



INSTITUTO POLITÉCNICO NACIONAL

**UNIDAD INTERDISCIPLINARIA DE
INGENIERÍA CAMPUS ZACATECAS**

PROGRAMACIÓN ORIENTADA A OBJETOS

SANDRA MIREYA MONREAL MENDOZA

PROYECTO FINAL

Equipo : “XD”

María José Borjón Ibarra (2017670052)

César Iván Martínez(2017670821)

María Fernanda Román(2017670482)

2CM1

Zacatecas, Zac . 05 de Diciembre de 2017



Índice :

portada.....	1
índice.....	2
Introducción.....	2-3
Objetivo.....	4
Desarrollo.....	4-6
Observaciones.....	6
Conclusiones.....	6
Referencias.....	7

Introducción :

La POO es una técnica para desarrollar soluciones computacionales utilizando componentes de *software* (objetos de *software*).

Objeto: Componente o código de *software* que contiene en sí mismo tanto sus características (campos) como sus comportamientos (métodos); se accede a través de su interfaz o signature.

La POO es un paradigma de la programación de computadores; esto hace referencia al conjunto de teorías, estándares, modelos y métodos que permiten organizar el conocimiento, proporcionando un medio bien definido para visualizar el dominio del problema e implementar en un lenguaje de programación la solución a ese problema.

La POO se basa en el modelo objeto donde el elemento principal es el objeto, el cual es una unidad que contiene todas sus características y comportamientos en sí misma, lo cual lo hace como un todo independiente pero que se interrelaciona con objetos de su misma clase o de otras clase, como sucede en el mundo real.

Es un sistema que almacena datos que están relacionados.

Es un repositorio en donde guardamos información integrada que podemos almacenar y recuperar.

- Un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos. [1].

BASE DE DATOS

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente



los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico.

Las bases de datos tradicionales se organizan por campos, registros y archivos. Un campo es una pieza única de información; un registro es un sistema completo de campos; y un archivo es una colección de registros. Por ejemplo, una guía de teléfono es análoga a un archivo. Contiene una lista de registros, cada uno de los cuales consiste en tres campos: nombre, dirección, y número de teléfono.

A veces se utiliza DB, de *database* en inglés, para referirse a las bases de datos.

Objetivo:

Generar un sistema para llevar a cabo la gestión de una biblioteca, que permita el registro de diferentes tipos de libros , y usuarios . Y además que permita llevar a cabo el préstamo de libros y consulta interna (únicamente un registro de los ítems que son consultados en las instalaciones) que únicamente permite conocer los volúmenes más .

Desarrollo :

Planteamiento final:

Sistema de Gestión de una biblioteca:

El sistema de gestión bibliotecario permitirá el registro de materiales disponibles para sus préstamos: libros y tesis. También se tiene un registro de los usuarios, y el profesores .

Para los ítems requiere, la clave de identificación, nombre, autor, año de publicación, lugar de publicación , los libros tienen un : número de edición, editorial, número de páginas, y categoría así como su disponibilidad; para las tesis es necesaria el registro de el o los asesores, el tema y categoría.

Para los usuarios se requiere tener un registro de clave única de usuario, nombre, apellido materno, apellido paterno, dirección, teléfono, si son alumnos se requiere de una matrícula, programa académico, grupo y semestre; y para los profesores, además de los datos pertenecientes a cualquier “ usuario”, posee una clave y una academia a la que pertenece .



Al momento de realizar un préstamo es necesario el usuario, la fecha de salida y la fecha de vencimiento del préstamo. El préstamo del material solicitado tendrá un máximo de 5 días para su devolución en caso de necesitar una renovación se permitirá un máximo de tres veces. Al momento regresar el material se debe considerar el préstamo al que pertenece, fecha de devolución

En el caso de las consultas es únicamente un registro interno de los préstamos que se hicieron.

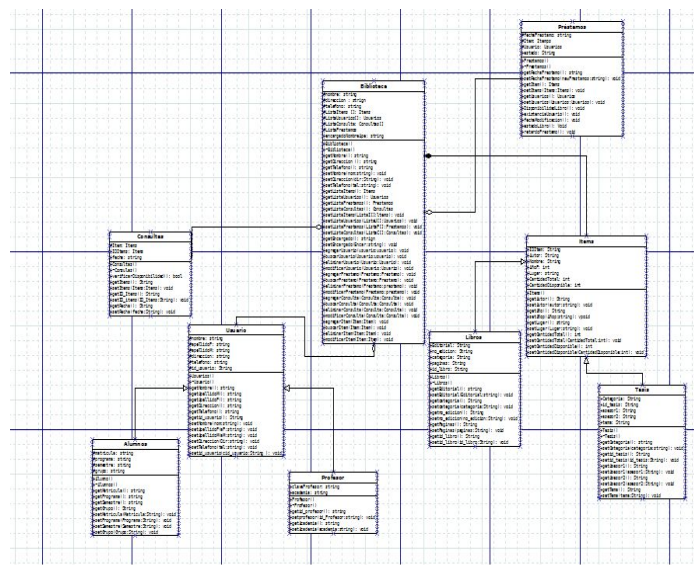
Diseño de las clases :

Tomando en cuenta el planteamiento inicial realizamos la siguiente separación para el POO.

Los , existen diferentes usuarios , pero tienen ciertos atributos, es el mismo caso de los items, pues tenemos diferentes ítems que comparten algunos atributos básicos, motivo por el cual hicimos la relación de herencia entre ellos (cada uno de los grupos por separado) .

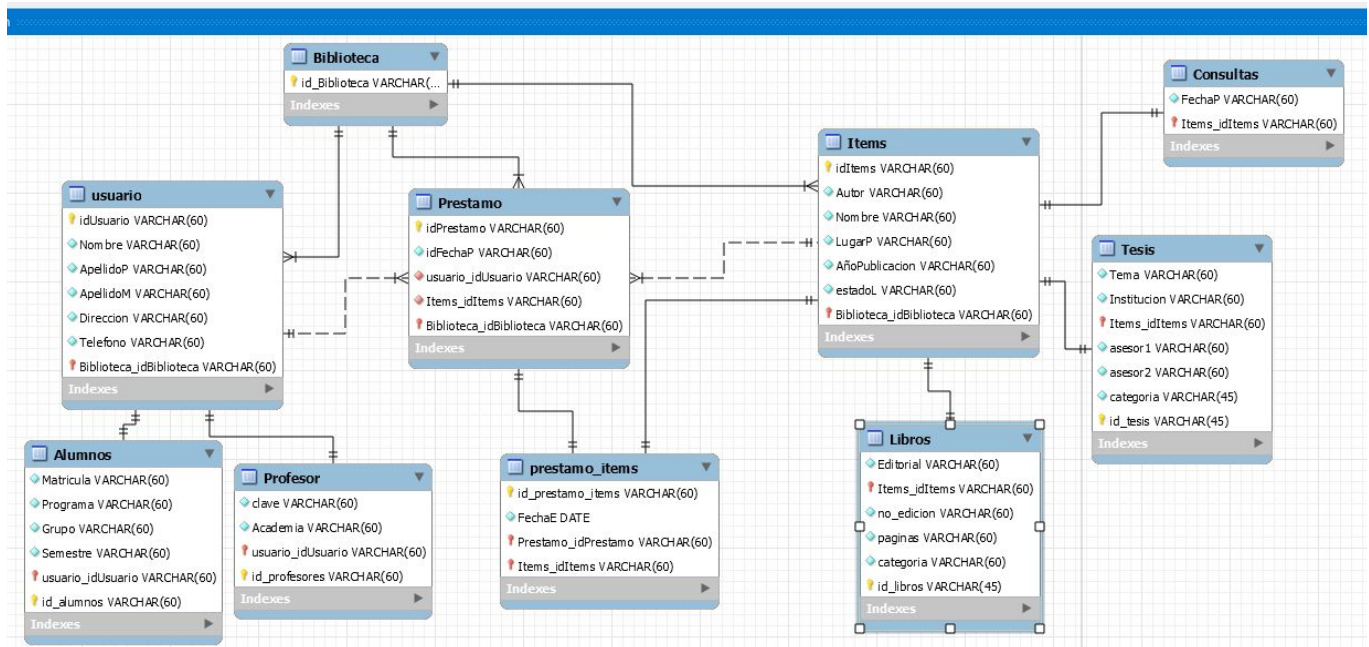
Tenemos además una clase prestamo donde nos encargaremos de gestionar el manejo y guardado de los préstamos, que implica la relación entre ítems y el usuario .

Al igual que las consultas , implican una relación que implica manejar objetos que no pertenecen al objeto que se está construyendo . Lo cual implica un tipo de relación de agregación o de composición.





Diseño de la base de datos :



Diseño de la base de datos :

Para el diseño de la base de datos para la biblioteca, se identificaron las principales entidades y funcionalidad con la cual cumpliría, en primer instancia la definición del tipo de "Usuarios" es decir, dentro de la biblioteca existen dos tipos de usuarios los cuales son "Alumnos" y "Profesores" los cuales comparte ciertos atributos como:

- *id_Item
- *Autor.
- *Nombre.
- *lugar de publicación.
- *año de publicación.
- *estado

Sin embargo, algunos de las de las diferencias son:

(Alumnos)	(Profesores)
----- -----	
Editorial	clave
No. de edición.	categoría
páginas	Academia
categoría.	
----- -----	



Aunque para la realización de la base de datos necesitábamos la creación de ambas tablas, pero los "id" de cada una de las tablas se concatenaron en "usuarios" para facilitar las búsquedas, mantener la información de ambos sin afectar la herencia y atributos individuales.

Realizamos algo similar para los Items, ya que existía una relación de herencia para los libros y las tesis, en este caso las tablas tanto de libros y tesis llevaban como llave foránea el "id_Items", para mantener la relación y los atributos que cada uno contienen, de esta manera fue mucho más fácil generar la tabla de préstamos la cual contiene su llave principal "id prestamos" y a su vez se lleva las llaves foraneas de usuarios(en la cual ya lleva el tipo de usuario), la de items (lleva el tipo de ítem) así como la fecha en la que se ha realizado el préstamo.

En el caso de las consultas se toma como referencia la Fecha con la que se registro préstamo y contiene la llave foránea de la biblioteca ya que es que es en donde se realizan las búsquedas. Por último la generamos una tabla de "prestamos item" la cual nos permitirá realizar las modificaciones o búsquedas de préstamos de los libros y las tesis, esta tabla contiene la llave identificadora de la tabla "id prestamos items", " id préstamo ", "it_items" y la fecha en la que el material solicitado debe ser devuelto a la biblioteca.

Observaciones

Decidir el proyecto: Al discutir las diferentes las diferentes posibilidades para el desarrollo del proyecto final y un poco ignorando los temas que veríamos durante el curso de programación orientada a objetos nos pareció la mejor idea elegir algo que modele una situación del mundo real, y además nos permitiera reforzar conocimientos de niveles previos(bases de datos).

En el proceso de planeación nos dimos cuenta de que al unir una base de datos y un proyecto de java pese a tener conceptos similares no podemos hacer la separación de entidades/ clases , de la misma manera, pues pertenecen a diferentes paradigmas.

Creamos entorno con las ideas generales para la diferentes partes (bases de datos , código y parte gráfica) , pero por más detalles que intentamos prever al momento de unir las diferentes partes se presentaron situación imprevistas e incluso que requerían del conocimiento de conceptos que no conocemos.

Conclusión :



Las cuestiones más importantes que pudimos rescatar (al menos para este tipo de proyectos) es que no debemos asumir que la separación de las entidades en base desde la base de datos y la de las clases no es la misma. Si bien conservan aspectos similares debemos recordar que son “ paradigmas” diferentes.

En cuestión con la utilización del paradigma de POO (a lo largo del curso) , presentó una perspectiva diferente de como realizar las cosas. Nos permite generalizar y ahorrar código, pero para que cumpla su función de la manera es necesario entender bien el concepto y mucha práctica, con práctica la separación de clases, y el aprendizaje de los distintos conceptos o métodos que nos ofrece el paradigma y en nuestro caso el lenguaje java.

Referencias :

[1] "Programación Orientada a Objetos", Fecha de búsqueda: "04/Diciembre/2017",[Online],Available: <https://msdn.microsoft.com/es-es/library/bb972232.aspx>

[2],"Bases de Datos", Fecha de búsqueda: "04/Diciembre/2017",[Online],Available: <http://www.monografias.com/trabajos34/base-de-datos/base-de-datos.shtml#ixzz50M3ABpHn>

[3],"Conexion a mysql desde java", Fecha de búsqueda: "04/Diciembre/2017",[Online],Available: http://monillo007.blogspot.mx/2012/04/conexion-consulta-y-ejecucion-de_19.html