

Modelar y diseñar: ¿es lo mismo?

Sitio: Agencia de Habilidades para el Futuro
Curso: Modelado y Diseño de Software 1º D
Libro: Modelar y diseñar: ¿es lo mismo?

Imprimido por: Eduardo Moreno
Día: jueves, 20 de marzo de 2025, 00:36

Tabla de contenidos

1. Introducción

2. Planificar para construir

3. El plano de un software

4. Modelo del diseño

4.1. ¿Qué artefactos hay que diseñar?

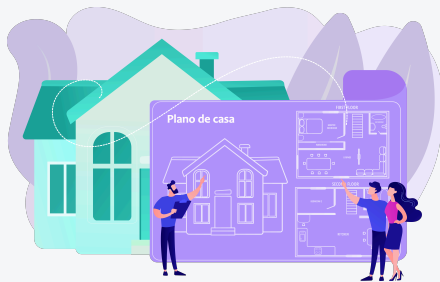
4.2. Nuestro primer modelo



¿Cuál es la relación entre diseño de software y modelado? ¿Y con el desarrollo?

El cliente se acerca a nosotros/as para pedirnos un software que responda a sus necesidades. Esto implica que tenemos que darle un desarrollo, es decir, un producto software. Pero para llegar a ese desarrollo antes hay que hacer un diseño. Y para diseñar, debemos modelar. Entonces, ¿qué es modelar? Esta pregunta supone empezar a conocer este concepto central: modelo. Esta palabra no es rara ni desconocida; sin embargo, es importante precisar su significado en este contexto. Por eso, es útil diferenciar entre "modelo" y un "modelo de diseño".

Cuando ves un desfile de moda, ves "modelos". ¿Alguna vez te preguntaste por qué se llaman así? La respuesta es porque es el ideal que se muestra al cliente o comprador de ropa para que apruebe la vestimenta ofrecida.



Lo mismo sucede con el arquitecto, que muestra el modelo de la casa para que el cliente apruebe o compre su proyecto.

Un **modelo** es una construcción ideal, en la que se destacan sus elementos fundamentales.

Los **desarrolladores de software** también trabajan con modelos. En este caso los modelos son **planos del software** y sirven para que el cliente nos compre el producto: "el producto software".



Ahora vamos a ingresar al mundo del modelado en el desarrollo de software.

El **modelado** es el análisis y diseño de aplicaciones software antes de escribir el código. Al igual que el arquitecto/a con su plano, el desarrollador/a crea un **conjunto de modelos** que permiten especificar aspectos del sistema, como los requisitos, la estructura y el comportamiento.

- Requisitos es lo mismo que requerimientos: lo que nuestro cliente quiere obtener del software.
- La estructura se refiere a los componentes, los elementos que forman el software (el código, las pantallas o interfaces por ejemplo).

- El comportamiento es cómo reacciona el software cuando se interactúa con él. Cuando se realiza alguna tarea con el programa software.

En el **índice** de la derecha podrás recorrer los contenidos revisándolos todas las veces que lo necesites.



Diseñar supone un proceso para planificar

El **diseño de software** es el proceso de construcción para la planificación de una solución informática.

Siempre que se realiza un proyecto, se realiza una planificación. Incluso en nuestras tareas cotidianas planificamos. Por ejemplo, si vamos a realizar un viaje pensamos a qué lugar nos vamos, dónde vamos a alojarnos, qué lugares visitaremos.

En un proyecto software también se planifica. Cuando se diseña, este proceso ayuda a los programadores/as para que puedan manejar la complejidad que significan la mayoría de los programas informáticos; esto además ayuda a disminuir el riesgo de errores a la hora de ser puestos en marcha.

Si los errores se producen una vez desarrollado el software, el costo y el tiempo de trabajo son muy altos, y podemos perder al cliente si no está satisfecho.

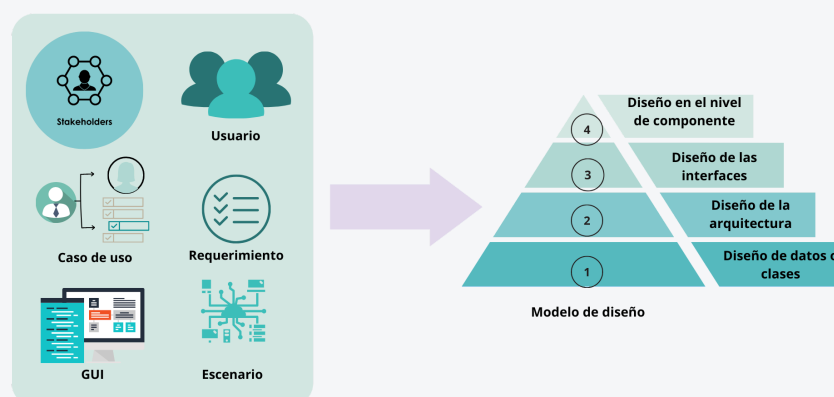


El diseñar ayuda a modificar elementos que se utilizan para armar el formato que tendrá el programa. La importancia del diseño del software se puede definir en una sola palabra: **calidad**; ya que dentro del diseño es donde se establece la calidad del proyecto.

El diseño es la única manera de materializar con precisión los requerimientos del cliente.

Una vez que se analizan los requerimientos del cliente, el diseño del software es la primera de las actividades técnicas que se realizan con el fin de construir y verificar el software .

El flujo de información durante el diseño del software se muestra en la imagen siguiente. Los requisitos del software, que se representan por los modelos de datos o modelo E-R (accedé al libro haciendo clic [aquí](#)) y de comportamiento, alimentan la tarea del diseño. Mediante uno de los muchos métodos de diseño que veremos en las siguientes semanas, la tarea de diseño produce: un diseño de datos, un diseño arquitectónico, un diseño de interfaz y un diseño de componentes. Todos ellos conforman el **modelo de diseño**.



¿Por qué es importante el diseño de software?

¿Diseñar es lo mismo que modelar o modelar es una de las fases del diseño?

- Diseñar permite modelar el sistema o producto que se va a construir.
- Evaluar el antes el modelo en calidad y necesidad de mejora previo a generar el código.
- Efectuar pruebas que involucran a los usuarios finales.

Todo error conceptual será detectado en este momento y no cuando tengamos el producto terminado. Sin un diseño, corremos el riesgo de construir un sistema inestable -un sistema que fallará cuando se lleven a cabo cambios; un sistema que puede resultar difícil de comprobar; y un sistema cuya calidad no puede evaluarse hasta muy avanzado el proceso, sin tiempo suficiente y con mucho dinero gastado en él.



Desarrollar un software supone crear modelos o planos de software

Como ya mencionamos, cuando queremos desarrollar un software (grande como un sistema o pequeño como un algoritmo) se crea un conjunto de **modelos** o “**planos del software**” .

Los modelos te ayudan a:

- **Razonar** sobre el sistema o producto software.
- **Favorecer** la comunicación con el cliente.
- **Documentar** las decisiones sobre el producto software.

Como muestra la primera imagen, cuando se realiza el modelo o el plano, **hay mucho dibujo y poco texto** . Se divide al plano en pequeñas partes para entender mejor lo que el cliente solicita. Esas partes están relacionadas entre sí y de esa manera, evitamos la **redundancia** (o repetición de procesos). Al cliente no le interesa **cómo se codifica el software para hacer la tarea sino qué tareas** va a realizar el software (por ejemplo: calcular la ganancia, mostrar los clientes que no pagan, etc.)

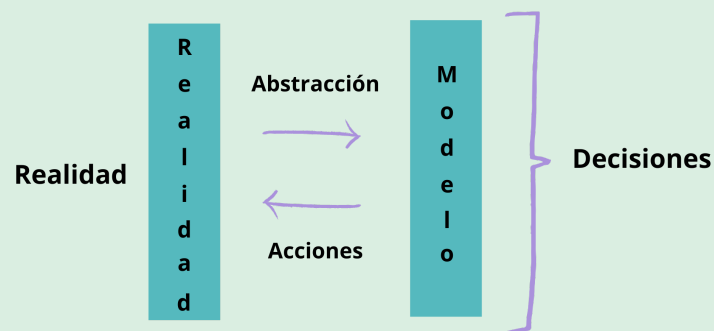
Características de los modelos

Gráfico más texto

- Segmentar el sistema y permitir distintos niveles de complejidad.
- Reducir la redundancia al mínimo.
- Ayudar a la comunicación.
- Ser transparente para el lector.



Un **modelo** es una simplificación de la realidad. Es resultado de un **proceso de abstracción** y ayuda a comprender y razonar sobre una realidad.



Una vez que se realiza el modelo, se toman las decisiones necesarias para comenzar a codificar el software.



¿Qué es el modelo de diseño?

El producto final del diseño, tal como venimos viendo, es el **modelo del diseño**.

El diseño de software, entonces, es tanto un proceso como un modelo.

El proceso de diseño es una secuencia de pasos que hacen posible que el diseñador/a describa todos los aspectos del software que se va a construir: esto culmina en un modelo de diseño, que luego se llevará a la realidad para el desarrollo que queremos lograr. Sin embargo, es importante destacar que el proceso de diseño no es un recetario, no hay una receta para hacer un diseño, depende de la persona que diseñe.

El modelo de diseño es el equivalente a los planos de un arquitecto para una casa. Comienza representando la totalidad de todo lo que se va a construir (por ejemplo, una representación en tres dimensiones de la casa) y refina lentamente lo que va a proporcionar la guía para construir cada detalle (por ejemplo, el diseño de cañería de agua y gas).

De manera similar, el modelo de diseño que se crea, separa el software y proporciona diversas visiones diferentes del mismo producto software. El modelo del diseño

- Agrupa las representaciones arquitectónicas.
- Agrupa las interfaces en el nivel de componentes.
- Presenta despliegue del producto final diseñado.

El **modelo de análisis** que describe la estructura del software que se está modelando, sirve de entrada al modelo del diseño, el cual contiene una serie de elementos divididos en cuatro grupos:

1. Elementos de arquitectura
2. Elementos de la interfaz
3. Elemento en el nivel de componentes
4. Elementos en el nivel de despliegue

¡Todos estos elementos los iremos viendo a lo largo de la cursada!



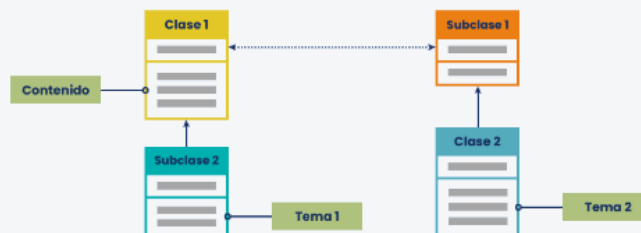


Un **artefacto** es un producto resultante del **proceso de desarrollo de software**. Algunos artefactos que vamos a ver son: los casos de uso, diagrama de clases u otros modelos UML que ayudan a la descripción de la función o tarea que realizará el software, la arquitectura o el diseño del software.

Como desarrollador/a utilizarás estos artefactos cuando realices tu diseño. Lo veremos paso a paso a lo largo de la cursada de la materia, pero te damos un adelanto.

Para el diseño de datos y/o clases se usa:

- Diagramas de clases
- Modelo lógico y físico de la base de datos

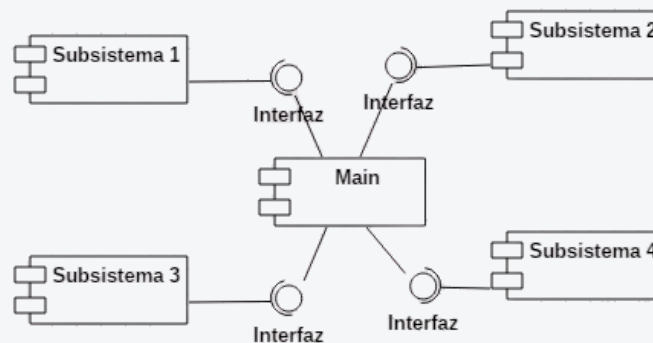


Para el diseño de la arquitectura del sistema o plano del software se usan:

- Los elementos del diseño de la arquitectura que dan la visión general del software.
- Estilos y patrones arquitectónicos.

El diseño en el nivel de los componentes describe por completo los detalles internos de cada componente.

- Utiliza diagrama de componentes.



El diseño en las interfaces del sistema que permite que la información fluya hacia dentro y fuera del sistema.

Muestra la

- Interfaz de usuario
- Interfaces internas
- Interfaces externas

El diseño del despliegue de la aplicación que indican la forma en la que se acomodarán la funcionalidad del software y los subsistemas dentro del ambiente físico.

- Se utiliza el diagrama de despliegue

iNo te preocupes, lo iremos viendo semana a semana!



Hacemos nuestro primer modelo

¿Te acordás de la historia de Manuel, que es dueño de la empresa Viajes SA?

En esa historia detectamos que el inicio de todo el proceso es conocer la **información sobre el cliente**.

Pero, ¿cómo reconocemos qué **información** es útil y qué **información** se puede descartar?

Para eso te damos un ejemplo sobre una empresa de alquiler de taxi

Vamos a usar herramientas que viste en Base de Datos (modelo E-R o modelo entidad relación).

Cambiamos de cliente para el ejemplo, así después, en la práctica, trabajás con la empresa de viajes.

En este caso es una empresa que se encarga de alquilar vehículos taxímetro a choferes.

El **cliente** se acerca para solicitarnos el desarrollo de un programa que le permita administrar y controlar los taxímetros y los choferes al comienzo del turno de trabajo.

El **sistema** gestiona la entrega de taxis a los choferes de acuerdo al horario de trabajo de los mismos y la devolución de los coches por el chofer.



¿Cuál es el **requerimiento de cliente**? ¿Qué **información** necesita que el software le dé?

El desafío es ver si los modelo anteriores están bien y cómo modelas este requerimiento.