aula MENTOR

GOBIERNO DE ESPAÑA

MINISTERIO DE EDUCACIÓN, CULTURA Y DEPORTE

Google Web Toolkit

AULA MENTOR

educacion.es

ISBN: 978-84-369-5429-6
Nipo: 030-12-323-6

Autores:
Clodoaldo Robledo
David Robledo

Edición y maquetación de contenidos:
Natalia Saavedra Mendoza

Diseño gráfico e ilustración de portada:
María Guija Medina

INTRODUCCIÓN:

¿QUÉ ES GWT?

ÍNDICE

1.1 ¿QUÉ ES GWT?

GWT (del inglés ***Google Web Toolkit***) es un *framework* (módulos dedesarrollo de aplicaciones) creado por Google que permite desarrollar fácilmente aplicaciones Web usando la tecnología AJAX.

Es compatible con la mayoría de los navegadores del mercado, pero no es fácil implementar aplicaciones en todos, ya que cada navegador necesita código específico en *JavaScript* distinto para el correcto funcionamiento de una aplicación web.

El concepto de Google Web Toolkit es bastante sencillo, pues se basa en usar el lenguaje Java con cualquier entorno de desarrollo (IDE) de Java; posteriormente, el compilador de GWT traduce el código Java a HTML y *JavaScript* que interpreta el navegador del usuario (lado cliente) y, en el servidor Web (lado servidor) se ejecuta el código Java compilado directamente.

1.1.1 SU HISTORIA

El entorno GWT fue creado a finales de 2006 por Google, y anunciado en la conferencia JavaOne de 2006. Versiones disponibles:

- ✓ GWT 1.0 Mayo 2006
- ✓ GWT 1.1 Agosto 2006
- ✓ GWT 1.2 Noviembre 2006
- ✓ GWT 1.3 Febrero 2007
- ✓ GWT 1.4 Agosto 2007
- ✓ GWT 1.5 Agosto 2008
- ✓ GWT 1.6 Abril 2009
- ✓ GWT 1.7 Julio 2009
- ✓ GWT 2.0 Diciembre 2009
- ✓ GWT 2.0.1 Febrero 2010
- ✓ GWT 2.0.2 Febrero 2010
- ✓ GWT 2.0.3 Febrero 2010
- ✓ GWT 2.0.4 Julio 2010
- ✓ GWT 2.1.0 Octubre 2010
- ✓ GWT 2.1.1 Diciembre 2010
- ✓ GWT 2.2.0 Febrero 2011
- ✓ GWT 2.3.0 Mayo 2011

Se puede observar en el listado anterior que se han publicado muchas versiones de GWT en un mismo año. Esto se debe a que Google ha implementado nuevas funcionalidades

en este entorno de desarrollo. Además, los navegadores disponibles en el mercado van evolucionando y es necesario actualizar el motor de GWT.

La buena noticia es que el código de una versión anterior se puede volver a recompilar con la nueva versión sin necesidad de modificar el código anterior.

La versión 2.3 es la última disponible en la fecha de elaboración del curso y es la que vamos a usar a lo largo del mismo. Además, en el código fuente disponible a lo largo del curso se usarán funciones normalizadas y estándares para que puedan ser compiladas en versiones futuras.

GWT es un conjunto de módulos de código abierto, por lo que no requiere la adquisición de ninguna licencia. Además, Google permite que los usuarios contribuyan al mismo con mejoras.

En agosto de 2010 Google adquirió la empresa Instantiations, responsable del entorno de desarrollo rápido de Java "Eclipse", disponible para plataformas Windows, Linux y Mac. Google ha introducido recientemente las librerías necesarias en este entorno para desarrollar aplicaciones Web con GWT. Por lo tanto, éste será el entorno de desarrollo que usaremos durante el curso.

1.1.2 ¿QUÉ ES AJAX?

Def.

AJAX, acrónimo de "*Asynchronous JavaScript And XML*" (*JavaScript* y XML asíncronos, es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el navegador del usuario y mantienen comunicación asíncrona con el servidor en segundo plano. De esta forma, es posible realizar cambios sobre la misma página sin necesidad de recargarla desde el principio (únicamente se baja del servidor la porción de información de la página que el usuario solicita). Esto significa aumentar la interactividad, velocidad y usabilidad.

AJAX es una combinación de estas tres tecnologías ya existentes:

- ✓ XHTML o HTML y hojas de estilos en cascada CSS para el diseño que acompaña a la información.
- ✓ Document Object Model (DOM) al que el usuario accede con un lenguaje de scripting, especialmente mediante implementaciones como *JavaScript* y JScript, para mostrar e interactuar dinámicamente con la información presentada. Es decir, se usa *JavaScript* para leer los contenidos de la página cargada e interactuar con el servidor web.
- ✓ Objeto XMLHttpRequest para intercambiar datos asincrónicamente con el servidor Web. XML es el formato usado comúnmente para la transferencia

de datos del navegador al servidor. Si bien, cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML. En la práctica siempre se usa XML y se utilizan otros formatos en situaciones muy especiales.

Es importante tener en cuenta que AJAX no constituye una tecnología en sí misma, sino que es un término que engloba a un grupo de tecnologías que trabajan conjuntamente.

1.1.3 ¿POR QUÉ ES ÚTIL GWT?

La computación en la nube (del inglés Cloud computing) es un modelo muy extendido en el que la aplicación que utiliza un usuario se encuentra físicamente en Internet. Únicamente es necesario disponer de un navegador para acceder a la misma.

A este concepto hay que añadir el de Aplicaciones con interfaces gráficas complejas en Internet (del inglés, Rich Internet Applications: RIAs). Es decir, la aplicación tiene el aspecto y la funcionalidad de las clásicas aplicaciones que se instalan en un ordenador convencional, sólo que en este caso se "instalan" en Internet.

GMail o Google Maps son ejemplos clásicos de este tipo de aplicaciones.

Existen múltiples herramientas para desarrollar aplicaciones de este tipo: applets de Java, Adobe Flash, Microsoft Silverlight, PHP, ASP y un largo etcétera.

Si has programado alguna vez aplicaciones Web, sabrás que su creación es un proceso que requiere mucho esfuerzo y es propenso a errores.

A veces, los desarrolladores emplean el 90% de su tiempo estudiando las características de los navegadores, sobre todo el de su motor de *JavaScript*. Además, la creación, la reutilización y el mantenimiento de una gran cantidad de componentes AJAX y de código *JavaScript* son tareas complejas y delicadas.

Google Web Toolkit (GWT) facilita estas tareas al ofrecer a los desarrolladores la posibilidad de crear y mantener rápidamente aplicaciones *JavaScript* con interfaces complejas, pero de gran rendimiento, en el lenguaje de programación Java.

1.1.4 PRINCIPALES CARACTERÍSTICAS DE GWT

- ✓ Usa Java como lenguaje base para el desarrollo de la aplicación Web, si bien, luego, el compilador de GWT traduce a *JavaScript* el código del lado cliente y a Java bytecode el del lado servidor (al estilo Servlet de Java). Es muy importante esta característica, ya que el programador acostumbrado a programar en Java tarda poco tiempo en adquirir los conocimientos necesarios para desarrollar aplicaciones con GWT.
- ✓ Dispone de componentes gráficos dinámicos y reutilizables (Widget). Los programadores pueden usar estas clases prediseñadas para implementar componentes y/o comportamientos que, de otra manera, tendrían que crear, tales como botones, cuadros de texto, arrastrar y soltar o menús en árbol.
- ✓ Permite un comunicación sencilla con el servidor web. GWT admite un conjunto indefinido de protocolos de transferencia de información, como JSON y XML. Para ello, se usa el mecanismo de llamada a procedimiento remoto (del inglés RPC: Remote Procedure Call) de GWT que permite el establecimiento de comunicaciones Java de una forma sencilla y eficaz.
- ✓ Al igual que ocurre con el mecanismo de invocación de métodos remotos (del inglés RMI: Remote Method Invocation) tradicional de Java, únicamente hay que crear una interfaz que especifique los métodos remotos que se quieran ejecutar. Después GWT se encarga de hacer todo por nosotros.
- ✓ Acepta el depurado del código Java. Mediante un depurador (debugger) incluido en GWT.
- ✓ Habilita el control de diferentes características del navegador por parte del desarrollador para mostrar la aplicación con determinadas características o aspectos.
- ✓ Se integra con JUnit. JUnit es un software de Java que permite realizar la ejecución de clases Java de manera controlada, para evaluar si el funcionamiento de cada uno de los métodos de la clase funciona tal y como se esperaba en el desarrollo y así se puede realizar una depuración automatizada del código fuente escrito en Java.

- ✓ Ofrece la posibilidad de Internacionalizar las aplicaciones Web. Es posible desarrollar aplicaciones que muestren información en diferentes idiomas en función de la procedencia del visitante.
- ✓ Se puede mezclar código escrito en *JavaScript* dentro del código Java creado para GWT. Para ello, se puede utilizar la Interfaz Nativa *JavaScript* (JSNI son las siglas en inglés).
- ✓ Tiene soporte para las API´s de Google (inicialmente, soporte para Google Gears, ya abandonado por Google).
- ✓ GWT es de código abierto.
- ✓ Permite el desarrollo de aplicaciones orientado a objetos (POO).
- ✓ Los errores comunes en *JavaScript* son controlados en tiempo de compilación, como la discrepancia de tipos de datos, simplificando mucho el desarrollo de funciones de *JavaScript* complejas.
- ✓ El código *JavaScript* creado puede ser ofuscado para optimizar el rendimiento y evitar que otros usuarios usen el código fuente.
- ✓ Se puede encontrar un amplio conjunto de bibliotecas desarrolladas por Google y terceros que mejoran las funcionalidades de GWT.

1.1.5 ALGUNAS DESVENTAJAS DE GWT

- ✓ Las aplicaciones desarrolladas con GWT no son indexables por los buscadores de Internet debido a que sus contenidos se cargan dinámicamente en el navegador. Existen soluciones que generan la información necesaria que necesitan los buscadores, pero son difíciles de usar y requieren bastante dedicación por parte del programador.
- ✓ Las aplicaciones GWT pueden no funcionar en un navegador muy antiguo. Este tipo de aplicaciones basan su funcionalidad en descargar en el lado cliente (navegador) un fichero *JavaScript*. Si el navegador es muy antiguo, la aplicación GWT no arrancará.
- ✓ Las aplicaciones GWT no suelen funcionar en los teléfonos móviles (salvo Android) debido a las limitaciones del motor *JavaScript* de los navegadores de los móviles, si bien muchas veces es posible instalar otros navegadores que sí cargan bien las aplicaciones. En el caso de móviles de tipo Android (de Google), sí suelen cargar sin problemas las aplicaciones.
- ✓ GWT no evita los problemas de seguridad. Como ocurre en cualquier plataforma de desarrollo, existen debilidades en el desarrollo que puede aprovechar un hacker para atacar nuestra aplicación Web. No obstante, las últimas versiones de

GWT ya incluyen muchas mejoras que evitan este tipo de errores y mejoran la seguridad.

1.1.6 COMPILADOR DE GWT A *JAVASCRIPT*

El componente más importante de GWT es el compilador Java-a-*JavaScript*. Este código *JavaScript* será el que se ejecute en el navegador del usuario.

Este compilador es el encargado de transformar el código Java escrito en Eclipse y produce distintas versiones en *JavaScript* equivalentes que se pueden ejecutar en todos los navegadores soportados por GWT: en el momento de la escritura de este curso, todas las versiones de Safari y Firefox, Opera (al menos hasta las versiones 9.x), y las versiones de 6 a 9 de Internet Explorer, Google Chrome, etcétera.

En realidad, el número de versiones generadas por el compilador de GWT a código *JavaScript* puede ser mucho mayor si la aplicación utiliza internacionalización, es decir, la aplicación se traduce automáticamente en función del idioma del navegador del usuario. En la Unidad 7 estudiaremos sus posibilidades.

Es posible minimizar el tamaño de código *JavaScript* para que la descarga al navegador sea más rápida. También se puede separar en bloques permitiendo que el navegador del usuario se descargue el código *JavaScript* necesario en trozos más pequeños. En la Unidad 8 se muestra más información sobre esto. Además, el compilador GWT hace varias optimizaciones de código durante la compilación, con objeto de producir código *JavaScript* de alta calidad, muchas veces superando el código desarrollado por programadores experimentados.

> *Normalmente, el código JavaScript está oculto, pero es posible pedir una compilación "Limpia" ("Pretty") de la salida para entender mejor lo que el compilador hace.*

Las optimizaciones aplicadas más importantes son:

- ✓ Eliminación de código sin uso (*Dead Code*): se trata del código que no se usa en la aplicación y no se incluye en el fichero *JavaScript* creado por GWT. Por ejemplo, si desarrollas una clase con diez métodos, pero sólo usas un par de ellos, el compilador GWT no genera el código para el resto de ellos. Además, si creas una clase a partir de otra (esto en POO se llama herencia) con varias docenas de métodos, el código de salida se genera solamente para los métodos realmente necesarios.

- ✓ Cálculo automático de constantes (*Contant Folding*): si un valor de una expresión puede ser conocido en tiempo de compilación, la expresión se calcula de antemano, y el resultado se utiliza directamente. Por ejemplo, si escribes el código Window.alert("Hola" + "Mundo"); se generará el código *JavaScript* $wnd.alert("Hola Mundo");. Ten en cuenta que este código se ejecuta un poco más rápido ya que la concatenación de la cadena ya está hecha.
- ✓ Propagación de copias (*Copy Propagation*): se trata de una extensión del cálculo automático de constantes, que permite calcular el valor de una variable si se puede conocer de antemano en tiempo de compilación.

Por ejemplo, el código Java int a = 10; int b = a * a, será compilado como si hubiéramos escrito int b = 100.

- ✓ Eliminación de cadenas repetidas (String Interning): evita la creación de cadenas repetidas por el programador.
- ✓ Inclusión de código (Code Inlining): en los casos en que el código de un método es corto y sencillo, GWT sustituye el código de este método en el sitio donde aparece la llamada original.

Todas estas optimizaciones significan que el código *JavaScript* final será, en general, muy eficiente. Como inconveniente, GWT no hace compilaciones parciales. GWT estudia la totalidad del código fuente y realiza una compilación monolítica para maximizar el número de posibles optimizaciones. Esta característica de GWT hace que se pierdan ventajas como la reutilización de los módulos previamente compilados. Se trata de una característica de diseño del equipo de desarrollo de Google donde se buscaba un mayor rendimiento en las aplicaciones creadas con GWT.

1.1.7 SOBRE LA LIBRERÍA GWT JRE DE EMULACIÓN

Mientras que en una aplicación normal de Java se pueden utilizar todas las clases y métodos disponibles, por la forma de funcionar del compilador GWT, éste requiere tener acceso al código fuente real de cualquier clase que se desee usar.

Esta librería se usa para compilar el código Java que escribe el programador a JavaScript. Por eso, es necesario disponer del código fuente de las clases Java para que se lleve a cabo una traducción correcta y coherente.

Esta exigencia se extiende a la JRE (*Java Runtime Environment*), que es un conjunto de utilidades que permite la ejecución de programas Java. Por esto, GWT proporciona una

implementación especial y parcial que se denomina “Biblioteca de emulación JRE” (*JRE Emulation Library*).

Esta librería sólo contiene cuatro paquetes:

- ✓ java.io (¡muy restringida!)
- ✓ java.lang
- ✓ java.sql (bastante limitada)
- ✓ java.util

1.1.7.1 java.io

Este paquete proporciona la funcionalidad del sistema de flujos de datos, de serialización y del sistema de archivos.

La razón de la limitación de este paquete es simple: el código de GWT compilado a *JavaScript* se ejecuta en un “apartado” del navegador que no puede acceder a los archivos locales o impresoras. Esto podría cambiar un poco con algunas características de HTML 5, si bien, por ahora, no hay nada implementado.

1.1.7.2 java.lang

Proporciona las clases que son fundamentales para el diseño de aplicaciones en lenguaje de programación Java. Incluye las excepciones, las clases y los métodos de utilidad general y las interfaces de algunos de éstos.

EXCEPCIONES	`IndexOutOfBoundsException`
	`NegativeArraySizeException`
	`NullPointerException`
	`NumberFormatException`
	`RuntimeException`
	`StringIndexOutOfBoundsException`
	`Throwable`
	`UnsupportedOperationException`
	`ArithmeticException`
		`ArrayIndexOutOfBoundsException`
		`ArrayStoreException`
		`AssertionError`
		`ClassCastException`
		`Error`
		`Exception`
		`IllegalArgumentException`
		`IllegalStateException`

CLASES	`Character`
	`Boolean`
	`Byte`
	`Double`
	`Float`
	`Class`
	`Object`
	`Short`
		`String`
		`StringBuffer`
		`StringBuilder`
		`Integer`
		`Long`
		`Number`

UTILIDADES	`Maths`	`System`

INTERFACES	`Appendable`
	`CharSecuence`
	`Cloneable`
		`Comparable`
		`Iterable`
		`Runnable`

1.1.7.3 java.sql

Este paquete proporciona la API para acceder y procesar los datos almacenados en una fuente de datos (normalmente una base de datos).

Se incluyen únicamente tres clases muy útiles para procesar variables de tipo fecha/tiempo. Desde un punto de vista de seguridad, no es recomendable ni conveniente conectarnos directamente a la base de datos SQL de un usuario que nos visita.

Clases	Date Time TimeStamp

1.1.7.4 java.util

Proporciona varias colecciones de herramientas útiles en el desarrollo de aplicaciones Java.

EXCEPCIONES	ConcurrentModificationException EmptyStackException	MissingResourceException NoSuchElementException TooManyListenersException
CLASES	AbstractCollection AbstractHashMap AbstractList AbstractMapEntry AbstractMap AbstractQueue EventObject HashMap HashSet IdentityHashMap LinkedHashMap LinkedHashSet AbstractSequentialList AbstractSet	ArrayList Arrays Collections Date EnumMap EnumSet LinkedList MapEntrylmpl PriorityQueue Stack TreeMap TreeSet Vector
INTERFACES	Collection Comparator Enumeration EventListener Iterator Listlterator List	Map Queue RandomAccess Set SortedMap SortedSet

GWT dispone también de ciertos paquetes que proporcionan otra funcionalidad adicional, que los programadores de Java dan por supuesto:

- ✓ `com.google.gwt.il8n.client.DateTimeFormat` y `com.google.gwt.il8n.client.NumberFormat` proporcionan las funciones de formato.
- ✓ `com.google.gwt.core.client.Duration` se puede utilizar para la sincronización.
- ✓ `com.google.gwt.core.client.Random` se usa para generar procesos aleatorios y proporciona un sustituto para java.util.Random.
- ✓ `com.google.gwt.user.client.Timer` se puede utilizar en lugar de java.util.Timer para usar temporizadores en nuestras aplicaciones de cliente.

Como consejo general, antes de usar una clase o excepción de Java muy específica en el lado cliente, hay que comprobar si es compatible con el compilador Java-a-*JavaScript* de GWT. De todas formas, al compilar el proyecto, Eclipse informará de este tipo de error.