

# FUNDAMENTOS DE JAVA

**15 consejos prácticos sobre Java que  
cambiarán la forma en que codificas**



**Eudris Cabrera**

@eudriscabrera





# Eudris Cabrera

Graduado de Ingeniería Telemática en la PUCMM. Entusiasta de la tecnología y el software libre, desarrollador Java, instructor ocasional (Java / Linux), aficionado del béisbol/sóftbol y los bailes latinos.

En el tiempo libre, escribe artículos, realiza charlas, participa en eventos tecnológicos y ayuda en la revisión de documentación técnica en el mundo Java y el código abierto.



¿De dónde soy?





*Twitter:*  
*@JavaDominicano*



*<https://jconfdominicana.org>*

# AGENDA



- 15 consejos prácticos sobre características fundamentales que todo programador Java debe tener en cuenta.
- Conocer técnicas y mejores prácticas para producir software con menos bug en Java.
- Puntos Extra (Bonus)





## Asuntos Legales


Los conceptos emitivos son basado en mi experiencia laboral y están bajo mi responsabilidad, no se puede considerar como una posición oficial de mis empleadores.

Todas las marcas registradas, así como todos los logotipos, imágenes, fotografías, audio y vídeos mostrados en esta presentación son propiedad de sus respectivos propietarios y/o representantes.


Su utilización es solamente para fines ilustrativos.

# Fundamentos de Java



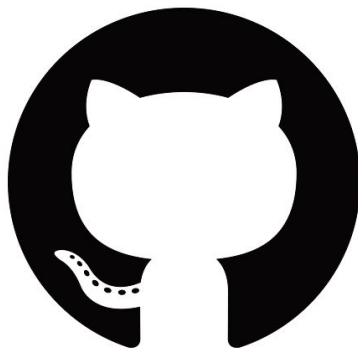
- 
1. Conocer los tipos de datos y sus valores por defecto.
  2. Concatenar las cadenas de caracteres de forma correcta
  3. Para cadenas grandes de caracteres usar **StringBuilder** en lugar de StringBuffer
  4. Evite Raw Type y en su lugar utilice tipos genericos
  5. Usar tipos enumerados (enum)
  6. Conocer los paquetes más utilizados
  7. Comprender el uso y manejo de excepciones
  8. Manejar los logs de forma más inteligente



- 
9. Tratar los recursos (de archivos, conexiones a base de datos, socket, etc.) de forma correcta
  10. Usar Varargs
  11. Siempre tener en cuenta que micro optimizaciones pueden mejorar el rendimiento de nuestra aplicación
  12. Utilizar Paths y Files
  13. Aprender cómo y por qué usar Expresiones Lambdas
  14. Utilizar el API Stream y las expresiones Lambda en conjunto para proporcionar potentes capacidades de filtrado
  15. Entender, aceptar y aprovechar Optional

# Código fuente de los ejemplos en GitHub

<https://github.com/ecabrerar/java-fundamentos-tips.git>



---

## Puntos Extra (Bonus)



# Herramientas para analizar tu código

Las herramientas de análisis estático de código se utilizan para cuidar la calidad del código y evitar la deuda técnica.

Las siguientes herramientas están disponibles para Eclipse y también para otros IDEs.

## 1. SonarLint

Es una extensión que ayuda a detectar y corregir problemas de calidad a medida que escribes código. Al igual que un corrector ortográfico, evita fallas para que puedan repararse antes de hacer push de tu código.

## 2. PMD

Detecta cosas como código duplicado, código muerto (variables, parámetros o métodos sin usar), complejidad de métodos (if innecesarios, etc.)



# Herramientas para analizar tu código

## 3. Findbugs

Identificar cientos de tipos de errores potenciales. Opera a nivel de bytecode en lugar de código fuente

## 4. Check Style

Se utiliza para comprobar que el código analizado cumple con una serie de reglas de estilo.



# Leer código

Intentar comprender cosas de proyectos más avanzados, descubrir la organización(flujos), siempre te ayudará a crecer como desarrollador.

1. Estudiar código de otros programadores y/o proyectos.
2. Leer código de proyectos de código abierto.
3. Leer código del JDK o de tu IDE favorito.
4. Leer código que tiene que ver con tu proyecto(Librerías, frameworks,etc)



# Participar en comunidades de desarrolladores

1. Socializar con otros desarrolladores.
2. Conocer un mentor.
3. Ser mentor de alguien.
4. Dar una charla en un evento relacionado con la comunidad.
5. Resolver incidencias en proyecto open source.





# Preguntas ?



**¡GRACIAS**  
**P O R**  
**ACOMPañARNOS!**

# Referencias

---

## Enlaces:



<https://www.codacy.com/blog/review-of-java-static-analysis-tools/>

<http://www.vogella.com/tutorials/Logging/article.html>

<https://www.baeldung.com/java-logging-intro>

<https://howtodoinjava.com/best-practices/java-exception-handling-best-practices/>

<https://github.com/ecabrerar/java-8-mas-alla-de-las-expresiones-lambda>

Effective Java 3rd Edition

[https://www.amazon.com/-/es/gp/product/B078H61SCH/ref=dbs\\_a\\_def\\_rwt\\_hsch\\_vapi\\_tkin\\_p1\\_i0#](https://www.amazon.com/-/es/gp/product/B078H61SCH/ref=dbs_a_def_rwt_hsch_vapi_tkin_p1_i0#)



**@eudriscabrera**

<https://www.linkedin.com/in/eudriscabrera>



**@eudris**

<http://www.slideshare.net/eudris>



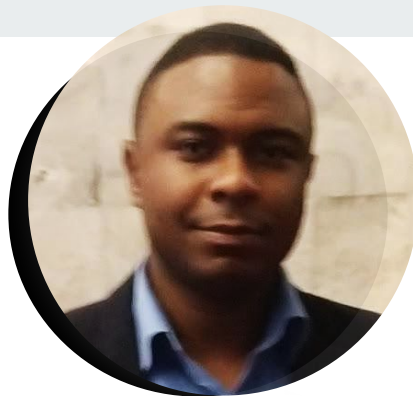
**@ecabrerar**

<https://github.com/ecabrerar>



**@eudriscabrera**

<https://twitter.com/eudriscabrera>



**Eudris Cabrera Rodríguez**

Ingeniero Telemático

*Desarrollador de Software / Consultor Informático*

*eudris@gmail.com*