

Contenidos

- 3.1. Desarrollo de las clases.
 - 3.1.1. Visual Paradigm.
 - 3.1.2. Restricciones.
- 3.2. Diseño del sistema.
 - 3.2.1. Diagramas de clase.
 - 3.2.2. Diagramas de secuencia.
- 3.3. Entregables.
- 3.4. Referencias.

0. Desarrollo de las clases

0.1. Visual Paradigm

Visual Paradigm



- Es una herramienta de soporte al modelado UML.
- Incluye una gran cantidad de diagramas: clase, secuencia, estados, actividad, componentes, etc.
- Cuenta con una versión gratuita, conocida como *Community Edition*.
- Está incluido en los ordenadores de la UCO, en Linux.

0.2. Restricciones

Restricciones

- El proyecto se realizará haciendo uso del sistema de control de versiones Git y la plataforma GitHub para el almacenamiento del repositorio de forma remota.
- Todos los diagramas serán subidos a Git, con lenguaje Markdown.
- El tipo de uso que se hará de GitHub será el mismo que el realizado en la práctica anterior.

1. Diseño del sistema

1.1. Diagrama de clases

Diagrama de clases

- Diagrama de estructura estática.
- Una clase es una estructura de datos que tiene unos atributos y realiza unas acciones.
- Muestra las clases, el sistema, atributos, métodos y relaciones entre ellos.
- Ejemplo:
 - Clase: Ordenadores
 - Atributos: placa base, procesador, etc.
 - Acciones: encender, jugar, programar, etc.

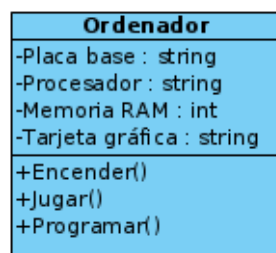


Diagrama de clases

- Asociaciones: relaciones estáticas entre clases.

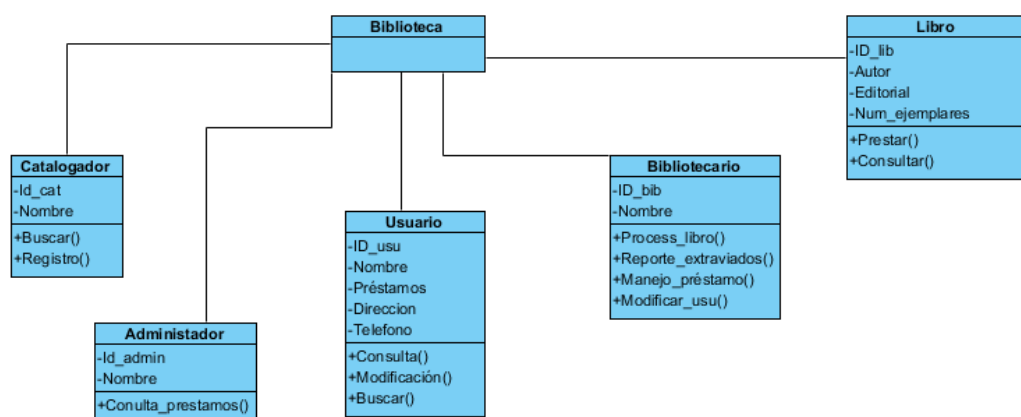


Diagrama de clases

- **Multiplicidad:** indican el número de instancias de una clase que muestran relación a otro número de instancias de otra clase.

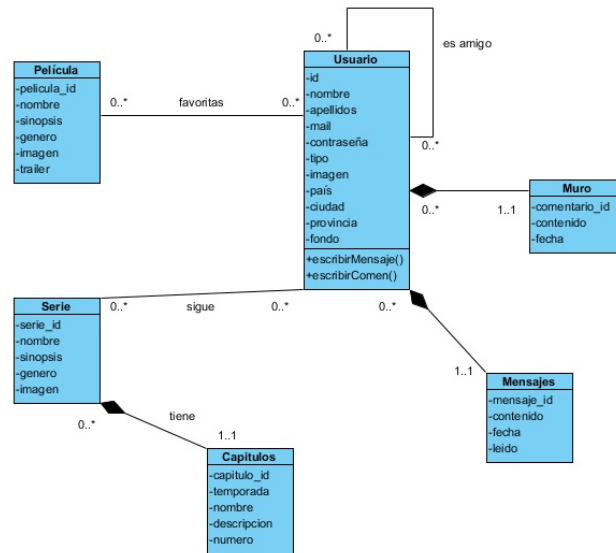


Diagrama de clases

- **Agregación:** indica que una clase forma parte de otra.
- **Ejemplo:** de la clase *Student* a la clase *School*.
- **Composición:** indica una posesión intensa. Si el compuesto se destruye, el componente también lo hace.
- **Ejemplo:** de la clase *Serie* a la clase *Capítulos*.

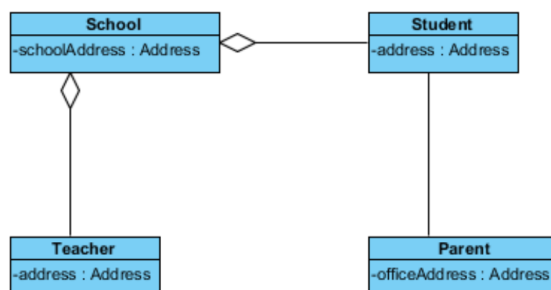


Diagrama de clases

- **Generalización:** relación entre dos clases donde una clase es una versión especializada de la otra.

- **Ejemplo:** de la clase Araña, Pez y Gato a la clase Animal.

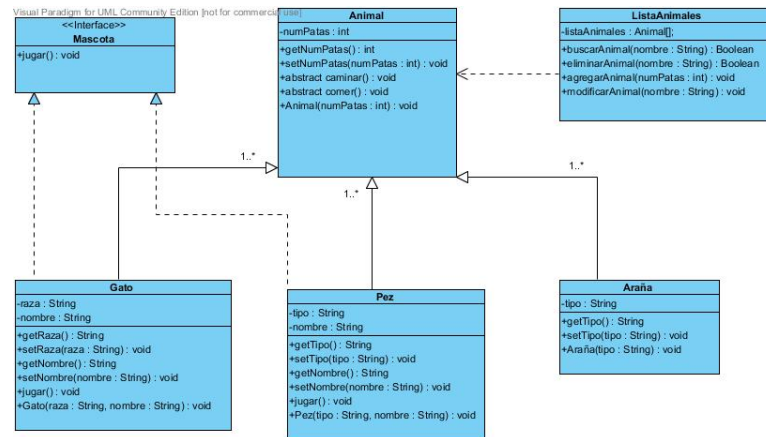
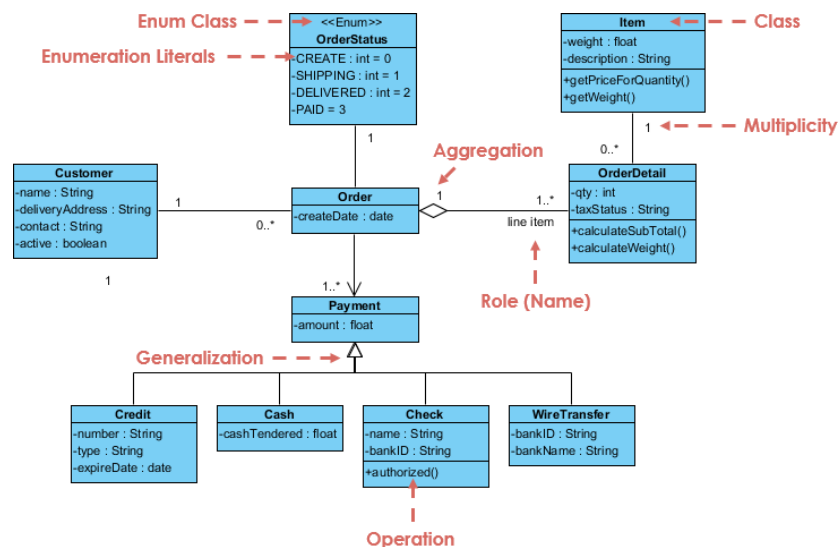


Diagrama de clases



1.2. Diagramas de secuencia

Diagrama de secuencia

- Describe el comportamiento dinámico del sistema.
- Muestra la mecánica de la interacción durante una acción concreta.
- **Ejemplo:** El usuario quiere ver un mensaje.

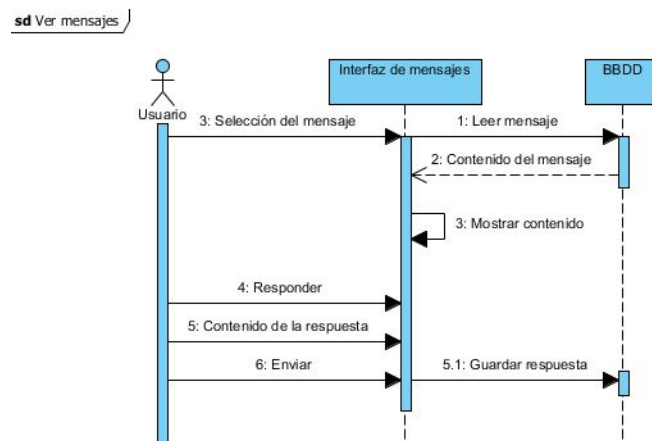


Diagrama de secuencia

- **Rol de la clase:** describe el comportamiento de un objeto.
- **Activación:** representan el tiempo que un objeto necesita para hacer una tarea.
- **Lineas de vida:** indican la presencia del objeto.
- **Mensajes:** muestran las comunicaciones entre objetos.

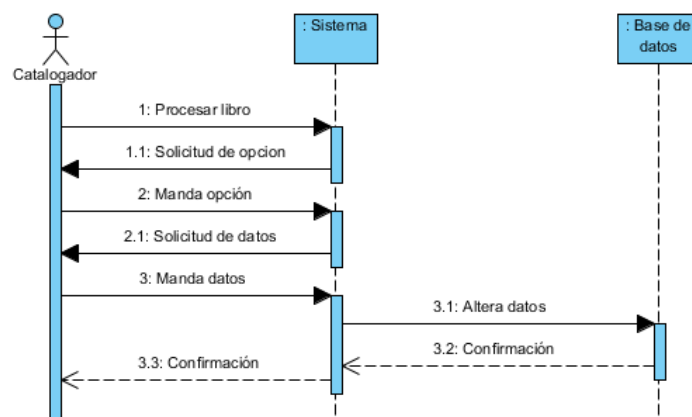


Diagrama de secuencia – Flujo alt

- **Ejemplo:** El cliente quiere realizar una búsqueda por autor.

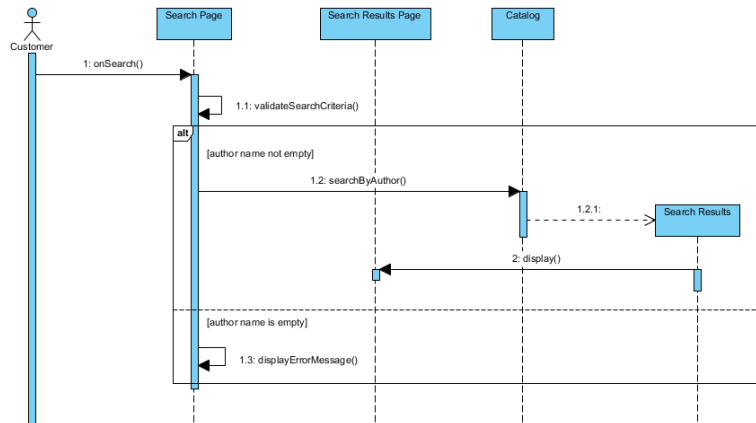


Diagrama de secuencia – Flujo opt

- **Ejemplo:** Un miembro realiza un pedido.

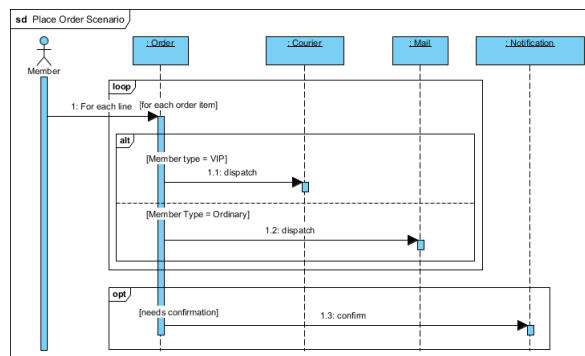


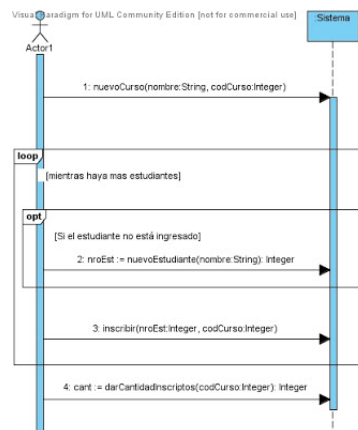
Diagrama de secuencia – opt / alt

Las diferencias entre opt y alt son las siguientes:

- **alt:** son flujos alternativos de los cuáles siempre se ejecuta uno y solo uno.
- **opt:** es un flujo opcional que a veces se ejecuta y a veces no, dependiendo de una condición.

Diagrama de secuencia – Flujo bucle

- **Ejemplo:** El actor1 crea un nuevo curso e introduce a los alumnos.



2. Entregables

Entregables

Todos los entregables serán subidos a Git, con lenguaje Markdown. Se entregarán los tipos de diagramas, explicados anteriormente:

1. Diagrama de clase: un archivo Markdown con la descripción de las clases y un diagrama de clase que agrupe todos los elementos del problema.
2. Diagrama de secuencia: un archivo Markdown que contenga todos los diagramas de secuencia que existan en el problema.

Fecha límite de entrega: 08 de noviembre a las 23.59h.

3. Referencias

Referencias

Referencias

[Arlow] Arlow, J. y Neustadt, I. (2016). Programación UML 2. Programación (Anaya Multimedia) Anaya Multimedia.

[Catedra] Resumen diagramas UML: http://www.teatroabadia.com/es/uploads/documentos/iagramas_del_uml.pdf

[Debrauwer] Debrauwer, L., y Van der Heyde, F. (2016). UML 2.5: iniciación, ejemplos y ejercicios corregidos. Ediciones ENI.

[Miles] Miles, R., y Hamilton, K. (2006). Learning UML 2.0. O'Reilly Media, Inc. 2