

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Pabellón de Arteaga



Unidad 4

Lecturas 6,7,8,9

Presenta:

Juan Ignacio Ornelas Carreón

Docente:

Eduardo Flores Gallegos

Contenido

Lectura 6.....	2
Bibliografía.....	3
Lectura 7.....	3
Bibliografía.....	5
Lectura 8 y 9.....	5
Bibliografía.....	6

No hagas esto en producción

En el mes de marzo de 2017 recibí una llamada solicitando una revisión del código de un producto que se lanzará. Tuvo muchas empresa tuvo problemas con fugas de memoria, fallas espontáneas, carga lenta, aumento de CPU y tuvo que lanzar en un par de semanas.

En el momento en que se llevó a cabo la revisión del código se tomó alrededor de medio día descubrir la fuente de los problemas conocidos y otro medio día para redactar un documento para que su equipo de ingenieros los arreglara.

El problema que se presentó era la falta de experiencia, que realmente solo podían resolver si seguían construyendo y creciendo. Me he encontrado con eso antes. Es común, y creo que es saludable, bueno, la mayoría del tiempo, se vio en la necesidad de contratar al primer desarrollador, el cual se dio a la tarea de examinar al segundo desarrollador, y así hasta que tuvieron un equipo de desarrollo.

Resulta que es una muy mala idea “Muévete rápido y rompe cosas” cuando su negocio depende de un pequeño número de grandes clientes.

Se encuentra un gran equilibrio entre el movimiento rápido y el movimiento lento, difícil comunicar ese equilibrio porque cada tipo de producto exige un equilibrio diferente. Supongo que la intuición proviene de la experiencia, que es una respuesta terrible para alguien que intenta aprender.

[1]

Bibliografía

[1] «square,» 19 agosto 2018. [En línea]. Available: <https://stephenmann.io/post/dont-do-this-in-production/>. [Último acceso: diciembre 10 2018].

[2] Valerio Adrián Anacleto, 21 noviembre 2013. [En línea]. Available: http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=40. [Último acceso: 10 diciembre 2018].

Lectura 7

Java se ha convertido en una carga, la codificación con Node.js está llena de alegría

El resultado de un diseñador que paso varios años perfeccionando herramientas u objetos, creando nuevas cosas eliminando cosas innecesarias, con un objeto con los atributos correctos para este propósito.

Uno de los principales propósitos de spring, es una pila Java EE preconfigurada y fácil de usar.

Spring se ocupa de los muchos detalles en lo que otra persona se ocupa de la codificación, lo que lo hace un poco mas rápido. La clase Repositorio JPA sintetiza consultas de base de datos para métodos con nombres como "findUserByFirstName": no escribe ningún código, simplemente agregue métodos nombrados de esta manera a la definición del Repositorio y Spring se encargará del resto.

En lo que es este nivel de abstracción obviamente requiere un poco de lógica para encontrar todo y ejecutar las solicitudes.

Ryan Dahl fue el que dijo que usar hilos hechos para un sistema complejo de peso pesado.

El resultado fue un sistema ligero, un único hilo de ejecución, un ingenioso uso de funciones anónimas de JavaScript para devoluciones de llamadas asíncronas y una biblioteca de tiempo de ejecución que ingeniosamente implementó asynchronicity.

Después se encuentra el lenguaje JavaScript, los programadores parecen tener la estética de eliminar el texto de manera que la intención de los programadores pueda brillar claramente.

Los problemas en JavaScript uno de los mas comunes es Callback Hell, donde cada nivel de anidamiento complica el código, lo que dificulta aún más el manejo de errores y resultados.

Claridad supuesta a través de tipos e interfaces bien definidos.

Es un problema que se considera un defensor de Java que se murió en la lana fue que la comprobación de tipos estricta permite escribir aplicaciones enormes.

un defensor de Java que se murió en la lana fue que la comprobación de tipos estricta permite escribir aplicaciones enormes.

La única manera de mantener al programador Java en su sano juicio, es desplegando los campos disponibles en un objeto, describiendo los parámetros del método, ayudando a construir clases, ayudando con la refactorización, y todo el resto de las facilidades ofrecidas por Eclipse, NetBeans, e IntelliJ.

Node.js anima a los programadores a dividir sus programas en unidades pequeñas, el módulo.

Un modulo es:

Autocontenido.

Límite fuerte.

Exportaciones explícitas.

Importaciones explícitas.

Potencialmente independiente.

Más fácil de razonar.

Más fácil de probar.

En la gestión de paquetes de Maven existen y funciona bastante bien, y supuestamente también funcionan en Gradle. Pero no es tan útil / útil / potente como el sistema de administración de paquetes para Node.js.

En Node.js hay dos excelentes sistemas de gestión de paquetes que funcionan en estrecha colaboración. Al principio, npm, y el repositorio de npm, fue la única herramienta de este tipo.

Tanto Java como JavaScript tienen enormes incentivos para correr rápido. En Java y Node.js, el incentivo es un código rápido del lado del servidor. En el navegador JavaScript, el incentivo es un mejor rendimiento de la aplicación del lado del cliente; consulte la siguiente sección sobre aplicaciones de Internet enriquecidas.

Después de tanto tiempo de la idea de RIA se esta comenzando a dar frutos. Con Node.js en el servidor ahora podemos tener ese nirvana, pero con JavaScript en ambos extremos del cable. Todavía se pueden desarrollar otros tipos de aplicaciones de escritorio de Java, y la competencia entre los IDE de NetBeans y Eclipse sigue vigente y, como resultado, sigue vigente.

[2]

Bibliografía

[1] «square,» 19 agosto 2018. [En línea]. Available: <https://stephenmann.io/post/dont-do-this-in-production/>. [Último acceso: diciembre 10 2018].

[2] d. herron. [En línea]. Available: <https://blog.sourcerer.io/why-is-a-java-guy-so-excited-about-node-js-and-javascript-7cfc423efb44>. [Último acceso: 10 diciembre 2018].

[3] Valerio Adrián Anacleto, 21 noviembre 2013. [En línea]. Available: http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=40. [Último acceso: 10 diciembre 2018].

Lectura 8 y 9

Las reglas de diseño para la web moderna son prácticas muy inmutables, y debes seguirlas porque soy un defensor de desarrolladores, por otro lado son utilizadas por algunos de los mejores sitios web de hoy en día.

Los usuarios cuando por primera vez visitan su sitio web principalmente lo que se debe hacer es mostrarles un diálogo modal gigante que le indique que deberían instalar su aplicación móvil.

Es de gran importancia Informar a los usuarios sobre las cookies para que se dé cuenta que al ingresar a un sitio web estamos utilizando cookies para rastrearlas.

Los europeos hoy en día cuentan con algunas leyes molestas que enturbian las aguas en lo que podemos rastrear, en cambio Dinamarca es un estado socialista fallido, ya que sus usuarios no merecen la pena, para ayudar a nuestros usuarios, hagamos que el enlace de cancelación sea muy pequeño para minimizar las posibilidades de que lo toquen accidentalmente.

El proporcionar que los usuarios tengan la oportunidad de optar por no participar es de suma importancia ya que permitamos que los usuarios desactiven la opción de aplicaciones móviles.

[3]

Bibliografía

- [1] «square,» 19 agosto 2018. [En línea]. Available: <https://stephenmann.io/post/dont-do-this-in-production/>. [Último acceso: diciembre 10 2018].
- [2] d. herron. [En línea]. Available: <https://blog.sourcerer.io/why-is-a-java-guy-so-excited-about-node-js-and-javascript-7cfc423efb44>. [Último acceso: 10 diciembre 2018].
- [3] c. beyer, «Cómo diseñar para la web moderna,» agosto 2018. [En línea]. Available: <https://medium.com/s/silicon-satire/how-to-design-for-the-modern-web-52eaa926bae2>. [Último acceso: 10 diciembre 2018].
- [4] Valerio Adrián Anacleto, 21 noviembre 2013. [En línea]. Available: http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=40. [Último acceso: 10 diciembre 2018].