

ANTEPROYECTO DEL TRABAJO DE FIN DE GRADO

INFORMACIÓN GENERAL

Alumno/a	Cheng Deng				
Titulación:	Ingeniería del Software				
Tutor/es:	Javier Troya Castilla				
Título	Generación automática de interfaces web Angular y conjuntos de pruebas a partir de especificaciones web textuales y sencillas.				
Subtítulo <i>(solo si en grupo)</i>					
Título en inglés	Automatic generation of Angular web interfaces and test sets from lightweight textual web specifications.				
Subtítulo en inglés <i>(solo si en grupo)</i>					
Trabajo en grupo:	Sí	<input type="checkbox"/>	No	X	
Otros integrantes del grupo:					

INTRODUCCIÓN

Contextualización del problema a resolver. Describir claramente de dónde surge la necesidad de este TFG y el dominio de aplicación. En caso de que el TFG se base en trabajos previos, debe aclararse cuáles son las aportaciones del TFG.

Contextualización del problema

El paradigma de *Ingeniería del Software Dirigida por Modelos (ISDM)* [1] proporciona fundamentos y herramientas para seguir una metodología de desarrollo software centrada en los modelos. A la hora de desarrollar aplicaciones software, aplicar ISDM significa reducir las tareas de escritura de código y, en su lugar, emplear modelos en el desarrollo.

A pesar de que ISDM proporciona varios fundamentos a nivel teórico, la aplicación de esta metodología en la industria no está muy consolidada. Uno de los motivos es la inmadurez de las herramientas para aplicar ISDM, debido en parte a la heterogeneidad de los sistemas software a desarrollar. De hecho, es común que se empleen lenguajes específicos de dominio (DSL del inglés *domain-specific language*) [2] para especificar soluciones concretas a problemas específicos.

Por otro lado, la proliferación de internet desde hace varias décadas lleva dando lugar, desde hace años, a un desarrollo de software basado principalmente en la nube y en aplicaciones web. De hecho, cada vez existen más *frameworks* para el desarrollo de páginas y aplicaciones web. Uno de estos *frameworks* es *Angular* [3], una herramienta Javascript potente y muy adecuada para el desarrollo de aplicaciones *frontend* modernas, de complejidad media o elevada. El *framework* Angular ofrece una base para el desarrollo de aplicaciones robustas, escalables y optimizadas, que promueve además las mejores prácticas y un estilo de codificación homogéneo y de gran modularidad.

El problema que ocurre con Angular, al igual que con otras herramientas de desarrollo web, es que requiere un aprendizaje inicial, sobre todo si se comienza a desarrollar una página web desde cero. Además, no es trivial garantizar que la página web que desarrollamos cumple con su especificación. Esto significa que es importante realizar pruebas sobre las páginas web que se desarrollan. La metodología conocida como *Behavior-Driven Development* promueve el desarrollo de las pruebas a medida que se va desarrollando el software. *Jasmine* [4] es una suite de pruebas que sigue esta metodología, y está especialmente indicada para definir pruebas unitarias y de integración. Las pruebas definidas en *Jasmine* se pueden ejecutar en *Karma* [5], un ejecutor de pruebas desarrollado directamente por el equipo de *Angular*, lo que lo convierte en el ejecutor más indicado para las pruebas

sobre código *Angular*. El problema, al igual que ocurre con *Angular*, es que no es trivial definir pruebas en *Jasmine* por parte de usuarios no expertos.

Para abordar el problema del desarrollo web y sus pruebas por usuarios no expertos, podemos aprovechar las tareas y herramientas de automatización que proporciona la metodología ISDM. Así, haciendo uso de un DSL, sería ideal que un usuario pudiera definir una especificación de su aplicación web con un lenguaje textual sencillo, y que la aplicación web, así como sus pruebas, se generasen de manera automática.

OBJETIVOS

Descripción detallada de en qué consistirá el TFG. En caso de que el objeto principal del TFG sea el desarrollo de software, además de los objetivos generales deben describirse sus funcionalidades a alto nivel.

Objetivos

Este TFG se centra en el desarrollo de un entorno para la generación automática de páginas web en *Angular* (la parte del *frontend*), así como de un conjunto de pruebas asociado especificadas en *Jasmine* que podrán ejecutarse en *Karma*. Esta generación automática se realizará a partir de una especificación textual sencilla, y cercana al lenguaje humano, de la página web.

Para poder desarrollar el entorno, primero se definirá una gramática del lenguaje textual con el que los usuarios podrán especificar páginas web de manera sencilla e intuitiva. Para ello, se utilizará la herramienta *Xtext*, integrada en *Eclipse Modeling Framework* [6]. De acuerdo con el paradigma ISDM que se aplicará, la gramática definida será equivalente a un metamodelo, por lo que las páginas especificadas con este lenguaje serán modelos conformes a dicho metamodelo.

El segundo paso consiste en desarrollar una transformación modelo a texto que tome como entrada las especificaciones textuales de las páginas web y genere el código *Angular*, así como sus pruebas en *Jasmine*. Un proyecto *Angular* se compone de varios archivos, por lo que esta transformación deberá generar de manera automática varios archivos. No se persigue que la página web generada en *Angular* sea completa, pero sí que sea una versión inicial que se pueda completar a posteriori de manera manual. El conjunto de pruebas *Jasmine* generado deberá servir para probar dicha versión inicial de la web con el ejecutor de pruebas *Karma*. Para desarrollar la transformación modelo a texto se utilizará *Xtend*, también integrado en *Eclipse Modeling Framework*.

Tareas a desarrollar para cumplir los objetivos.

T1. Estudiar los frameworks *Xtext* y *Xtend*, así como *Angular*, *Jasmine* y *Karma*. Dichos frameworks no se han visto durante la carrera. Además, *Xtext* y *Xtend* son específicos del paradigma ISDM, y *Jasmine* es específico de la metodología *Behavior-Driven Development*.

T2. Estudiar qué tipo de pruebas *Jasmine* suelen aplicarse a páginas web desarrolladas con *Angular*.

T3. Definir un lenguaje específico de dominio con *Xtext* para poder especificar páginas web de manera sencilla. Este lenguaje se centrará en la parte del *frontend*.

T4. Desarrollar una transformación modelo a texto que tome como entrada especificaciones de páginas web realizadas con el lenguaje de la tarea T3 y genere una versión inicial de una página web en *Angular*. La transformación también generará un conjunto de pruebas especificadas en *Jasmine*.

ENTREGABLES

Listado de resultados que generará el TFG (aplicaciones, estudios, manuales, etc.)

E1. Lenguaje específico de dominio para definir páginas web de manera sencilla

E2. Transformación modelo a texto que tome como entrada especificaciones de páginas web realizadas con el lenguaje del resultado E1 y genere una versión inicial de una página web en *Angular*. La transformación también generará un conjunto de pruebas especificadas en *Jasmine*.

E3. Manual de usuario del entorno desarrollado, donde se detalle cómo definir páginas web de manera sencilla para obtener código *Angular* de las mismas, así como su conjunto de pruebas de manera automática. Deberá explicarse cómo ejecutar las pruebas generadas con el ejecutor *Karma*.

E4. Memoria del TFG, donde se detalle el estudio realizado, la metodología empleada, las fases del desarrollo y los resultados obtenidos.

MÉTODOS Y FASES DE TRABAJO

METODOLOGÍA:

Descripción de la metodología empleada en el desarrollo del TFG. Especificar cómo se va a desarrollar. Concretar si se trata de alguna metodología existente y, en caso contrario, describir y justificar adecuadamente los métodos que se aplicarán.

El proyecto seguirá una metodología iterativa incremental. Debido a que se definirá un lenguaje completamente nuevo, se empezará por una versión muy simple de su gramática, para ir aumentando su complejidad en iteraciones sucesivas. Respecto a la transformación modelo a texto, se comenzará transformando aspectos sencillos del lenguaje, para ir incrementando en iteraciones posteriores hasta realizar la transformación completa, incluida la generación de pruebas.

Al final de cada iteración, el autor del proyecto se reunirá con el tutor para valorar el resultado obtenido y establecer los objetivos de la siguiente iteración. Como punto adicional y por la tutorización, se aceptarán consultas entre autor y tutor en cualquier momento de la iteración.

FASES DE TRABAJO:

Enumeración y breve descripción de las fases de trabajo en las que consistirá el TFG.

F1. Estudio de las herramientas *Xtext*, *Xtend*, *Angular*, *Jasmine* y *Karma*.

F2. Estudio de las pruebas *Jasmine* más comunes para aplicaciones web desarrolladas en *Angular*.

F3. Diseño y especificación de la gramática de un lenguaje para la definición de páginas web de manera textual y sencilla.

- Entregable E1.

F4. Desarrollo de una transformación modelo a texto que tome como entrada especificaciones de páginas web realizadas con el lenguaje de la fase F3 y genere una versión inicial de una página web en *Angular*. La transformación también generará un conjunto de pruebas especificadas en *Jasmine*.

- Entregable E2.

F5. Escritura del manual de usuario, de la memoria y del informe final. Conclusiones.

- Entregable E3.
- Entregable E4.

TEMPORIZACIÓN:

La siguiente tabla deberá contener una fila por cada una de las fases enumeradas en la sección anterior. En caso de tratarse de un trabajo en grupo, se añadirá una columna HORAS por cada miembro del equipo. Debe especificarse claramente el número de horas dedicado por cada alumno/a y la suma de horas individual deberá ser también de 296.

FASE	HORAS
Estudio de las herramientas <i>Xtext</i> , <i>Xtend</i> , <i>Angular</i> , <i>Jasmine</i> y <i>Karma</i> .	40
Estudio de las pruebas <i>Jasmine</i> más comunes para aplicaciones web desarrolladas en <i>Angular</i>	15
Diseño y especificación de una gramática para la definición de páginas web de manera textual y sencilla	80
Desarrollo de una transformación modelo a texto	126
Escritura del manual de usuario, de la memoria y del informe final. Conclusiones	35
	296



ENTORNO TECNOLÓGICO

TECNOLOGÍAS EMPLEADAS:

Enumeración de las tecnologías utilizadas (lenguajes de programación, frameworks, sistemas gestores de bases de datos, etc.) en el desarrollo del TFG.

Xtext

Xtend

Eclipse Modeling Framework (EMF)

Angular

Typescript

Jasmine

Karma

RECURSOS SOFTWARE Y HARDWARE:

Listado de dispositivos (placas de desarrollo, microcontroladores, procesadores, sensores, robots, etc.) o software (IDE, editores, etc.) empleados en el desarrollo del TFG.

Ordenador personal

Eclipse IDE

Angular

REFERENCIAS

Listado de referencias (libros, páginas web, etc.)

- [1] S. Kent, "Model Driven Engineering". Proc. of Int. Conference on Integrated Formal Methods. 2002.
- [2] M. Fowler, "Domain-Specific Languages". Pearson Education. 2010.
- [3] Y. Fain, A. Moiseev, "Angular Development with Typescript". Simon and Schuster. 2018
- [4] Jasmine, Behavior-Driven JavaScript: <https://jasmine.github.io/>
- [5] Karma, JavaScript Test-Runner: <https://karma-runner.github.io/latest/index.html>
- [6] Xtext, Integration with EMF: https://www.eclipse.org/Xtext/documentation/308_emf_integration.html

Málaga, 2 de Marzo de 2022

Firma tutor/tutora:

Firma cotutor/a:

Firma tutor/a coordinador/a: