Instituto Tecnológico de Pabellón de Arteaga

Desarrollo, Aplicación y Consultoría de Sistemas de Información

Tics 9no.

Lectura 7

Prof. Eduardo Flores Gallegos

Denise Kirstie Martínez Santana

7 de Diciembre del 2019

INSTITUTO TECNOLÓGICO de Pabellón de Arteaga

TEC

LECTURA 7

Desde 2010, he escrito extensamente sobre la programación de Node.js. A saber, cuatro ediciones de Node.js Web Development, además de otros libros y numerosas publicaciones de blog tutoriales sobre la programación de Node.js. Eso es MUCHO tiempo explicando Node.js y los avances en el lenguaje JavaScript.

Mientras trabajaba para Sun Microsystems creía en All Things Java. Presenté sesiones en JavaONE, co-desarrollé la clase java.awt.Robot, ejecuté el Concurso de Regresiones de Mustang (el concurso de búsqueda de errores para la versión Java 1.6), ayudé a lanzar la " *Licencia de Distribución para Java*" que era la versión previa a OpenJDK. Responda a las distribuciones de Linux para distribuir compilaciones JDK, y luego desempeñó un pequeño papel en el lanzamiento del proyecto OpenJDK. En el camino encontré un blog en java.net (un sitio web ahora desaparecido), escribiendo 1–2 veces a la semana durante aproximadamente 6 años discutiendo eventos en el ecosistema de Java. Un gran tema era defender Java contra aquellos que predicen la muerte de Java.

- Java está lleno de código repetitivo que oculta la intención de los programadores
- Spring & Spring Boot enseña una lección: empapelar sobre la complejidad crea más complejidad
- Java EE fue un proyecto de "diseño por comité" que cubre todo en el desarrollo de aplicaciones empresariales, por lo que es extremadamente complejo
- La experiencia de la programación de Spring es excelente hasta que no lo es, hasta que ese día aparece una oscura excepción imposible de entender desde las profundidades de un subsistema del que nunca has oído hablar que requiere más de 3 días solo para resolver el problema.
- ¿Cuál es la sobrecarga requerida en el marco para permitir que los codificadores escriban código cero?
- Si bien los IDE como Eclipse son potentes, son un síntoma de la complejidad de Java.
- Node.js fue el resultado de un individuo que perfeccionó y refinó la visión de una arquitectura ligera basada en eventos, hasta que Node.js se reveló.
- La comunidad de JavaScript parece apreciar la eliminación de boilerplate que permite a los programadores brillar
- La solución para Callback Hell, la función async / await, es un ejemplo de eliminación de boilerplate para que brille la intención de los programadores.
- Codificar con Node.js es una alegría
- JavaScript carece de la estricta comprobación de tipos de Java, que es una bendición y una maldición. El código es más fácil de escribir pero requiere más pruebas para garantizar la corrección
- El sistema de gestión de paquetes npm / hilado es excelente y de uso alegre, en comparación con la abominación que es Maven
- Tanto Java como Node.js ofrecen un rendimiento excelente, en contra del mito de que JavaScript es lento y, por lo tanto, el rendimiento de Node.js debe ser malo

- El rendimiento de Node.js está a la vanguardia de la inversión de Google en V8 para acelerar el navegador Chrome

- La intensa competencia entre navegadores hace que JavaScript sea cada vez más potente cada año, lo que beneficia a Node.js