

Grado en Ingeniería Informática
Computación

Trabajo de Fin de Grado

**Transpilador de un lenguaje de modelado
personalizado de sistemas de simulación
dinámicos discretos a Python**

Autor

Baldwin David Rodríguez Ponce

2022

BORRADOR

Grado en Ingeniería Informática
Computación

Trabajo de Fin de Grado

**Transpilador de un lenguaje de modelado
personalizado de sistemas de simulación
dinámicos discretos a Python**

Autor

Baldwin David Rodríguez Ponce

Directora

María Begoña Losada Pereda

BORRADOR

Tabla de contenidos

1. Gestión del proyecto	1
1.1. Alcance	1
1.1.1. Objetivos	2
1.1.2. Requisitos	3
1.1.3. EDT	3
1.2. Metodología	5
1.2.1. Significado de las listas	5
1.2.2. Significado de etiquetas	5
1.2.3. Formato de las cartas	6
1.3. Tareas y estimación de dedicaciones	6
1.3.1. Descripción de tareas a realizar	6
1.3.2. Estimación de dedicación	9
1.4. Análisis de riesgos y viabilidad	10
1.4.1. Métricas	10
1.4.2. Tabla de riesgos	10
1.5. Sistema de información y comunicaciones	10
1.5.1. Sistema de Información	10
1.5.2. Comunicaciones	10
Bibliografía	11

1.CAPÍTULO

Gestión del proyecto

1.1. Alcance

El alcance de este proyecto incluye el trabajo necesario para diseñar, implementar y documentar un framework de modelado y desarrollo de simulación de sistemas dinámicos discretos basados en eventos. Dicho framework se dividirá en tres módulos principales:

- **Lenguaje:** Un lenguaje de modelado específico pensado para ser usado por usuarios que no tengan mucha experiencia en programación. Se hará uso de las herramientas de desarrollo de compiladores Flex y Bison para generar un transpilador que traduzca ficheros de este lenguaje a código Python. A través de él se plantea:
 - Permitir la rápida implementación de este tipo de modelos a través de grafos de sucesos.
 - Permitir que el programador del lenguaje se encargue sólo de realizar las implementaciones pertinentes al sistema de simulación que desee desarrollar:
 - Especificación de las variables globales, variables de entrada, contadores estadísticos y medidas de rendimiento propias del modelo.
 - Inclusión de eventos adicionales y sus acciones correspondientes.
 - Creación y eliminación de eventos en función de tiempo y condiciones lógicas.
 - Inclusión de código adicional escrito directamente en Python en caso de

Cambiar redacción

ser necesario.

- **Núcleo:** Una serie de módulos que implementarán un microframework de simulación de este tipo de sistemas en específico para Python, pensado para ser usado por programadores y para acotar la traducción del nuevo lenguaje. A través de él se plantea:
 - Permitir que la traducción del lenguaje incluya dentro del fichero generado las estructuras de datos, funciones y procedimientos que tienen en común todos los sistemas dinámicos discretos:
 - Generadores de datos aleatorios para distintos tipos de distribuciones.
 - Reloj y temporizador de simulación para ejecutar los eventos.
 - Estructura de datos para almacenar los sucesos según deben ocurrir en el tiempo.
 - Las respectivas implementaciones mínimas de los dos eventos que siempre formarán parte de todos los modelos: “Inicio” y “Fin”.
 - Generador de informes final que se ejecutará al finalizar la simulación y mostrará los resultados que se deseaban estudiar con ésta.
- **CLI:** Una interfaz de comandos por terminal que se usará para gestionar, configurar y ejecutar los proyectos desarrollados con este framework.

1.1.1. Objetivos

Objetivo general

Generar un lenguaje de modelado de simulación de sistemas dinámicos discretos basados en eventos junto con un transpilador que lo traduzca a Python, un microframework para acotar la traducción y un CLI para gestionar proyectos desarrollados con este producto.

Objetivos específicos

1. Diseñar un nuevo lenguaje de modelado y simulación de sistemas dinámicos discretos basados en eventos usando las herramientas de desarrollo de procesadores de lenguaje Flex y Bison.

2. Diseñar, implementar y verificar una serie de módulos y funcionalidades realizadas en Python con el fin de generar un microframework para simular estos mismos sistemas.
3. Implementar y verificar un transpilador que traduzca el lenguaje del producto a Python usando las características del objetivo anterior.
4. Diseñar, implementar y verificar una serie de operaciones accesibles desde una CLI de cara a ser usadas para la gestión, configuración y ejecución parametrizada de simulaciones realizadas con el producto.
5. Implementar distintas técnicas de análisis de salidas, experimentación y optimización de modelos dentro del núcleo del framework.
6. Desarrollar un manual de usuario que contendrá toda la documentación necesaria para hacer uso del framework.

1.1.2. Requisitos

El requisito base principal del proyecto consiste en cumplir con un tiempo de dedicación total máximo de 300 horas.

1.1.3. EDT

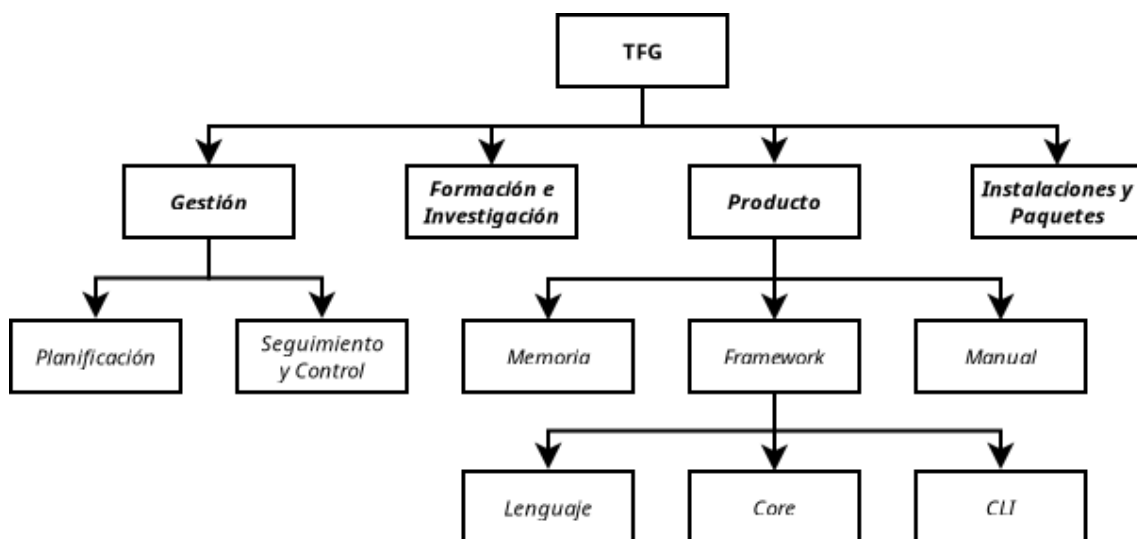


Figura 1.1: Esquema de Descomposición de Trabajo del Proyecto

Rama de Gestión

El paquete de trabajo **Gestión (Ge)** contendrá:

- **Planificación (Ge.P):** Este paquete de trabajo agrupará todas las tareas relacionadas a la realización de la planificación y preparación inicial del proyecto.
- **Seguimiento y Control (Ge.S):** Este paquete de trabajo agrupará todas las tareas necesarias para asegurar que el seguimiento del proyecto está realizando como se plantea en la planificación y, en caso de no ser así, controlar las consecuencias de las desviaciones emergentes.

Rama de Formación e Investigación

El paquete de trabajo **Formación e Investigación (FeI)** contendrá todas las tareas relacionadas con el aprendizaje de herramientas, búsqueda de referencias y recolección de información externa necesaria para el desarrollo del proyecto.

Rama de Producto (Pr)

El paquete de trabajo **Producto (Pr)** contendrá todas las tareas relacionadas al propio diseño, implementación y verificación de los entregables principales del proyecto. Podemos desglosarlo en tres paquetes más pequeños:

- **Memoria (Pr.Me):** Este paquete de trabajo agrupará todas las tareas necesarias para la realización de la memoria final del proyecto.
 - **Framework (Pr.Fr):** Este paquete de trabajo agrupará todas las tareas relacionadas al desarrollo del framework planteado en el alcance del proyecto. Podemos separarlo en sus tres partes principales:
 - **Lenguaje (Pr.Fr.Le):** Este paquete contendrá todas las tareas relacionadas al diseño y desarrollo del nuevo lenguaje de modelado.
 - **Core (Pr.Fr.Co):** Este paquete contendrá todas las tareas relacionadas al diseño y desarrollo del propio núcleo del framework.
 - **CLI (Pr.Fr.Cli):** Este paquete contendrá todas las tareas relacionadas al diseño y desarrollo de la interfaz de comandos proporcionada para facilitar la gestión de proyectos realizados con el a generar.
-

- **Manual (Pr.Ma):** Este paquete de trabajo agrupará todas las tareas necesarias para la realización del manual de usuario que se generará para facilitar el uso del producto a desarrollar.

Rama de Instalación y Paquetes

El paquete de trabajo **Instalación y Paquetes (IyP)** agrupará todas las tareas relacionadas a la preparación del entorno de desarrollo y la instalación de paquetes y dependencias para la realización del producto principal del proyecto.

1.2. Metodología

Explicar Kanban

Por último, la gestión del proyecto se realizará siguiendo la metodología de desarrollo ágil Kanban para permitir construir de manera iterativa las distintas funcionalidades del proyecto. Hacer uso del método Kanban para el desarrollo y gestión del proyecto.

1.2.1. Significado de las listas

- **Backlog:**
- **To Do:**
- **Doing:**
- **Testing:**
- **Done:**
- **Approved:**

1.2.2. Significado de etiquetas

- **Bug:**
 - **Bloqueado:**
 - **Pendiente:**
 - **No aceptada:**
-

1.2.3. Formato de las cartas

1.3. Tareas y estimación de dedicaciones

No se puede predecir porque usamos Kanban. Sólo conocemos las fechas finales de entrega.

1.3.1. Descripción de tareas a realizar

Gestión

Planificación

1. Definición de la metodología de gestión
2. Generación de la planificación inicial.
3. Automatización de herramientas de gestión con Trello.
4. Cambios y actualizaciones de la planificación

Seguimiento y Control

1. Recogida de información sobre el desarrollo del proyecto
2. Contraste de la información de seguimiento con los planes, identificación de desviaciones significativas y actuación ante riesgos emergentes.
3. Aseguramiento de las condiciones para el éxito del proyecto.
4. Reuniones de fin de iteración

Formación e Investigación

1. Investigación sobre Kanban
 2. Investigación teórica sobre simulación de sistemas
 3. Recolectar referencias de simulación de sistemas
 4. Recolectar referencias de compilación
-

5. Investigar sobre herramientas de generación de CLIs
6. Investigación sobre lenguajes de modelado y simulación
7. Investigación sobre antecedentes del proyecto

Producto

Memoria

1. Maquetación y diseño del informe
2. Definición de la estructura
3. Estructurar informe
4. Estructura de la introducción
5. Estructura del contexto
6. Redactar motivaciones
7. Redactar contexto
8. Redactar antecedentes
9. Redacción del resumen
10. Redacción del marco teórico
11. Redacción del desarrollo
12. Redacción de conclusiones
13. Redacción de anexos de la memoria

Producto

■ **Lenguaje:**

1. Generación de tokens básicos del lenguaje
 2. Diseño básico de la sintaxis del lenguaje
 3. Implementación del analizador léxico
 4. Implementación del analizador sintáctico
-

5. Diseño del léxico del lenguaje
6. Diseño de la sintaxis del lenguaje
7. Diseño de la semántica del lenguaje
8. Implementación del analizador léxico
9. Implementación del analizador sintáctico
10. Implementación del análisis semántico y la tabla de símbolos
11. Pruebas del lenguaje

■ **Núcleo:**

1. Diseño de los módulos del core
2. Implementación de los generadores de datos
3. Implementación del reloj, temporizador y lista de sucesos
4. Implementación del generador de informes
5. Implementación del gestor de datos de entrada
6. Diseño inicial del prototipo del framework
7. Implementación inicial del prototipo del framework
8. Pruebas del core

■ **CLI:**

1. Diseño del CLI
2. Implementación del gestor de proyectos
3. Implementación del lanzador de modelos traducidos
4. Pruebas del CLI

Manual

1. Maquetación y preámbulo del manual
 2. Definición de estructura y contenido
 3. Redacción de la introducción
-

4. Explicación de requisitos y dependencias
5. Redacción del proceso de instalación
6. Explicación de la creación y configuración de proyectos
7. resumen del lenguaje generado
8. Redacción de anexos del manual

Instalación y Paquetes

1. Preparación del entorno de trabajo
2. Instalación de paquetes para el desarrollo del lenguaje
3. Instalación de paquetes para el desarrollo del núcleo
4. Instalación de paquetes para el desarrollo del CLI

1.3.2. Estimación de dedicación

Paquete	Nombre	Estimación (Horas)
Ge.P	Gestión - Planificación	20
Ge.S	Gestión - Seguimiento	25
FeI	Formación e Investigación	50
Pr.Me	Proyecto - Memoria	50
Pr.Fr.Le	Proyecto - Framework - Lenguaje	48
Pr.Fr.Co	Proyecto - Framework - Core	52
Pr.Fr.Cli	Proyecto - Framework - CLI	25
Pr.Ma	Proyecto - Manual	15
IyP	Instalaciones y Paquetes	15

Tabla 1.1: Estimación de tiempos de dedicación por paquetes de trabajo

1.4. Análisis de riesgos y viabilidad

1.4.1. Métricas

1.4.2. Tabla de riesgos

1.5. Sistema de información y comunicaciones

1.5.1. Sistema de Información

Estructura

Aquí debemos explicar la estructura

TFG: Directorio principal del proyecto. Se divide en:

- **TFG.Proyecto:** Aquí residen los entregables del proyecto.
 - **Memoria:** Los ficheros relacionados a la memoria
 - **Framework:** Los ficheros relacionados al framework
 - **Lenguaje:** Los ficheros relacionados al lenguaje del framework
 - **Core:** Los ficheros relacionados al núcleo del framework
 - **CLI:** Los ficheros relacionados al CLI del framework
 - **Manual:** Los ficheros relacionados al manual
- **TFG.Gestión:** Aquí residen los ficheros relacionados a la gestión del proyecto
 - **Planificación:** Los ficheros relacionados a la planificación
 - **Seguimiento:** Los ficheros relacionados al seguimiento y control
 - **Actas:** Las actas de cada reunión realizada

1.5.2. Comunicaciones

- [Reuniones a través de Webex]
 - [Comunicaciones para responder dudas a través de email]
-

Bibliografía

- Buss, A. (1996). Modeling with event graphs. *Winter Simulation Conference*, 153-160.
<https://doi.org/10.1109/WSC.1996.873273>
- Maria, A. (1997). Introduction to modeling and simulation. *Proceedings of the 29th conference on Winter simulation - WSC '97*. <https://doi.org/10.1145/268437.268440>
- Perros, H. G. (2009). *Computer simulation techniques: the definitive introduction!*
- Banks, J. (2010). *Discrete-event system simulation*. Prentice Hall.
- Wainer, G. (2011). *Discrete-event modeling and simulation: theory and applications*. CRC Press.
- MATLAB. (2017a). Understanding Discrete Event Simulation, Part 1: What Is Discrete Event Simulation.
- MATLAB. (2017b). Understanding Discrete Event Simulation, Part 2: Why Use Discrete Event Simulation.
- MATLAB. (2017c). Understanding Discrete Event Simulation, Part 3: Leveraging Stochastic Processes.
- MATLAB. (2017d). Understanding Discrete Event Simulation, Part 4: Operations Research.
- MATLAB. (2017e). Understanding Discrete Event Simulation, Part 5: Communication Modeling.

