

Modelo Orientado a Objetos

Equipo:

Ariel Alonso Tabasco

Roberto Carlos Cetina Dzul

Arabely Silvestri Hau May

Nancy Gabriela Par Puc

Modelo Orientado a Objetos



Conceptos básicos



¿Qué es?



Proceso de desarrollo



Análisis



Diseño



Modelos de diseño



Método



Patrones de diseño

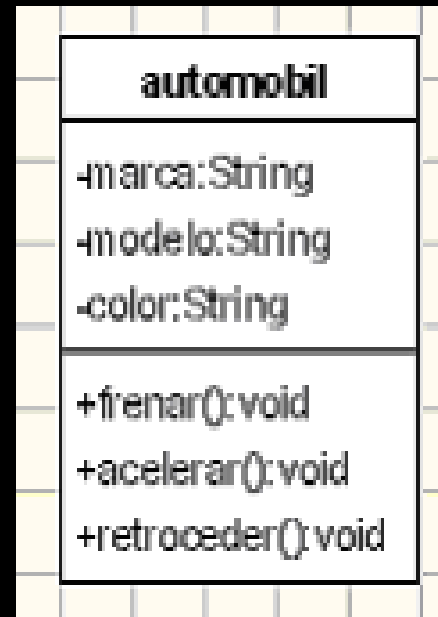


Conclusión



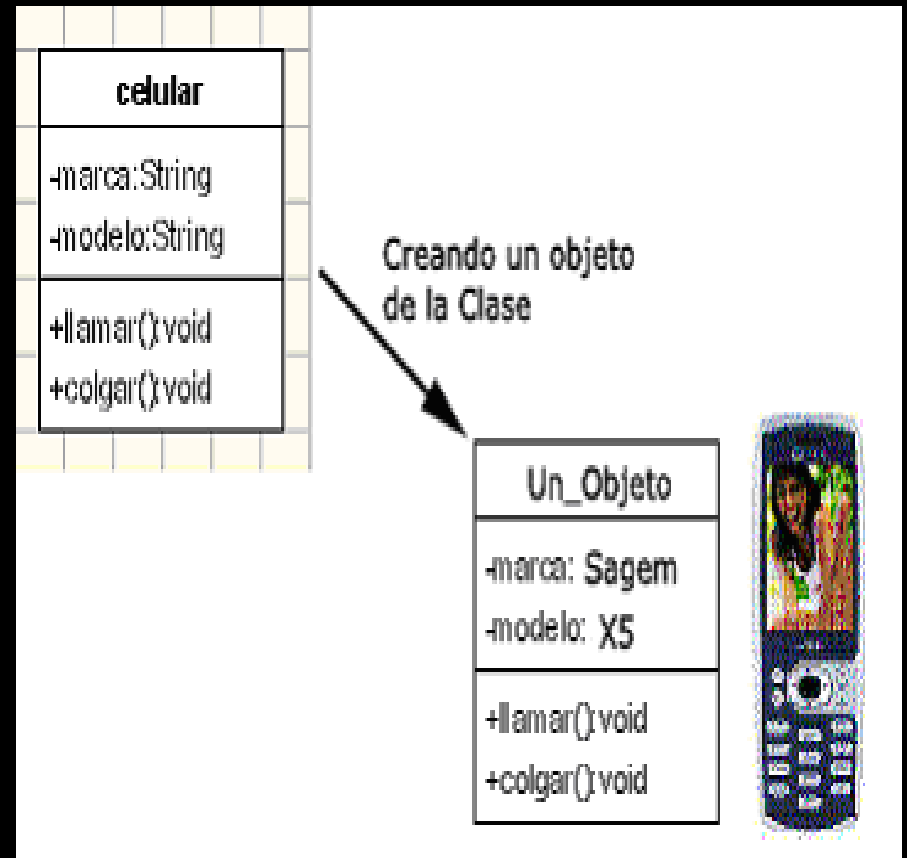
Objetos

- Todo el programa está construido en base a diferentes componentes (Objetos)
- Todo objeto del mundo real tiene 2 componentes: características y comportamiento.



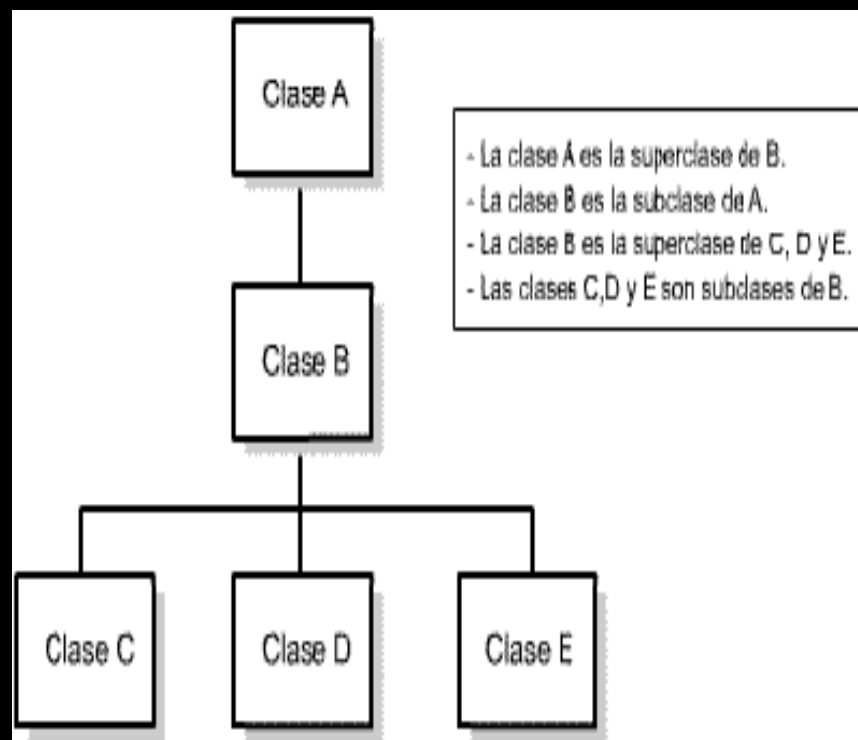
Clase

- Una clase es una plantilla genérica para un conjunto de objetos de similares características.



Herencia

- La herencia básicamente consiste en que una clase puede heredar sus variables y métodos a varias subclases



Envío de mensajes

- Los mensajes son invocaciones a los métodos de los objetos.



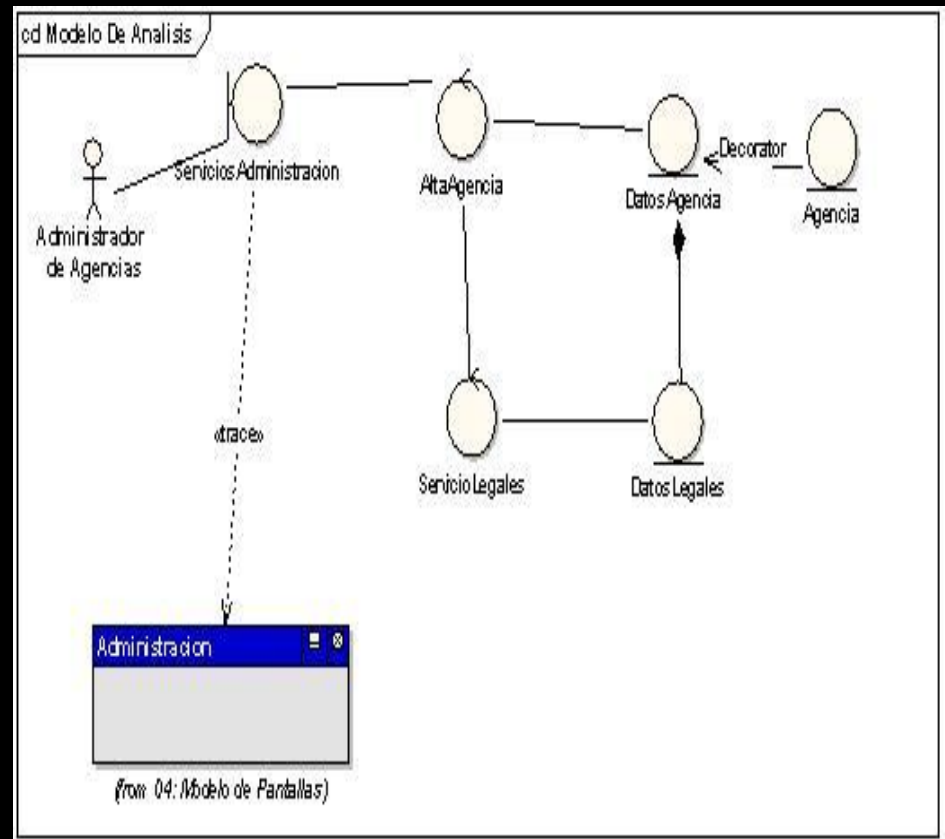
¿Qué es?

- Es una técnica de diseño, la cual se caracteriza por la determinación y delegación de responsabilidades.



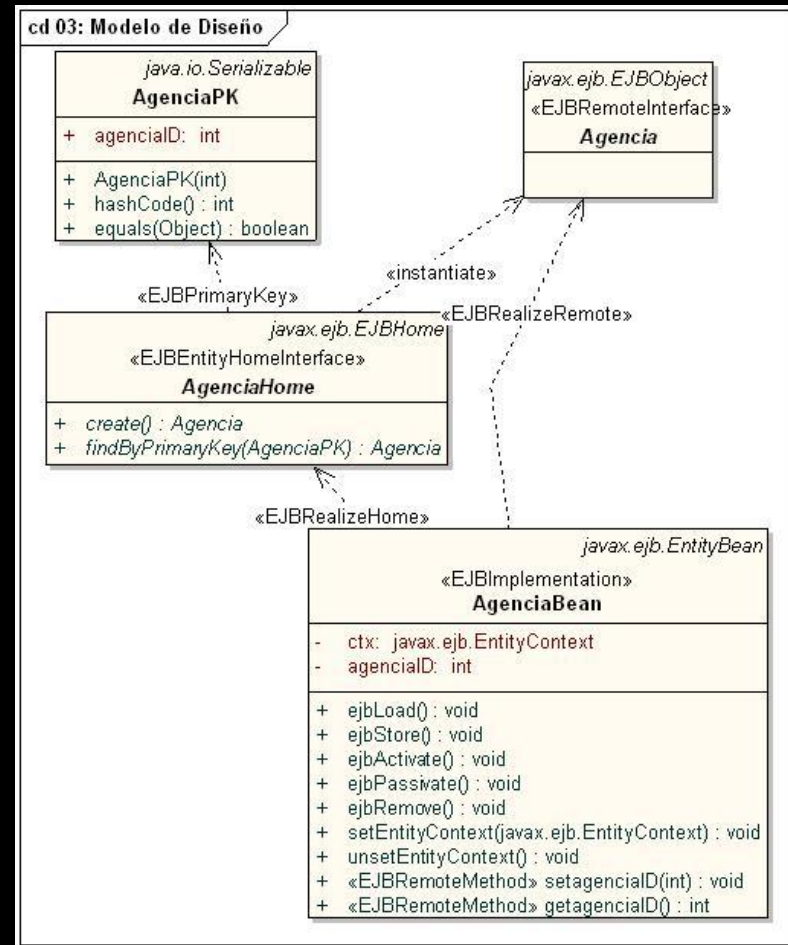
Análisis orientado a objetos

El modelo del análisis orientado a objetos ilustra información, funcionamiento y comportamiento.



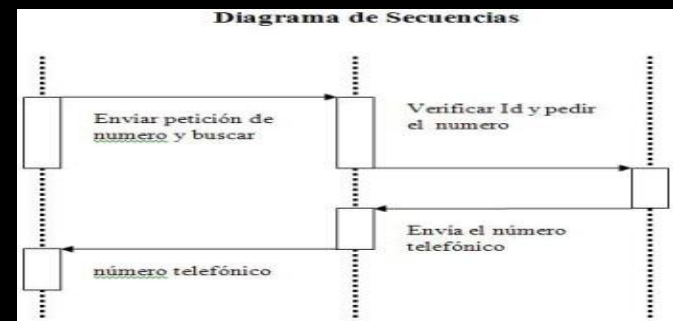
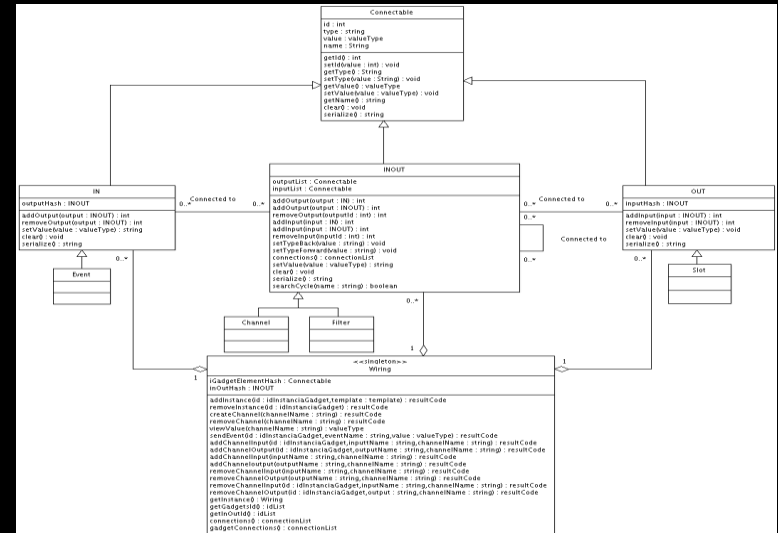
Diseño orientado a objetos

- El diseño orientado a objetos transforma el modelo del análisis en un modelo de diseño que sirve como anteproyecto para la construcción de software.



Modelos del diseño

- **Estáticos.** Estructura de subsistemas y/o clases y sus relaciones.
- **Dinámicos.** Se describen las estructuras que muestran la interacción entre objetos. Ejemplos de UML: diagramas de secuencia, diagramas de estado



Patrones del diseño

- Son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.

Tipos: de creación, estructurales, de comportamiento.



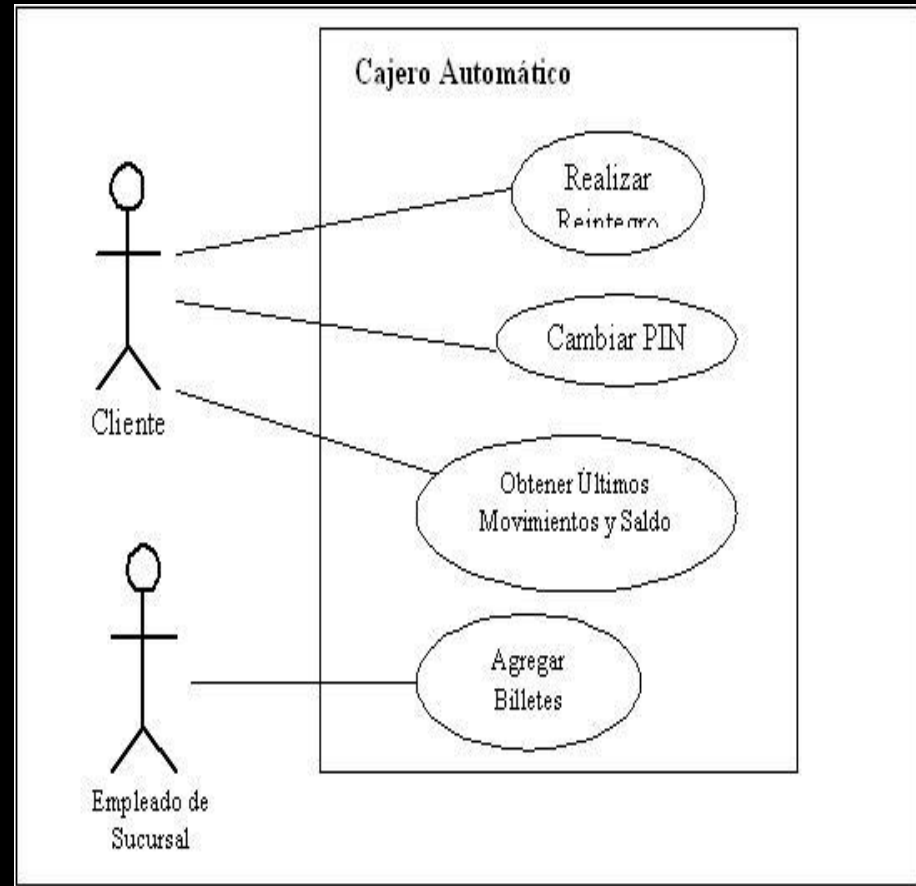
Métodos

- **El método de Booch:** este método abarca un micro proceso de desarrollo y un macro proceso de desarrollo tanto para el análisis como para el diseño. El nivel micro define un conjunto de tareas de análisis que se reaplican en cada etapa en el macro proceso.
- **El método de Rumbaugh:** Este método mejor conocido como OMT, se utiliza para el análisis, diseño del sistema y diseño a nivel de objetos.
- **El método de Jacobson:** también llamado OOSE (que en español significa ingeniería del Software Orientada a Objetos este método, en el análisis, se diferencia de los otros por la importancia que da al caso de uso.



Paso 1: Realizar un Diagrama de Casos de Uso.

- Hay que tener en cuenta que cada Caso de Uso representa una funcionalidad del software que vamos a construir.



Paso 2: *Priorizar los Casos de Uso a trabajar.*

- Pongamos los casos de uso en una lista colocando los mas importantes al inicio y los menos importantes al final.

NOMBRE DEL CASO DE USO
1. Registra usuario, claves y asignar roles
2. Solicitar pedido
3. Revisa existencias
4. Emite fallo
5. Envía pedido
6. Evalúa pedido
7. Registra Orden de Compra
8. Envía Orden de Compra
9. Elige opción reportes
10. Procesa información e imprime



Paso 3: *Generar los Documentos de Caso de Uso.*

- Ese documento debe ser generado por el Analista del proyecto y debe tener mas o menos la siguiente estructura:
 - Descripción Breve.
 - Precondiciones.
 - Flujo Básico.
 - Flujo Alternativo.
 - PostCondiciones.
 - Interfaz Gráfica.

Diagrama de Formato Expandido

CASO DE USO
PEDIDOS

ADMON. DE

ACTORES:

Administradores, Usuarios, Sistema, Proveedores y Área de compras.

TIPO DE ACTORES:
Secundarios (Proveedor)

Primarios,

DESCRIPCIÓN:

El usuario hace un pedido por medio del sistema el cual valida si tiene derechos o no para realizar la operación, si los tiene se revisa el stock existente del producto y si ya no hay producto en el almacén se envía la petición al evaluador, el cual examina el pedido para que este no exceda en 10 productos emitiendo un fallo favorable o desfavorable.

En caso de ser favorable se envía la OC a compras para aprobar o no el pedido, de aceptarlo, se envía la OC al proveedor para que los surta.

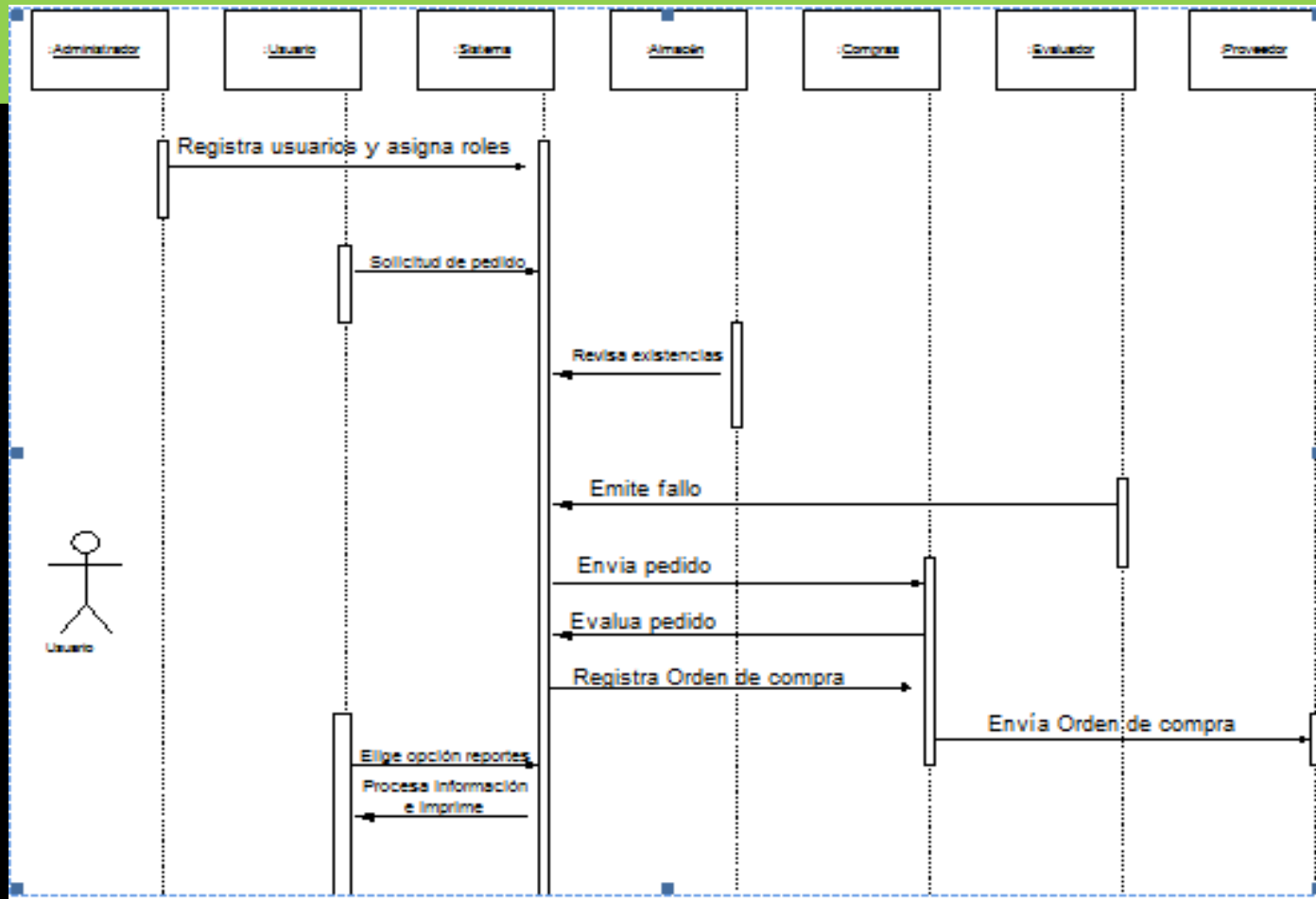
El usuario puede pedir distintos reportes al sistema.

REFERENCIAS CRUZADAS

El sistema registra claves de usuario, recibe fallo favorable o desfavorable, recibe evaluación del pedido, registra orden de compra, procesa reportes e imprime.

Paso 4: *Generar los Diagramas de Secuencia.*

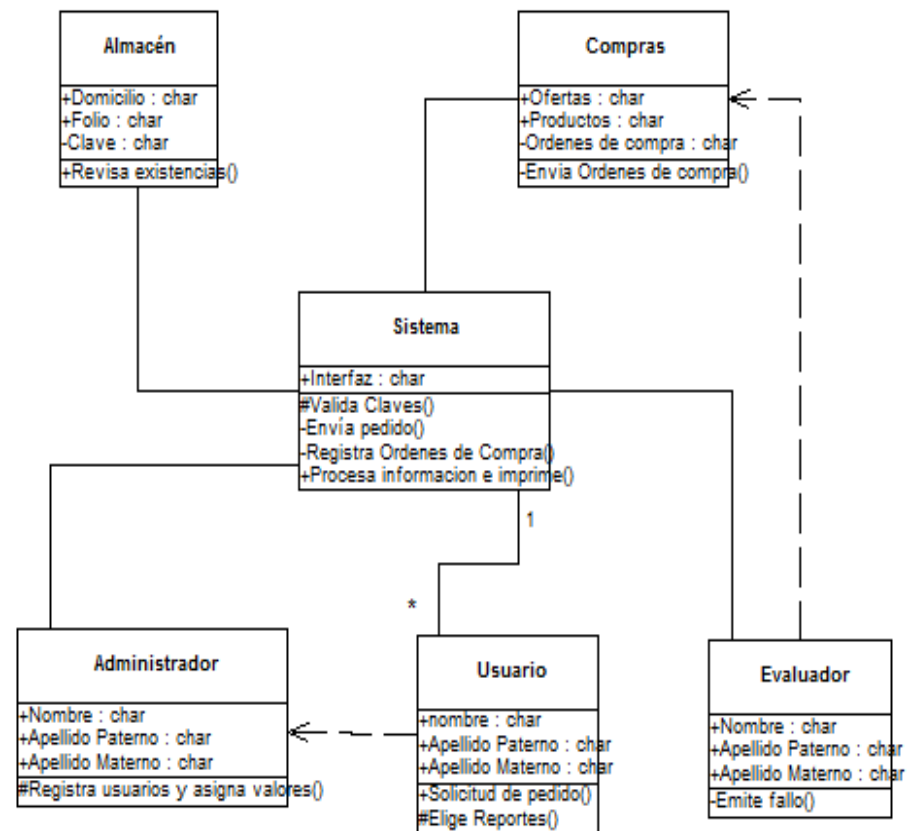
- Permiten conocer la forma en la que los objetos se comunicarán en una pantalla para cumplir su objetivo.



Paso 5: *Diseñar el Framework del proyecto.*

- Esto significa que el Arquitecto de Software del proyecto hará su trabajo, el cual consiste en diseñar las clases que se usarán en todo el Software.

DIAGRAMA DE CLASES



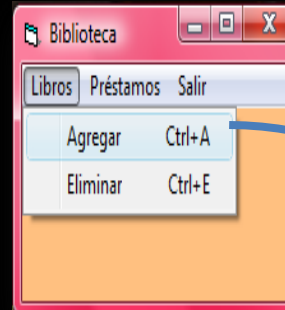
Paso 6: *Creación de la Base de Datos.*

- El diagrama de Clases de UML puede servir como Base para el diseño de la Base de Datos del proyecto, claro, solo utilizando la Capa de Datos, es decir, las clases de estereotipo Entidad.



Paso 7: *Construir la máscara*

- Mientras los pasos 4, 5 y 6 se están realizando, el personal a cargo del Diseño del Sistema, pueden ir desarrollando las plantillas para la creación del sistema.



Agregar Nuevos Libros

Nombre Del Libro	<input type="text"/>	Clave	<input type="text"/>
Autor	<input type="text"/>	Editorial	<input type="text"/>
Clasificación	<input type="text"/>	Edición	<input type="text"/>
N° De Páginas	<input type="text"/>	Cantidad	<input type="text"/>


Dar De Alta

Salir

Paso 8: *Programar las funcionalidades de los Casos de Uso.*

- Trabajo de los programadores.
- Escribirán el código necesario para que el caso de uso funcione.

```
1 public class VariableLocal{
2     public static void main(String args[]){
3         VariableLocal vl=new VariableLocal();
4
5         /**
6          * La variable entera i es local pues está declarada e inicializada
7          * dentro de un bucle for. Si se usa fuera del código del for se
8          * producirá un error de compilación
9          */
10        for(int i=0;i<=5;i++){
11            System.out.println(i);
12            System.out.println("Repeticion "+i);
13        }
14        vl.mostrarVariable();
15    }
16    void mostrarVariable(){
17
18        //Línea que provoca error de compilación.
19        //Acceso incorrecto a una variable local
20        System.out.println(i+1);
21    }
22 }
```



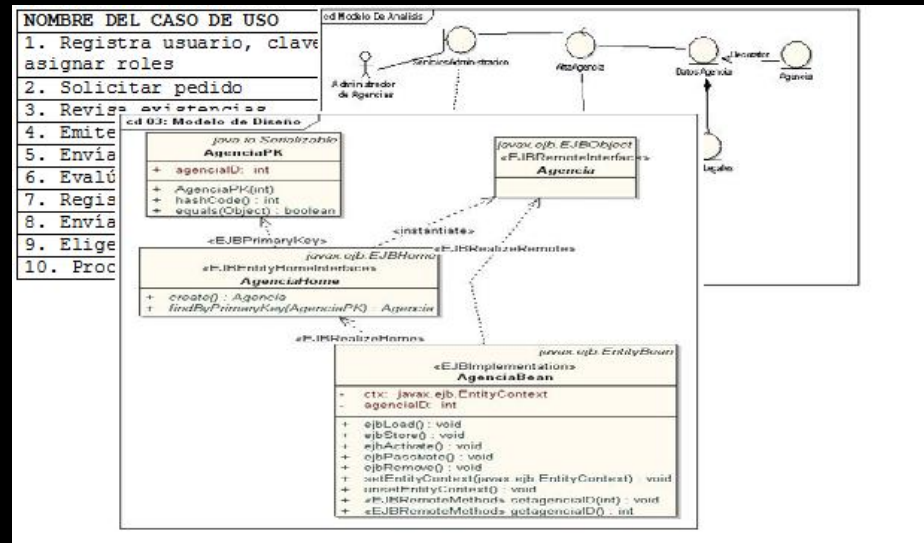
Paso 9: *Probar los Requerimientos del Software.*

- No deben ser hechos por las personas que programaron los Casos de Uso, es mucho mejor que lo haga otra persona.



Paso 10: Integrar los requerimientos concluidos.

- Ahora si, ya es momento de unir lo que se hizo y ponerlo a disposición de los usuarios.



Usuarios

Nombre Usuario

Contraseña

Permiso

Salir



Conclusión

- Existen varias metodologías orientadas a objetos, a pesar que tienen variantes entre ellas, todas trabajan con el mismo paradigma por tanto se basan en los mismo fundamentos de modelación de objetos.
- han surgido tantas metodologías que tratan este modelo de programación, llegando a destacarse un enfoque de Modelación de Lenguaje Unificado (UML) como uno de los más prácticos y eficientes.



Páginas consultadas

- <http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php>
- <http://www.portalfox.com/index.php?name=Sections&req=viewarticle&artid=31>
- <http://www.di.uniovi.es/~cernuda/pfc/aoo.pdf>
- <http://hancocchi.net/proceso-desarrollo-software-orientado-objetos/>
- <http://www.portalfox.com/index.php?name=Sections&req=viewarticle&artid=31>
- <http://www.di.uniovi.es/~cernuda/pfc/aoo.pdf>