

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. Trello como herramienta de gestión | 5 |
| 1.3. Tareas y estimación de dedicaciones | 6 |
| 1.3.1. Descripción de tareas a realizar | 6 |
| 1.3.2. Estimación de dedicación | 10 |
| 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 | 11 |
| 1.5.1. Sistema de Información | 11 |
| 1.5.2. Comunicaciones | 11 |
| Bibliografía | 13 |

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

ser necesario.

Cambiar redacción

- **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:

Aquí te falta la posibilidad de poner las cosas variables también

- 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

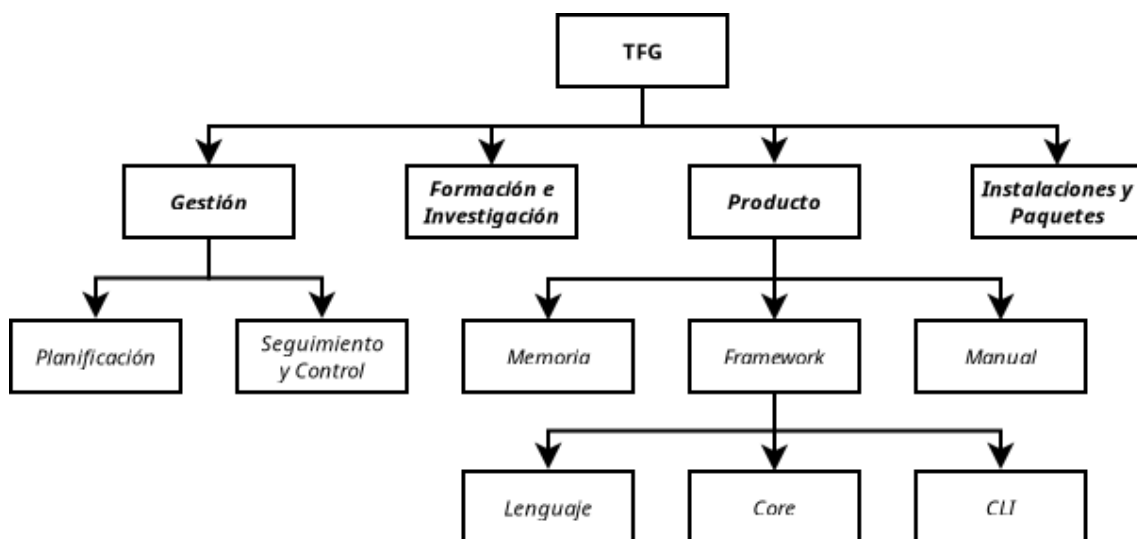


Figura 1.1: Esquema de Descomposición de Trabajo del Proyecto

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.

Desglosar más lo que es Kanban

1.2.1. Trello como herramienta de gestión

Significado de las listas

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

Significado de etiquetas

- **Bug:**

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

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. Preparación de documentos de cara a la próxima reunión.
4. Reuniones de fin de iteración

Formación e Investigación

1. Investigación y formación sobre la metodología Kanban.
-

2. Profundización sobre conceptos de simulación de sistemas.
3. Profundización sobre conceptos de compilación.
4. Investigación y formación sobre herramientas de desarrollo de CLIs.
5. Investigación y formación sobre generación de paquetes instalables para Python.
6. Investigación y formación sobre funciones de alto nivel y paquetes relacionados de Python.
7. Investigación sobre lenguajes de modelado y simulación.
8. Investigación sobre antecedentes del proyecto.
9. Exploración de alternativas al módulo de lenguaje.

Producto

Memoria

1. Diseño y preparación de la estructura
2. Redacción del resumen
3. Redacción de la introducción
4. Redactar contexto
5. Redacción de la gestión del proyecto
6. Redacción del marco teórico
7. Redacción del desarrollo
8. Redacción de conclusiones
9. Preparación e inclusión de referencias bibliográficas
10. Redacción de anexos
11. Validación y corrección de los índices
12. Maquetación y diseño de la memoria

Framework

■ Lenguaje:

1. Diseño del léxico del lenguaje.
2. Diseño de la sintaxis del lenguaje.
3. Diseño de la semántica del lenguaje.
4. Diseño de módulos de cara a la implementación del lenguaje.
5. Implementación del analizador léxico.
6. Implementación del analizador sintáctico.
7. Implementación de la tabla de símbolos.
8. Implementación del análisis semántico.
9. Documentación de los ficheros del lenguaje.
10. Pruebas y verificación del lenguaje.

■ **Núcleo:**

1. Diseño de los módulos y paquetes del núcleo.
 2. Implementación de la estructura de datos para almacenar eventos.
 3. Implementación del reloj y el proceso de temporización de sucesos.
 4. Implementaciones de los procesos de inicialización y finalización.
 5. Implementación del generador de informes final.
 6. Implementación de distintos generadores de datos aleatorios.
 7. Implementación de la inicialización de las variables globales, variables de entrada, contadores estadísticos y medidas de rendimiento para cualquier modelo.
 8. Implementación de la inclusión de nuevos tipos de eventos y sus funciones de ejecución correspondientes.
 9. Implementación de distintas estrategias de generación y eliminación de eventos dentro de la lista de sucesos.
 10. Implementación de la especificación de una función de parada definida por el usuario.
 11. Implementación de la funcionalidad de configuración de la simulación.
 12. Documentación de los ficheros del núcleo.
-

13. Pruebas y verificación del núcleo.

■ **CLI:**

1. Diseño de las funcionalidades de la CLI.
2. Implementación del generador de proyectos.
3. Implementación del gestor de configuraciones del proyecto.
4. Implementación del lanzador de modelos de simulación.
5. Documentación de los ficheros de la CLI.
6. Pruebas y verificación de la CLI.

Manual

1. Diseño y preparación de la estructura
2. Redacción de la introducción al producto
3. Especificación de dependencias y requisitos
4. Redacción del proceso de instalación
5. Redacción de la explicación del módulo del lenguaje
6. Redacción de la explicación del núcleo del framework
7. Redacción de la explicación de la CLI
8. Generación de ejemplos de uso
9. Preparación e inclusión de referencias bibliográficas
10. Redacción de anexos
11. Validación y corrección de los índices
12. Maquetación y diseño del manual

Instalación y Paquetes

1. Preparación de paquetes para LaTeX.
 2. Preparación del entorno de trabajo.
-

3. Instalación de paquetes para el desarrollo del lenguaje.
4. Instalación de paquetes para el desarrollo del núcleo.
5. Instalación de paquetes para el desarrollo de la CLI.

1.3.2. Estimación de dedicación

| Paquete | Nombre | Estimación (Horas) |
|-----------|---------------------------------|--------------------|
| Ge.P | Gestión - Planificación | 25 |
| Ge.S | Gestión - Seguimiento | 20 |
| 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: No entender la metodología de gestión y desarrollo (Kanban)
 - 2: Falta de tiempo
 - 3: ¿Pérdida de tiempo?
 - 4: Problemas de comunicación
 - 5: Malas propuestas de módulos del producto (Piensa en el lenguaje, por ejemplo).
-

1.5. Sistema de información y comunicaciones

1.5.1. Sistema de Información

Estructura

TFG: Directorio principal del proyecto. Se divide en:

Aquí debemos
explicar la
estructura

- **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.

