



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



TFG del Grado en Ingeniería Informática
título del TFG



Presentado por Nombre del alumno
en Universidad de Burgos — 7 de mayo de 2017
Tutor: nombre tutor



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería en Informática



D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Nombre del alumno, con DNI dni, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 7 de mayo de 2017

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android . . .

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
Objetivos del proyecto	2
Conceptos teóricos	3
3.1. Secciones	3
3.2. Referencias	3
3.3. Imágenes	3
3.4. Listas de items	4
3.5. Tablas	5
Técnicas y herramientas	6
4.1. GWT:	6
4.2. JSweet:	7
4.3. DukeScript:	7
4.4. Vaadin:	7
Aspectos relevantes del desarrollo del proyecto	8
5.1. Parser JavaCC	8
5.2. Algoritmos	10
5.3. Internacionalización	10
Trabajos relacionados	11

<i>ÍNDICE GENERAL</i>	IV
Conclusiones y Líneas de trabajo futuras	12
Bibliografía	13

Índice de figuras

3.1. Autómata para una expresión vacía	4
5.2. Arquitectura RPC simple.	9
5.3. Diagrama sobre la comunicación entre el cliente y el servidor. . . .	10

Índice de tablas

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto .	5
---	---

Introducción

Descripción del contenido del trabajo y del estructura de la memoria y del resto de materiales entregados.

Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuales son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir y los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto.

Conceptos teóricos

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de \LaTeX ¹.

3.1. Secciones

Las secciones se incluyen con el comando `section`.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.2. Referencias

Las referencias se incluyen en el texto usando `cite` [2]. Para citar webs, artículos o libros [1].

3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de \LaTeX , pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:

¹Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz



Figura 3.1: Autómata para una expresión vacía

3.4. Listas de ítems

Existen tres posibilidades:

- primer ítem.
- segundo ítem.

1. primer ítem.
2. segundo ítem.

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikTeX					X
TeXMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Primer ítem más información sobre el primer ítem.

Segundo ítem más información sobre el segundo ítem.

■

3.5. Tablas

Igualmente se pueden usar los comandos específicos de \LaTeX o bien usar alguno de los comandos de la plantilla.

Técnicas y herramientas

Dentro de las diferentes herramientas que utilizaré para la realización de este trabajo, la más esencial es aquella con la cual, realizaré la conversión a JavaScript. Es por ello esencial hacer una buena elección comparando y analizando la diferentes posibilidades a elegir.

Como principales herramientas para la conversión de Java a JavaScript he podido encontrar GWT (Google Web Toolkit), JSweet y DukeScript aunque también hay otras de las que hablaré más adelante. Comenzaré por GWT.

4.1. GWT:

Google Web Toolkit es un framework ampliamente conocido por los desarrolladores web, entre otras cosas, gracias a ser de código abierto, de su gran utilidad y calidad además de ser completamente gratuito². Contiene una SDK que proporciona un conjunto de APIs de Java que permiten el desarrollo de aplicaciones AJAX escritas en Java. Posteriormente compila el código en JavaScript ya optimizado dando robustez a la aplicación web.

Básicamente, permite a los desarrolladores compilar código JAVA en archivos JavaScript ya optimizados de forma autónoma, proporcionando así todas las ventajas de las aplicaciones escritas en este último lenguaje.

GWT permite compartir código escrito en Java en la parte del servidor con código JavaScript en la parte del cliente lo que nos lleva pensar que la aplicación resultante será fiel a la idea inicial del Thoth.

²<http://www.gwtproject.org/>

4.2. JSweet:

JSweet³ es básicamente un «transpiler» es decir un compilador que traduce un código en un lenguaje a otro lenguaje. Al igual que GWT esta orientado a objetos, que proporciona una programación segura gracias a que usa un sistema de «tipado» Java.

La diferencia fundamental con GWT es que al ser un «transpiler» hace una traducción directa entre Java y JavaScript posicionando el código a un lado o al otro del Cliente y el servidor. Esto, claramente, tiene sus ventajas y sus inconvenientes dependiendo del uso que se le quiera dar.

4.3. DukeScript:

Se define así mismo como una tecnología para la creación de aplicaciones Java «multiplataforma» que internamente hacen uso de tecnologías HTML5 y JavaScript para el renderizado.⁴ Al igual que en los casos anteriores «sólo» se necesita desarrollar la aplicación en Java para después transformarla. Y digo «sólo» porque eso es en la teoría ya que como hemos podido ver, y en parte es lógico, la traducción no suele requerir, por lo menos, realizar ajustes del lenguaje para un buen funcionamiento.

DukeScript se centra sobre todo en el desarrollo de aplicaciones «multiplataforma» llevadas a cabo en Java, más que en el paso de Java a JavaScript. Da la posibilidad de que alguien con conocimientos, digamoslo así, en Java pueda llevar a cabo un proyecto en lenguajes pensados para aplicaciones móviles o web. Esto no quita que se puedan realizar aplicaciones de escritorio con JavaScript.

4.4. Vaadin:

Vaadin es un «framework» de Java de código abierto, para crear aplicaciones web ⁵. Se programa en Java o cualquier otro lenguaje de JVM.

³<http://http://www.jsweet.org/>

⁴<https://dukescript.com/>

⁵<https://vaadin.com/home>

Aspectos relevantes del desarrollo del proyecto

Lo más relevante de este proyecto es el transformar la última versión de Thoth a una aplicación web hecha por medio de GWT lo que supone comprender el funcionamiento interno del Thoth original, «desmigajando» para poder adaptarlo a las condiciones de un proyecto hecho con GWT. Estas condiciones limitan un poco la aplicación y nos obligan a reformar partes que antes eran más sencillas.

Es el caso, por ejemplo del núcleo de la gramática de Thoth que no puede ser llevado a un proyecto en GWT tal cual y necesita ser adaptado para que funcione como en la aplicación original.

5.1. Parser JavaCC

GWT esta compuesto por una parte cliente que es la que será traducida a JavaScript y otra servidor escrita en Java. Lo ideal en una aplicación de este estilo sería incluir todo en el lado del cliente y olvidarse del resto, traduciendo todo a JavaScript para una ejecución fluida en la web. El caso es, que como no es fácil traducir de un lenguaje a otro, hay partes del Thoth antiguo que no pueden ir en el lado del cliente y por consiguiente traducirse a dicho lenguaje. Una de estas partes es el parser JavaCC.

Originalmente en el núcleo de Thoth dentro del directorio «grammar» colgaba un sub-directorio llamado «parserjavacc» en el cual se encontraba el parser para la gramática. Es la parte más importante dentro de la gramática ya que permite reconocer si esta está bien construida o no. Ahora no procede explicar que es, pero si en que ha cambiado esta versión con respecto a la anterior. Para poder incluir esa parte en el proyecto decidimos hacerlo en el lado del servidor, ya que al no hacerse la traducción no habría ningún problema.

Ahora bien lo más complicado de todo esto es la comunicación entre esas dos partes.

Comunicación cliente-servidor

Una vez alojado en el servidor el parser habrá que establecer una comunicación entre la parte del cliente en la que habrá un método que llame a la funcionalidad que se desee del servidor.

Para hacerlo, GWT utiliza la técnica denominada por sus siglas en inglés RPC o llamada de procedimiento remoto (Remote Procedure Call) que tiene una estructura similar a esta:

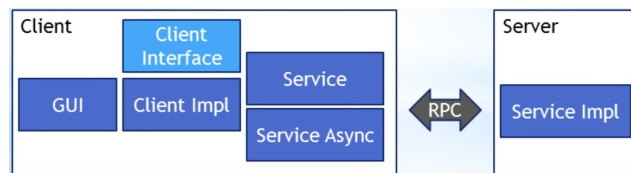


Figura 5.2: Arquitectura RPC simple.

Del lado del cliente tenemos, en este caso, una interfaz gráfica de usuario (GUI), una clase dedicada a hacer las llamadas RPC (ClientImpl), dos interfaces que definen los métodos (Service y ServiceAsync), una clase con los métodos en el lado del servidor (ServiceImpl) y por último una interfaz, que no es esencialmente necesaria (ClientInterface).

En términos generales, para utilizar el parser de la gramática, que es quizá la clave de todo, hay que llamar al servidor y establecer una comunicación. Es ahí donde utilizamos la clase «GrammarServiceClientImp» que hace las llamadas a las interfaces necesarias hasta llegar al servidor. Esta clase es un nexo de unión entre la parte del núcleo, que ese encuentra en el servidor y la parte del cliente que contiene lo visual y las acciones de los botones entre otros. La comunicación queda representada en este diagrama.

Dentro de la clase «GrammarServiceClientImp» he creado dos constructores, uno al que le paso una url y otro una url y además una gramática para que construya, valga la redundancia, una nueva interfaz gráfica de usuario con dicha gramática. La url es simplemente la localización del «servlet» para la comunicación.

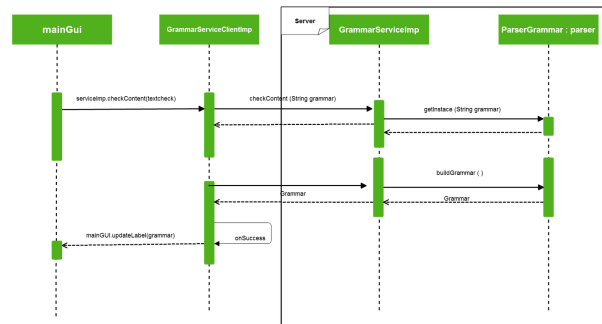


Figura 5.3: Diagrama sobre la comunicación entre el cliente y el servidor.

5.2. Algoritmos

5.3. Internacionalización

La aplicación cuenta con la funcionalidad de la internacionalización. Dentro del menú se pueden elegir entre varios idiomas a los que se traducirán los diferentes elementos. Los idiomas en los que está disponible la versión web de Thoth son: Alemán, castellano o español, francés y por supuesto inglés. Consideramos que esta funcionalidad es muy importante para poder llegar a diferentes países en el caso de que fuera necesario.

La internacionalización de la aplicación es un poco diferente a la utilizada en la versión de escritorio de Thoth. En primer lugar es necesario incluir una interfaz con los métodos para la internacionalización y los mensajes «por defecto» asociados a cada uno. Cada vez que queramos hacer uso de esos mensajes hay que hacer una llamada al método de la interfaz. Esa interfaz se encuentra en el directorio «client.gui.utils» donde se encuentran también los ficheros «properties» asociados, donde se encuentran las diferentes traducciones según el mensaje. Estos mensajes son los mismos que los utilizados para la internacionalización de Thoth V2.

Para poder realizar el cambio de idioma es necesario hacer uso de las propiedades las clases «xml» y «html» de GWT, en concreto «locale» que es la que especificará la localidad, que determina el idioma. Por ello cada vez que elegimos un idioma, la aplicación se redirige a una nueva «URL» (llevando a cabo una nueva compilación) con el atributo «locale=» seguido de las siglas del idioma al que se quiere traducir.

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final grado no parece obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

- [1] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [2] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].