

LABORATORIO II - JAVA

TSP - 2017



COMISIÓN DE INGRESANTES

- Jueves de 21 a 23 hs – Cesar Pizzi (Teoría)
- Viernes de 21 a 23 hs – Facundo Matoff y Guido Benedetti (Práctica)

PARCIALES Y NOTAS

- 1 Parcial Teórico
- 1 Parcial Práctico
- 1 Recuperatorio Teórico
- 1 Recuperatorio Práctico
- 1 Trabajo Práctico Nivelador
- 1 Trabajo Práctico Integrador Final.
- Nota = 6 (60%) para Regularizar.
- Nota = 7 (75%) para Promocionar.

RÉGIMEN DE REGULARIDAD Y PROMOCIÓN

2 Parciales Aprobados

2 Trabajos Prácticos presentados y
aprobados



REGULAR

Rinde Final Teórico/Práctico.

2 Parciales Promocionados

2 Trabajos Prácticos presentados y
Trabajo Final Promocionado



PROMOCIONADO

Promociona la Materia sin Final

- Se pueden recuperar **ambos** Parciales para **Regularizar** la Materia.
- Se puede rendir solo **un** Parcial para **Promocionar** la Materia.

EMAIL

labii.tsp.frre.utn@gmail.com

CHAT TELEGRAM

[A cargo de Guido]

GUÍA PRÁCTICA y EJERCICIOS

- GitHub: labii-tsp-frre-utn/2017

<https://github.com/labii-tsp-frre-utn/2017>

HERRAMIENTAS A UTILIZAR



BIBLIOGRAFIA

- [Thinking in JAVA 4th](#)

OBJETIVOS

- Aplicar varios conceptos de Programación Orientada a Objetos y Patrones de Diseño.
- Aprender a desarrollar con el Lenguaje de Programación Orientado a Objetos JAVA
- Aprender sobre desarrollos de aplicaciones Web con JAVA y Unit Testing.

CONTENIDOS

- Introducción a la Programación Orientada a Objetos.
- Diseño de Sistemas con Programación Orientada a Objetos con JAVA
- Diseño de Sistemas y Configuraciones con JAVA
- Diseño de Cases de Tests con JAVA

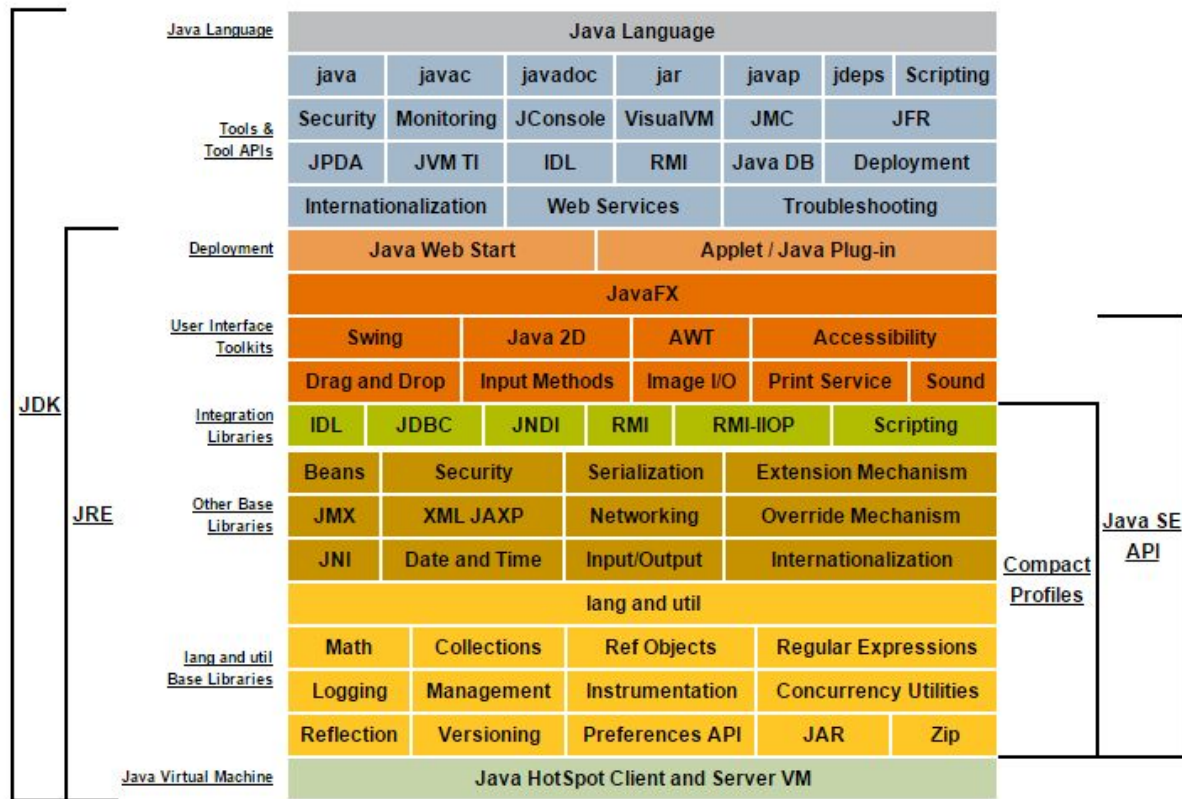
JAVA: LENGUAJE DE PROGRAMACIÓN

1. Instalación y configuración
2. JDK, JRE, JVM
3. Escribir un programa con JAVA.
4. Compilador: javac
5. Intérprete: java
6. Ejecutar de programa.

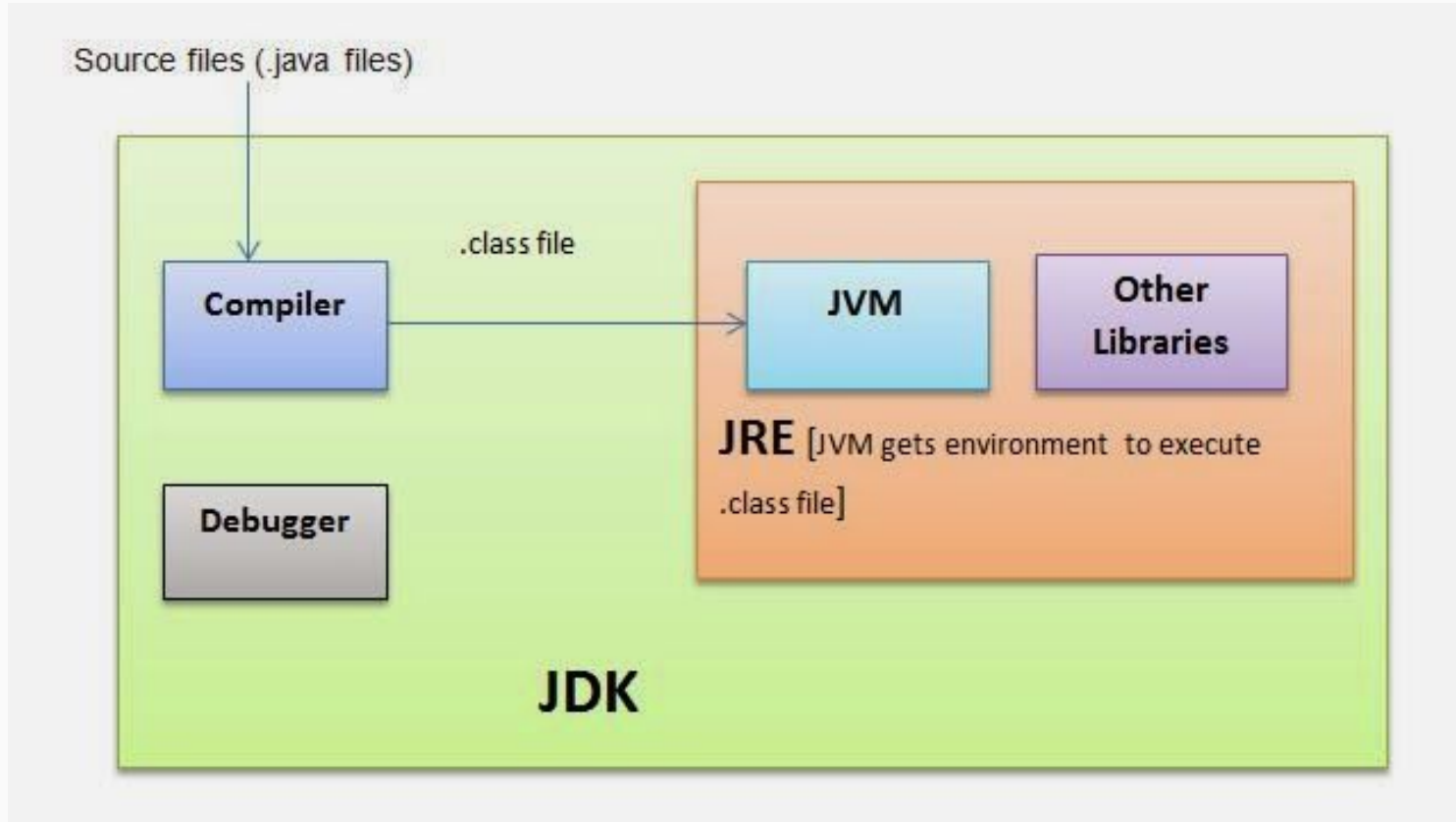


JAVA: Diagrama conceptual

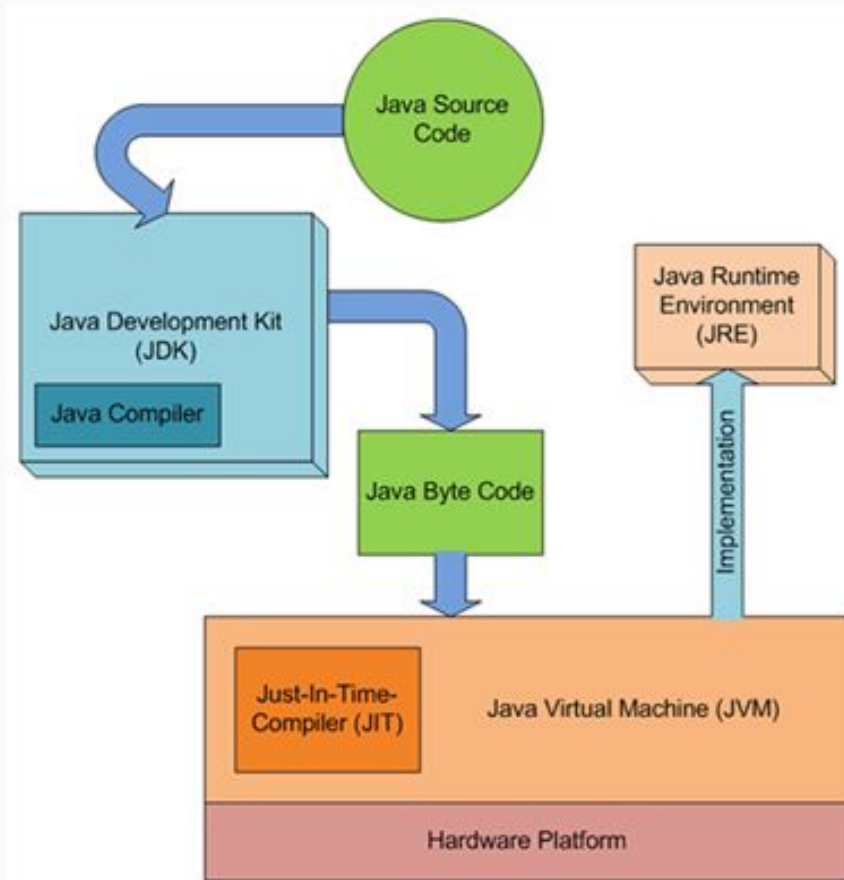
Description of Java Conceptual Diagram



JDK => JRE => JVM



JDK => JRE => JVM



JAVA: APLICACION DE CONSOLA

1. Leer valores introducidos por el Teclado.
2. Imprimir valores o resultados.
3. Scanner
4. System.in
5. System.out

JAVA: IDE IntelliJ

1. Instalación
2. Configuración
3. Crear una Aplicación de Consola.

TIPOS DE DATOS PRIMITIVOS

1. Tipos de datos:

- Enteros
- Flotantes
- Caracteres
- Cadenas de Caracteres (String)
- Booleans

ESTRUCTURAS SIMPLES

1. Variables
2. Constantes
3. Operadores
4. Métodos y Funciones
5. Argumentos
6. Alcance Local y Global
7. Palabras Reservadas

ESTRUCTURAS DE DECISIÓN

1. Operadores lógicos
2. Condicional simple
 - IF - ELSE - ELSEIF
3. Condicional múltiple
 - CASE

ESTRUCTURAS DE ITERACIÓN

1. Contadores
2. Acumuladores
3. FOR
4. WHILE
5. DO...WHILE

ARREGLOS Y MATRICES

1. Concepto y manejo de memoria.
2. Definición
3. Recorridos

INTRODUCCION A LA PROGRAMACIÓN ORIENTADA A OBJETOS

1. Concepto de Encapsulamiento.
2. Clases
3. Objetos
4. Instanciación
5. Atributos
6. Propiedades y modificadores
7. Métodos
8. Métodos de Clase

JAVA: Tipos Primitivos vs Tipos por Referencia

1. Tipos Primitivos
2. Tipos por Referencia
3. Strings y manipulacion
4. Clase Date y Times
5. Conversión de Tipos
6. Casting de Tipos
7. Garbage Collector

OOP: HERENCIA, POLIMORFISMO Y SOBRECARGA

1. Herencia simple
2. Herencia múltiple
3. Polimorfismo
4. Sobrecarga
5. Clases en JAVA

Modificadores de Acceso

1. Package
2. Public
3. Private
4. Protected

Abstract, Static y Final

1. Uso en Variables o Atributos

- a. Constantes

2. Uso en métodos

- a. Métodos abstractos
- b. Métodos Estáticos o de Clases
- c. Métodos Final

3. Uso en clases

- a. Clases Abstractas
- b. Clases Final

La Clase OBJECT

1. Propiedades de la Clase Object.

- a. Clase base o template

2. Métodos

- a. `toString()`
- b. `Equals()`
- c. `getClass()`
- d. `hashCode()`

INTERFACES

1. Introducción
 - a. Definir una interfaz
2. Diferencia con Clases Abstractas
3. Usos
4. Up-casting
 - a. Herencia de Clases

ENUMERACIONES

1. Introducción
2. Definiciones con enum.

DEBUGGING

1. Concepto de Debugging
2. Pasos para Debugger un Programa en Ejecución
 - a. Hacer un step-into en un método.
 - b. Validar contenido de variables y resultado de operaciones sobre y con variables.

EXCEPTIONS

1. Introducción
2. Tipos de Exceptions
3. Try..Catch..Finally
4. Custom Exceptions
5. Clase Exception, Throwable
6. Analisis del Stack Trace de una Exception

COLECCIONES

1. Introducción
2. Interfaces para Collections.
 - a. List
 - b. Set
 - c. Map
3. Tipos Principales de Collections
 - a. ArrayList
 - b. HashSet
 - c. HashMap
4. Operaciones sobre diferentes Collections

GENERICS

1. Introducción
2. Generics como la solución al Casting de tipos.
3. Generics en Collections
 - a. `ArrayList<E>`
 - b. `HashSet<E>`
 - c. `HashMap<E>`

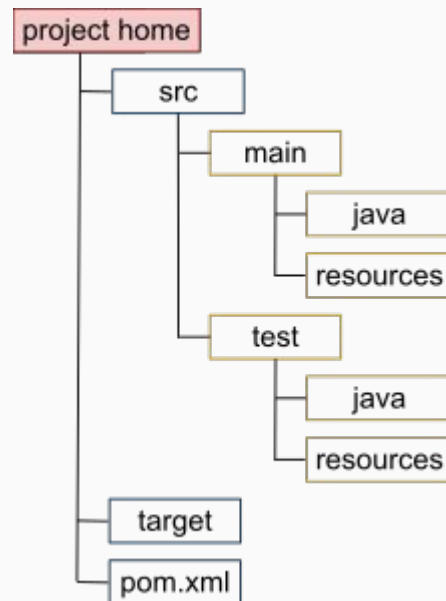
TESTS UNITARIOS (Unit Testing)

1. Introducción a Test Unitarios.
 - a. Qué son los Test Unitarios.
2. Introduccion a libreria JUnit4 para Tests Unitarios
3. Crear una Clase de Tests
4. Ejecutar Tests Unitarios

ESTRUCTURA DE PROYECTOS EN JAVA

1. Introducción a MAVEN
2. ¿Qué es el archivo POM?
3. Estructura de Carpetas
4. ¿Que es un Artifact?

maven



INTRODUCCIÓN A SERVIDOR WEB

1. Introducción a Servidor Web
2. ¿Qué es un Sitio Web?
3. ¿Qué es un Web Service o Servicio Web?
4. ¿Qué es un Cliente Web?
 - a. Chrome
 - b. Internet Explorer
 - c. Postman
5. ¿Qué es un Request y un Response?
 - a. HTTP

INTRODUCCIÓN LENGUAJE A XML y JSON

1. Introducción
2. Estructura
 - a. XML
 - b. Json
3. Usos comunes
 - a. Configuraciones
 - b. REST API

INTRODUCCION A SERVICIOS REST

1. ¿Qué es un Request y un Response?
2. Introducción a Servicios REST
3. ¿Qué es un Verbo HTTP?
 - a. GET
 - b. POST
 - c. PUT
 - d. DELETE
4. ¿Qué es REST?
5. ¿Qué es una API?
6. Probar Servicio REST
 - a. Postman

SPRING-BOOT

1. Introducción a Spring-Boot
2. Configuración de Sprint-Boot con MAVEN
 - a. Configure POM file
 - b. Download/Import dependencies
3. Crear una REST API con Spring-Boot
 - a. GET
 - b. POST