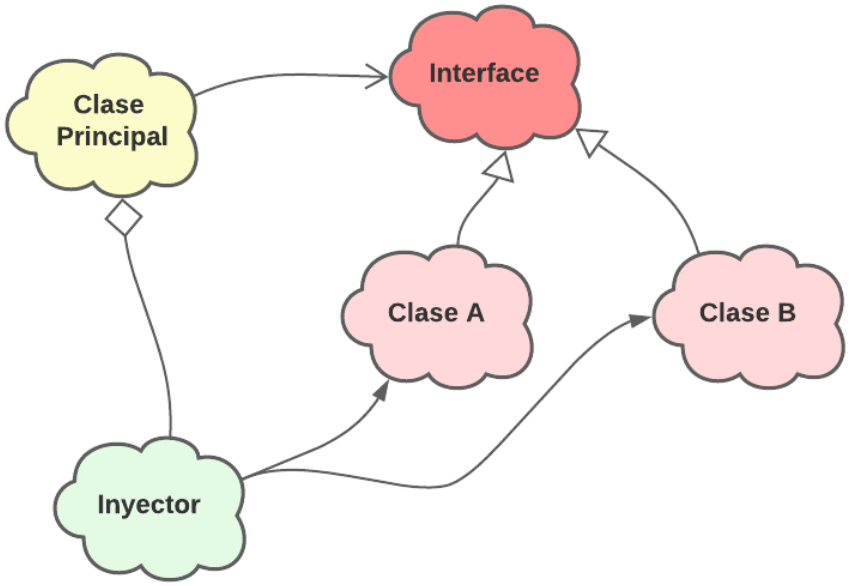
Inyección de dependencias

La inyección de dependencias sirve para eliminar la relación que tienen los elementos altamente acoplados. Esto quiere decir que los objetos no deben de crear aquellos objetos que necesitan para funcionar y deben trasladar esta responsabilidad a otros servicios, o sea, su creación dependerá de otros objetos.



Patrón Modelo – Vista – Controlador (MVC)

El modelo vista controlador es un patrón de arquitectura de software, comúnmente utilizado en aplicaciones de gran tamaño, su característica principal se encuentra en dividir cada uno de sus componentes y delegar tareas específicas a cada uno, con la intención de hacer las aplicaciones más escalables y sostenibles.

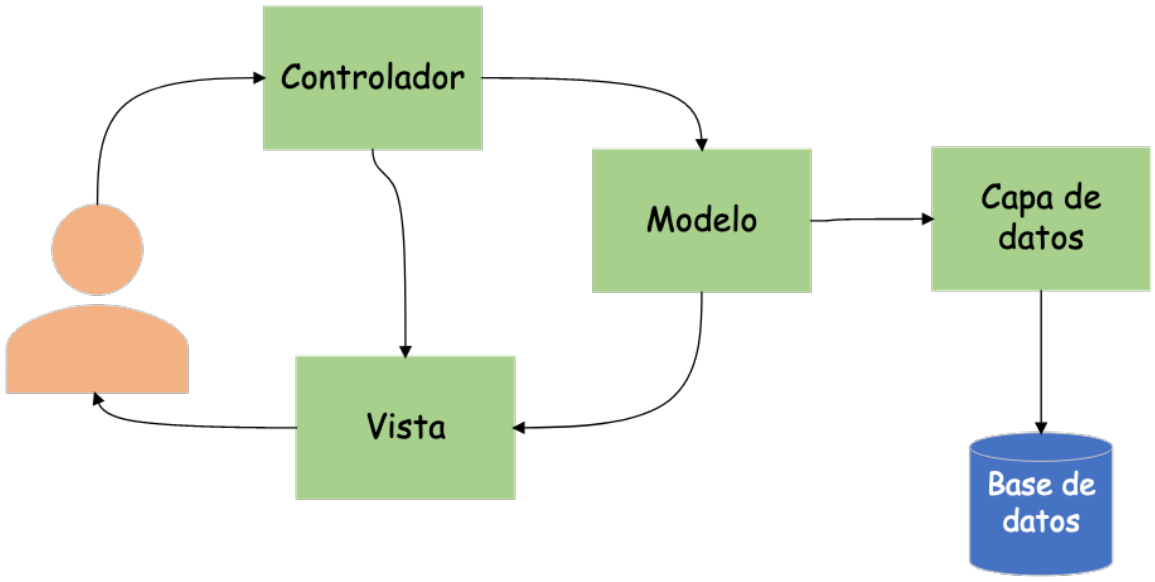
Este modelo implementa 3 componentes principales:

El modelo: se encarga de representar la información que contiene la aplicación, es probable que esta capa pueda acceder a la base de datos, pero no contiene ninguna lógica de cómo deben ser mostrados los datos al usuario.

Vista: Es la encargada de procesar los datos que provienen por el controlador o el modelo, esta contiene todos los elementos que serán mostrados al usuario, la vista no tiene comunicación con el modelo y es el controlador el que decidirá cuales serán los datos que serán mostrados al usuario.

Controlador: es el encargado de orquestar la distribución de la información, se encarga principalmente de recibir las peticiones de los usuarios y sirve como intermediario entre el

intercambio de información que se realiza entre el modelo y la vista.



¿Cómo funciona el patrón MVC?

El controlador recibe las peticiones http provenientes del usuario, entonces el controlador analiza la petición y se comunica con el modelo expresando los datos que necesita para que sean devueltos a la vista.

El modelo analiza la solicitud de la petición y devuelve los datos que son necesarios para que sean mostrados a la vista, el controlador recibe los datos del modelo y es quien decide que datos deberán ser enviados y mostrados en la vista.

Por último, la vista recibo los datos que deberán ser mostrados y los envía al navegador en donde deberán ser procesados para poder ser visualizados correctamente por los usuarios.

Tipos de excepciones que tenemos en java

Las excepciones en Java ocurren cunado ocurre un error que impide que el programa compile o siga funcionando, cuando una excepción ocurre, el programa dejara de funcionar y será responsabilidad de los programadores implementar las medidas necesarias para evitar que el programa se trunque y los usuarios puedan continuar haciendo uso de él.

La manera de proteger al programa de estos errores y evitar que el programa detenga su ejecución, es por medio del tratamiento de errores con try-catch. El try-catch es un bloque de código en el que tenemos que introducir el código que puede generar una excepción, de esta manera Java gestionará de una manera diferente si ocurre un error y evitará que el programa detenga su ejecución.

¿Cuáles son los tipos de errores que tenemos en Java?

Tiempo de compilación:

los errores que suceden en tiempo de compilación, son los que se pueden generar por parte de los programadores, este tipo de errores son generados cuando se olvida un punto y coma o se tiene un error en la sintaxis de alguna línea de código, estos errores normalmente impiden que el programa se pueda ejecutar.

Errores en tiempo de ejecución:

este tipo de errores ocurren cuando el programa ya se ha puesto en ejecución, normalmente no ocurren por algún error ocasionado por el programador, pero puede ser que ocurran por un problema de lógica en el código.

Dentro de este tipo de errores tenemos la siguiente clasificación:

Error:

este tipo de errores normalmente ocurren por un factor externo al programa y a la programación de su código, ocurre cuando un factor externo al programa impide que el programa se pueda ejecutar o que su ejecución sea detenida, puede que el programa se detenga por alguna falla en el hardware o hasta por que el equipo se haya quedado sin conexión a internet.

Normalmente cuando ocurre este tipo de excepciones no se puede hacer nada para solucionarlo.

Exception:

este tipo de errores son los que si pueden ser gestionados por java, cuando una excepción de este tipo ocurre o puede ocurrir, se pueden tomar medidas de prevención para que una vez que sea detectado por Java, impidamos que el programa termine su ejecución y dirigir su ejecución hacía un camino más amigable.

Runtime / Unchecked exceptions:

estos son un tipo de errores que pueden ser producidos por los programadores, por ejemplo, cuando se apunta a la posición de un arreglo que no ha sido declarada o cuando se quiere almacenar un tipo de dato en un lugar incorrecto. Java no obliga a los programadores a tratar el código con una excepción, por lo que puede ser más común que ocurran este tipo de excepciones.

IOException / Checked exception:

son denominadas excepciones comprobadas, por lo que Java siempre obligará al programador a tratarlas con un try-catch, este tipo de excepciones son ajenas a la lógica o a algún error cometido por el programador ya que pueden ocurrir, por ejemplo, cuando se cambia la ubicación de un archivo que es utilizado por el programa, lo que generará una excepción.

Procesos síncronos y asíncronos

Proceso síncrono: un proceso síncrono es aquel en el que una instrucción tiene que esperar a que termine la instrucción anterior para poder iniciar su ejecución.

Proceso asíncrono:

Los procesos asíncronos no esperan la confirmación de la instrucción anterior y continúan con su ejecución en el momento que es indicado, esto permite tener una mejor respuesta de los programas y reduce el tiempo de espera del cliente.