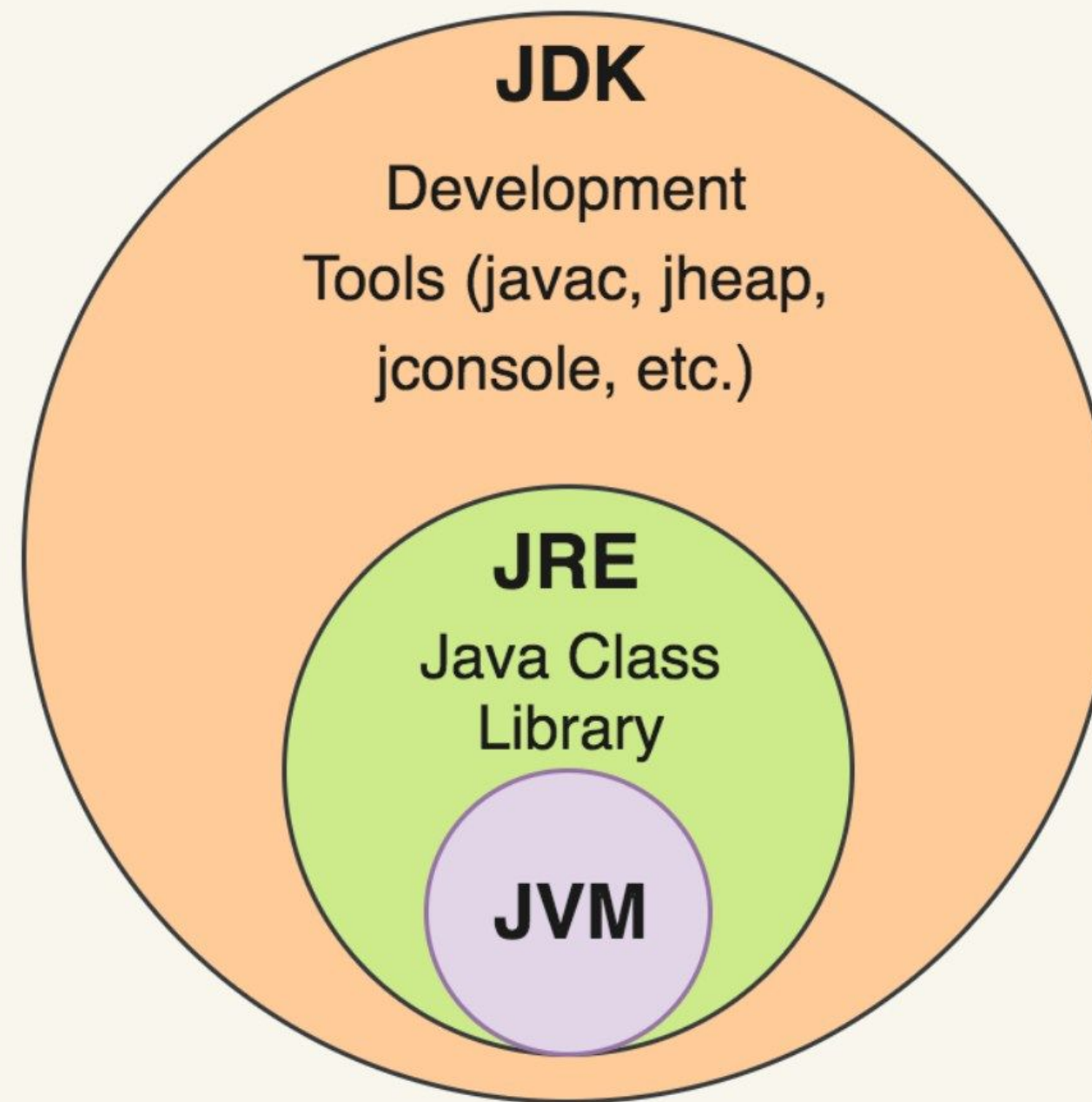




ARQUITECTURA JAVA


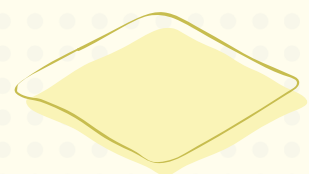
ESTEPHANY GONZALEZ
IPUZ

JDK, JRE y JVM

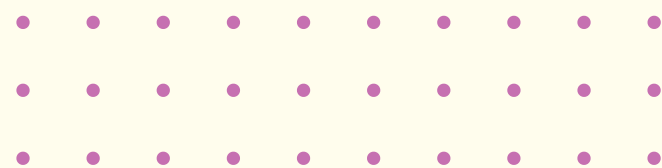





JDK (JAVA DEVELOPMENT KIT)



Se trata de un paquete de software que puede utilizar para desarrollar aplicaciones basadas en Java. El Entorno de ejecución de Java es un complemento necesario para ejecutar programas de Java. Java Development Kit es necesario para desarrollar aplicaciones de Java.





Herramientas de JDK para desarrolladores

- **JAVAC**

Compilador de Java ,que convierte el codigo fuente escrito en java en codigo Byte para su ejecucion en una JVM

- **JAVA**

Herramienta de linea de comandos que permite ejecutar aplicaciones java compiladas

- **JAVADOC**

Genera Documentacion automatica apartir de comentarios Incluidos en el codigo fuente.

- **JBC**

Depurador de java que permite al desarrollador encontrar y corregir errores de codigo.





Herramientas de JDK para desarrolladores

- **JAR**

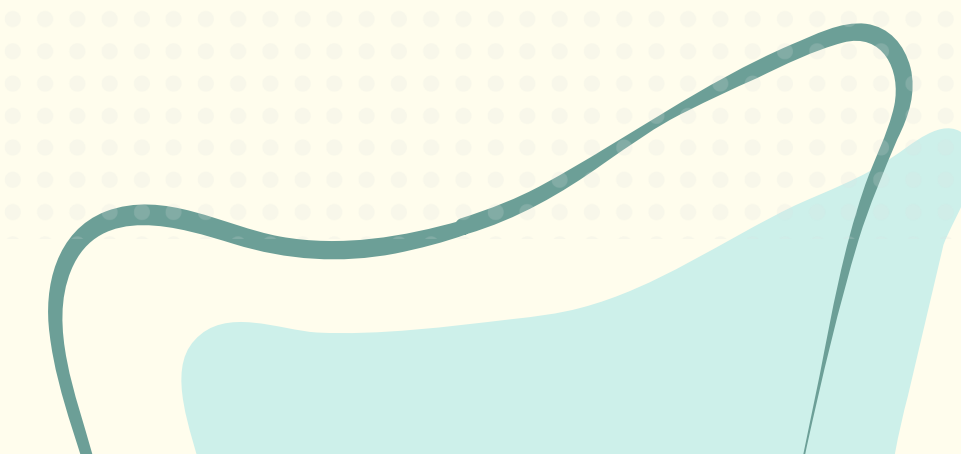
Crea y gestiona archivos JAR(Archivos Java Archive) que son contenedores de clase java y recursos relacionados.


- **JPS**

Lista los procesos de java en ejecución.

- **JAVAP**

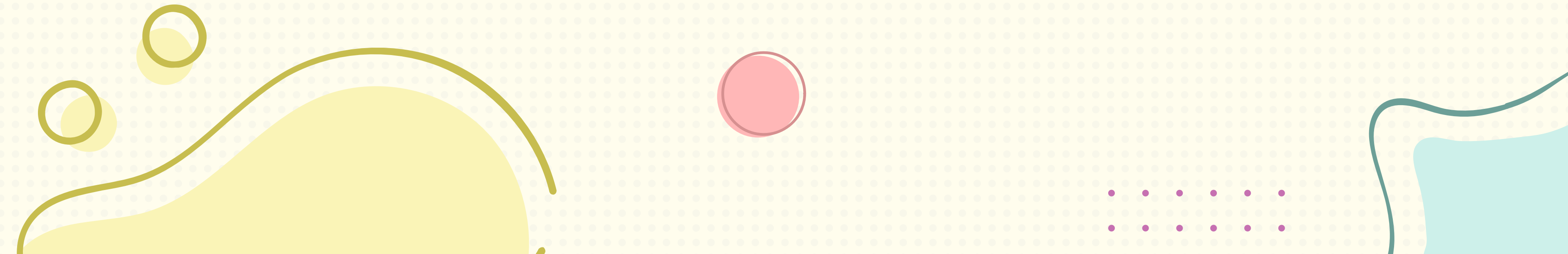
Desambla y visualiza información detallada sobre clases y métodos Java.






JRE (JAVA RUNTIME ENVIRONMENT)

es un software que los programas Java necesitan para ejecutarse correctamente . Java es un lenguaje informático que impulsa muchas aplicaciones web y móviles actuales. El JRE es la tecnología subyacente que se comunica entre el programa Java y el sistema operativo.





JVM (JAVA VIRTUAL MACHINE)

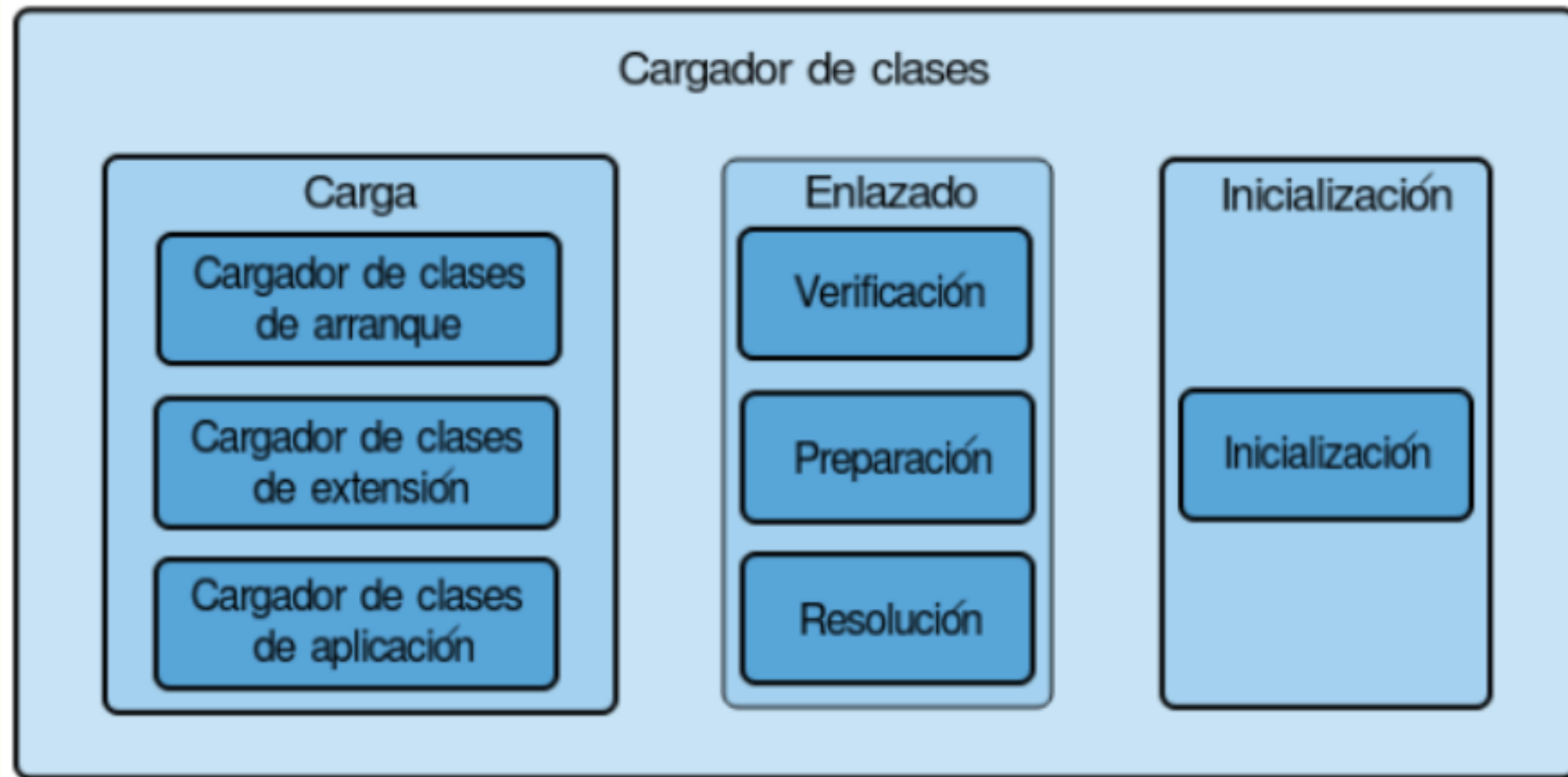
JVM Ejecuta el programa de java linea a linea. Los desarrolladores pueden configurar ajustes con la Jvm.

JVM es esencialmente una parte de JRE , No se puede descargar por separados, si no en cambio al instalar JRE se puede Instalar JVM.

LOS 3 COMPONENTES DE LA JVM

- Cargador de clases(Class Loader)
- Area de datos/ Memoria en tiempo de ejecucion (Runtime memory/Data area)
- Motor de ejecucion (Execution Engine)

Cargador de clases



Cargador de clases

se encarga de cargar las clases Java en memoria durante la ejecución del programa. su tarea principal es buscar, cargar y verificar las clases necesarias para que el programa se ejecute correctamente.

LA CLASE MAIN ES LA PRIMERA CLASE EN SER CARGADA.

EXISTEN 3 CARGADORES DE CLASES INCORPORADOS:

- **CARGADOR DE CLASES DE ARRANQUE (BOOTSTRAP CLASS LOADER)**

Carga los paquetes estándar de Java como java.lang, java.net, java.util, java.io, etc. localizados en el fichero rt.jar y otras bibliotecas fundamentales presentes en el directorio

- **CARGADOR DE CLASES DE EXTENSIÓN (EXTENSION CLASS LOADER)**

Carga las extensiones de las bibliotecas estándar de Java presentes en el directorio

- **CARGADOR DE CLASES DE APLICACIÓN (APPLICATION CLASS LOADER)**

es el responsable de encontrar y cargar los bytecodes que definen las clases. Una vez que se cargan, los bytecodes son verificados antes de que se puedan crear las clases reales

ENLAZADO

Después de que una clase haya sido cargada en memoria, se somete al proceso de enlazado. Enlazar una clase o una interfaz supone resolver las referencias externas y dependencias, integrando la clase en el conjunto del programa que hace uso de ella.



Verificación

esta fase verifica la corrección estructural del archivo.class contrastándolo frente a un conjunto de restricciones o reglas. Si la verificación falla, se produce `VerifyException`.

Ejemplo:

String Variable = 5 -> Verify Exception

Esta colocando un Int cuando se supone que debe asignar un valor tipo string = "hola";



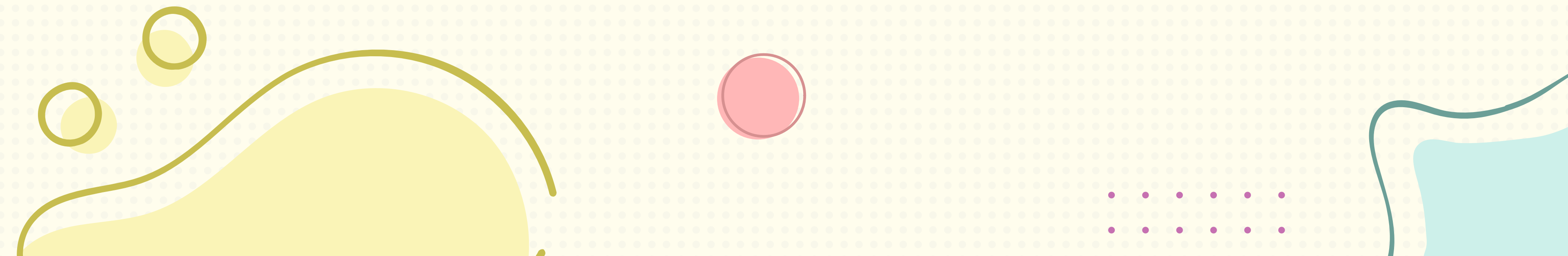


PREPARACIÓN

la JVM asigna memoria para los campos estáticos de una clase o interfaz, inicializándolos con valores por defecto.

Resolución

esta fase verifica la corrección estructural del archivo.class contrastándolo frente a un conjunto de restricciones o reglas. Si la verificación falla, se produce `VerifyException`.



MUCHAS
GRACIAS POR
LA ATENCIÓN!