

M4_AE3_ABPRO-Ejercicio grupal

Cristian Aranda

Darío Inostroza

Emily Quintana

Paso 1: Análisis y Conceptualización

1.1- Discutan en equipo qué es un diagrama de clases y para qué sirve.

Un diagrama de clases es una representación visual de las clases, sus atributos (propiedades) y métodos (funciones), y las relaciones entre ellas en un sistema orientado a objetos. En UML, los diagramas de clases son uno de los seis tipos de diagramas estructurales.

Dentro de cada clase (caja), se detalla la siguiente información:

Nombre de la clase: El nombre de la entidad que estás modelando (ej. Auto, Estudiante, CuentaBancaria).

Atributos: Las características o propiedades de la clase (ej. Auto podría tener color, velocidad, marca).

Métodos: Las acciones o comportamientos que la clase puede realizar (ej. Auto podría tener acelerar(), frenar(), encender()).

Este tipo de diagrama se utiliza en la modelización de sistemas para planificar y estructurar el diseño del software de manera más clara.

Durante las fases de análisis y diseño del ciclo de desarrollo, se pueden crear diagramas de clases para:

- Capturar y definir la estructura de clases
- Definir relaciones entre clases
- Ilustrar la estructura de un modelo mediante el uso de atributos, operaciones y señales.
- Mostrar las clases de implementación en un paquete
- Mostrar la estructura y el comportamiento de una o más clases
- Mostrar una jerarquía de herencia entre clases y clasificadores

Durante la fase de implementación de un ciclo de desarrollo de software, puede utilizar diagramas de clases para:

- Convertir sus modelos en código
- Convertir su código en modelos.

1.2- Investiguen la notación de un diagrama de clases y su uso en la representación de atributos, métodos, colaboración y composición.

En los diagramas de clases, la notación estándar sigue las convenciones del lenguaje UML (Unified Modeling Language). A continuación se detallan los elementos principales de la notación que se utilizan para representar una clase:

La notación básica de una clase es una caja rectangular dividida en tres secciones:

- Sección superior: Contiene el nombre de la clase. Se escribe en negrita y centrado.
- Sección intermedia: Contiene los atributos (variables o propiedades) de la clase.
- Sección inferior: Contiene los métodos (funciones o comportamientos) de la clase.

Dentro de cada sección, se utiliza una notación específica para describir la visibilidad, el nombre y el tipo de dato.

Visibilidad (nivel de acceso):

- + Público: El atributo o método es accesible desde cualquier otra clase.
- - Privado: El atributo o método sólo es accesible dentro de la propia clase.
- # Protegido: El atributo o método es accesible dentro de la clase y sus subclases (herencia).

Notación de relaciones:

- Colaboración (Asociación): indica que una clase "conoce" o "usa" a la otra. Se representa con una línea sólida.
- Composición: indica una relación de "parte-todo" donde la "parte" no puede existir sin el "todo". Es una relación fuerte: si el objeto "todo" es destruido, sus "partes" también lo son. Se representa con una línea que empieza en la clase contenida y termina en rombo relleno en la clase contenedora.
- Herencia: indica un tipo de relación en la que una clase asociada es hija de otra en virtud de asumir las mismas funcionalidades de la clase padre. Se representa con una línea terminada en flecha en la clase padre (dónde están definidos los atributos)

3.- Respondan en un documento:

¿Cómo puede ayudar un diagrama de clases a estructurar el sistema?

-Los diagramas de clases son útiles en tanto ayudan a organizar la estructura y modelado del software de forma más clara y concisa, buscando representar los atributos y métodos de las clases bajo el estándar impuesto por el lenguaje UML, permite su comprensión como si se tratara de los planos para crear un software.

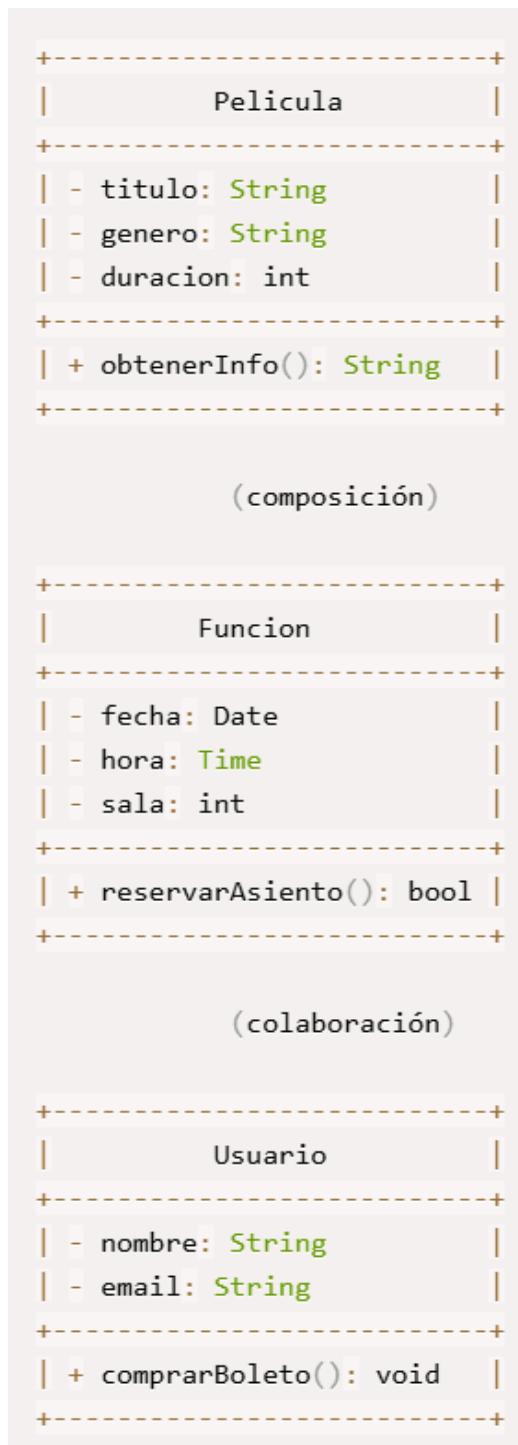
¿Qué ventajas ofrece antes de la implementación?

-Permiten la visualización y planificación de la programación orientada a objetos dentro de un grupo de trabajo, el tipo de interacción que poseen las clases entre sí y los atributos y métodos a utilizar.

Paso 2: Lectura de un Diagrama de Clases

Se les proporciona el siguiente diagrama de clases correspondiente al sistema

CINECLUB



Resuelvan en equipo:

¿Cuáles son las clases principales del sistema?

-Serían las clases Película, Función y Usuario.

¿Cómo se representan los atributos y métodos?

-Los atributos: están contenidos en el cuadro central, en primer lugar se agregar el símbolo (+) o (-) y/o (#) dependiendo de la visibilidad, luego el nombre del atributo y finalmente el tipo de dato al cual corresponde este atributo.

- Métodos: están escritos en el cuadro inferior del contenedor, caracterizando su visibilidad también por los símbolos “+”, “-” y “#”, seguido por el nombre del método y el tipo de dato que entrega.

¿Qué significa la composición entre Película y Función?

En esta relación de composición, la clase *Función* tiene una dependencia de la clase *Película*, debido a que sin la clase *Película*, no puede existir la clase *Función* debido a que esta recibe información desde *Película* para poder existir. Sin embargo, la *Película* sí puede existir por sí misma, ya que no recibe información de ninguna otra clase. Por lo tanto, la representación en el diagrama deberá ser de una línea sólida y una punta de rombo relleno apuntando hacia la clase *Película*.

¿Cómo se representa la colaboración entre Usuario y Función?

La colaboración entre las clases Usuario y Función deben ser representadas con una línea simple al tratarse de una asociación, puesto que ambas clases no requieren de la otra para existir, sino que trabajan en conjunto para la realización de un fin.

¿Cómo se relacionan estas clases con la plataforma web que desean construir?

Clase Película:

En la plataforma web, la información de esta clase (título, género, duración) será lo que se muestre en el catálogo de películas con toda la información necesaria para cada evento de forma ordenada y así los clientes puedan acceder a ella en un solo lugar. Para esto, por ejemplo, el método obtenerInfo() podría ser invocado por el backend para generar la página de detalles de una película.

Clase Funcion:

Se manifestará en la programación de proyecciones. Cada instancia de Función será una fila en la tabla de horarios del sitio, mostrando cuándo se proyecta una película. Esta información

es crucial tanto para “CINECLUB”, permitiéndoles mantener un orden y control respecto de las funciones que deberán proyectar. Así como también para los clientes para poder elegir cual función les acomoda más.

El método reservarAsiento() es la lógica que se ejecutará cuando un usuario haga clic en el botón "Reservar" en la página de horarios. Esto permitirá también mantener un control de cuántos boletos se han vendido para cada función y película.

Clase Usuario:

Esta clase se traduce en el perfil del usuario. Los atributos como nombre y email son la información básica que se guarda cuando el usuario se registra. El generar un usuario en la plataforma, permite asignar información precargada a cada compra que se pueda hacer en el futuro y abre la opción a otros elementos como pueden ser promociones.

Dentro de los métodos de esta clase se encuentra comprarBoleto(), que es la acción que se desencadena cuando el usuario completa el proceso de compra. La relación entre Funcion y Usuario indica que la acción de comprarBoleto() por parte de un Usuario afecta a la Funcion (por ejemplo, disminuyendo los asientos disponibles).

Paso 3: Creación de un Diagrama de Clases (60 min)

Cada equipo debe diseñar su propio diagrama de clases UML con base en el siguiente requerimiento:

- La plataforma debe permitir a los administradores agregar películas y funciones.**
- Los usuarios deben poder registrarse, consultar horarios y comprar boletos.**
- Se debe incluir una clase Boleto que almacene información de cada compra.**

El diagrama debe representar:

- Clases principales y sus atributos/métodos.**
- Relaciones entre las clases (colaboración y composición).**
- Notación UML adecuada.**