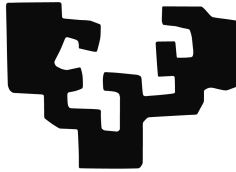


eman ta zabal zazu



Universidad Euskal Herriko
del País Vasco Unibertsitatea

Escuela Universitaria de Ingeniería Técnica Industrial de Bilbao

Trabajo Fin de Grado

Del Diseño a la Programación de Sistemas de Control para la
Automatización de Procesos Industriales Basados en PLC's

Autor: Helen Joana Paz Cortón

Grado: Ingeniería Informática de Gestión y Sistemas de Información

Septiembre 2015

Resumen

Actualmente las mayoría de sistemas de automatización industrial utilizan los Autómatas Programables (PLC - *Programmable Logic Controller*) como sistemas de control y el diseño de dicho sistema se realiza mediante el lenguaje de modelado Grafcet. SFCEdit es un editor de Grafcet que permite el diseño de estos sistemas de control y la exportación de los mismos en un formato XML.

Por otra parte tenemos la norma IEC 61131-3 que busca estandarizar los lenguajes de programación de los PLC de esta manera no se tendría que usar un lenguaje distinto por cada fabricante, y la organización PLCOpen rigiéndose por la norma ha creado un formato XML para cada lenguaje, en los cuales se basa el traductor.

Habitualmente el paso del diseño a la programación se realiza de forma manual. La automatización de este proceso es el objetivo general del proyecto realizado. En éste se ha desarrollado una herramienta que permite realizar la traducción del diseño gráfico al estándar IEC 61131-3 considerando las particularidades de algunos fabricantes de PLCs como Beckhoff (TwinCAT) y TSX Micro (PL7Pro). Además la herramienta realiza la traducción al formato XML de PLCOpen.

El traductor funciona recibiendo un fichero XML (exportado de SFCEdit) procesa su información y genera los ficheros en función de la compatibilidad que el usuario seleccione.

Índice general

Introducción	7
Objetivos del proyecto	8
Objetivos	8
Propósito	8
Ámbito	8
Contexto de Negocio	8
Definiciones, acrónimos, y abreviaturas	8
Alcance	9
Estructura de Descomposición del Trabajo (EDT)	10
Desglose de Tareas	10
Planificación Temporal	15
Herramientas	16
Herramientas de Gestión y Documentación	16
Herramientas de Desarrollo	17
Herramientas de Diseño y Programación de PLCs	18
Gestión de Riesgos	20
Evaluación Económica	21
Antecedentes	22
Situación Actual	22
Estudio de Diferentes Alternativas Existentes	22
Captura de Requisitos	23
Modelo de Casos de Uso	23
Jerarquía de Actores	23
Prototipos de Interfaces de Usuario	23
Descripción de la Arquitectura	25

Análisis y Diseño	28
Análisis	28
Diagrama de Clases	28
Diseño	34
Diagrama de Secuencias	34
Interfaz	34
Arquitectura	36
 Desarrollo	 37
 Verificación y Evaluación	 39
Pruebas del Software	39
Pruebas de Integración	39
Pruebas de Aceptación	41
Documentación de Pruebas	41
 Conclusiones y Trabajo Futuro	 43
 Bibliografía	 45
 ANEXO I.- Casos de Uso Extendidos	 47
Caso de Uso Leer Instrucciones	47
Descripción	47
Actores	47
Precondiciones	47
Requisitos No Funcionales	47
Flujo de eventos	47
Postcondiciones	47
Imágenes	48
Caso de Uso Iniciar Traductor	49
Descripción	49
Actores	49
Precondiciones	49
Requisitos No Funcionales	49
Flujo de eventos	49
Postcondiciones	50
Imágenes	50

Caso de Uso Iniciar Traductor	53
Descripción	53
Actores	53
Precondiciones	53
Requisitos No Funcionales	53
Flujo de eventos	53
Postcondiciones	54
Imágenes	54
ANEXO II.- Diagramas de Secuencia	55
ANEXO III.- Manual de usuario	63
Introducción	63
Requisitos del sistema	63
Funcionamiento de la aplicación	63

Índice de figuras

1.	Alcance - Estructura de Descomposición del Trabajo.	10
2.	Alcance - Planificación Temporal.	15
3.	Herramientas - Logo Dropbox	16
4.	Herramientas - Logo Git	16
5.	Herramientas - Logo Lyx	16
6.	Herramientas - Logo Microsoft Office	16
7.	Herramientas - Logo ObjectAid UML Explorer	17
8.	Herramientas - Logo Raspberry Pi	17
9.	Herramientas - Logo Cacao	17
10.	Herramientas - Logo Eclipse	17
11.	Herramientas - Logo Java	18
12.	Herramientas - Logo XML Copy Editor	18
13.	Herramientas - Logo Visual Paradigm	18
14.	Herramientas - Logo PC Worx	18
15.	Herramientas - Logo PL7 Pro	19
16.	Herramientas - Logo SFCEdit	19
17.	Herramientas - Logo TwinCat	19
18.	Captura de Requisitos - Modelo de Casos de Uso	23
19.	Captura de Requisitos - Jerarquía de Actores	23
20.	Captura de Requisitos - Prototipos de IU - Pantalla de Configuración	24
21.	Captura de Requisitos - Prototipos de IU - Pantalla de Gestión de Variables	24
22.	Captura de Requisitos - Descripción de la Arquitectura - Estructura Básica de un GRAFCET (<i>diseñado en SFCEdit</i>)	25
23.	Análisis y Diseño - Análisis - Diagrama de Clases	28
24.	Análisis y Diseño - Análisis - Clases Parte I	29
25.	Análisis y Diseño - Análisis - Clases Parte II	30
26.	Análisis y Diseño - Análisis - Clases Parte III	31

27.	Análisis y Diseño - Análisis - Clases Parte IV	32
28.	Análisis y Diseño - Análisis - Clases Parte V	33
29.	Análisis y Diseño - Diseño - Interfaz Menú	34
30.	Análisis y Diseño - Diseño - Interfaz Instrucciones	35
31.	Análisis y Diseño - Diseño - Interfaz Configuración	35
32.	Análisis y Diseño - Diseño - Interfaz Variables Detectadas	36
33.	Diseño - Arquitectura	36
34.	Desarrollo - Traductor	37
35.	Desarrollo - Diagrama del Formato de PLCOpen	38
36.	Pruebas del Software - Pruebas de Integración Modulo 1	39
37.	Pruebas del Software - Pruebas de Integración Modulo 2	40
38.	Pruebas del Software - Pruebas de Integración Modulo 3	40
39.	Pruebas del Software - Pruebas de Integración Modulo 4	40
40.	Caso de Uso Leer Instrucciones	47
41.	Casos de Uso Extendidos - Interfaz Menú	48
42.	Casos de Uso Extendidos - Interfaz Leer Instrucciones	48
43.	Caso de Uso Iniciar Traductor	49
44.	Casos de Uso Extendidos - Interfaz Menú	50
45.	Casos de Uso Extendidos - Interfaz Configuración	51
46.	Casos de Uso Extendidos - Interfaz Error XML no encontrado.	51
47.	Casos de Uso Extendidos - Interfaz Error XML con formato incorrecto.	51
48.	Casos de Uso Extendidos - Interfaz Error Directorio	51
49.	Casos de Uso Extendidos - Interfaz Error Rellenar Configuración	52
50.	Casos de Uso Extendidos - Interfaz de Finalizado	52
51.	Caso de Uso Gestionar Variables	53
52.	Casos de Uso Extendidos - Interfaz Configuración	54
53.	Casos de Uso Extendidos - Interfaz Error Rellenar Variables	54
54.	Casos de Uso Extendidos - Interfaz de Finalizado	54
55.	Análisis y Diseño - Diseño - Diagrama de Secuencia PreProceso I	55
56.	Análisis y Diseño - Diseño - Diagrama de Secuencia PreProceso II	56
57.	Análisis y Diseño - Diseño - Diagrama de Secuencia PreProceso III	57
58.	Análisis y Diseño - Diseño - Diagrama de Secuencia PreProceso IV	58
59.	Análisis y Diseño - Diseño - Diagrama de Secuencia Imprimir PLC Beckhoff	59
60.	Análisis y Diseño - Diseño - Diagrama de Secuencia Imprimir PLC TSX Micro	60
61.	Análisis y Diseño - Diseño - Diagrama de Secuencia Imprimir PLCOpen I	61
62.	Análisis y Diseño - Diseño - Diagrama de Secuencia Imprimir PLCOpen II.	62
63.	Manual de usuario - Menú	64
64.	Manual de usuario - Configuración del traductor	64
65.	Manual de usuario - Gestión de Variables	65

Índice de cuadros

1.	Alcance - Desglose de Tareas - Documentación.	10
2.	Alcance - Desglose de Tareas - Norma IEC 61131-3.	11
3.	Alcance - Desglose de Tareas - SFCEdit.	11
4.	Alcance - Desglose de Tareas - PLCOpen.	11
5.	Alcance - Desglose de Tareas - Plataformas de compatibilidad.	12
6.	Alcance - Desglose de Tareas - Alternativas de lector de XML.	12
7.	Alcance - Desglose de Tareas - Diseño de Diagramas.	12
8.	Alcance - Desglose de Tareas - Diseño de Pruebas.	13
9.	Alcance - Desglose de Tareas - Implementación de la Lógica de Negocio.	13
10.	Alcance - Desglose de Tareas - Implementación de Interfaz de Gráfica.	13
11.	Alcance - Desglose de Tareas - Implementación de pruebas.	14
12.	Alcance - Desglose de Tareas - Depuración del Código.	14
13.	Alcance - Desglose de Tareas - Manual.	14
14.	Gestión de Riesgo - Riesgo 1	20
15.	Gestión de Riesgo - Riesgo 2	20
16.	Gestión de Riesgo - Riesgo 3	20
17.	Gestión de Riesgo - Riesgo 4	21
18.	Gestión de Riesgo - Riesgo 5	21
19.	Gestión de Riesgo - Riesgo 6	21
20.	Evaluación Económica - Cuantificación de horas semanales por tarea	21
21.	Evaluación Económica - Cuantificación económica de horas	22
22.	Documentación de Pruebas I	41
23.	Documentación de Pruebas II	42

Introducción

La automatización industrial consiste en el uso de sistemas computarizados y electromecánicos para controlar procesos industriales.

Durante el diseño de los sistemas de control se desarrollan, revisan y documentan los refinamientos progresivos de las estructuras de datos, de la estructura del programa y de los detalles procedimentales. Para el diseño de software de control en el ámbito de la ingeniería de automatización se necesitan métodos y estándares que describan los procesos de control considerando las operaciones a realizar, la información procedente del proceso, y la secuenciación de las operaciones.

Existen distintos lenguajes de modelado que ayudan a representar gráficamente el proceso a automatizar, entre ellos tenemos:

Redes de petri: Definida en 1960 por CarlAdam Petri. Es un grafo dirigido que nos permite representar uno o distintos procesos para llegar a un mismo objetivo. Se representa con dos tipos de nodos: plazas o lugares (circunferencia) que representan las condiciones necesarias para que se ejecute un proceso y transiciones (segmento rectilíneo) que representan un procesos del sistema, y a su vez los arcos relacionan las condiciones y los procesos.

Grafcet (Grphe Fonctionnel de Commande Etape Transition): es un grafo o diagrama evolucionado a partir de las redes de Petri que contempla entradas, acciones a realizar, y los procesos intermedios que provocan estas acciones. En principio solo se usaba para documentar la parte secuencia de proceso, pero dado al uso constante que le dieron distintos fabricantes en sus autómatas se convierte en un potente lenguaje de programación que soluciona sistemas secuenciales.

Statechart: consta de estados, eventos y acciones, donde cada uno de ellos representa los aspectos relevantes, las transiciones y las operaciones a realizar.

Actualmente se usan estos lenguajes para modelar el proceso y luego se procede manualmente a la programación del autómata, anteriormente los distintos lenguajes de programación dependían de los fabricantes, pero hoy en día esto ha cambiado gracias a la norma IEC 61131-3.

La Norma IEC 61131-3 es la tercera parte del estándar internacional que establece los lenguajes de programación para PLC: Ladder Diagram (LD), Function Block Diagram (FBD), Structured Text (ST), Instruction List (IL) y Sequential Function Chart (SFC), para cada uno de ellos define las especificaciones de sintaxis y semántica, de esta forma la programación industrial sería independiente de cualquier fabricante. Por otra parte PLCOpen proporciona un formato para cada uno de estos lenguajes.

Lo que se busca con este Trabajo Fin de Grado (TFG) es, partiendo de los diseños de los sistemas de control desarrollados con el lenguaje de modelado Grafcet y utilizando la herramienta de diseño SFCedit, facilitar la programación generando de forma automática el código según el estándar.

Objetivos del proyecto

Objetivos

El TFG consistirá en desarrollar una herramienta que nos permita convertir un archivo XML exportado del editor SFCedit, en otro XML que se rija por el estándar IEC 61131-3.

Propósito

El propósito de este TFG es facilitar la programación de PLCs, permitiendo que se trabaje con un nivel de abstracción mayor en el desarrollo de los sistemas de control.

Ámbito

En la actualidad no hay una aplicación conocida que realice la transformación que ejecuta el traductor a desarrollar. Sin embargo hay teorías que demuestran que dicha transformación puede ser posible obteniendo el formato que establece PLCOpen.

La organización PLCOpen, que busca la eficiencia en la automatización industrial, enfoca su atención hacia la armonización de la programación de los sistemas de control, por lo que ha creado formatos de intercambio XML basados en el estándar IEC 61131-3.

Contexto de Negocio

Este proyecto puede afectar al campo de la industria ya que les permitirá una mayor abstracción en el desarrollo de los sistemas de control.

Esta herramienta es desarrollada como un TFG de la titulación de Grado en Ingeniería Informática de Gestión y Sistemas de Información, en previo acuerdo con los profesores que han realizado la propuesta de dicho proyecto y los tutores del mismo.

Para realizar el proyecto se usarán como guías la Norma IEC 61131-3 y las plantillas que proporciona PLCOpen para la transformación al estándar.

Definiciones, Acrónimos, y Abreviaturas

- *API (Application Programming Interface)* : conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
- *IEC (International Electrotechnical Commission)*: Comisión Electro-técnica Internacional que es una organización de normalización en los campos de lo eléctrico, electrónica y tecnologías relacionadas.
- *DON (Document Object Model)* : Es una API que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.
- *IDE (Integrated Development Environment)*: Entornos de Desarrollo Integrado.

- *Norma IEC 61131*: estándar de los autómatas programables y sus periféricos. Fue publicada por primera vez en diciembre de 1993 por la IEC y su edición actualizada fue publicada en febrero del 2013.
- *Patron Singleton*: diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto, garantizando que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Se implementa creando en nuestra clase un método que crea una instancia del objeto para asegurar que la clase no puede ser instanciada nuevamente, se regula el alcance del constructor con atributos como protegido o privado.
- *PLC (Programmable Logic Controller)* : Controlador lógico programable es una computadora utilizada en la ingeniería automática o automatización industrial, para automatizar procesos electromecánicos, tales como el control de la maquinaria de la fábrica en líneas de montaje o atracciones mecánicas.
- *PLCOpen*: es una organización independiente que busca proporcionar eficiencia a la automatización industrial, basándose principalmente en la Norma IEC 61131-3.
- *SLIP (Extensible Stylesheet Language)* : Es una familia del lenguaje XML que permite describir como debe estar el contenido de un XML.
- *SAX (Simple API from XML)* : API para usar XML en lenguajes de programación Java.
- *TAD (Tipo Abstracto de Datos)*: es una especificación de un conjunto de datos y un conjunto de operaciones que pueden realizarse sobre esos datos.
- *TFG*: Trabajo Fin de Grado.
- *XML (eXtensible Markup Language)*: es un lenguaje basado en marcas que se utiliza para almacenar datos de forma legible.
- *XSD (XML Schema Definition)*: es un lenguaje de esquema, que se usa para describir la estructura de un XML.
- *XSLT (Transformaciones SLIP)* : Estándar que sirve para transformar ficheros XML en otro fichero ya sea XML o no.

Alcance

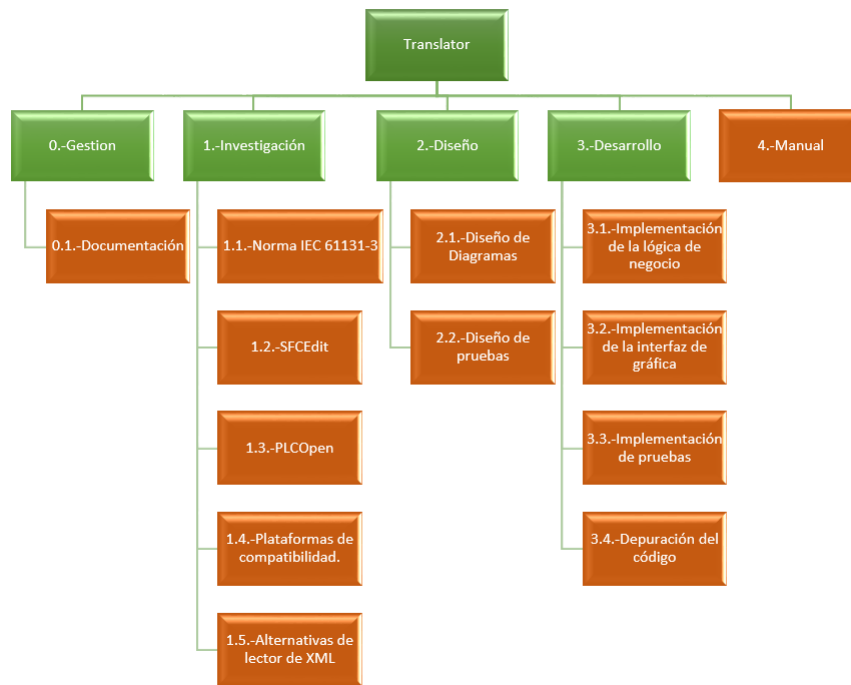
En el traductor a desarrollar el usuario podrá:

- Importar un fichero XML obtenido de SFCedit.
- Seleccionar el directorio en el que desea que se guarden los ficheros que generará el traductor.
- Seleccionar el lenguaje al que se quiere realizar la traducción.
- Decidir el tipo de cada una de las señales encontradas, clasificándolas por tipo de datos y tipo de variable.

El algoritmo de esta herramienta debe conseguir:

- Validar el fichero XML importado para comprobar que tenga el formato específico de SFCedit.
- Recorrer el fichero guardando los datos obtenidos en objetos.
- Generar una lista de las señales que ha detectado en el fichero.
- Generar los ficheros correspondientes con el lenguaje seleccionado y que se base en el estándar IEC 61131-3.

Estructura de Descomposición del Trabajo (EDT)



1. Figura: Alcance - Estructura de Descomposición del Trabajo.

Desglose de Tareas

<p>Paquete De Trabajo: 0.1.- Documentación. Responsables: Helen J.Paz Cotón. Esfuerzo: 26 Horas / Persona. Duración: 26 Horas.</p>
<p>Descripción: Tarea que consta de la redacción de la documentación, así como los distintos puntos que esta compone. (En esta tarea no se incluye el manual de usuario). Entrada: Información correspondiente a la documentación, figuras, cuadros y diagramas del proyecto. Salidas/Entregables: Documento que contiene la información referente al detalle de la realización del proyecto. Recursos necesarios: Editor de documentos y una persona. Precedencias: Conocer el contenido de los distintos apartados de la documentación.</p>

1. Cuadro: Alcance - Desglose de Tareas - Documentación.

Paquete De Trabajo: 1.1.- Norma IEC 61131-3.

Responsables: Helen J.Paz Cotón.

Esfuerzo: 16 Horas / Persona.

Duración: 16 Horas.

Descripción: Investigación e información de los distintos lenguajes que establece la Norma IEC 61131-3.

Entrada: Ninguna.

Salidas/Entregables: Conocimiento de los distintos lenguajes establecidos por la Norma IEC 61131-3 y como implementarlos.

Recursos necesarios: Documentación de la Norma IEC 61131 y una persona.

Precedencias: Ninguna.

2. Cuadro: Alcance - Desglose de Tareas - Norma IEC 61131-3.

Paquete De Trabajo: 1.2.- SFCEdit.

Responsables: Helen J.Paz Cotón.

Esfuerzo: 6 Horas / Persona.

Duración: 6 Horas.

Descripción: Investigar los distintos elementos que contiene un gráfico exportado de SFCEdit.

Entrada: Ninguna.

Salidas/Entregables: Conocimientos de los elementos que contiene un XML exportado del software SFCEdit, decisión de como usarlos.

Recursos necesarios: software SFCEdit y una persona.

Precedencias: Ninguna.

3. Cuadro: Alcance - Desglose de Tareas - SFCEdit.

Paquete De Trabajo: 1.3.- PLCOpen.

Responsables: Helen J.Paz Cotón.

Esfuerzo: 40 Horas / Persona.

Duración: 40 Horas.

Descripción: Investigación a fondo de la documentación que presenta PLCOpen y su estructura que proponen para cada uno de los distintos lenguajes que componen la Norma IEC 61131-3.

Entrada: Ninguna.

Salidas/Entregables: Conocimientos sobre la estructura que propone PLCOpen para cada lenguaje y decisión de como implementarlo.

Recursos necesarios: Documentación de PLCOpen y una persona.

Precedencias: Ninguna.

4. Cuadro: Alcance - Desglose de Tareas - PLCOpen.

Paquete De Trabajo: 1.4.- Plataformas de compatibilidad.
Responsables: Helen J.Paz Cotón.
Esfuerzo: 16 Horas / Persona.
Duración: 16 Horas.

Descripción: Investigación sobre la programación del lenguaje literal estructurado en las plataformas PLC TSX Micro (PL7Pro) y PLC Beckhoff (TwinCAT).
Entrada: Ninguna.
Salidas/Entregables: Conocimientos de las estructuras de programación que se usarán para la transformación en cada caso de la aplicación.
Recursos necesarios: Documentación de PLC TSX Micro (PL7Pro), PLC Beckhoff (TwinCAT) y una persona.
Precedencias: Ninguna.

5. Cuadro: Alcance - Desglose de Tareas - Plataformas de compatibilidad.

Paquete De Trabajo: 1.5.- Alternativas de lector de XML.
Responsables: Helen J.Paz Cotón.
Esfuerzo: 20 Horas / Persona.
Duración: 20 Horas.

Descripción: Analizar y comparar las distintas alternativas DON y SAX que sirven para la lectura de ficheros XML usando java.
Entrada: Ninguna.
Salidas/Entregables: Conocimientos de DON (DON Analizador) y SAX (Simple API for XML) y elección de SAX para la extracción de información de un fichero XML.
Recursos necesarios: Documentación de las distintas alternativas.
Precedencias: Ninguno.

6. Cuadro: Alcance - Desglose de Tareas - Alternativas de lector de XML.

Paquete De Trabajo: 2.1.- Diseño de Diagramas.
Responsables: Helen J.Paz Cotón.
Esfuerzo: 30 Horas / Persona.
Duración: 30 Horas.

Descripción: Desarrollo de los distintos diagramas; diagrama de clases, diagramas de casos de uso, diagramas de secuencia, los cuales permitirán la comprensión de atributos y métodos de las aplicación.
Entrada: Planificación del proyecto y bocetos.
Salidas/Entregables: Diagrama de clases, diagrama de secuencia, diagramas de casos de uso y prototipos de interfaz gráfica.
Recursos necesarios: Aplicación Visual Paradigm, Cacao y una persona.
Precedencias: Diseño de bocetos de la aplicación.

7. Cuadro: Alcance - Desglose de Tareas - Diseño de Diagramas.

Paquete De Trabajo: 2.2.- Diseño de Pruebas.

Responsables: Helen J.Paz Cotón.

Esfuerzo: 16 Horas / Persona.

Duración: 16 Horas.

Descripción: Analizar distintos métodos a implantar en la aplicación para decidir cual de estos es necesario probar y así corregir posibles errores.

Entrada: métodos existentes en el diagrama de clases.

Salidas/Entregables: Documento que refleje las pruebas a realizar por cada uno de los métodos analizados.

Recursos necesarios: Editor de documentos y una persona.

Precedencias: 2.1- Diseño de Diagramas.

8. Cuadro: Alcance - Desglose de Tareas - Diseño de Pruebas.

Paquete De Trabajo: 3.1.- Implementación de la Lógica de Negocio.

Responsables: Helen J.Paz Cotón.

Esfuerzo: 100 Horas / Persona.

Duración: 100 Horas.

Descripción: Implementación de cada uno de los métodos necesarios para el correcto funcionamiento de la aplicación.

Entrada: Diseño de diagramas.

Salidas/Entregables: Aplicación funcional, sin interfaz gráfica.

Recursos necesarios: IDE eclipse con java y una persona.

Precedencias: 2.1.- Diseño de Diagramas. 2.2.- Diseño de Pruebas.

9. Cuadro: Alcance - Desglose de Tareas - Implementación de la Lógica de Negocio.

Paquete De Trabajo: 3.2.- Implementación de Interfaz de Gráfica.

Responsables: Helen J.Paz Cotón.

Esfuerzo: 40 Horas / Persona.

Duración: 40 Horas.

Descripción: Implementación de la interfaz gráfica que servirá para que el usuario interactúe con la aplicación.

Entrada: Prototipo de las pantallas.

Salidas/Entregables: Documento que contiene la información referente al detalle de la realización del proyecto.

Recursos necesarios: IDE eclipse con java y una persona.

Precedencias: 3.1.- Implementación de la lógica de negocio.

10. Cuadro: Alcance - Desglose de Tareas - Implementación de Interfaz de Gráfica.

Paquete De Trabajo: 3.3.- Implementación de pruebas.
Responsables: Helen J.Paz Cotón.
Esfuerzo: 20 Horas / Persona.
Duración: 20 Horas.

Descripción: Implementación de las pruebas especificadas en el diseño previo de las mismas.
Entrada: Diseño de pruebas.
Salidas/Entregables: Implementación de los métodos necesarios para realizar las pruebas de la aplicación.
Recursos necesarios: IDE eclipse con java y una persona.
Precedencias: 2.2.- Diseño de Pruebas.

11. Cuadro: Alcance - Desglose de Tareas - Implementación de pruebas.

Paquete De Trabajo: 3.4.- Depuración del Código.
Responsables: Helen J.Paz Cotón.
Esfuerzo: 30 Horas / Persona.
Duración: 30 Horas.

Descripción: Ejecución de las pruebas para corregir posibles errores en la logica de negocio.
Entrada: Lógica de negocio e interfaz gráfica.
Salidas/Entregables: Aplicación con la funcionalidad completa y correcta.
Recursos necesarios: IDE eclipse con java y una persona.
Precedencias: 3.1.- Implementación de la Lógica de Negocio,
3.2.- Implementación de Interfaz de Gráfica. 3.3.- Implementación de pruebas.

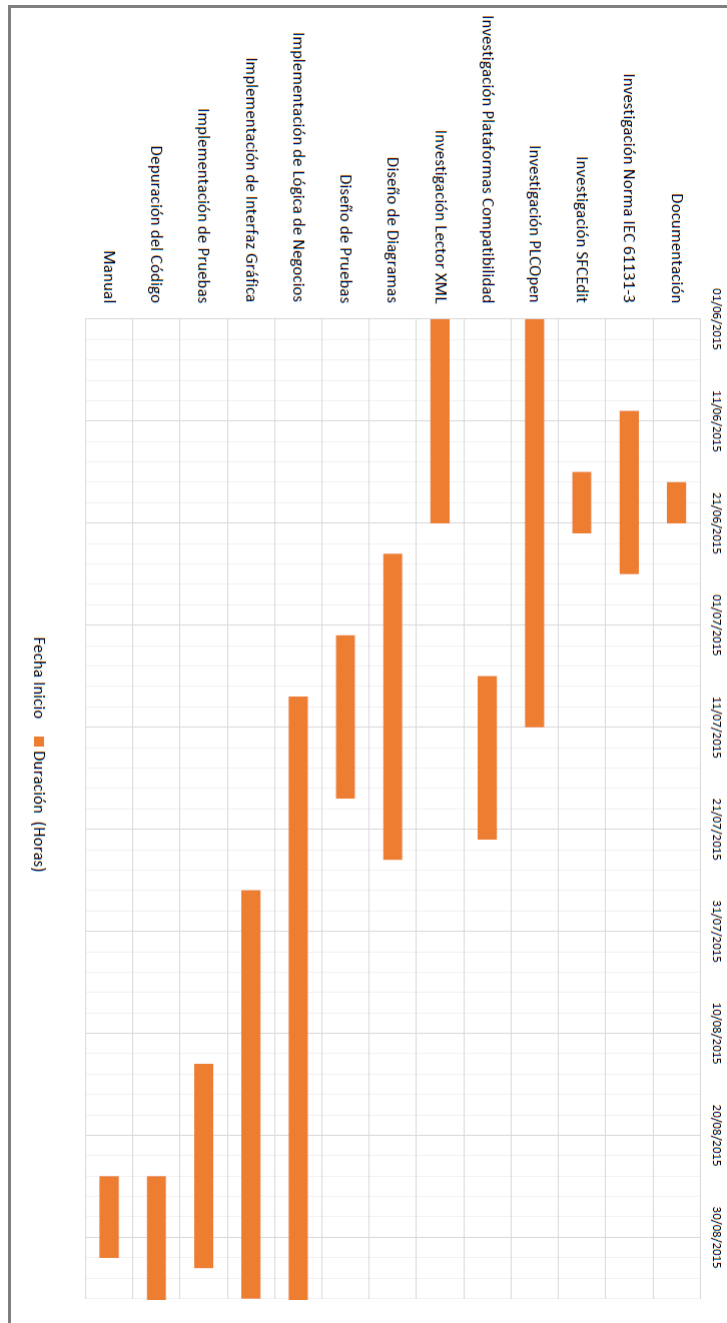
12. Cuadro: Alcance - Desglose de Tareas - Depuración del Código.

Paquete De Trabajo: 4.- Manual.
Responsables: Helen J.Paz Cotón.
Esfuerzo: 8 Horas / Persona.
Duración: 8 Horas.

Descripción: Documentación con la explicación detallada de como usar la aplicación paso a paso.
Entrada: Aplicación de la cual se explicara el funcionamiento.
Salidas/Entregables: Manual de usuario.
Recursos necesarios: Editor de documentos y una persona.
Precedencias: 3.1.- Implementación de la Lógica de Negocio,
3.2.- Implementación de Interfaz de Gráfica. 3.3.- Implementación de Pruebas.
3.4.- Depuración del Código.

13. Cuadro: Alcance - Desglose de Tareas - Manual.

Planificación Temporal



2. Figura: Alcance - Planificación Temporal.

Herramientas

Las herramientas utilizadas para el desarrollo del proyecto, se han clasificado en tres grupos: un primer grupo utilizadas para la gestión y documentación del mismo; un segundo grupo utilizado para el desarrollo del proyecto; y un último grupo, pertenecientes al entorno de los sistemas de automatización, utilizadas para la definición y pruebas del proyecto.

Herramientas de Gestión y Documentación

- Dropbox



3. Figura: Herramientas - Logo Dropbox

Dropbox es una herramienta de almacenamiento en la nube. Se ha utilizado para guardar toda la información referente a la documentación e investigación.

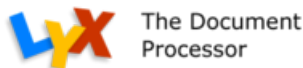
- Git



4. Figura: Herramientas - Logo Git

Git es un software de control de versiones diseñado por Linus Torvalds. Se ha utilizado para el control de versiones de la aplicación desarrollada.

- Lyx



5. Figura: Herramientas - Logo Lyx

Lyx es un editor de documentos que permite la inserción de código \LaTeX . Se ha usado para desarrollar la documentación del proyecto.

- Microsoft Word 2013



6. Figura: Herramientas - Logo Microsoft Office

Microsoft Word es una herramienta de procesamiento de texto y documentación. Gracias a su facilidad de uso y agradable apariencia final, ha sido la herramienta escogida para realizar los diagramas jerárquicos y diagrama de Gant.

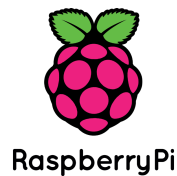
- ObjectAid UML Explorer



7. Figura: Herramientas - Logo ObjectAid UML Explorer

ObjectAid UML Explorer es una herramienta que muestra el código y las librerías java, en clases UML y diagramas de secuencia que son automáticamente actualizadas. Se ha utilizado para realizar el diagrama de clases y los diagramas de secuencia.

- Raspberry Pi



8. Figura: Herramientas - Logo Raspberry Pi

Raspberry Pi es una placa de computadora de bajo coste desarrollada por la fundación Raspberry Pi. Se ha utilizado como servidor que aloja el programa de control de versiones del proyecto.

Herramientas de Desarrollo

- Cacao



9. Figura: Herramientas - Logo Cacao

Cacao es una herramienta de diseño online. Se ha usado para realizar los prototipos de pantallas de la aplicación y el diseño de la arquitectura.

- IDE Eclipse



10. Figura: Herramientas - Logo Eclipse

Eclipse es una herramienta usada comúnmente como entorno de desarrollo integrado para java. Se ha usado para realizar la programación de la aplicación.

- Java



11. Figura: Herramientas - Logo Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Se ha usado para realizar toda la programación e interfaz gráfica de la aplicación.

- XML Copy Editor



12. Figura: Herramientas - Logo XML Copy Editor

XML Copy Editor es un editor especializado para ficheros XML. Esta herramienta ha sido usada en el proyecto para generar un esquema XSD a partir de un XML específico.

- Visual Paradigm versión 10.0



13. Figura: Herramientas - Logo Visual Paradigm

Visual Paradigm es un software de modelado UML que permite analizar, diseñar, codificar, probar y desplegar, puede incluso generar código fuente a partir de diagramas. Se ha usado para realizar los casos de uso.

Herramientas de Diseño y Programación de PLCs

Herramientas utilizadas para la definición y realización de pruebas.

- PC Worx



14. Figura: Herramientas - Logo PC Worx

PC Worx es un software de ingeniería universal para todos los sistemas de control de Phoenix Contact. Une la programación según el estándar IEC 61131.

- PL7 Pro



15. Figura: Herramientas - Logo PL7 Pro

PL7 Pro software para la programación de PLC que usa los lenguajes de programación de los autómatas Micro, Premium y Atrium.

- SFCEdit



16. Figura: Herramientas - Logo SFCEdit

SFCEdit es una herramienta de diseño de Grafcet. Con esta herramienta se crearon los ejemplos usados para conocer las etiquetas de la herramienta y realizar las pruebas de la aplicación desarrollada.

- TwinCat



17. Figura: Herramientas - Logo TwinCat

TwinCat es una herramienta que convierte prácticamente cualquier PC compatible en un controlador en tiempo real con un sistema multi-PLC, en un entorno de programación y funcionamiento de la estación.

Gestión de Riesgos

En todo proyectos existe la posibilidad de que ocurran situaciones inesperadas que impidan seguir la planificación del mismo. Para intentar evitar que esto suceda se analizarán los distintos riesgos a los que podría ser expuesto el proyecto y de esta forma establecer un plan de contingencia en caso de que sucediese alguno.

El riesgo se mide asumiendo una determinada vulnerabilidad frente a cada tipo de peligro, para realizar este estudio, acotaremos las probabilidades de ocurrencia y el impacto de la siguiente manera:

- Medidas de probabilidad de ocurrencia.
 - Improbable [0-25 %].
 - Poco Probable [25-50 %].
 - Probable [50-75 %].
 - Muy Probable [75-100 %].

- Medidas de impacto sobre la planificación del proyecto.
 - Bajo.
 - Medio.
 - Alto.

Descripción	Retraso del proyecto debido a la falta de experiencia en la planificación de proyectos.
Prevenición	Inclusión de tiempos de holgura.
Plan de contingencia	Realización de horas extra.
Probabilidad	Probable.
Impacto	Alto.

14. Cuadro: Gestión de Riesgo - Riesgo 1

Descripción	Retraso del proyecto debido a interpretaciones erróneas del proyecto a realizar o falta de detalle del producto deseado.
Prevenición	Realizar reuniones continuamente con el director del proyecto, para discutir los avances del proyecto, creación de prototipos e inclusión de tiempos de holgura en las horas destinadas al proyecto.
Plan de contingencia	Negociaciones por ambas partes y realizar horas extra.
Probabilidad	Muy probable.
Impacto	Alto.

15. Cuadro: Gestión de Riesgo - Riesgo 2

Descripción	Baja de la persona que realiza el proyecto, ya sea por enfermedad, accidente, problemas personales, otros.
Prevenición	No se pueden prevenir este tipo de riesgos.
Plan de contingencia	Realización de horas extra o modificación de fechas de entrega en caso de que sea necesario.
Probabilidad	Improbable.
Impacto	Alto.

16. Cuadro: Gestión de Riesgo - Riesgo 3

Descripción	Perdida del equipo debido a accidentes como pueden ser; rotura del software, robo, otros.
Prevención	Realizar copias de seguridad en la nube y discos externos disponer de un equipo de reserva con el software necesario.
Plan de contingencia	Restauración de la copia de seguridad y uso del equipo de sustitución.
Probabilidad	Poco probable.
Impacto	Medio.

17. Cuadro: Gestión de Riesgo - Riesgo 4

Descripción	Perdida del equipo debido a un fallo del software (virus informático).
Prevención	Realizar copias de seguridad en la nube y discos externos, disponer de un equipo de reserva y de un software para malware actualizado.
Plan de contingencia	Restaurar la copia de seguridad y usar el equipo de sustitución hasta recuperar el equipo afectado, en caso de que se pueda.
Probabilidad	Poco probable.
Impacto	Bajo.

18. Cuadro: Gestión de Riesgo - Riesgo 5

Descripción	Retraso del desarrollo del proyecto debido a errores encontrados en una etapa avanzada del mismo.
Prevención	Analizar bien cada uno de los puntos a programar e ir probando ejecutando pruebas cada cierto tiempo, para localizar errores a tiempo.
Plan de contingencia	Realizar horas extras.
Probabilidad	Probable.
Impacto	Medio.

19. Cuadro: Gestión de Riesgo - Riesgo 6

Evaluación Económica

Ya que se conoce de antemano que es un proyecto universitario, tiene un tiempo limitado y será realizado mayormente usando los recursos del hogar o de la universidad, no se vio necesario realizar los cálculos para determinar un salario mínimo anual, los gastos generales y administrativos (la cuota de autónomo, alquiler de espacio de trabajo, equipos, material de oficina, entre otros). Por estas razones simplemente se estipula un precio/hora de 12€ por persona para realizar los cálculos económicos del desarrollo del proyecto según el análisis realizado y la previsión del tiempo que se le dedicará.

Tarea / Semana	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Total
Documentación			6			8							6	6	26 Horas
Investigación	30	30	16			12				10					98 Horas
Diseño				6	20	20									46 Horas
Desarrollo							20	30	40	20	20	20	20	20	190 Horas
Manual													8		8 Horas

20. Cuadro: Evaluación Económica - Cuantificación de horas semanales por tarea

Semana	1	2	3	4	5	6	7
Horas	40	30	22	6	20	40	40
Coste Hora	12€	12€	12€	12€	12€	12€	12€
Ingresos	0€	0€	0€	0€	0€	0€	0€
Coste Semana	480€	360€	264€	72€	220€	480€	480€
Coste Acumulado	-480€	-840€	-1104€	-1176€	-1396€	-1876€	-2356€

Semana	8	9	10	11	12	13	14
Horas	30	40	20	20	40	14	6
Coste Hora	12€	12€	12€	12€	12€	12€	12€
Ingresos	0€	0€	0€	0€	0€	0€	0€
Coste Semana	360€	480€	220€	220€	480€	168€	72€
Coste Acumulado	-2716€	-3196€	-3416€	-3636€	-4116€	-4284€	-4356€

21. Cuadro: Evaluación Económica - Cuantificación económica de horas

Como se puede observar en las tablas, el proyecto realizado tendría un coste de 4.356,00€ desarrollándose en un periodo de aproximadamente 4 - 5 meses. Que sería un coste relativamente económico tomando en cuenta que solo es el coste de persona/hora.

Antecedentes

Situación Actual

En la actualidad la programación de los sistemas se realiza de forma manual a partir de los diseños, por un lado, se programa la parte secuencial en la que se define la secuencia del proceso con sus pasos y transiciones correspondientes y, por otro lado, se programa la parte combinacional donde se generan las acciones a realizar.

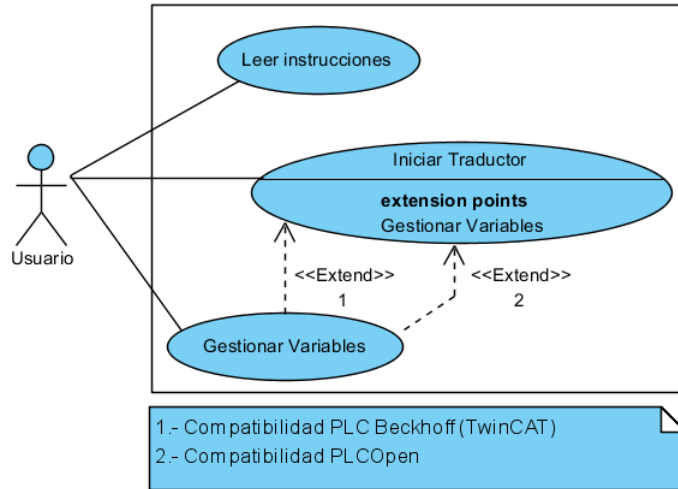
Estudio de Diferentes Alternativas Existentes

Actualmente, no existe aún una alternativa, que realice lo que hace el traductor creado en este proyecto, ya que mediante este traductor, se le brinda al usuario la posibilidad de pasar un diagrama secuencial directamente a la programación de un autómata programable. Inicialmente este proceso de traducción se realiza a uno de los cinco lenguajes establecidos en el estándar, pero tiene la posibilidad de ampliarse de manera que traduzca en los distintos lenguajes y en compatibilidad con distinto software.

Una vez estudiado con detenimiento el tema, se podrían tener distintos enfoques, en este proyecto usamos la herramienta SFCEdit ya que en esta se pueden desarrollar Grafcet de manera correcta, de forma clara, fácil y a su vez nos permite exportar obteniendo un fichero XML, lo que al parecer sería fácil de transformar en otro XML que siga las plantillas de PLCOpen. Si embargo la limitación de la información que nos proporciona el SFCEdit no nos permite hacer esta transformación de manera correcta en todos los lenguajes existentes en la Norma, por lo que se podría enfocar de distinta manera usando otra herramienta gráfica que nos proporcione la información necesaria o desarrollando la herramienta gráfica y luego adaptar el traductor a ella.

Captura de Requisitos

Modelo de Casos de Uso



18. Figura: Captura de Requisitos - Modelo de Casos de Uso

Jerarquía de Actores

Esta aplicación tendrá un único usuario, dado que no es necesario registrarse para ser usada, por lo que su jerarquía de actores sería la siguiente:



19. Figura: Captura de Requisitos - Jerarquía de Actores

Prototipos de Interfaces de Usuario

- Pantalla de configuración del traductor.

La pantalla de configuración, nos permitirá seleccionar el fichero XML del cual queremos realizar la traducción, así como seleccionar la carpeta en la que queremos que se guarden los ficheros, que exportará la aplicación una vez terminado el proceso de traducción. En esta pantalla también se puede seleccionar el lenguaje, al que traducir de una lista, basada en los lenguajes que establece la Norma IEC 61131-3 y por último se puede escoger un software de compatibilidad, para que el código de los ficheros exportados pueda ser usado en este software directamente, sin necesidad de tener que realizar la programación manualmente.

Traductor

Indica el fichero .xml que deseas traducir.

Explorar

Carpeta donde se guardaran los ficheros generados.

Explorar

Lenguaje de salida

IL (Lista de Instrucciones)

Compatibilidad con

Radio1
 Radio 2
 Radio1

Iniciar

20. Figura: Captura de Requisitos - Prototipos de IU - Pantalla de Configuración

■ Pantalla de gestión de variables.

La pantalla de gestión de variables le permitirá al usuario decidir a que tipo corresponde cada una de las variables que ha detectado el traductor. Dentro de las elecciones que se pueden hacer tendremos el tipo de variable, el tipo de variable según la procedencia de los datos, tipo de temporizador y tipo de contador. También existe la opción de indicar que dicho elemento no es un temporizador o un contador o una señal del sistema, en dicho caso este no será tomado en cuenta en el proceso de traducción.

Traductor

Señales

Indica a que tipo cooresponde:

Señal	Entrada	Salida	Name	Name	Name	No es una señal
Sample 1	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sample 2	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sample 3	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sample 4	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Sample 5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sample 6	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Temporizadores

Indica a que tipo cooresponde:

Temporizador	TON	TOF	Name	No es un temporizador
Sample 1	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sample 2	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Sample 3	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Contadores

Indica a que tipo cooresponde:

Contador	tipo1	tipo2	Name	No es un contador
Sample 1	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sample 2	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Sample 3	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

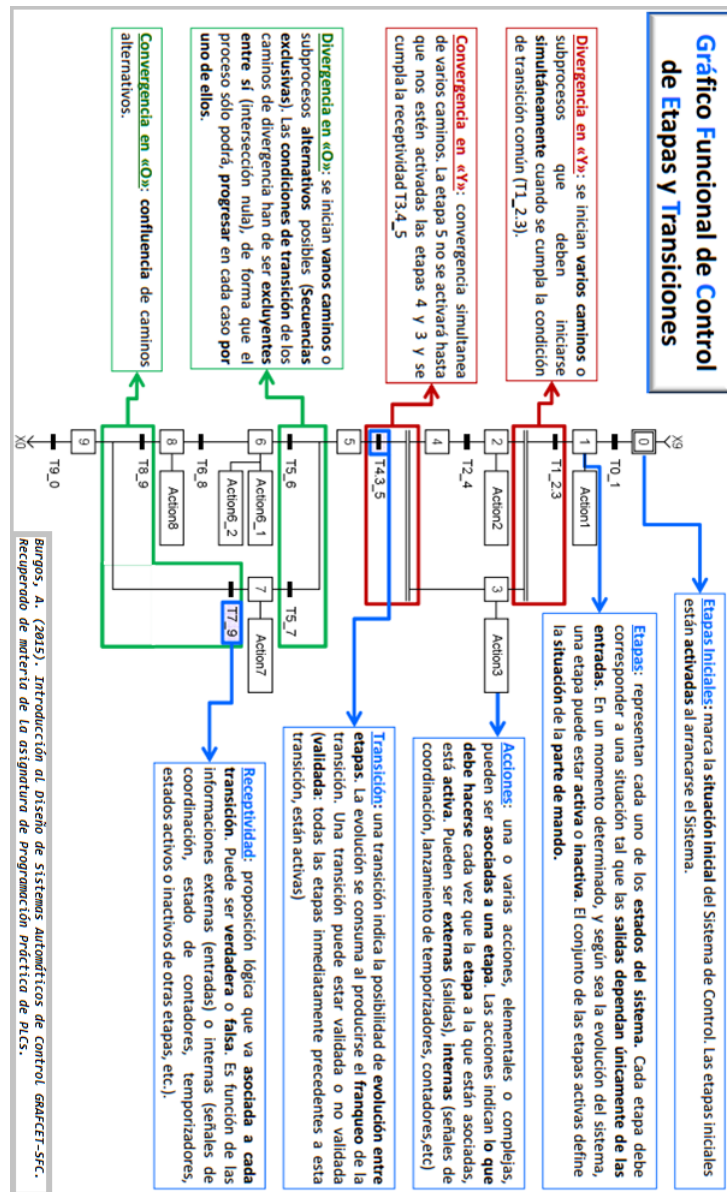
Cancelar Guardar

21. Figura: Captura de Requisitos - Prototipos de IU - Pantalla de Gestión de Variables

Descripción de la Arquitectura

Para determinar la arquitectura de la aplicación se estudian los distintos modelos a tener en cuenta para el análisis de los datos, dentro de los cuales tenemos:

- GRAF CET** que permite describir gráficamente el proceso, es importante tener en cuenta y entender cada una de sus partes, ya que a través de ellas obtendremos los datos necesarios para generar las salidas de la aplicación a realizar. La estructura básica de un GRAFCET contiene una serie de pasos que se explican en la Figura 22 basados en esta estructura se puede pasar a la implementación del diseño obteniendo la parte combinacional y la parte secuencial del mismo.



22. Figura: Captura de Requisitos - Descripción de la Arquitectura - Estructura Básica de un GRAFCET (diseñado en SFCEdit)

Parte Combinacional: es la generación de las salidas en función de las etapas activas, las salidas dependen únicamente del estado de sus entradas.

Parte Secuencial: indica cuando se activa y desactiva una etapa, las salidas dependerán de los estados anteriores y su evolución.

Otro dato importante a tener en cuenta en el momento de realizar la programación de un diseño son los temporizadores o contadores.

- *Contadores:* son usados en las acciones para inicializar y actualizar el contador y en las transiciones para consultar y comparar su valor.
 - *Temporizadores:* son usados en las acciones para inicializar su valor y en las transiciones para consultar si ha finalizado la temporización.
- *SFCEdit*, esta herramienta no solo permite la creación del diseño si no que también ofrece la opción de exportarlo en distintos formatos, dentro de estos tenemos el XML que se usará como entrada en la aplicación a desarrollar, por lo que es necesario conocer como describe cada una de las partes del GRAFCET para poder generar el modelo con el que trabajara la aplicación.

Tras analizar las líneas de código XML generadas por SFCEdit se realiza una clasificación de cada dato a tomar en cuenta:

- *Project:* dentro de este se encontrara uno o más grafcet y los atributos propios del proyecto.
- *Grafcet:* dentro de él se encontrara una o más Sequence, hlink y jump, además de sus atributos propios en los cuales podemos observar tres tipos distintos de Grafcet; normal, macro y enclosure.
- *Sequencia :* dentro de cada secuencia podemos encontrar ninguna, una o varias etapas (step) y transiciones (transition), durante el estudio de esta etiqueta se observa que es una agrupación de distintas etapas y transiciones las cuales convergen, divergen o saltan a otra secuencia, su atributo id nos indica el número de secuencia que representa.
- *Step :* una etapa puede contener o no acciones (action) asociados, así como comentarios, además de sus atributos propios en los cuales se encuentran distintos tipos de etapas;
 - Step.
 - Initial Step.
 - Task Step.
 - Macro Step: es una representación simbólica de una parte de un grafcet, su finalidad es descomponer el grafcet en partes más pequeñas para mejorar su comprensión. Llama a un grafcet de tipo macro.
 - Enclosing Step: representa un conjunto de etapas encapsuladas dentro de otras, las etapas de este tipo tienen asociado un grafcet de tipo enclosure y dicho grafcet se activará o desactivará según la activación o des-activación de la etapa de encapsulado.
 - Initial Enclosing Step: si además de ser una etapa encapsulada es una etapa inicial indica que al iniciarse el grafcet global activa a su vez el grafcet de encapsulado.

En la aplicación a realizar solo se tendrán en cuenta las etapas normales y las iniciales.

- *Action:* la etiqueta que viene dentro de cada acción varia dependiendo del tipo de acción que pueden ser:
 - Action on activation.
 - Action on desactivation.
 - Conditional action.

- Action on event.
- Forcing order.
- Join to previous action.

En la aplicación se tendrán en cuenta la acción normal y de forzado que contendrán una etiqueta de texto dentro y la acción condicional que tendrá una etiqueta de condición.

- Transition: una transición siempre contendrá una receptividad (condition) en la que se encontrara la expresión lógica necesaria para el flanqueo de la transición y también puede tener o no un comentario asociado.
- Hlink: tiene atributos propios como el tipo que indicará si es convergencias o divergencias de caminos alternativos o simultáneos y en el atributo seqid tendremos el número de secuencia de la cual converge o diverge según sea el tipo. También contendrá dos o más nodos (node) en los que se indicarán las secuencias de las que parte o a las que va la secuencia que indica el atributo seqid.
- Node: cada nodo contendrá un único atributo seqid que indicará un número de secuencia.
- Jump: como mínimo en cada grafcet debe haber un salto que vaya de la última transición a la etapa inicial. Sin embargo pueden haber más saltos que nos llevan a distintas etapas o transiciones del grafcet.

Aunque la aplicación a realizar no tenga en cuenta todos los tipos de grafcet, step o action, quedara abierta la opción de añadir estas consideraciones en mejoras futuras.

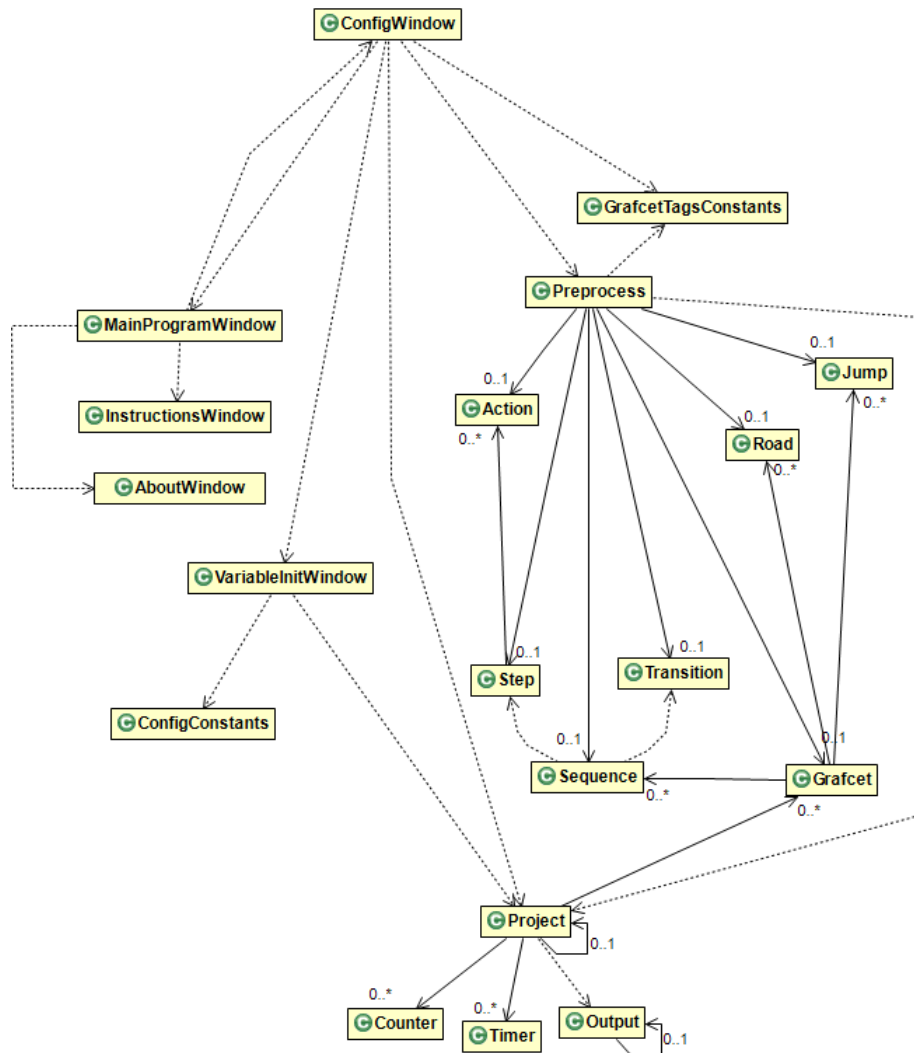
- *Modelo final*, una vez realizado el análisis de los datos con los que se va a trabajar, se decide usar una arquitectura local, la información se almacena en clases que inicialmente serán las siguientes:
 - Clases que serán de Tipo Abstracto de Datos (TAD) para guardar en ellas los atributos y métodos necesario.
 - Step: contendrá sus atributos y una lista de acciones.
 - Transition: tendrá como atributos una condición y un comentario.
 - Action: incluirá como atributos, el tipo de acción, el texto de dicha acción y comentarios.
 - Road: en esta clase se guardará el tipo de camino si es convergencia o divergencia, el número de secuencia de la que viene o a la que va según sea el tipo y una lista con las distintas secuencias que tiene asociadas.
 - Jump: contendrá como atributos el número de secuencia final y el numero de secuencia final del salto.
 - Sequence: englobará una lista de objetos que pueden ser tanto transiciones como pasos y un número de secuencia como atributo.
 - Grafcet: comprenderá los atributos del grafcet y una lista de secuencias.
 - Timer: incluirá los atributos necesarios de un temporizador, de los cuales algunos de ellos se les pedirán al usuario.
 - Counter: contendrá los atributos necesarios de un contador, de los cuales algunos de ellos se les pedirán al usuario.
 - Una clase Project con patrón singleton ya que será única y no podrá ser instanciada por ninguna otra clase, esta tendrá tres listas:
 - Lista de Grafcet.
 - Lista de Timer.
 - Lista de Counter.

Análisis y Diseño

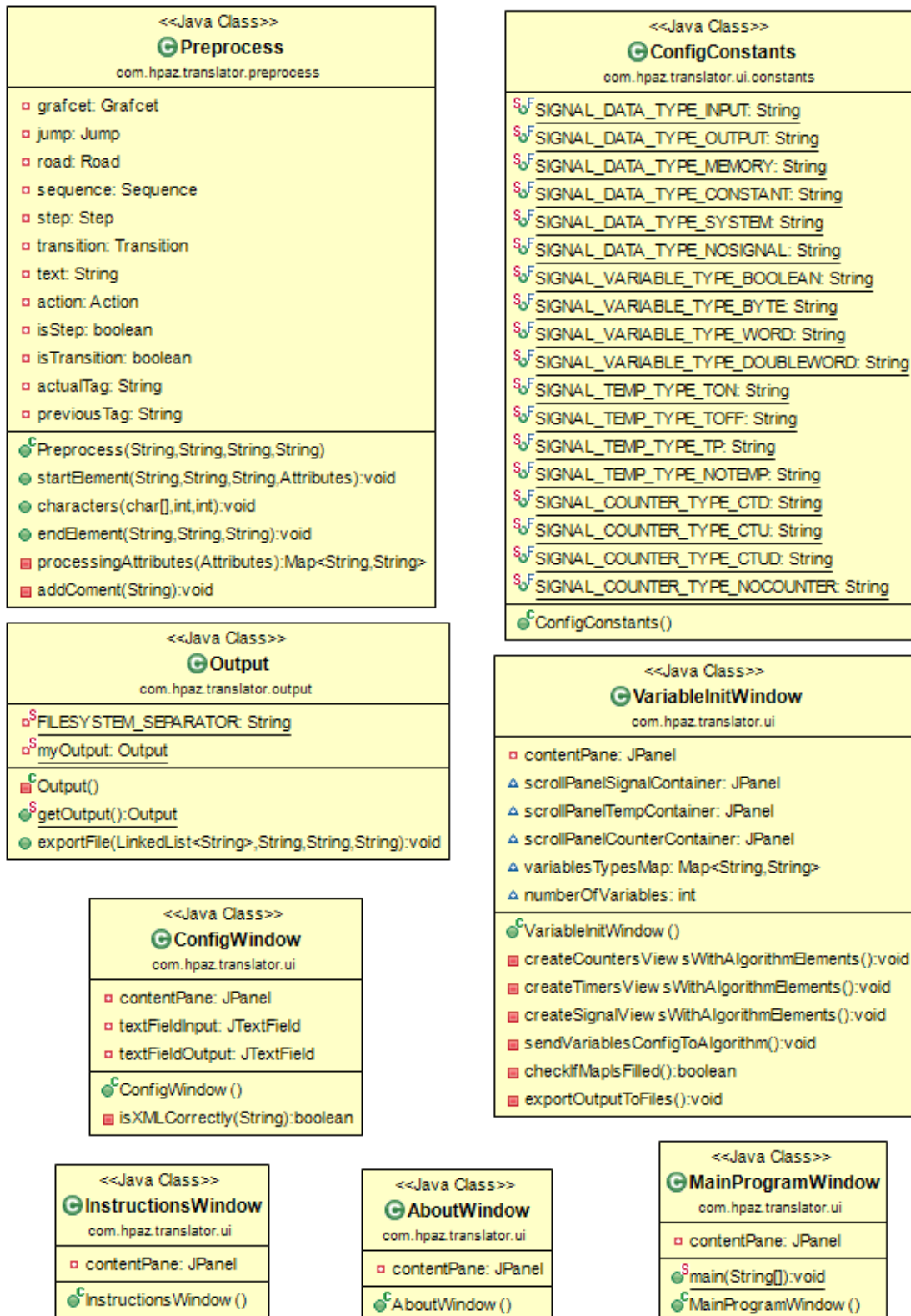
Análisis

Diagrama de Clases

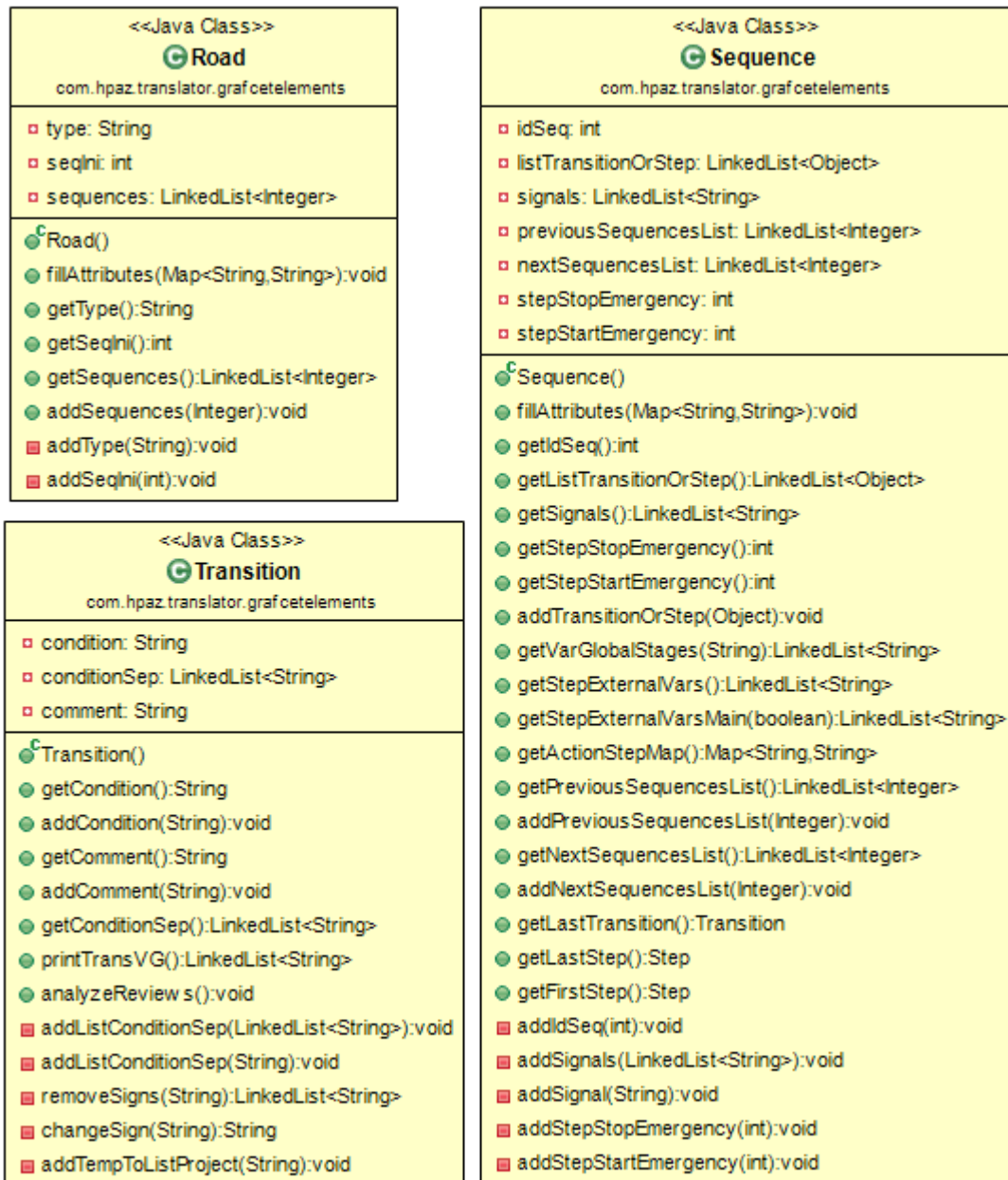
En el siguiente diagrama se pueden observar las conexiones y dependencias relevantes entre cada clase generada para el desarrollo del proyecto.



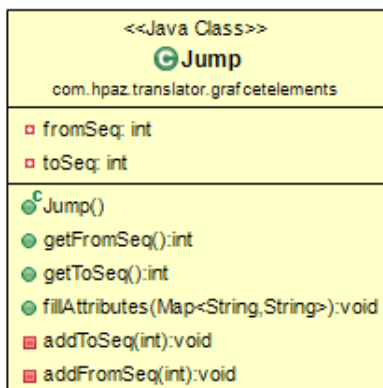
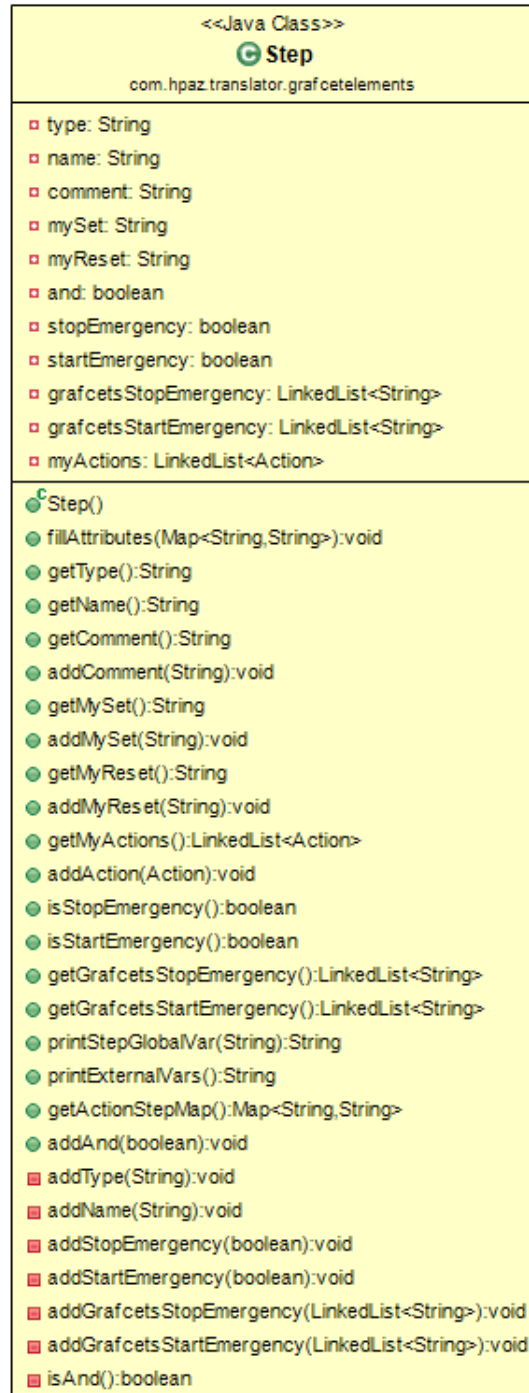
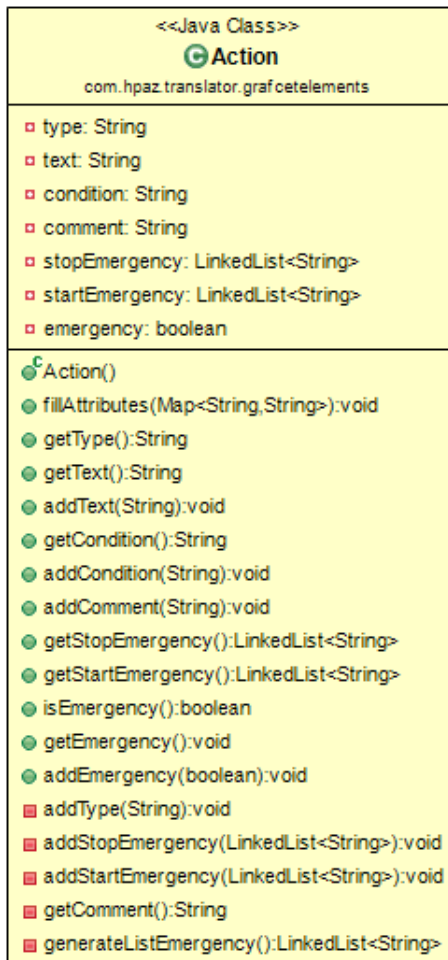
23. Figura: Análisis y Diseño - Análisis - Diagrama de Clases



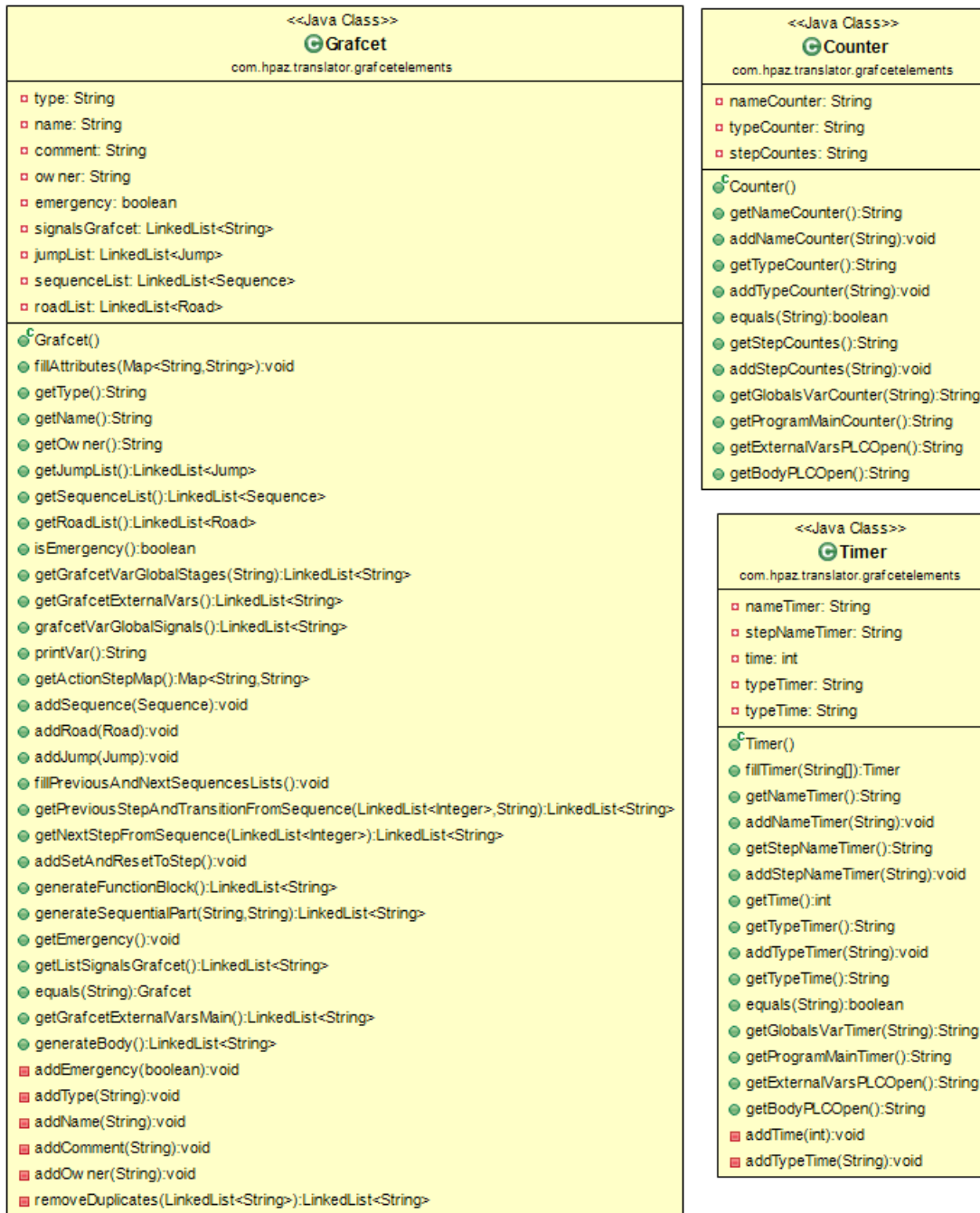
24. Figura: Análisis y Diseño - Análisis - Clases Parte I



25. Figura: Análisis y Diseño - Análisis - Clases Parte II



26. Figura: Análisis y Diseño - Análisis - Clases Parte III



27. Figura: Análisis y Diseño - Análisis - Clases Parte IV

```

<<Java Class>>
Project
com.hpaz.translator.grafcetelements

name: String
language: String
program: String
outputDir: String
listGrafcet: LinkedList<Grafcet>
listTimers: LinkedList<Timer>
listCounters: LinkedList<Counter>
listTimersUI: LinkedList<String>
listCountersUI: LinkedList<String>
list_FE_and_RE: LinkedList<String>
assignments: Map<String,String>
listUI: Map<String,String>
actionStepMap: Map<String,String>
listEmergencyStop: LinkedList<String>
listEmergencyStart: LinkedList<String>
stepStopEmergency: String
stepStartEmergency: String
signalsProject: LinkedList<String>
project: Project

Project()
getProgram():String
getProject():Project
addOutputPath(String):void
addName(String):void
addLanguage(String):void
addProgram(String):void
addGrafcet(Grafcet):void
getListTimers():LinkedList<Timer>
addTimer(Timer):void
getListCounters():LinkedList<Counter>
addCounter(Counter):void
getListTimersUI():LinkedList<String>
getListCountersUI():LinkedList<String>
addStepStopEmergency(String):void
addStepStartEmergency(String):void
addListEmergencyStop(LinkedList<String>):void
addListEmergencyStart(LinkedList<String>):void
addProjectVariablesFromUs erInterface(Map<String,String>):void
generateActionStepMap(Map<String,String>):void
print():void
getSignalsProject():LinkedList<String>
equalsTimer(String):int
equalsCount(String):int
getList_FE_and_RE():LinkedList<String>
add_FE_and_RE(String):void
addAssignments(String,String):void
getName():String
getListG():LinkedList<Grafcet>
getListEmergencyStop():LinkedList<String>
getListEmergencyStart():LinkedList<String>
getStepStopEmergency():String
getStepStartEmergency():String
getListUI():Map<String,String>
getActionStepMap():Map<String,String>
addActionStepMap(String,String):void
addListTimers(Timer):void
addListTimersUI(String):void
addListCounters(Counter):void
addListCountersUI(String):void
getAssignments():Map<String,String>
generateGlobalVars():LinkedList<String>
generateVarGlobalP LCOpen():LinkedList<String>
addListUI(Map<String,String>):void
getProgramTSXMicroCP():LinkedList<String>
getProgramTSXMicroSP():LinkedList<String>
generateProgramMain():LinkedList<String>
compareStartAndStopLists():boolean
generateListEmergency(MainP LCOpen(LinkedList<String>,LinkedList<String>,boolean,String,String):LinkedList<String>
getProgramMain(LinkedList<String>,LinkedList<String>):LinkedList<String>
getGlobalVars(LinkedList<String>):LinkedList<String>
removeDuplicates(LinkedList<String>):LinkedList<String>
generateSignalsProject():void
generateBodyMain(LinkedList<String>):LinkedList<String>
generatePOUMain():LinkedList<String>
generatePousP LCOpen():LinkedList<String>

```

28. Figura: Análisis y Diseño - Análisis - Clases Parte V

Diseño

Diagramas de Secuencias

Se han realizado diagramas de secuencias del proceso de extracción de datos del fichero XML, almacenaje en las distintas clases, así como los procesos de exportación de cada una de las distintas opciones de compatibilidad. Los diagramas de secuencia se encuentran en el *Anexo II* que se puede encontrar en la parte final de esta documentación.

Interfaz

A continuación se explica cada una de las interfaces que se han desarrollado.

- Menú o ventana principal.

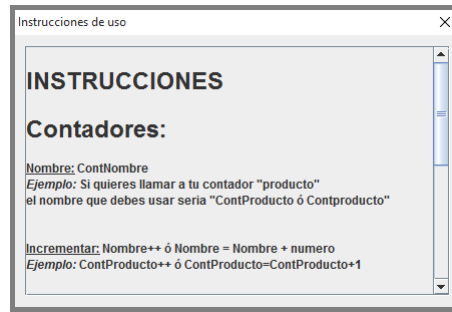
Se planteó la opción de hacer una barra de menú en la parte superior de la pantalla, pero por comodidad para el usuario se decidió colocar una pantalla con botones. Se ha colocado un logo que identifique a la aplicación y se le ha dado una apariencia amistosa que de la bienvenida al usuario, en la cual tendrá la opción de leer las instrucciones de la aplicación o iniciar el traductor directamente.



29. Figura: Análisis y Diseño - Diseño - Interfaz Menú

- Instrucciones de uso.

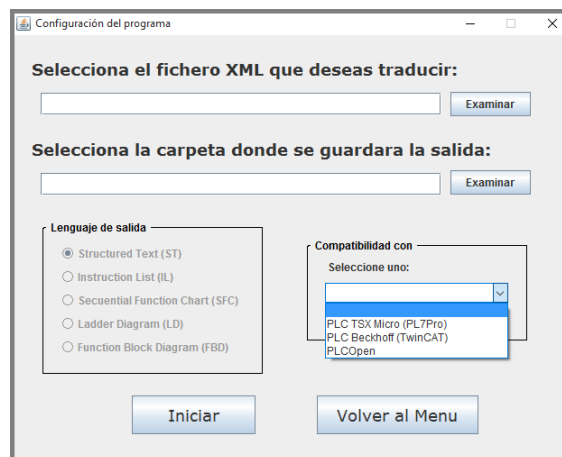
La pantalla de instrucciones se ha creado con la finalidad no solo de asesorar a usuarios primerizos, si no también para que ayude a solucionar dudas en posibles errores de funcionamiento, debido al mal uso de la aplicación. Se ha usado un texto plano con tamaño intermedio, subrayados, espaciados e incluyendo ejemplos de su sintaxis para que el usuario comprenda fácilmente. Esta pantalla tiene barras deslizadoras tanto horizontales como verticales para facilitar la lectura de los textos largos.



30. Figura: Análisis y Diseño - Diseño - Interfaz Instrucciones

- Configuración del Traductor.

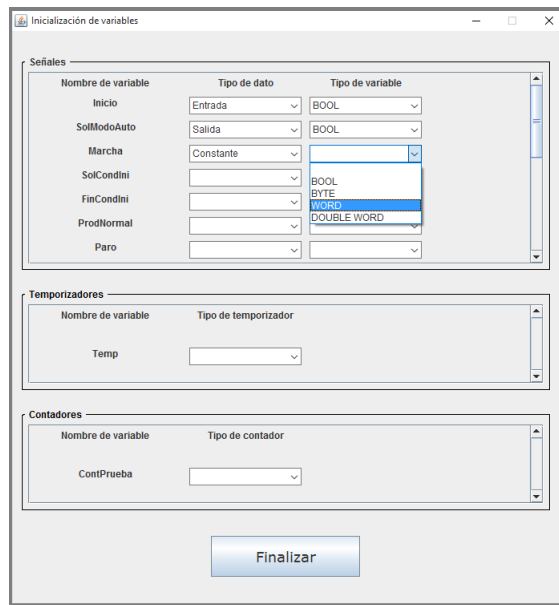
La pantalla de configuración, a la vez de permitir al usuario seleccionar de manera fácil los ficheros y directorios que desee use el traductor, limita las alternativas para evitar errores de edición y sintaxis a la hora de especificar nombres de programas o lenguajes. Esto permite al usuario ver toda la información y seleccionar fácilmente la opción que desee ejecutar.



31. Figura: Análisis y Diseño - Diseño - Interfaz Configuración

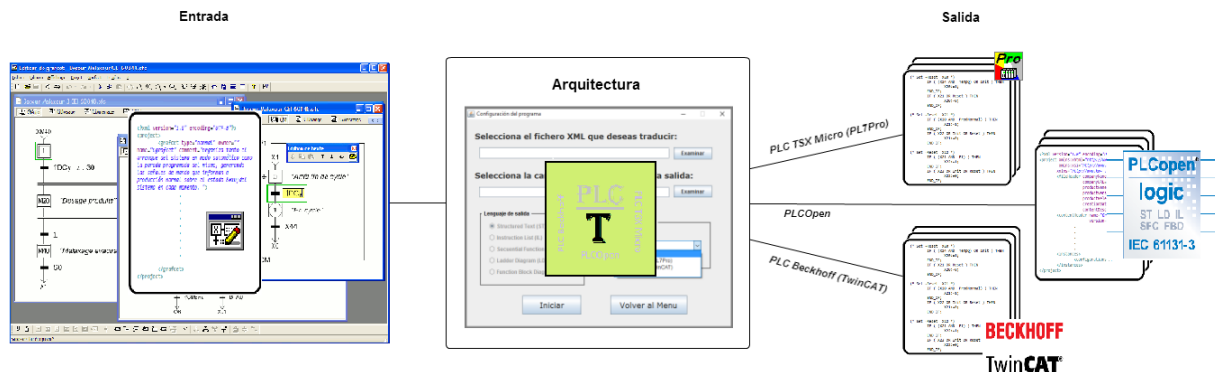
- Configuración de variables.

La pantalla de configuración de variables se ha creado para facilitarle trabajo futuro al usuario, esta ventana se usará dependiendo de la opción de compatibilidad que el usuario seleccione en la configuración del traductor, ya que no todos los software de compatibilidad permiten inicializar las variables mediante código, por ejemplo el PLC TSX Micro (PL7Pro) tiene su propio menú en el cual se indican los tipos de variables, así como los tipos de datos. Sin embargo para los software que si aceptan la inicialización mediante código, la pantalla de configuración de variables nos facilita el tener que hacerlo manualmente, simplemente seleccionando un tipo de variable y un tipo de dato.



32. Figura: Análisis y Diseño - Diseño - Interfaz Variables Detectadas

Arquitectura



33. Figura: Diseño - Arquitectura

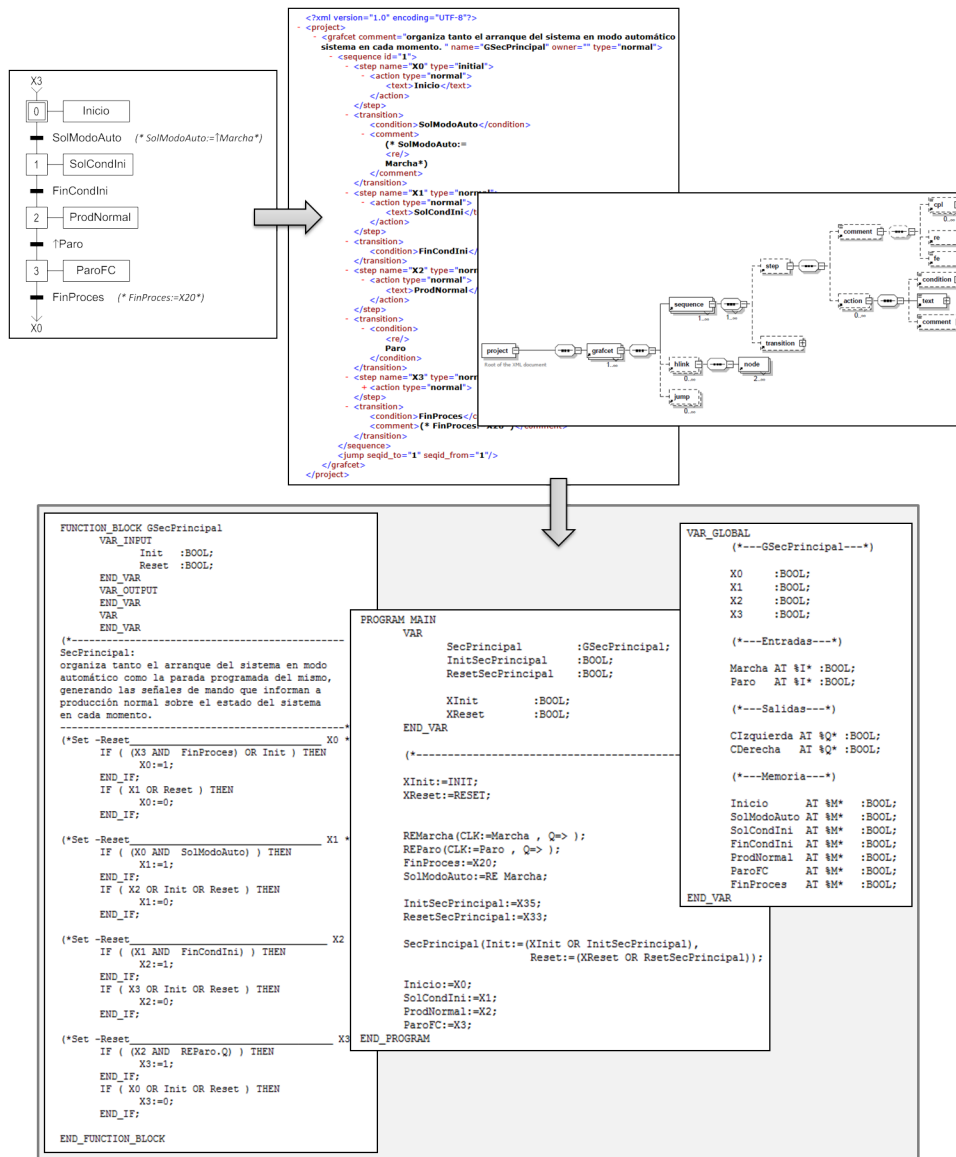
Al tener una arquitectura local, esta sería únicamente la aplicación desarrollada, sin embargo en la imagen se reflejan las entradas y las salidas para ayudar a la comprensión de lo que realiza la misma.

Entrada: tendremos un único fichero XML con un formato exportado por la aplicación de diseño SFCEdit.

Salidas: la aplicación puede contener distintas salidas dependiendo de la opción que seleccione el usuario en la compatibilidad.

Desarrollo

El software desarrollado recibe un fichero XML, generado a partir de un Grafset de diseño creado en SFCEdit (ver Figura 34); e información adicional proporcionada por el usuario. A partir de estas entradas se genera automáticamente el código necesario para ejecutar en un PLC. Para la realización de este proyecto se requirió una investigación y análisis previo ya que el campo a tratar es amplio y existen muchos detalles a tener en cuenta.



34. Figura: Desarrollo - Traductor

Inicialmente se pretendía hacer la transformación de los XML a través de las hojas de estilo XSLT, se realizaron algunas pruebas, pero al ser una plantilla basada en reglas solo permitía leer la información y transformarla al momento, según lo indicaban las reglas descritas, sin embargo para realizar la traducción a un lenguaje específico es necesario tener los datos y recorrer algunos de ellos más de una

vez o simplemente guardar datos para usarlos posteriormente, por lo que usando XSLT nos veríamos en la obligación de recorrer el fichero XML varias veces y así tal vez podríamos obtener la información necesaria con el formato indicado. Otra razón por la que se decidió no usar XSLT es la falta de experiencia en dicho lenguaje. Por lo que se planteo la posibilidad de usar el lenguaje Java, ya que este permite guardar la información y clasificarla, recorriendo el fichero XML una única vez y luego usar los datos de las clases para generar los ficheros que exportará la aplicación. Se discutió el tema con la directora del proyecto, la cual estuvo de acuerdo con la decisión y fue así cuando se comenzó a buscar procesadores de XML para Java.

Entre los procesadores de XML para Java se encontraron dos; DON que genera un árbol de objetos con sus dependencias en memoria, permitiendo de esta manera acceder a cualquier posición las veces que se quiera y SAX que lee el fichero XML secuencialmente usando un proceso de métodos callback cada vez que se detectan determinado patrones en el documento, de esta manera se puede encontrar la información deseada y almacenarla donde queramos.

La diferencia entre estos dos tipos de procesamiento es que DON para XML muy grandes, consumiría mucha memoria, sin embargo SAX no almacena información en memoria.

Como los ficheros XML con los que trabajaremos podrían ser desde muy pequeños hasta muy grandes se decide usar SAX y guardar todos los datos en clases creadas únicamente a modo de almacenamiento temporal para ordenar la información de la manera requerida.

Una vez decida la forma de obtener la información del fichero XML, se comienza a estudiar la herramienta SFCedit y todas las posibles variables que se pueden obtener de cada Grafcet generado y exportado, de esta manera quedan aclarados los datos con los que el traductor tendrá que trabajar. Por otra parte ya que el programa solo trabaja con fichero exportados de SFCedit era necesario validar el fichero XML para que contenga el formato esperado, para esto se uso el validado que proporciona SAX y se creo un fichero XSD para usarlo como plantilla.

Una vez que se tiene la información con la que se podrá trabajar, es necesario conocer los modelos de PLCOpen por lo que se realiza una investigación a fondo sobre los distintos modelos que proporciona PLCOpen basados en el estándar de la norma IEC 61131- 3. Para comprender mejor la composición de los formatos que proporciona PLCOpen ,se realiza un diagrama conectando cada una de las variables y métodos de los cuales se compone cada fichero, se clasificó cada tipo de dato , cuales de estos son opcionales y cuales obligatorios.(Figura 35)



35. Figura: Desarrollo - Diagrama del Formato de PLCOpen

Una vez analizados los formatos de salida y de entrada que debe tener la aplicación, se llega a la conclusión de que el formato de entrada no nos suministra las variables necesarias para todos los lenguajes que proporciona la norma IEC 61131- 3, por lo que se procede a realizar el software con el lenguaje literal estructurado (ST) del cual si tendremos la información necesaria. Para que la herramienta desarrollada tenga más atractivo se hace compatible con dos softwares para ejecutar PLC además de PLCOpen, por lo que el siguiente paso continuo con más investigación, para determinar que ficheros deben exportar dependiendo del software de compatibilidad que seleccione el usuario.

Se estudiaron dos software de compatibilidad distintos: PLC TSX Micro (PL7Pro), PLC Beckhoff (TwinCAT) y para el formato de PLCOpen también se tomo como modelo los ficheros exportados por la herramienta PC WORX.

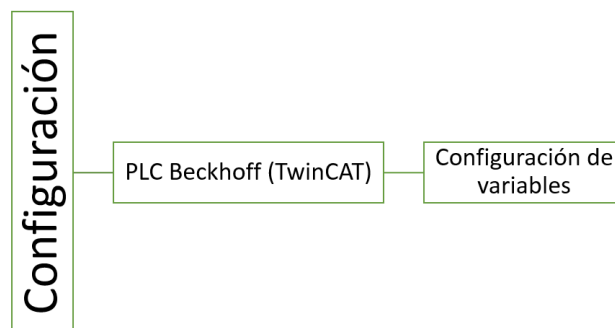
Finalmente para desarrollar los ficheros con dicha compatibilidad se estudio la información requerida por cada software y su formato, y aunque la información sea la misma para cada uno de ellos, sus diferencias radican en la forma de mostrarla y en las reglas sintácticas que tiene cada software en específico.

Verificación y Evaluación

Pruebas del Software

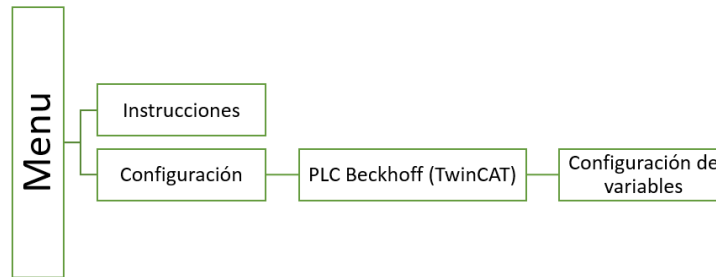
Pruebas de Integración

En la herramienta desarrollada, las pruebas de integración se han ido realizando de forma descendente, comenzando por el modulo raíz e ir integrando cada uno de los distintos módulos que le siguen.



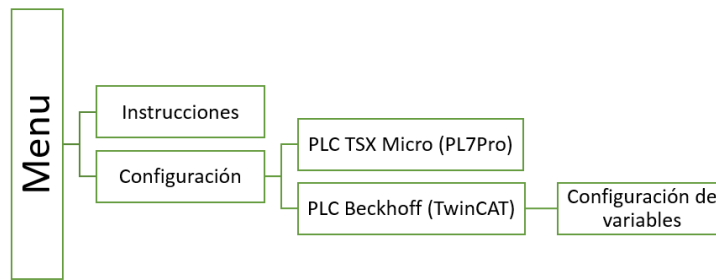
36. Figura: Pruebas del Software - Pruebas de Integración Modulo 1

- El primer módulo a desarrollar fue el de configuración seguido del apartado de PLC Beckhoff (TwinCAT) y la configuración de variables. Se comprueba que las entradas se analicen correctamente y que las salidas sean las esperadas para la opción seleccionada.



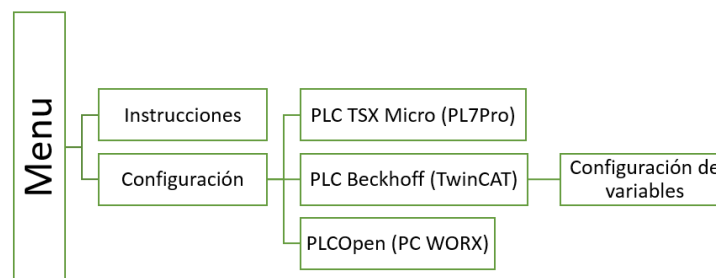
37. Figura: Pruebas del Software - Pruebas de Integración Modulo 2

- Al comprobar el primer módulo, se añade un menú inicial en el cual se obtendrán dos opciones leer las instrucciones de uso e ir a la pantalla de configuración, por lo que el segundo módulo probado fue que las instrucciones de uso se lean correctamente.



38. Figura: Pruebas del Software - Pruebas de Integración Modulo 3

- El tercer módulo consiste en comprobar que en el caso de seleccionar la opción PLC TSX Micro (PL7Pro) los ficheros de salida sean los correcto y sigan funcionando correctamente todos los módulos ya probados.



39. Figura: Pruebas del Software - Pruebas de Integración Modulo 4

- El cuarto y último módulo en probar ha sido el de la opción de PLCOpen, en este caso se comprueba que los ficheros exportado sean los esperados tanto en este módulo como en todos los anteriores.

Pruebas de Aceptación

- Fase Alfa

El desarrollador ha ido probando el programa constantemente, cada vez que se realizaban tanto grandes como pequeños cambios para verificar que el funcionamiento de la aplicación sea la correcta.

- Fase Beta I

En la fase Beta I se ha dado la aplicación al director del proyecto para que realice pruebas, tomando en cuenta que el director tiene conocimientos avanzados del funcionamiento y de los resultados que deben obtenerse.

- Fase Beta II

Se deja la aplicación a un usuario que desconoce del tema y funcionamiento, para saber si es capaz de usarla y entender lo que hace. De esta forma podremos obtener información de mejoras que desconocemos ya que hasta el momento cada usuario que ha probado la aplicación ha tenido conocimientos previos de la misma. Este usuario nos podrá dar observaciones como, si la aplicación es fácil de usar, si la información dada para su uso hace que sea capaz de usarla correctamente, entre otras.

- Fase Beta III

Una vez mejoradas las sugerencias de la Fase Beta II se vuelve a dar la aplicación a distintos usuarios, esta vez la corrección de errores es mínima, por lo que se decide que no es necesario, realizar una siguiente fase de prueba.

Documentación de Pruebas

Descripción	Resultado Esperado	Resultado Obtenido
Pulsar el botón «Iniciar» de pantalla de configuración del programa sin haber rellenado todas las opciones.	Mensaje de error y te permite continuar rellanando los campos.	Mensaje de error y te permite continuar rellanando los campos.
Seleccionar un fichero XML que no exista.	Mensaje de error y te permite modificar los campos.	Mensaje de error y te permite modificar los campos.
Indicar el nombre de un fichero que no tenga extensión XML.	Mensaje de error y te permite modificar los campos.	Mensaje de error y te permite modificar los campos.
Seleccionar un fichero XML que no tenga el formato adecuado	Mensaje de error y te permite modificar los campos.	Mensaje de error y te permite modificar los campos.
Seleccionar un directorio de salida que no exista.	Mensaje de error y te permite modificar los campos.	Mensaje de error y te permite modificar los campos.

22. Cuadro: Documentación de Pruebas I

Descripción	Resultado Esperado	Resultado Obtenido
Pulsar el botón «Iniciar» de pantalla de configuración del programa con todas las opciones correctas con compatibilidad PLC TSX Micro.	Mensaje de que se ha realizado la traducción correctamente y se han generado los ficheros en el directorio indicado.	Mensaje de que se ha realizado la traducción correctamente y se han generado los ficheros en el directorio indicado.
Pulsar el botón «Iniciar» de pantalla de configuración del programa con todas las opciones correctas con compatibilidad PLC Beckhoff.	Se inicia la pantalla de gestión de variables detectadas por la aplicación.	Se inicia la pantalla de gestión de variables detectadas por la aplicación.
Pulsar el botón «Iniciar» de pantalla de configuración del programa con todas las opciones correctas con compatibilidad PLCOpen.	Se inicia la pantalla de gestión de variables detectadas por la aplicación.	Se inicia la pantalla de gestión de variables detectadas por la aplicación.
En la gestión de variables que se pulse «Finalizar» sin haber rellenado todos los campos.	Mensaje de error y permite continuar rellenando los campos.	Mensaje de error y permite continuar rellenando los campos.
En la gestión de variables que se pulse «Finalizar» habiendo rellenado todos los campos.	Mensaje de que se ha realizado la traducción correctamente y se han generado los ficheros en el directorio indicado.	Mensaje de que se ha realizado la traducción correctamente y se han generado los ficheros en el directorio indicado.

23. Cuadro: Documentación de Pruebas II

Conclusiones y Trabajo Futuro

En este proyecto se ha desarrollado una herramienta que permite realizar la traducción de diseños de sistemas de control a la programación siguiendo el estándar IEC 61131- 3. Esta herramienta, que a partir de los diseños proporciona los programas para los PLCs, permitirá trabajar con un mayor nivel de abstracción en el desarrollo de los sistemas de control.

Para el desarrollo de esta herramienta se ha realizado un análisis de los diseños en el lenguaje de modelado Grafset y la información que la herramienta SFCEdit proporciona de los mismos en el formato XML.

Se ha realizado un análisis de la norma IEC 61131-3 y de los modelos XML de la norma proporcionados por PLCOpen, centrando el desarrollo en el lenguaje de programación estructurado.

Por último, se ha realizado el análisis de los procedimientos de transformación de los diseños a los programas, identificando, por una parte, las reglas que deben cumplir los diseños para poder realizar su traducción, y por otra parte, los elementos y unidades de programación que han sido necesarias implementar.

Tras el análisis se han establecido los requisitos de la herramienta de traducción, se ha realizado la implementación y pruebas las pruebas correspondientes.

La herramienta desarrollada es un traductor, que recibe un fichero XML, lo procesa y devuelve el código en lenguaje literal estructurado (ST). Se ha preparado para que pueda realizarse la traducción en compatibilidad con dos software de programación de PLC y el formato de PLCOpen.

Para el desarrollo del traductor se planteó realizar las transformaciones con hojas de estilo XSLT. El XSLT, como se ha explicado anteriormente, es una plantilla que permite obtener un XML con un formato prediseñado en base a reglas. Esta opción fue descartada por no ajustarse a las necesidades del desarrollo, decidiéndose realizar el proyecto en el lenguaje de programación Java, que además de ser multiplataforma, permite hacer una interfaz cómoda y fácil de usar.

Una vez realizada la investigación previa y conociendo las referencias necesarias para generar los ficheros XML de PLCOpen, se procede a estudiar los datos que proporciona el XML exportado por la herramienta de diseño SFCEdit, llegando a la conclusión de que la información con la que se trabajará no dispone de los requisitos necesarios para los lenguajes gráficos que especifica el estándar, por esta razón se comienza a desarrollar el proyecto con la transformación a uno de los lenguajes de texto plano (ST) y se añade como dificultad el que sea compatible con distintos softwares, dejando abierta la posibilidad de ampliar el proyecto para añadir más adelante la traducción a los lenguajes restantes.

Con respecto a la planificación, se ha añadido holgura en la mayoría de los casos de investigación, debido a la falta de experiencia con proyectos de este ámbito. En la parte del desarrollo, la estimación ha sido más ajustada, debido a que la aplicación se ha desarrollado partiendo de las especificaciones necesarias para cada software de compatibilidad y la diferencia de sintaxis entre ellos ha implicado rehacer y modificar métodos ya existentes.

El proyecto realizado puede ser de gran utilidad en el campo de la industria, ya que no sería necesario tener amplios conocimientos informáticos para realizar la programación de un sistema de control, simplemente podría diseñarse el proyecto con un lenguaje de modelado a través de la herramienta SFCEdit y usar el traductor desarrollado para generar el código a usar en el software seleccionado. Lo que lograría programas mejor estructurados, libres de errores lo que implica reducir el tiempo de codificación y trabajar en un nivel de abstracción más alto.

Durante la realización de este proyecto ha surgido un retraso debido a pérdida de software, a pesar de ser uno de los riesgos previstos, nunca esperas que suceda, esto supuso tensión emocional y gasto económico ya que hubo que sustituir el disco duro, sin embargo esto no generó ninguna pérdida del contenido del proyecto. Debido a que toda la información de documentación estaba guardada en la

nube y el código en un servidor independiente con copia de seguridad, realizadas en distintos equipos y en la nube. Por otra parte también surgían problemas a medida que se iba desarrollando la aplicación; generados por las limitaciones que imponían los softwares de compatibilidad, debido a las distintas maneras de usar cada uno de ellos y la forma en la que exigía expresarse su contenido.

Para finalizar he de decir que el traductor realizado es solo una pequeña parte de un gran proyecto que se podría seguir desarrollando en el futuro, así como también se podría añadir mejoras, entre ellas; la opción a que acepte ficheros de distintos softwares, mejorar la distribución y comentarios del contenido que se exporta para que sea más intuitivo al usuario, que permita traducir a los cuatro lenguajes que faltan para completar los que establece la Norma IEC 61131-3, entre otras.

Bibliografía

Alex Guerra (Xela) (2008).

“Java y XML: SAX” en La Tasca de Xela, 8 de julio.

Implementacion SAX

[Consulta: 12 de junio del 2015]

Automática e Instrumentación (2011).

Colaboración entre Automation ML y PLCopen.

Documentación Automation ML y PLCopen

[Consulta: 24 de agosto del 2015]

Facultad de Ingeniería. Universidad de la República. (Uruguay)

Unidad Didáctica 4: Statechart

Documentación Statechart

[Consulta: 5 de junio del 2015]

dtrinf (2012)

Como escribir eñes “Ñ” y acentos en Java

Documentacion Java

[Consulta: 5 de septiembre del 2015]

Ingénierie en automatisme et informatique industrielle.

Don't waste more time to draw your grafquets!.

Descarga y documentacion de SFCedit.

[Consulta: 4 de junio del 2015]

Mercedes Granda (2012/2013).

Redes De Petri: Definición, Formalización Y Ejecución.

Master en Computación. Santander: Universidad de Cantabria.

Documentacion Redes Petri

[Consulta: 4 de junio del 2015]

Mkyong (2013).

How To Read XML File In Java – (DON Parser).

Documentación para parsear XML en Java

[Consulta: 12 de junio del 2015]

Pablo San Segundo. Universidad Politécnica de Madrid UPM

Introducción al modelado GRAFCET

Documentación GRAFCET

[Consulta: 5 de junio del 2015]

Rafa Morales (2014)

Qué son los esquemas XSD

Documentación Esquema XSD

[Consulta: 30 de agosto del 2015]

Raspberry Shop.

Raspberry Pi.

Especificaciones Raspberry Pi

[Consulta: 14 de agosto del 2015]

Wikipedia®.

Controlador lógico programable.

Documentación PLC

[Consulta: 14 de agosto del 2015]

Wikipedia®.

IEC 61131-3.

Documentación IEC 61131-3

[Consulta: 14 de junio del 2015]

Wikipedia®.

Git.

Documentación Git

[Consulta: 24 de agosto del 2015]

Wikipedia®.

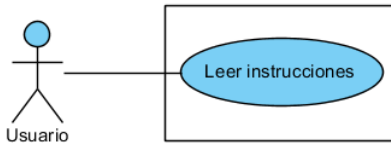
Interfaz de programación de aplicaciones

Documentación API

[Consulta: 30 de agosto del 2015]

ANEXO I.- Casos de Uso Extendidos

Caso de Uso Leer Instrucciones



40. Figura: Caso de Uso Leer Instrucciones

Descripción

Pantalla con las instrucciones relacionadas con el uso de la aplicación, condiciones y requisitos.

Actores

Usuario.

Precondiciones

Ninguna.

Requisitos No Funcionales

Ninguno.

Flujo de eventos

- 1 Pulsar en el menú el botón de «Leer Instrucciones».
- 2 Aparece una ventana con instrucciones de uso de la aplicación. (Figura: Casos de Uso Extendidos - Interfaz Leer Instrucciones).
 - 2.1 [Si se pulsa el botón de cerrar la ventana «x»].
 - 2.1.1 Se regresará a la pantalla del menú inicial.(Figura: Casos de Uso Extendidos - Interfaz Menú)

Postcondiciones

El usuario habrá leído las instrucciones de uso de la aplicación y tendrá los conocimientos necesarios para comenzar a usarla de manera correcta.

Imágenes

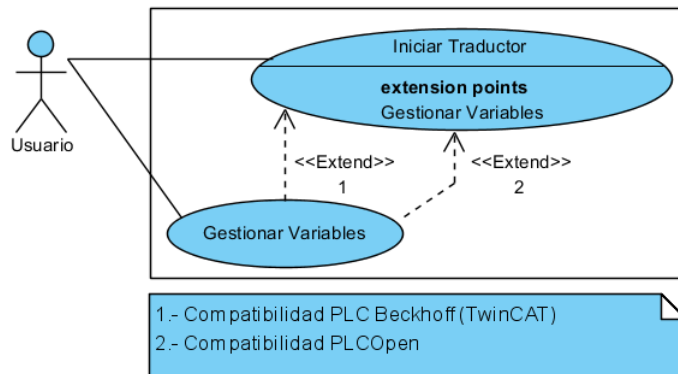


41. Figura: Casos de Uso Extendidos - Interfaz Menú



42. Figura: Casos de Uso Extendidos - Interfaz Leer Instrucciones

Caso de Uso Iniciar Traductor



43. Figura: Caso de Uso Iniciar Traductor

Descripción

Aparecerá una pantalla en la que el usuario podrá configurar el traductor.

Actores

Usuario.

Precondiciones

Haber pulsado el botón «Iniciar Traductor».

Requisitos No Funcionales

Ninguno.

Flujo de eventos

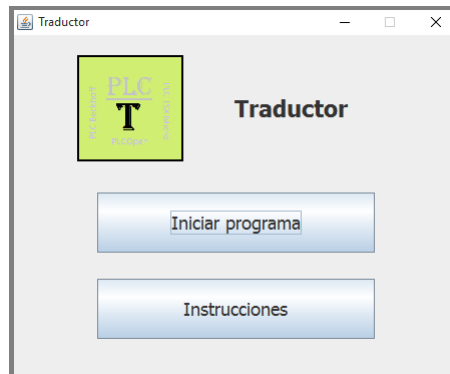
- 1 Pulsar en el menú el botón de «Iniciar Traductor».
- 2 Aparece una ventana con los campos que son necesarios rellenar para poder iniciar la traducción. (Figura: Casos de Uso Extendidos - Interfaz Configuración).
- 3 El usuario pulsa el botón «Examinar» para buscar en el equipo una carpeta en la que se desee guardar los ficheros que generará la aplicación.
- 4 El usuario selecciona un Lenguaje entre las opciones que permiten ser seleccionadas.
- 5 El usuario selecciona el software de compatibilidad entre las opciones de la lista.
 - 6.1 [Si se pulsa el botón de «Iniciar»]
 - 6.1.1 [Si el nombre del fichero XML no se encuentra]

- 6.1.1.1 Aparecerá un mensaje de error y el usuario podrá modificar los datos.(Figura: Casos de Uso Extendidos - Interfaz Error XML no encontrado).
- 6.1.2 [Si el directorio de salida no es correcto]
 - 6.1.2.1 Aparecerá un mensaje de error y el usuario podrá modificar los datos.(Figura: Casos de Uso Extendidos - Interfaz Error Directorio).
- 6.1.3 [Si algún dato no ha sido rellenado]
 - 6.1.3.1 Aparecerá un mensaje de error y el usuario podrá rellenar los datos. (Figura: Casos de Uso Extendidos - Interfaz Error Rellenar Configuración).
- 6.1.4 [Si todos los datos se rellenan correctamente]
 - 6.1.4.1 [Si el fichero XML no contiene el formato necesario]
 - 6.1.4.1.1 Aparecerá un mensaje de error y el usuario podrá modificar los datos. (Figura: Casos de Uso Extendidos - Interfaz Error XML con formato incorrecto).
 - 6.1.4.2 [Si la compatibilidad es PLC TSX Micro (PL7Pro)]
 - 6.1.4.2.1 Aparecerá un mensaje indicando que los ficheros se han generado correctamente y se volverá al menú principal. (Figura: Casos de Uso Extendidos - Interfaz de Finalizado).
 - 6.1.4.3 [Si la compatibilidad es PLC Beckhoff (TwinCAT)]
 - 6.1.4.3.1 EXTEND Gestionar Variables.
 - 6.1.4.4 [Si la compatibilidad es PLCOpen (PC WORX)]
 - 6.1.4.4.1 EXTEND Gestionar Variables.
- 6.2 [Si se pulsa el botón de «Volver al Menú»]
 - 6.2.1 Se regresará a la pantalla del menú inicial.(Figura: Casos de Uso Extendidos - Interfaz Menú).
- 6.3 [Si se pulsa el botón de cancelar «x»]
 - 6.3.1 Se cerrará la aplicación.

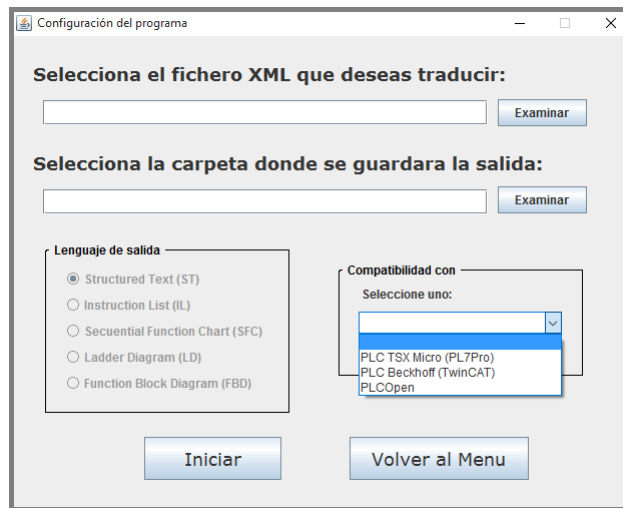
Postcondiciones

El usuario habrá iniciado las funciones del traductor.

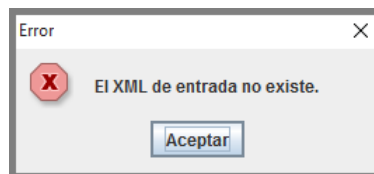
Imágenes



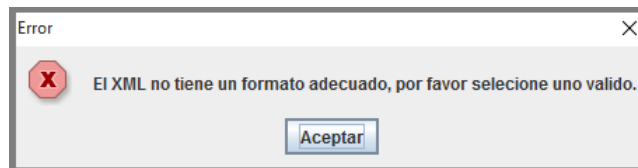
44. Figura: Casos de Uso Extendidos - Interfaz Menú



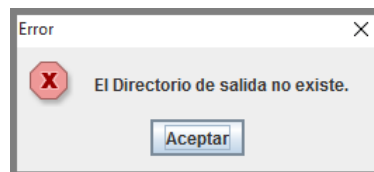
45. Figura: Casos de Uso Extendidos - Interfaz Configuración



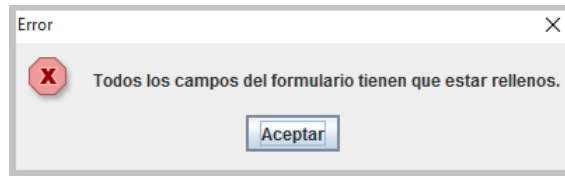
46. Figura: Casos de Uso Extendidos - Interfaz Error XML no encontrado.



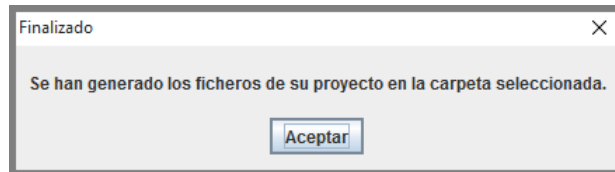
47. Figura: Casos de Uso Extendidos - Interfaz Error XML con formato incorrecto.



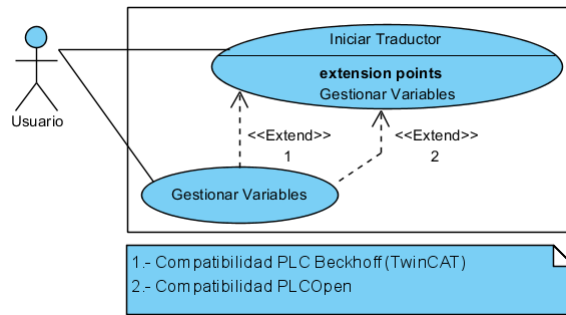
48. Figura: Casos de Uso Extendidos - Interfaz Error Directorio



49. Figura: Casos de Uso Extendidos - Interfaz Error Rellenar Configuración



50. Figura: Casos de Uso Extendidos - Interfaz de Finalizado



51. Figura: Caso de Uso Gestionar Variables

Caso de Uso Gestionar Variables

Descripción

Este subcaso de uso nos ofrece una ventana donde se podrán gestionar las variables detectadas por el traductor.

Actores

Usuario

Precondiciones

Haber pulsado el botón «Finalizar» y que todos los campos en pantalla se hallan relleno correctamente.

Requisitos No Funcionales

Ninguno.

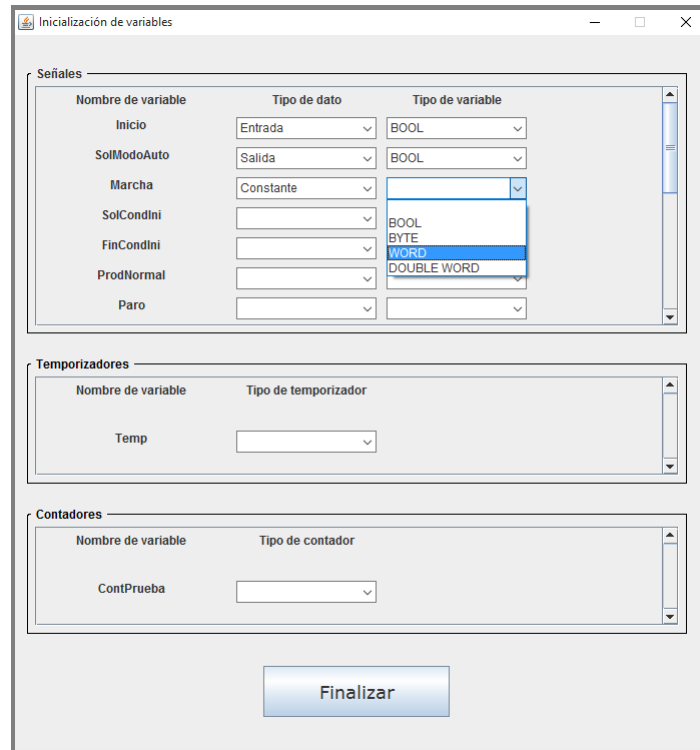
Flujo de eventos

- 1 Aparece una ventana con las variables encontradas por el traductor para que se les asigne su correspondiente tipo. (Figura: Casos de Uso Extendidos - Interfaz de Variables).
 - 1.1 [Si se pulsa el botón «Finalizar»]
 - 1.1.1 [Si se han relleno todos los campos]
 - 1.1.1.1 Aparecerá un mensaje indicando que los ficheros se han generado correctamente y se volverá al menú principal. (Figura: Casos de Uso Extendidos - Interfaz de Finalizado).
 - 1.1.2 [Si No se han relleno todos los campos]
 - 1.1.2.1 Aparecerá un mensaje de error y el usuario podrá rellenar los datos. (Figura: Casos de Uso Extendidos - Interfaz Error Rellenar Variables).
 - 1.2 [Si se pulsa el botón de cerrar la ventana «x»]
 - 1.2.1 Se cerrará la aplicación.

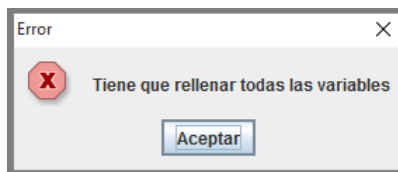
Postcondiciones

El usuario habrá usado el traductor y se habrán generado los ficheros en los casos que se requerían.

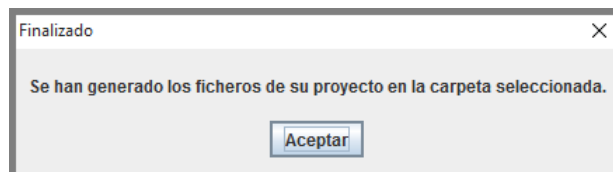
Imágenes



52. Figura: Casos de Uso Extendidos - Interfaz Configuración

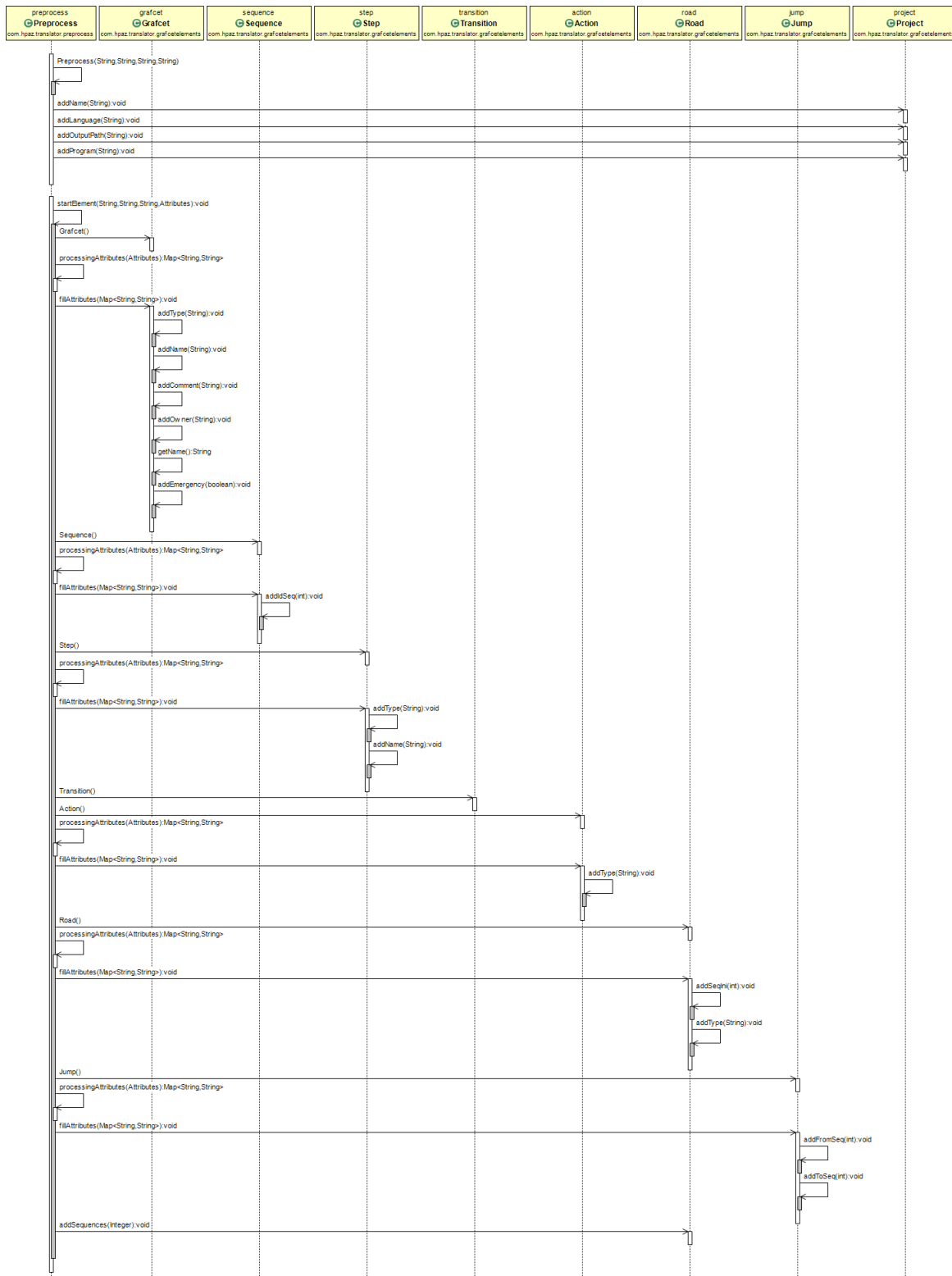


53. Figura: Casos de Uso Extendidos - Interfaz Error Rellenar Variables

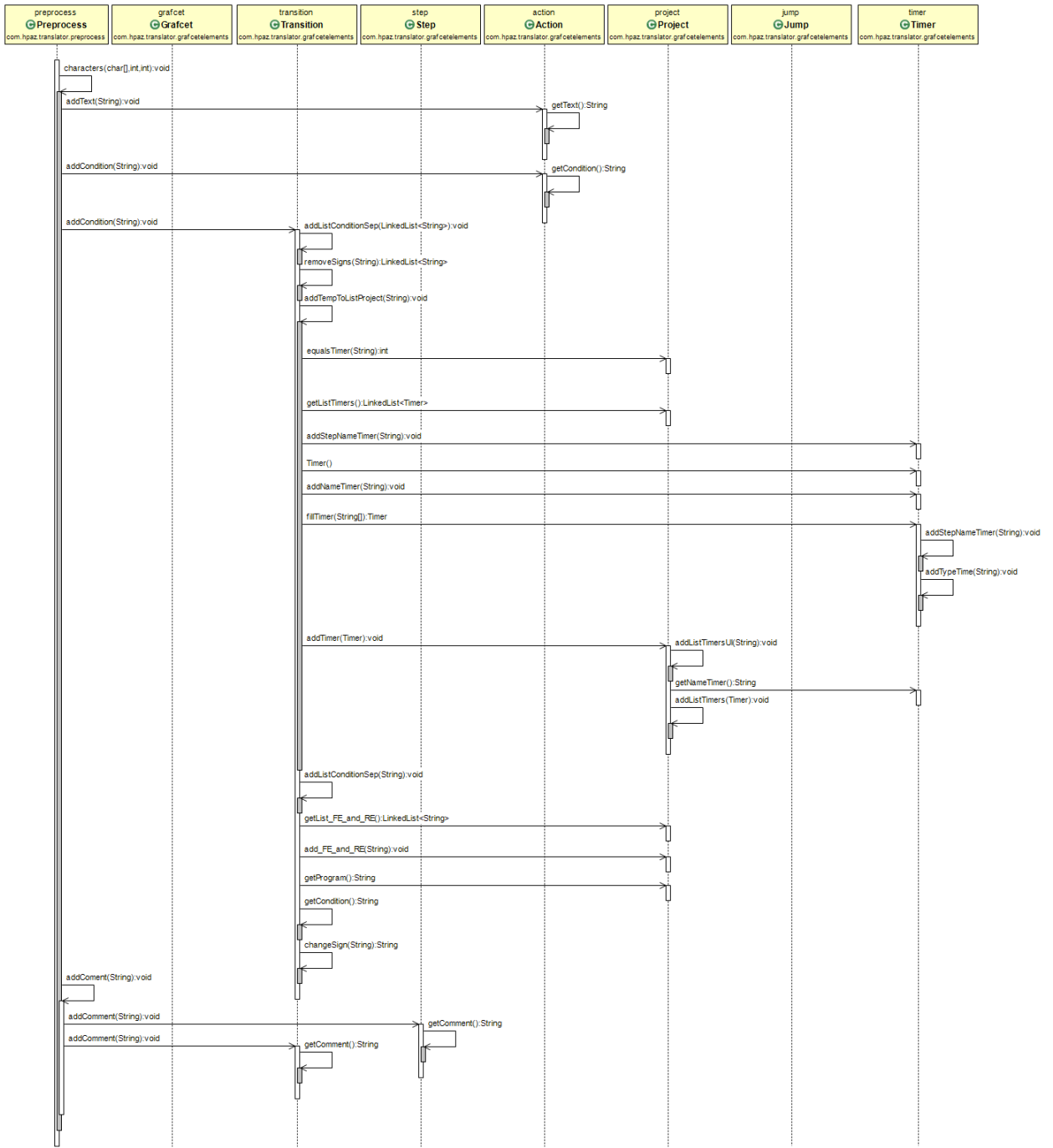


54. Figura: Casos de Uso Extendidos - Interfaz de Finalizado

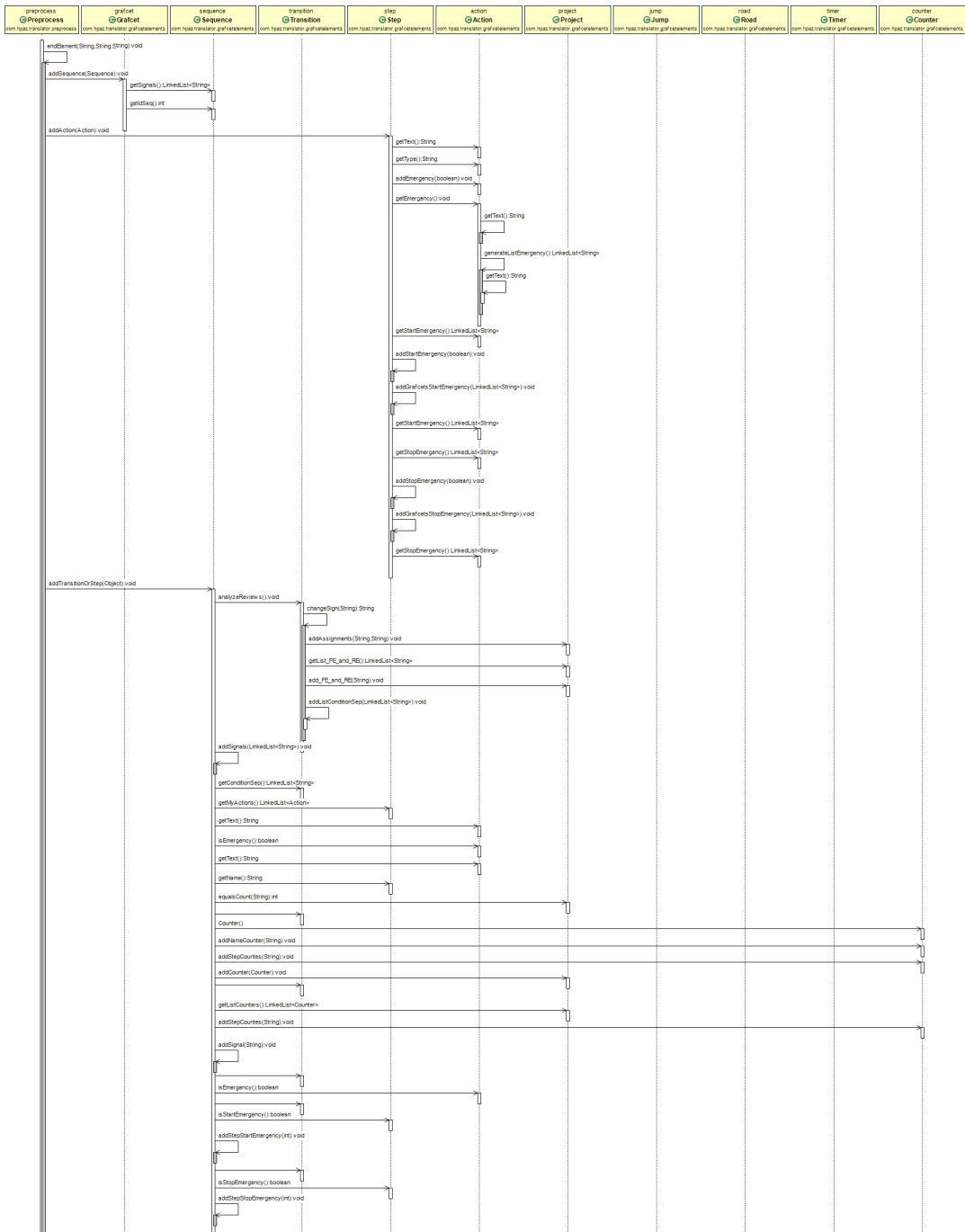
ANEXO II.- Diagramas de Secuencia



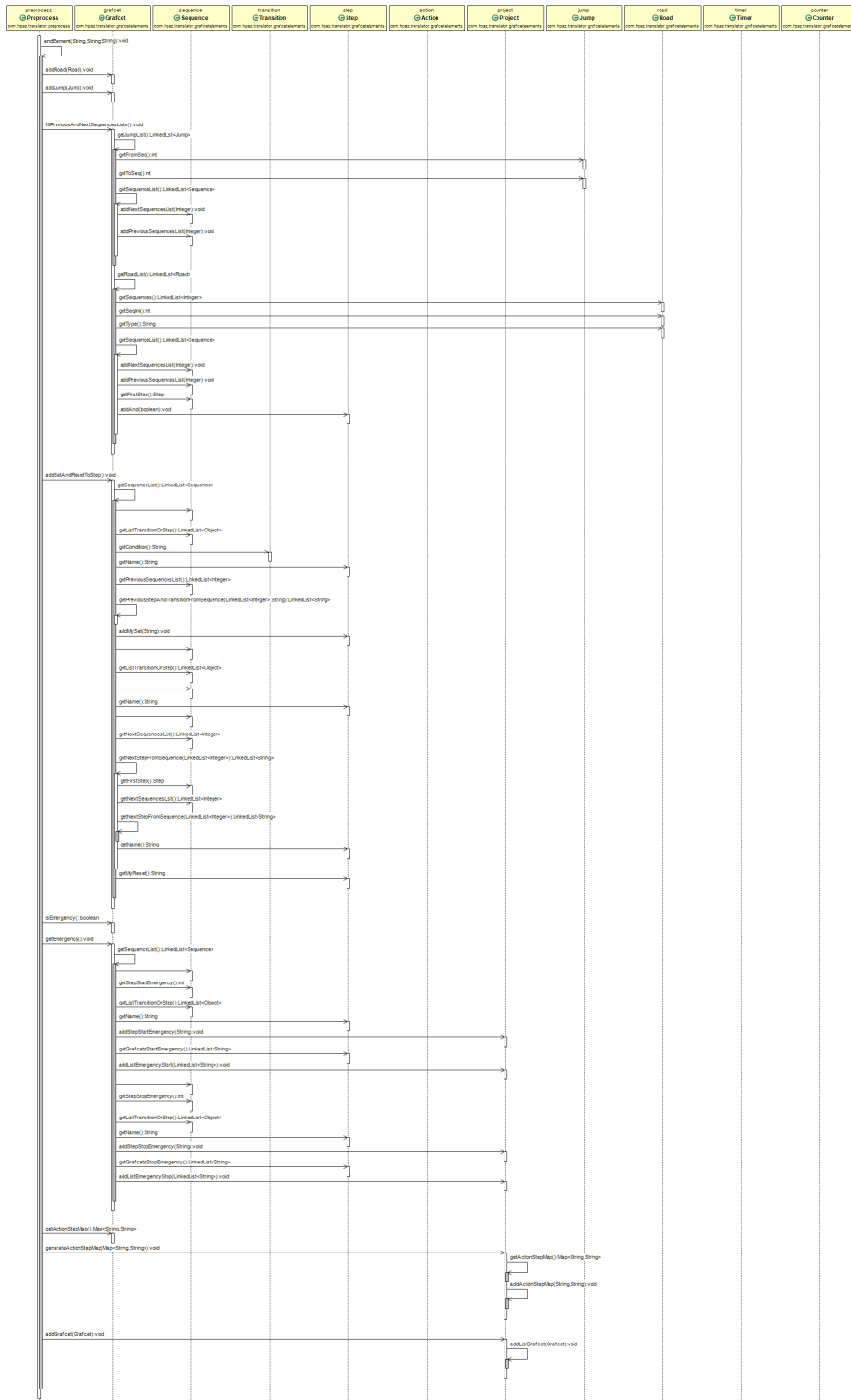
55. Figura: Análisis y Diseño - Diseño - Diagrama de Secuencia PreProceso I



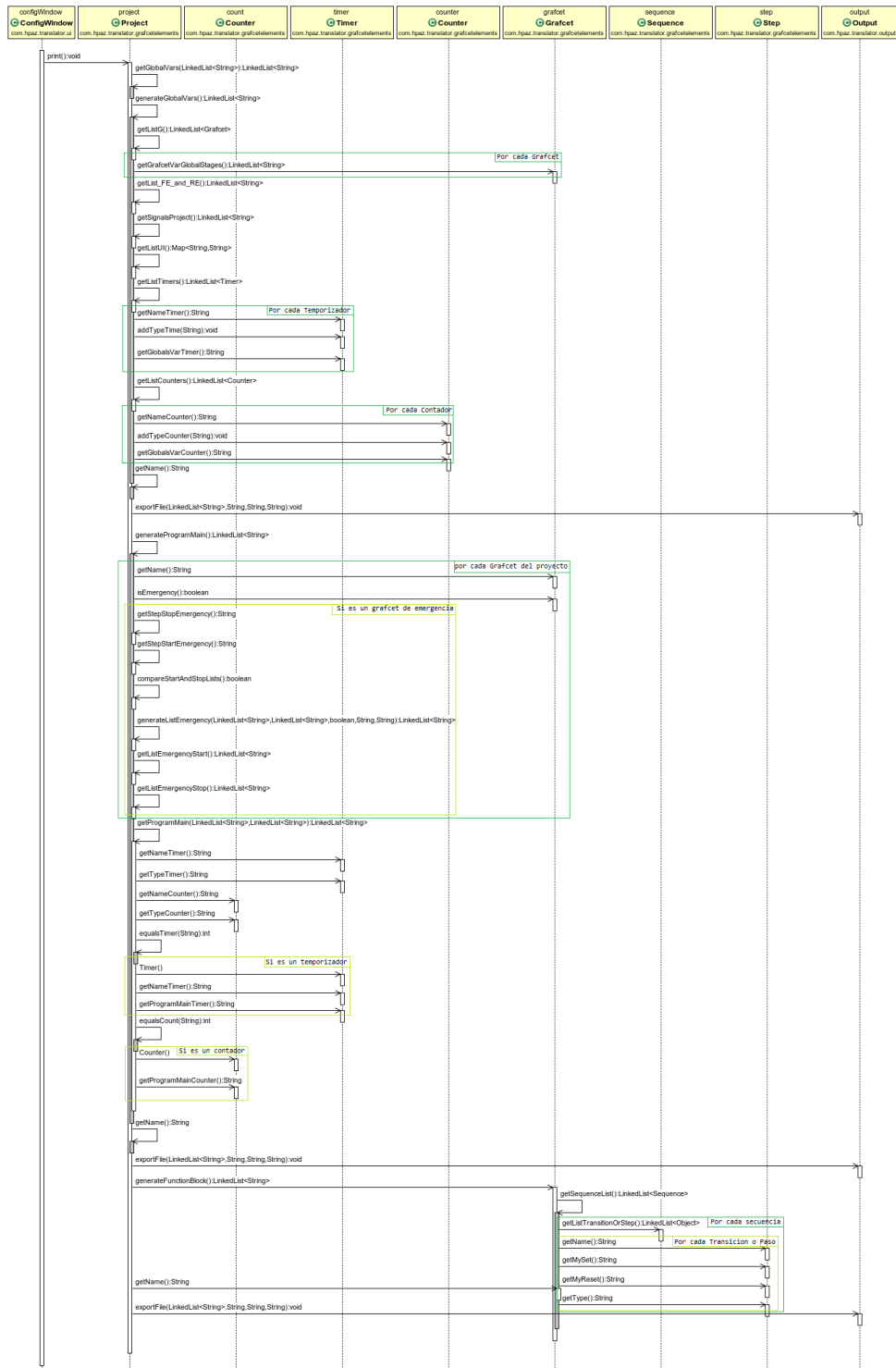
56. Figura: Análisis y Diseño - Diseño - Diagrama de Secuencia PreProceso II



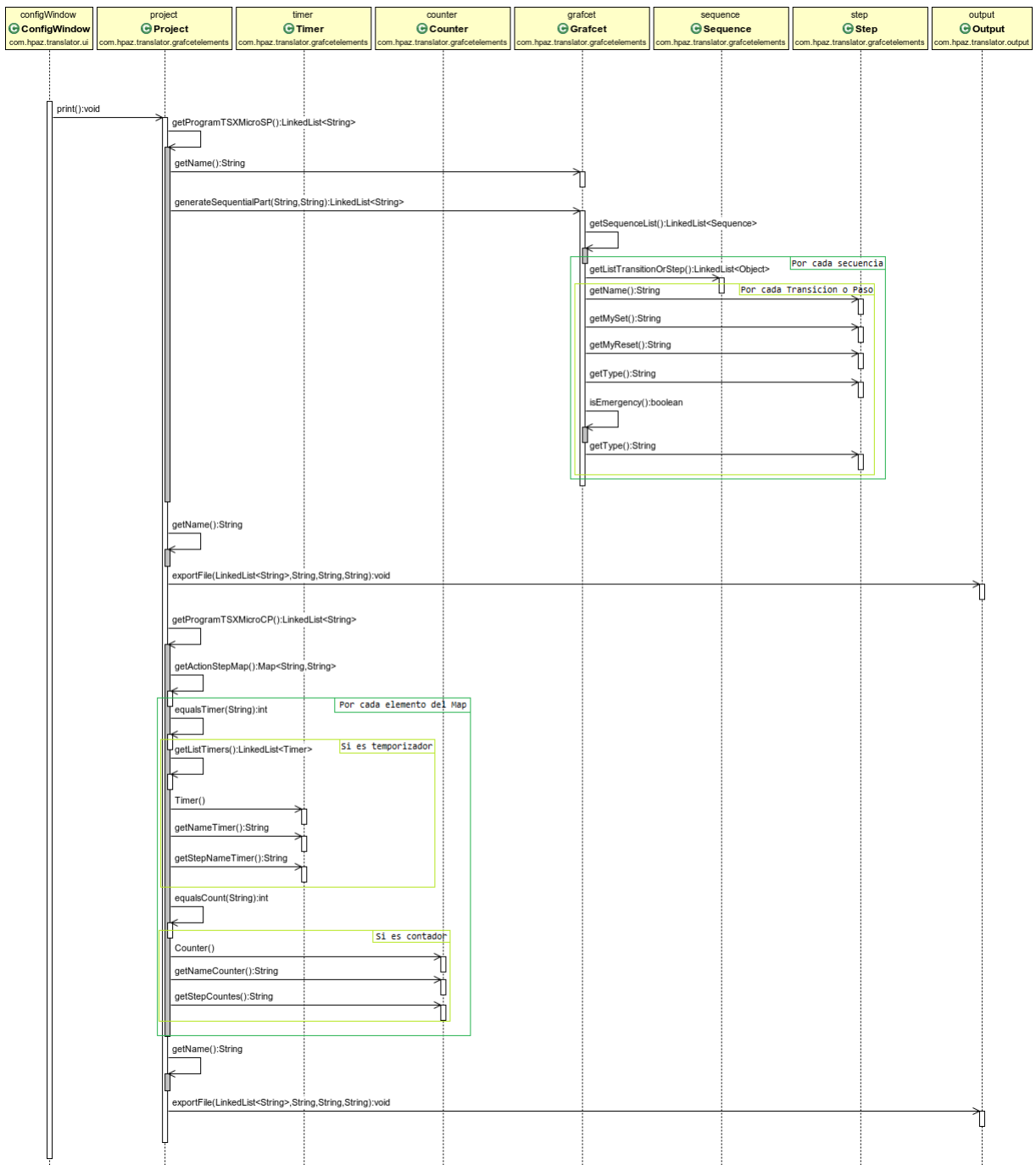
57. Figura: Análisis y Diseño - Diseño - Diagrama de Secuencia PreProceso III



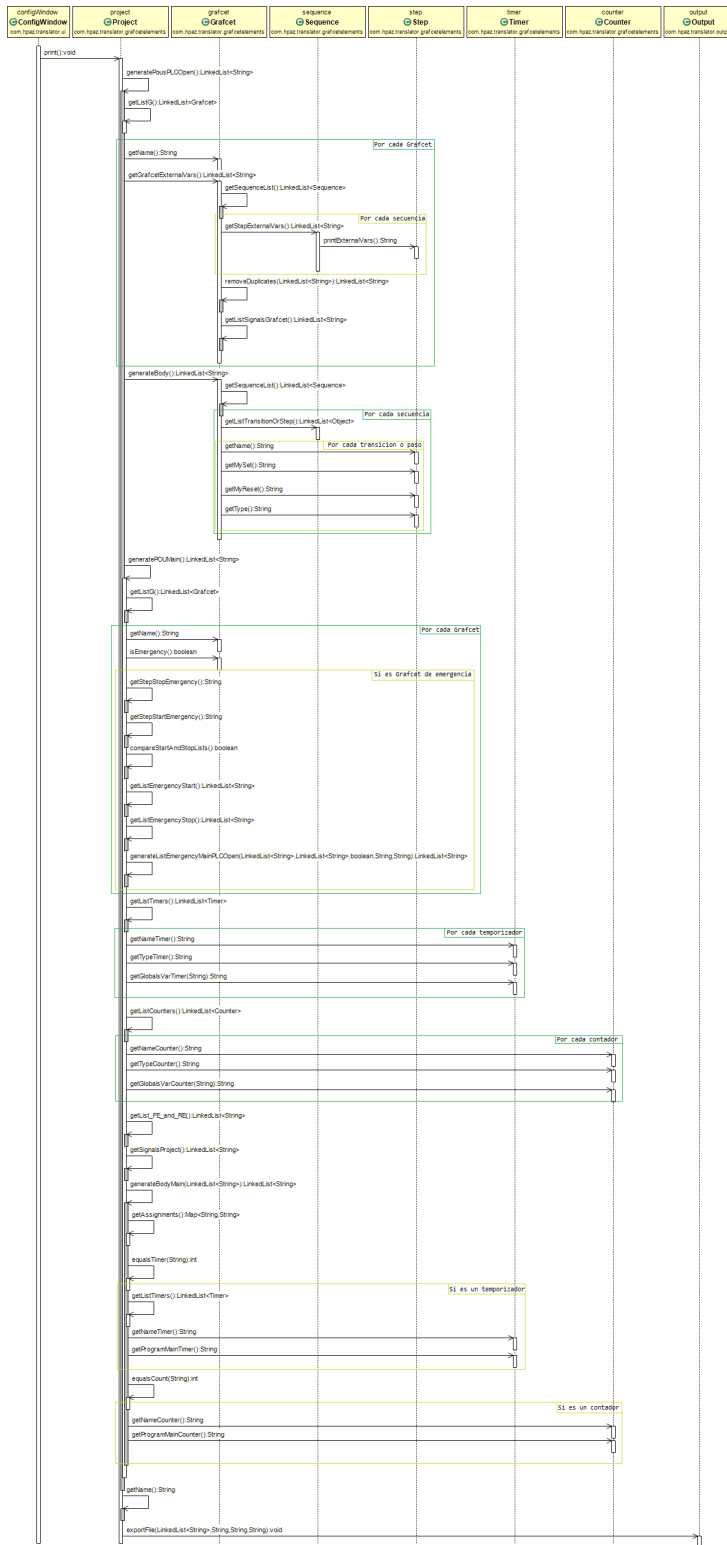
58. Figura: Análisis y Diseño - Diseño - Diagrama de Secuencia PreProceso IV



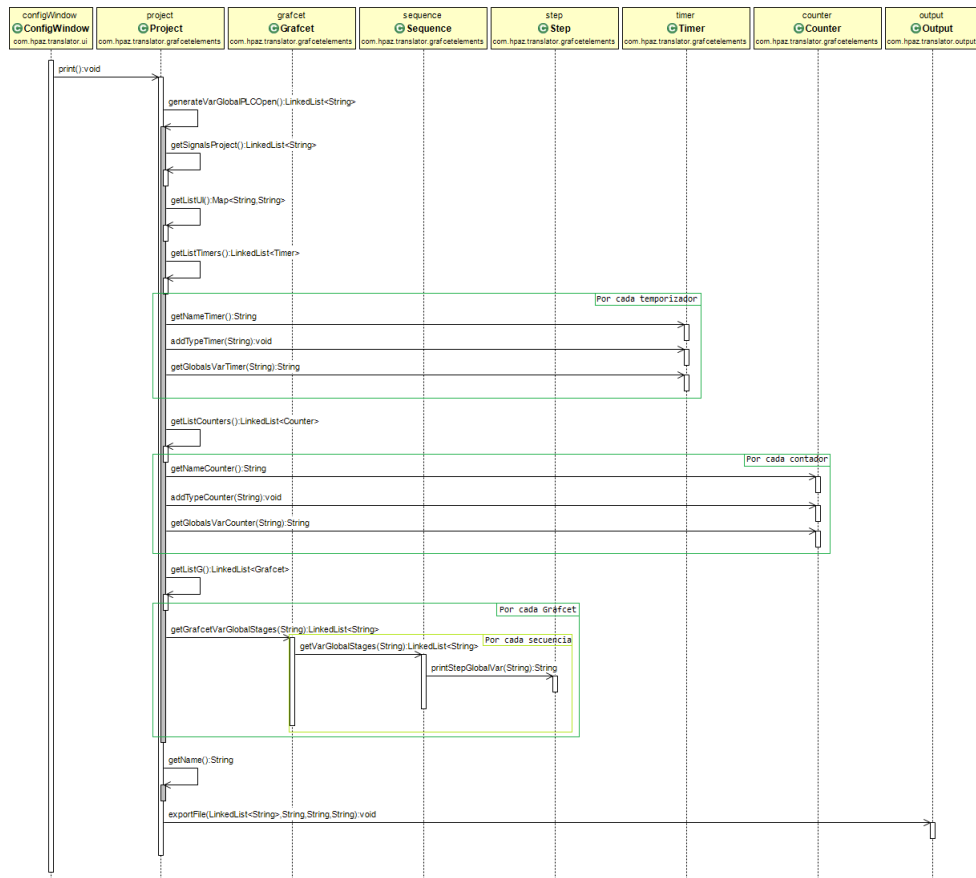
59. Figura: Análisis y Diseño - Diseño - Diagrama de Secuencia Imprimir PLC Beckhoff



60. Figura: Análisis y Diseño - Diseño - Diagrama de Secuencia Imprimir PLC TSX Micro



61. Figura: Análisis y Diseño - Diseño - Diagrama de Secuencia Imprimir PLCOpen I



62. Figura: Análisis y Diseño - Diseño - Diagrama de Secuencia Imprimir PLCOpen II

ANEXO III.- Manual de usuario

Introducción

El software ha sido desarrollado con el lenguaje de programación Java, con esto se busca que la aplicación sea multiplataforma y que pueda ser usada independientemente de la máquina o el sistema operativo que se tenga instalado. Hoy en día muchas aplicaciones y sitios web no funcionan a menos que se tenga Java instalado, por lo que es muy probable que este instalado ya en el equipo a usar, de no ser así se puede acceder y descargarlo gratuitamente desde la pagina java.com.

Este manual tiene como finalidad instruir a los usuarios del funcionamiento de la herramienta, de manera que puedan llegar a conocerla y usarla de forma cómoda y fácil.

Requisitos del sistema

La aplicación desarrollada prácticamente puede ser usada por cualquier ordenador y sistema operativo, aún así al ser implementado con el lenguaje de programación Java, las limitaciones serán equivalentes a las limitaciones de la máquina virtual de Java que tenga instalada el equipo a usar.

Se recomienda usar Máquina Virtual de Java (MVJ) 1.6 o superior y una memoria RAM de 1GB o superior.

Sin embargo dependiendo de la velocidad de su procesador, la ejecución de la aplicación será más o menos fluida.

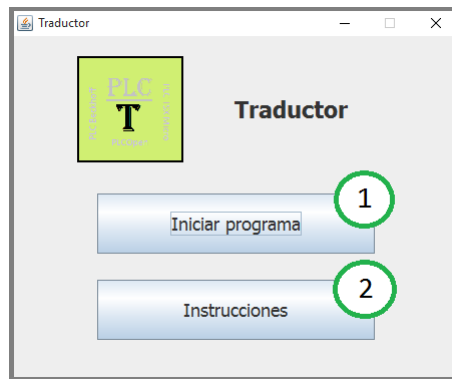
Funcionamiento de la aplicación

Para ejecutar el programa se puede hacer con doble click en el ícono de la aplicación o mediante consola con el siguiente comando:

```
java -jar [nombre de la aplicación].jar
```

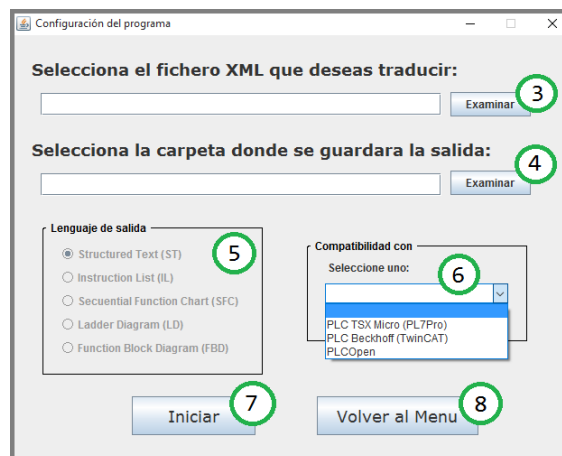
Se han enumerado las distintas acciones que se pueden realizar en la aplicación:

1. El botón de «Iniciar Programa» nos llevara a la ventana de la configuración de la aplicación para comenzar a usar la aplicación (Se recomienda leer las instrucciones antes, si es primera vez que se usa).
2. Botón de «Instrucciones» este botón nos abrirá una ventana en la cual se explican las necesidades sintácticas para que la aplicación sea ejecutada de manera correcta.



63. Figura: Manual de usuario - Menú

3. Botón «Examinar» al pulsarlo se abre un navegador de archivos en el que se puede buscar y seleccionar el fichero XML a usar.
4. Botón «Examinar» al pulsarlo se abre un navegador de archivos en el que se puede buscar y seleccionar el directorio en el que se archivarán los ficheros generados por la aplicación.
5. En esta opción se debe seleccionar el lenguaje en el que se generarán los ficheros de salida. Por el momento solo se encuentra funcional el lenguaje literal estructurado (ST).
6. En esta opción hay un menú desplegable en el que se seleccionará el software al que debe ser compatible el código de los ficheros generados por la aplicación.
7. El botón «Iniciar» se pulsa una vez acabada la configuración para que se comience el proceso de generar los ficheros. Dependiendo del software de compatibilidad que se selecciones aparecerá la ventana de configuración de variables o no.
8. Botón «Volver al Menú» permite volver a la pantalla anterior (Hay que tener en cuenta que al volver a la ventana anterior se perderán todos los datos que se han rellenado en la pantalla actual).



64. Figura: Manual de usuario - Configuración del traductor

9. En esta opción se debe seleccionar el tipo de dato de cada una de las variables que ha detectado la aplicación. *Si se selecciona que no es una señal, esta no será tomada en cuenta para la generación del código.*
10. En esta opción se debe seleccionar de que tipo será cada una de las variables que ha detectado la aplicación.
11. En esta opción se debe seleccionar de que tipo de temporizador de cada uno de los temporizadores que ha detectado la aplicación. *Si se selecciona que no es un temporizador, este no será tomada en cuenta para la generación del código.*
12. En esta opción se debe seleccionar de que tipo de contador de cada uno de los contadores que ha detectado la aplicación. *Si se selecciona que no es un contador, este no será tomada en cuenta para la generación del código.*
13. Botón «Finalizar» devolverá un mensaje informando sobre la situación del proceso ya sea que faltan datos por rellenar o que se ha finalizado el proceso generación de ficheros.

The screenshot shows a window titled "Inicialización de variables" with three main sections: "Señales", "Temporizadores", and "Contadores".

- Señales:** A table with columns "Nombre de variable", "Tipo de dato", and "Tipo de variable".

Nombre de variable	Tipo de dato	Tipo de variable
Inicio	Entrada (9)	BOOL (10)
SolModoAuto	Salida	BOOL
Marcha	Constante	[Dropdown menu]
SolCondiini	[Dropdown menu]	BOOL
FinCondiini	[Dropdown menu]	BYTE
ProdNormal	[Dropdown menu]	WORD
Paro	[Dropdown menu]	DOUBLE WORD
- Temporizadores:** A table with columns "Nombre de variable" and "Tipo de temporizador".

Nombre de variable	Tipo de temporizador
Temp	[Dropdown menu] (11)
- Contadores:** A table with columns "Nombre de variable" and "Tipo de contador".

Nombre de variable	Tipo de contador
ContPrueba	[Dropdown menu] (12)

At the bottom of the window is a "Finalizar" button (13).

65. Figura: Manual de usuario - Gestión de Variables