nos permite mapear una clase JAVA con una tabla de base de datos lo que facilita el trabajo con la persistencia de objetos. La persistencia es la capacidad para guardar y recuperar datos desde un medio de almacenamiento que en nuestro caso sería la base de datos en MYSQL. Involucra: mapeo entre Clases Java y las tablas de Bases de Datos, conocido como ORM (Object-Relational Mapping).

**MYSQL:** Es un sistema de gestión de base de datos relacionales. Se utilizó para guardar, actualizar y recuperar la información que utiliza nuestra aplicación.

**CONCLUSIONES**

REQUERIMIENTO NO FUNCIONALES:

A continuación se mostraran algunos de los atributos con los que deberá contar el producto final que se le mostrara al cliente:

|  |  |
| --- | --- |
| **ATRIBUTO** | **APLICACION** |
| **Confiabilidad** | De acuerdo a las pruebas realizadas el sistema tiene un alto nivel de confiablidad dado que toda la información que se maneja va a estar disponible en todas las funcionalidades del sistema, además se garantiza su buen funcionamiento durante todo el período de su ejecución. |
| **Disponibilidad** | Desde el punto de vista del software la aplicación va a estar disponible cuando el usuario lo requiera, pero ya en coordinación con el hardware, dado la naturaleza de que se realizó una aplicación de escritorio, la misma estará instalada en un equipo que solamente se ubicará en el edificio de la UGTI en el área de Ciencias de la Computación. |
| **Seguridad** | El sistema garantizará la seguridad de los datos que se manejan en el sistema, dado que existen varios tipos de usuarios lo que permite mantener un control detallado de la información y módulos a los que debe acceder cada tipo de usuario. |
| **Mantenibilidad** | El mantenimiento y modificaciones que se puedan hacer en el sistema, dependerá de las aplicaciones con las que se quiera integrar y las plataformas donde se quiera poner. |
| **Eficiencia** | El sistema se ha de comportar de manera eficiente y rápida, tanto en el registro de propuestas, y calificación de las mismas hasta convertirse en proyectos. |
| **Fácil Manejo** | La aplicación consiste en una interfaz suficientemente amigable, intuitiva y fácil de usar para cualquier tipo de usuario. |
| **Almacenamiento** | La base de datos garantiza la integridad de los datos tanto de estudiantes, docentes y propuestas y proyectos de fin de titulación. |