

UML para el manejo de eventos



Introducción

Cuando queremos desarrollar un **proyecto de software**, en ocasiones, lo primero en lo que pensamos es en la **codificación**, sin considerar lo que haremos y solo confiamos en nuestro conocimiento del problema. Esto puede llevarnos a modificaciones complicadas y costosas de último momento, incluso al fracaso del proyecto.

Para una correcta elaboración del *software*, podemos hacer uso del lenguaje de modelo unificado o UML.

Lenguaje de modelado unificado UML

El modelado es parte esencial en el desarrollo de cualquier proyecto de *software*; ya que a través de éste, podemos visualizar la estructura del proyecto, como se haría, por ejemplo, con los planos de un edificio.



El uso del modelado le garantiza a los responsables del proyecto el éxito y la seguridad de que su *software* tendrá la funcionalidad deseada.

Los modelos nos permiten:

- Trabajar en un nivel más abstracto.
- Mostrar los detalles que necesitamos ver en el momento, o bien, resaltar algún panorama en específico.
- Apreciar detalles importantes del prototipo.



El Unified Modeling Language (UML) es de gran ayuda para visualizar documentar y especificar modelos de sistemas de software; su diseño y estructura y así, poder unificar todos sus requisitos.

Enlace:

<http://uml.org/>





El caso de uso describe el objetivo al que esperamos llegue el actor y describe los pasos a seguir para cumplir dicho objetivo.



Caso 1

Caso 2

Caso 3

Caso 4

Una regla para escribir el nombre de los casos de uso, es empezar con un verbo que denota acción, estos verbos deben estar en infinitivo (con terminación en: ar, er o ir).



Observa el siguiente ejemplo:

Revisar estado
Agregar cliente
Eliminar empresa



Cliente

Consultar
precio

Vender

Comprar



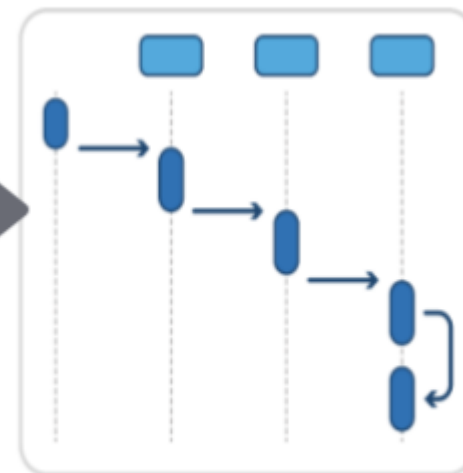
Vendedor

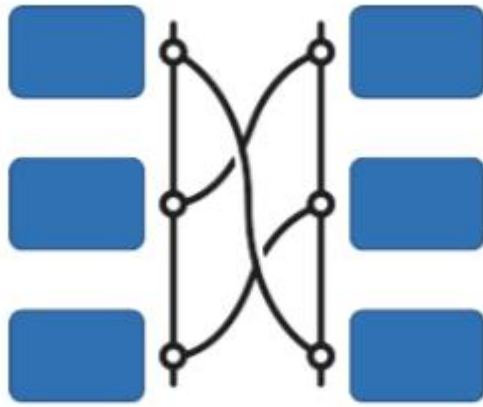
Diagrama de secuencias

Determina la interacción que tienen entre sí los objetos del sistema, ya que muestra gráficamente cómo se relacionan éstos.



Dicho diagrama muestra los **usuarios** y **objetos** en la parte superior, mientras que con flechas, se aprecia su relación, acompañados de un texto que representa un mensaje o un método perteneciente a otro objeto.

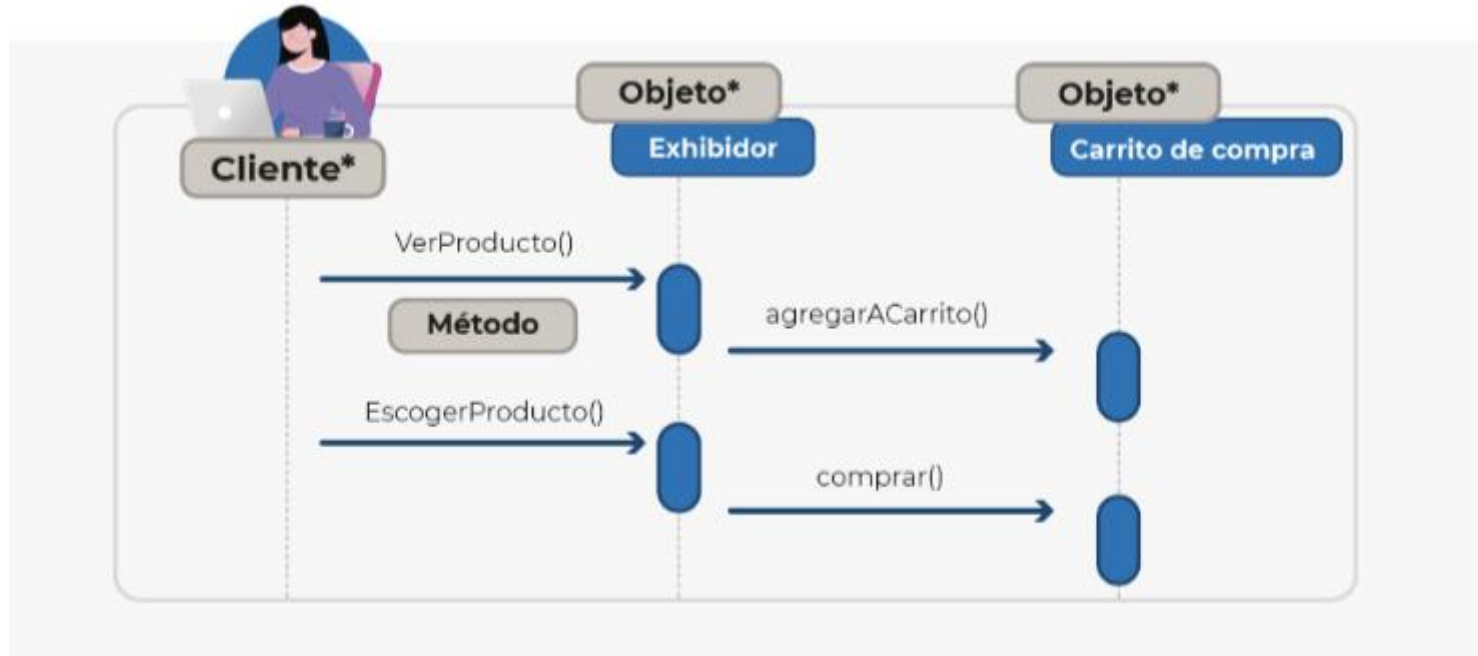




Los diagramas de secuencia nos ayudan a:

- Conocer las relaciones entre los componentes.
- Visualizar todos los pasos que tendrán los escenarios del sistema.

Es una de las primeras cosas que debemos considerar en el desarrollo de *software*.

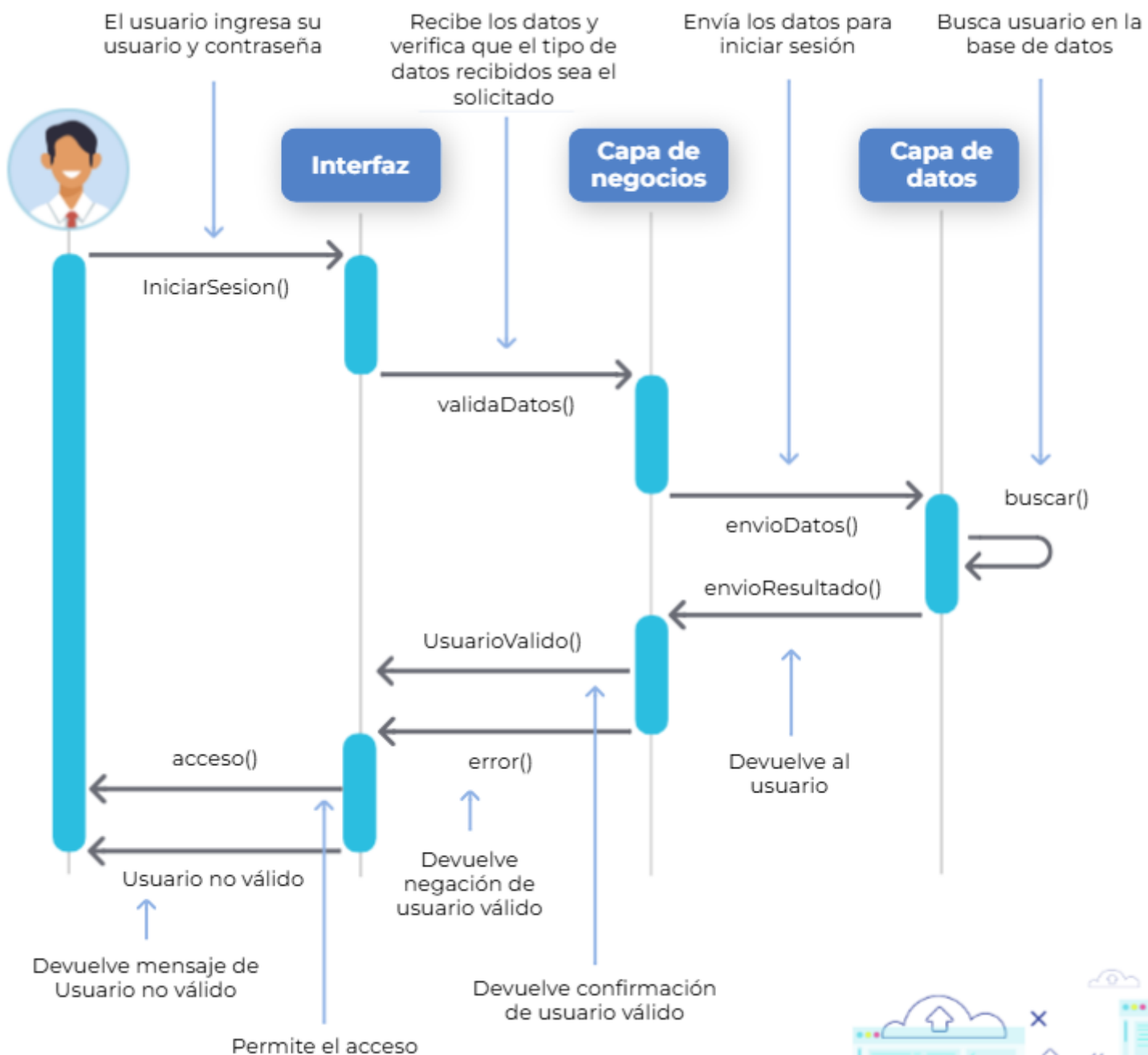




A continuación, se muestra un **diagrama de secuencia** más elaborado donde se representa el inicio de sesión de una *página web*.



Te invito a observar ese ejemplo y centrar tu atención en el lenguaje utilizado en su desarrollo y los contenidos ya explicados, para este tipo de diagramas.



Como hemos visto hasta ahora, el modelado de un programa de *software* es un proceso necesario en el alcance de la calidad de éste, ya que permite:

- Visualizar cada módulo necesario.
- Prever futuros conflictos que, si se consideran a tiempo, podrían ahorrar tiempo y dinero.



Introducción

Cuando queremos desarrollar un proyecto de software, en ocasiones, lo primero en lo que pensamos es en la codificación, sin considerar lo que haremos y solo confiamos en nuestro conocimiento del problema. Esto puede llevarnos a modificaciones complicadas y costosas de último momento, incluso al fracaso del proyecto.

Para una correcta elaboración del software, podemos hacer uso del lenguaje de modelo unificado o UML.

Lenguaje de modelado unificado UML

El modelado es parte esencial en el desarrollo de cualquier proyecto de software; ya que, a través de éste, podemos visualizar la estructura del proyecto, como se haría, por ejemplo, con los planos de un edificio.

El uso del modelado le garantiza a los responsables del proyecto el éxito y la seguridad de que su software tendrá la funcionalidad deseada.

Los modelos nos permiten:

- Trabajar en un nivel más abstracto.
- Mostrar los detalles que necesitamos ver en el momento, o bien, resaltar algún panorama en específico.
- Apreciar detalles importantes del prototipo.

Para aprender más del tema, pulsa en el siguiente ícono de enlace.

Siguiente página

Diagrama de casos de uso

Nos permite modelar y visualizar un panorama más amplio del software.

Los casos de uso definen la participación que tendrán los usuarios con el sistema mediante uso de actores y los pasos a realizar para completar una acción.

Los actores pueden ser:

- Una persona
- Un dispositivo o alguna pieza de software independiente del programa que elaboramos

El caso de uso describe el objetivo al que esperamos llegue el actor y describe los pasos a seguir para cumplir dicho objetivo.

Una regla para escribir el nombre de los casos de uso es empezar con un verbo que denota acción, estos verbos deben estar en infinitivo (con terminación en: ar, er o ir).

Observa el siguiente ejemplo:

Cliente

Agregar cliente

Eliminar empresa

Consultar precio

Vender

Comprar

Vendedor

Revisar estado

Página anterior

Siguiente página

Diagrama de secuencias

Determina la interacción que tienen entre sí los objetos del sistema, ya que muestra gráficamente cómo se relacionan éstos.

Dicho diagrama muestra los usuarios y objetos en la parte superior, mientras que con flechas, se aprecia su relación, acompañados de un texto que representa un mensaje o un método perteneciente a otro objeto.

Los diagramas de secuencia nos ayudan a:

- Conocer las relaciones entre los componentes.
- Visualizar todos los pasos que tendrán los escenarios del sistema.

Es una de las primeras cosas que debemos considerar en el desarrollo de software.

Página anterior

Siguiente página

A continuación, se muestra un diagrama de secuencia más elaborado donde se representa el inicio de sesión de una página web.

Te invito a observar ese ejemplo y centrar tu atención en el lenguaje utilizado en su desarrollo y los contenidos ya explicados, para este tipo de diagramas.

Interfaz

Capa de negocios

Capa de datos

El usuario ingresa su usuario y contraseña

Recibe los datos y verifica que el tipo de datos recibidos sea el solicitado

Envía los datos para iniciar sesión

Busca usuario en la base de datos

IniciarSesion()

validaDatos()

buscar()

envioDatos()

acceso()

UsuarioValido()

envioResultado()

Usuario no válido

error()

Devuelve mensaje de Usuario no válido

Permite el acceso

Devuelve negación de usuario válido

Devuelve confirmación de usuario válido

Devuelve al usuario

- Como hemos visto hasta ahora, el modelado de un programa de software es un proceso necesario en el alcance de la calidad de éste, ya que permite:
- Visualizar cada módulo necesario.
 - Prever futuros conflictos que, si se consideran a tiempo, podrían ahorrar tiempo y dinero.