

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Control de autenticación y
 acceso (Spring Security)
 UDDI

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Web Services
 Rest Services
 Social SSO
 SSO (Cas)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



E-mail Contraseña

» Estás en: Inicio Tutoriales Dart, el lenguaje de programación web del futuro creado por Google



Moisés Belchín

Amante de las nuevas tecnologías, está en constante evolución y apuesta por los lenguajes más novedosos y potentes del mercado, por esto, comenzó a trabajar con Google Dart desde su nacimiento, en 2011. Contribuyendo al desarrollo del lenguaje, colaborando con el equipo de desarrolladores de Google, desarrollando parches para la corrección de errores y creador de blogdart.es, el primer punto de referencia sobre este lenguaje en castellano para acercarlo a la comunidad de desarrolladores de habla hispana.

Ver todos los tutoriales del autor

Fecha de publicación del tutorial: 2014-06-23

Tutorial visitado 59 veces Descargar en PDF

Dart, el lenguaje de programación web del futuro creado por Google

- ¿Qué es Dart y cómo surge?
- Compatibilidad de Dart con los sistemas y navegadores web modernos
- Características que hacen de Dart un lenguaje especial
- Programación asíncrona
- Herramientas para trabajar con Dart
- Conclusiones

Mucho se está hablando de la web 2.0, la programación estructurada, el desarrollo ágil, y los cambios en el paradigma de programación donde cobra especial interés el desarrollo de aplicaciones en el lado cliente.



DART

Durante los últimos años hemos visto cómo han evolucionado las webs y aplicaciones modernas realizando peticiones asíncronas para recuperar información del servidor y componer la interfaz que utilizará el usuario.

Por todo lo comentado, es por lo que debemos prestar especial interés a Dart, el nuevo lenguaje *open source* desarrollado por Google.

¿Qué es Dart y cómo surge?

Dart es un lenguaje de código abierto, estructurado y flexible, orientado a objetos, basado en clases, con herencia simple y soporte de interfaces, clases abstractas y tipado opcional de datos.

Ha llegado para permitir que los desarrolladores puedan crear aplicaciones web más complejas, con un mantenimiento más estructurado y con un rendimiento muchísimo superior a lo que nos tienen acostumbrados las aplicaciones web.

Detrás de su creación están *Lars Bak* y *Kasper Lund* dos de los creadores del motor V8 de JavaScript para Google Chrome, por lo que a la hora de desarrollar Dart tanto el rendimiento como la eficiencia se han tenido muy presentes.

También involucrado en la creación de Dart está *Gilad Bracha* creador del lenguaje Newspeak y coautor de las especificaciones de Java y del desarrollo de la segunda edición de la máquina virtual de Java.

A finales de 2014 la previsión de usuarios de internet es de 3 billones (el 40% de la población mundial), cifra que se ha doblado en los últimos 5 años. El cloud computing y los dispositivos móviles con acceso a internet, han traído consigo el acceso masivo a las aplicaciones web y para poder procesar esta grandísima cantidad de solicitudes que son recibidas por los servidores, es necesario descargarlos de todos los procesos que puedan ser realizados desde el lado cliente.

Además, nuevas formas de ocio digital están ganando terreno, como el uso de contenidos en formato video o la reproducción en streaming. Las previsiones indican que para 2017, el 70% de la navegación por internet estará dirigida al consumo de estos dos contenidos, lo que supone un gran consumo de recursos para los servidores que es necesario gestionar del modo más eficiente posible, y ahí es donde destaca Dart.

De sintaxis parecida a Java y con una filosofía de trabajo similar JavaScript, no ha sido creado para sustituirlo, pero sí pretende ofrecer una opción alternativa más moderna en el desarrollo web y eficiente, sobre todo, para grandes proyectos web centrados en el lado cliente. JavaScript no es lo suficientemente robusto para estos grandes desarrollos, padece de grandes problemas de gestión de memoria y el código se vuelve demasiado complejo de gestionar y mantener.

Por estos motivos, los desarrolladores de Google al crear sus aplicaciones utilizando JavaScript consideraron que no era

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

- » Screencasts de programación narrados en Español
 - » Sorteo de entradas para APIdays Mediterranea
 - » Concurso del Día de la Madre:
 - » Aprende gratis ReactiveCocoa
 - » Checklist de Scrum de Autentia
- [Histórico de noticias](#)

Últimos Tutoriales

- » Cómo integrar en Gradle un servidor Jetty o Tomcat
- » Taller de Ventas
- » Introducción a Spring Data Hadoop
- » Desarrollando una aplicación de detección de iBeacons
- » SCRATCH, aprende a programar desde niño

el lenguaje ideal para hacerlo e intentaron buscar otras alternativas como GWT, con lo que consiguieron un enfoque estructurado, pero muy poco optimizado para la web.

De aquí surge Dart, un lenguaje que permita combinar ambos mundos, el dinamismo de JavaScript con la potencia y estructura de lenguajes como C# o Java.

Dart permite desarrollar tanto aplicaciones complejas, del mismo modo que con Java o C#, como crear pequeños scripts para realizar tareas sencillas y concretas para páginas web, del mismo modo que con JavaScript.

Compatibilidad de Dart con los sistemas y navegadores web modernos

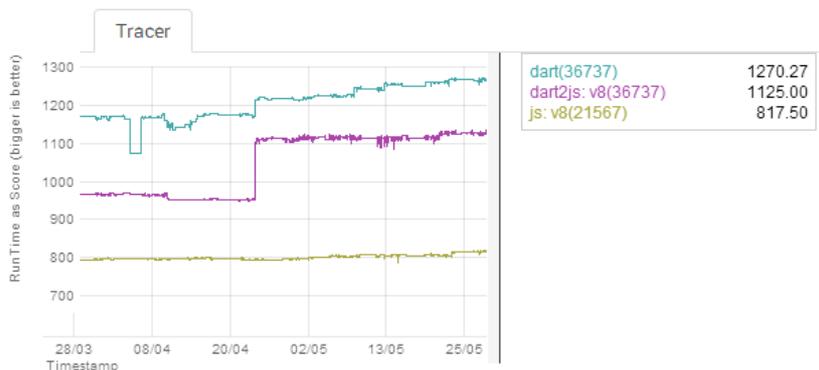
Dart utiliza la misma filosofía que Java, tiene su propia máquina virtual (DartVM) para ejecutarse, por lo que es multiplataforma y funciona perfectamente en MacOS, Linux y Windows.

Dart también es compatible con todos los navegadores web modernos porque puede ser compilado a código JavaScript y correr de manera nativa en los motores JavaScript de los navegadores.

En un futuro cercano Dart correrá de manera nativa en nuestros navegadores web cuando estos incluyan la máquina virtual de Dart.

Google Chrome planea incluir DartVM de forma nativa por lo que se podrán ejecutar aplicaciones Dart nativamente en el navegador y aprovechar todo el rendimiento que nos brinda este fantástico lenguaje.

Aún así, el rendimiento es sorprendente, Dart corriendo en su máquina virtual (DartVM) es del orden de dos veces más rápido que Java. Incluso compilado a JavaScript ya es más rápido que código nativo JavaScript corriendo en V8 en muchas de las pruebas de rendimiento.



<https://www.dartlang.org/performance/>

Características que hacen de Dart un lenguaje especial

Dart, es un lenguaje adelantado a su tiempo, ya que incluye gran parte de las funciones y APIs que incluirá la próxima versión 6 de JavaScript (esperada para finales de 2014).

Por ejemplo, estas son algunas de las novedades que veremos en JavaScript 6 y que Dart ya incorpora:

- Variables de ámbito de bloque.
- Valores por defecto en funciones.
- Parámetros opcionales por nombre.
- Interpolación de strings.
- Arrow functions.
- API Promises.

Las variables de ámbito de bloque son variables "vivas" sólo para ese ámbito de trabajo y fuera de él no existe. En JavaScript 6 se han creado para solucionar ciertos de los problemas de memoria de los que padece este lenguaje y se crean utilizando la palabra reservada let. En Dart no hace falta declararlas de ninguna forma especial, Dart aplica el ámbito de bloque de forma nativa.

JavaScript 6	Dart
<pre>var x = 1; console.log(x); // 1 if (true) { let x = 2; console.log(x); // 2 } console.log(x); // 1</pre>	<pre>var x = 1; print(x); // 1 if (true) { var x = 2; print(x); // 2 } print(x); // 1</pre>

Poder establecer valores por defecto en las funciones es algo muy interesante y útil.

JavaScript 6	Dart
<pre>function saludo(nombre='anónimo') { console.log('Bienvenido', nombre); }</pre>	<pre>void saludo({String nombre='anónimo'}) { print('Bienvenido a Dart, \${nombre}'); }</pre>

Los parámetros posicionales por nombre son extremadamente útiles, mejoran la auto documentación de llamada a las funciones o métodos y permiten llamar a las funciones sin necesidad de respetar el orden de los parámetros.

JavaScript 6	Dart
	<pre>void saludo({String nombre:'John', String apellidos:'Doe'}) {</pre>

<pre>function saluda({ nombre, apellidos }) { console.log(nombre, apellidos); }</pre>	<pre>print('\${nombre} \${apellidos}'); } void main() { saludo(); // John Doe saludo(nombre: 'Moises'); // Moises Doe saludo(apellidos: 'Belchin', nombre: 'Moises'); // Moises Belchin }</pre>
---	--

Poder realizar interpolación de cadenas trabajando con Dart o JavaScript es algo que ayudará mucho sobre todo a las personas nuevas que se acercan a cualquiera de estos dos lenguajes, les evitará muchos errores de aperturas y cierres de cadenas que nos han traído de cabeza a muchos de nosotros.

JavaScript 6	Dart
<pre>var x = 1; var y = 2; `\${x} + \${y} = \${x + y}` // "1 + 2 = 3"</pre>	<pre>var x = 1; var y = 2; print('\${x} + \${y}=\${x+y}'); // 1 + 2=3</pre>

Para funciones que tan solo contienen una simple expresión puedes utilizar *arrow functions*.

JavaScript 6	Dart
<pre>var sum = (n1, n2) => n1 + n2;</pre>	<pre>num sum(n1, n2) => n1 + n2;</pre>

Programación asíncrona

Para implementar el API Promises que tendrá JavaScript en su versión 6 Dart ha creado el API `dart:async` donde debemos prestar especial atención a la clase `Future` que permite crear código completamente asíncrono, pudiendo ejecutar múltiples tareas sin bloquear el sistema del usuario hasta que se reciba la respuesta.

Un objeto `Future` representa un proceso, función o valor de retorno que se recuperará en un futuro en modo diferido. Ejecutaremos una función, por ejemplo, y el valor de retorno de la función no lo obtendremos de manera inmediata sino en otro momento.

Cuando el objeto `Future` reciba el valor tras la ejecución de su tarea podremos trabajar con el resultado y mostrárselo al usuario, actualizar la interfaz o realizar cualquier otro proceso.

Veamos un sencillo ejemplo de uso de `Future` para realizar peticiones asíncronas.

```
var xhr = HttpRequest.getString('http://www.google.es')
..then((resp) => print(resp))
..catchError((e) => print('Error!'));
```

Herramientas para trabajar con Dart

Los ingenieros de Google han lanzado este nuevo lenguaje con un estupendo SDK. Puedes descargar *DartEditor*, el editor de Dart y ya tienes a tu disposición todas las herramientas para comenzar a trabajar con Dart, crear aplicaciones de línea de comandos, aplicaciones web en el lado cliente o aplicaciones que ejecuten en el lado del servidor.

El editor de Dart está basado en eclipse, tienes autocompletado de código, debug paso a paso y acceso directo a los paquetes de terceros que tengas instalados y al SDK de Dart por si necesitas consultar cualquier clase o método mientras desarrollas.

Dart también viene con *PUB* su gestor de paquetes. Esta herramienta permite localizar fácilmente una librería que necesites para tus proyectos, descargar el código y comenzar a utilizarlo.

PUB además se ocupa de las dependencias y de mantener actualizados tus paquetes a la versión que tú desees.

dart2js es otra de las herramientas que acompañan a Dart y es la herramienta encargada de realizar la compilación de código Dart a código JavaScript para que puedas ejecutar tus proyectos en cualquier navegador.

Como ya hemos comentado para Dart es importante el rendimiento, pero también es importante el rendimiento de tus aplicaciones y para que puedas realizar mediciones adecuadas y detectar posibles errores de memoria incorpora una nueva aplicación llamada *Observatory* que permite analizar el consumo de tiempo, pérdidas o fragmentación de memoria, las líneas de código ejecutadas y la memoria asignada.

Para completar este elenco Dart dispone de algunos extras muy interesantes:

- *dartfmt* herramienta para auto formatear código y que todos los proyectos tengan una estructura similar.
- *dartanalyzer* que se ocupa de analizar en tiempo real el código Dart que escribes para detectar errores en tiempo de desarrollo. También muestra información y avisos de interés para que puedas ser más productivo.
- *docgen* es el encargado de generar toda la documentación de tus proyectos a partir de los comentarios de documentación.

Conclusión

Como has podido ver Dart ya pone a tu disposición muy buenas características para que puedas comenzar a trabajar con él en el lado cliente, en el lado servidor o crear aplicaciones de línea de comandos.

Para muchos de nosotros, Dart se convertirá en el próximo estándar de programación y revolucionará el desarrollo web, marcando un antes y un después en el modo de estructurar y desarrollar aplicaciones web.



Desarrollador experto en tecnologías web y BigData. Trabaja con Dart desde su nacimiento en 2011 y colabora en su desarrollo y difusión entre la comunidad de desarrolladores hispano-hablantes. Es el creador de los primeros puntos de referencia sobre este lenguaje en castellano blogdart.es y autor del libro [Aprende Dart \(www.librodart.es\)](http://www.librodart.es).

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

[Share](#) |

[+1](#) [0](#)

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» [Regístrate](#) y accede a esta y otras ventajas «



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

IMPULSA Impulsores Comunidad [¿Ayuda?](#)

sin clicks **0 personas** han traído clicks a esta página

+ + + + + + + +

powered by [karmacracy](#)