



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

Automatización de alta de incidencias desde
correo al gestor

Alumno:
Roberto Gozalo Andrés
Tutores:
Diego García Álvarez
César Pablo Gutiérrez Martínez

Agradecimientos

Me gustaría dedicar este espacio para reflejar mi agradecimiento a todas esas personas que durante este largo periodo que ha sido mi estancia en este grado, de alguna manera, me han prestado su ayuda para resolver la infinidad de problemas que se me han ido planteando.

No quiero olvidarme de la empresa donde realicé las prácticas, Luce IT ya que el trato mostrado hacia mí fue excelente y en ella pude encontrar la motivación que había perdido sobre esta profesión así como la motivación para realizar este trabajo.

Me gustaría también agradecer el trato y la profesionalidad de mis dos tutores Diego y César Pablo que siempre han estado dispuestos a ayudarme en las mil y una dudas que me han ido surgiendo a lo largo del desarrollo de este proyecto.

Por último, agradecer el apoyo que me han brindado tanto mi familia como mis amigos de toda la vida y todos aquellos que he hecho a lo largo del grado. Sin ellos no hubiera sido posible el llegar hasta aquí.

Resumen

El siguiente trabajo [2] realizado en la Escuela de Ingeniería Informática de la Universidad de Valladolid, para el grado de Ingeniería Informática en su mención de Tecnologías de la Información, tiene como objeto el desarrollo de un bot para la automatización del proceso de inserción de incidencias, que son recibidas mediante correo electrónico, en una plataforma que se encarga de su gestión.

Para ello se empleará la plataforma de automatización Automation Anywhere [1] en su versión on premise y el sistema de gestión de incidencias Jira [5], en su versión gratuita, perteneciente a Atlassian. Como sistema gestor de correo electrónico se empleará Outlook [4].

Con este trabajo se pretende optimizar el gasto de recursos, tanto humanos como técnicos destinados a una tarea repetitiva, realizando para ello un bot que se encargue de abordar dicho cometido.

Abstract

The objective of the following essay which has been created in the Computer Engineering School of the Valladolid University is the development of a bot for automate the issue insert process which are received by email, in an issue management platform.

For this purpose it will be used the automation platform Automation Anywhere, using the on premise version and the management issue system called Jira in his free version which is part of the Atlassian company. For managing the emails Outlook will be used.

With the develop of this solution we pretend to optimize the technical and human resources which are involved in a repetitive process, developing a bot which can automate this process.

Índice general

Agradecimientos	I
Resumen	I
Abstract	I
Lista de figuras	VII
Lista de tablas	IX
1. Introducción, Objetivos y Motivación	1
1.1. Introducción	1
1.2. Objetivos	2
1.2.1. Objetivos de desarrollo	2
1.2.2. Objetivos personales	2
1.3. Motivación	3
2. Estado del arte	4
3. Planificación y desarrollo del proyecto	5
3.1. Metodología empleada: Metodología Scrum	6
3.1.1. Eventos	7
3.1.2. Artefactos	8
	II

3.2. Descripción detallada del Sistema	9
3.3. Planificación	10
3.3.1. HE01 Elicitación de requisitos:	11
3.3.2. HE02 Análisis funcional:	11
3.3.3. HE03 Diseño:	11
3.3.4. HE04 Implementación:	12
3.3.5. HE05 Pruebas:	12
3.3.6. Sprint Backlog	12
4. Tecnologías empleadas	16
4.1. Automation Anywhere	16
4.2. Jira	16
4.3. Trello	17
4.4. Overleaf	17
4.5. Microsoft Excel	17
4.6. Astah	17
5. Plan de riesgos y estimación de costes del proyecto	18
5.1. Planificación de riesgos	18
5.1.1. Pautas para la elaboración del plan de riesgos	18
5.1.2. Riesgos y oportunidades del proyecto	19
5.2. Presupuestación inicial	21
5.3. Análisis final del coste del proyecto	22
6. Seguimiento del proyecto	23
6.1. Introducción	23
6.2. Seguimiento de los diferentes sprints	23
6.2.1. Sprint 1	23

6.2.2. Sprint 2	24
6.2.3. Sprint 3	24
6.2.4. Sprint 4	24
6.2.5. Sprint 5	25
6.2.6. Sprint 6	25
6.2.7. Sprint 7	25
6.2.8. Sprint 8	26
7. Análisis	27
7.1. Introducción	27
7.2. Casos de Uso	27
7.2.1. Caso de Uso: Procesar Incidencias	29
7.2.2. Caso de Uso: Conexión al sistema gestor de correo	29
7.2.3. Caso de Uso: Procesado de la información del correo	30
7.3. Modelo de Dominio	31
7.3.1. Clases de dominio	31
7.4. Modelo de análisis	32
7.4.1. Clases de análisis	32
7.4.2. Realización de casos de uso de análisis	35
8. Diseño	38
8.1. Introducción	38
8.2. Arquitectura física	38
8.3. Arquitectura lógica: patrón arquitectónico	40
8.4. Patrones de diseño	41
8.4.1. Patrones de diseño para la lógica de dominio	41
8.4.2. Patrones de diseño para el acceso a datos	42
8.5. Realización de casos de uso en el modelo de diseño	42

9. Implementación del proyecto	44
9.1. Acciones de Automation Anywhere empleadas	44
9.1.1. Task Bot	46
9.1.2. XML start session	47
9.1.3. XML get single node	48
9.1.4. Try/Catch	49
9.1.5. Log to File	50
9.1.6. Message Box	51
9.1.7. Task Bot: Run	52
9.1.8. Email: Connect	53
9.1.9. Email: Change Status	53
9.1.10. String: Assign	54
9.1.11. String: Extract text	54
9.1.12. Email: Disconnect	56
9.1.13. String: Lenght	56
9.1.14. String: Substring	57
9.1.15. REST Web Services. POST	57
9.2. Organización de los diferentes Task Bots	59
9.2.1. readConfig	59
9.2.2. Orquestador	61
9.2.3. readOutlook	62
9.2.4. createIssue	65
10.Pruebas	67
10.1. Introducción	67
10.2. Pruebas desarrolladas	68

11. Conclusiones y líneas futuras	70
11.1. Conclusiones	70
11.2. Líneas futuras	71
A. Manual de usuario	72
B. Manual de despliegue y ejecución	73
B.1. Requisitos mínimos para el despliegue y ejecución del bot	73
B.2. Pasos previos a la ejecución del bot	73
B.3. Ejecución del bot	75
Bibliografía	76

Índice de figuras

3.1. Metodología Scrum. Fuente [9]	6
3.2. Captura del Trello empleado en el proyecto	15
7.1. Diagrama de casos de uso.	28
7.2. Diagrama de clases de dominio.	31
7.3. Diagrama de clases de análisis.	34
7.4. Diagrama de secuencia del caso de uso del procesado de incidencias	35
7.5. Diagrama de secuencia del caso de uso de la conexión al sistema gestor de incidencias	36
7.6. Diagrama de secuencia del caso de uso del procesado de la información recibida mediante correo electrónico	37
8.1. Modelo de despliegue de la arquitectura física del sistema.	39
8.2. Patrón Filtro Tubería aplicado al compilador de C. Fuente [34]	40
8.3. Patrón filtro tubería aplicado al sistema. Fuente elaboración propia	41
8.4. Diagrama de secuencia de diseño correspondiente al caso de uso <i>Procesado de la información del correo 7.3.</i>	43
9.1. Ejecucion del Task Bot. Fuente [1]	46
9.2. Iniciar sesion XML. Fuente [1]	47
9.3. Obtener datos de XML. Fuente [1]	48
9.4. Bloque Try/Catch. Fuente [1]	49

9.5. Añadir texto al log. Fuente [1]	50
9.6. Ventana emergente. Fuente [1]	51
9.7. Ejecucion de un task bot. Fuente [1]	52
9.8. Conectar sesion de correo. Fuente [1]	53
9.9. Modificar el estado de los correos. Fuente [1]	53
9.10. Asignar un valor a una variable. Fuente [1]	54
9.11. Extraer una porción de un string. Fuente [1]	55
9.12. Finalizar la sesion de correo. Fuente [1]	56
9.13. Obtener longitud de una cadena de texto. Fuente [1]	56
9.14. Obtener substring. Fuente [1]	57
9.15. Realizar llamada POST. Fuente [1]	58
9.16. Flujo de la lectura del archivo config. Parte 1. Fuente [1]	59
9.17. Flujo de la lectura del archivo config. Parte 2. Fuente [1]	60
9.18. Flujo del orquestador. Fuente [1]	61
9.19. Flujo de la lectura de correos.Parte 1. Fuente [1]	62
9.20. Flujo de la lectura de correos.Parte 2. Fuente [1]	63
9.21. Flujo de la lectura de correos.Parte 3. Fuente [1]	64
9.22. Flujo de la creación de incidencias.Parte 1. Fuente [1]	65
9.23. Flujo de la creación de incidencias.Parte 2. Fuente [1]	66
B.1. Instalación del software para ejecución del bot. Fuente [1]	74
B.2. Árbol de directorios necesarios para la ejecucion del sistema	74
B.3. Ejecución del bot. Fuente [1]	75

Índice de cuadros

5.1. Riesgo de retraso	19
5.2. Riesgo de ausencia de información	19
5.3. Riesgo de enfermedad	19
5.4. Riesgo de cambio de historias de usuario	20
5.5. Riesgo de ausencia del Scrum Master	20
5.6. Oportunidad de reutilización de conocimientos	20
5.7. Tabla de clasificación de riesgos	21
5.8. Calendario y Presupuesto de recursos humanos	21
5.9. Presupuesto de recursos materiales	22
5.10. Presupuesto final del proyecto	22
6.1. Seguimiento del sprint 1	23
6.2. Seguimiento del sprint 2	24
6.3. seguimiento del sprint 3	24
6.4. seguimiento del sprint 4	24
6.5. seguimiento sprint 5	25
6.6. seguimiento sprint 6	25
6.7. seguimiento sprint 7	25
6.8. seguimiento sprint 8	26

7.1. Caso de uso Procesar incidencias	29
7.2. Caso de uso Conexión al sistema gestor de correo	29
7.3. Caso de uso Procesado de la información del correo	30
7.4. Clases BCE. Fuente: [33]	32
7.5. Interacciones de las clases BCE. Fuente: [33]	33
10.1. Caso de prueba 01	68
10.2. Caso de prueba 02	69
10.3. Caso de prueba 03	69

NOTA: Este documento es un documento digital, por ello, gran parte de los diagramas que se muestran en él son imágenes vectoriales y pueden ser ampliados sin perder calidad para poder así observar con claridad todos los detalles de estos.

Capítulo 1

Introducción, Objetivos y Motivación

1.1. Introducción

En este documento se va a proceder a describir los diferentes detalles del proceso de creación de un sistema informático automatizado, también llamado bot. Este bot se encarga, de manera automática, bajo programación previa de su ejecución, de realizar la lectura de los emails recibidos por un buzón de correo electrónico [4]. En dichos correos, las diferentes empresas, envían incidencias a la empresa de soluciones informáticas.

El bot, después de leer dichos correos, procesa la información recibida e inicia el proceso de creación de la incidencia en el sistema de gestión de incidencias de la empresa de soluciones informáticas basado en la plataforma Jira [5].

Para crear dichas incidencias, el bot deberá registrarse en la plataforma, acceder al proyecto correspondiente y crear en él las diferentes incidencias.

Para determinar a qué proyecto corresponde cada incidencia se deberá determinar un set de palabras clave mediante el cual puedan ser categorizadas las incidencias en los diferentes proyectos de la empresa.

1.2. Objetivos

A continuación se van a describir los distintos objetivos que se pretenden cumplir con el desarrollo de este trabajo de fin de grado, tanto de desarrollo como personales.

1.2.1. Objetivos de desarrollo

El objetivo principal de este proyecto es el de dotar al mundo empresarial de una herramienta, totalmente automática, que permita, la creación de incidencias a través de los correos que son recibidos en una bandeja de entrada de correo electrónico. Creando esta herramienta, se optimizará el gasto de recursos de las empresas, tanto humanos como económicos, empleados en realizar una tarea repetitiva, susceptible a errores humanos.

1.2.2. Objetivos personales

Ya que el desarrollo de este proyecto tiene un marcado carácter educativo, puesto que es la realización de un trabajo de fin de grado, los objetivos personales están orientados a la adquisición y afianzamiento de conocimientos. A continuación, se detallará cada uno de ellos:

- Afianzar los conocimientos sobre el desarrollo de proyectos así como cada una de las partes en las que está compuesto.
- Conocer en profundidad el desarrollo de bots mediante la plataforma Automation Anywhere.[1]
- Familiarizarse con la herramienta de gestión de incidencias Jira.
- Entender y aplicar una metodología ágil como es Scrum [3] al desarrollo de un proyecto real. Ya que es una de las metodologías más empleadas en el mundo laboral.
- Adquirir las competencias necesarias para redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas. [2]
- Ser capaz de definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas, de acuerdo con los conocimientos adquiridos.
- Adquirir la capacidad para analizar y valorar el impacto social y medioambiental de las soluciones técnicas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico en Informática.

1.3. Motivación

La idea de desarrollar este proyecto surgió tras la realización de las prácticas de grado en la empresa. En ellas se observó que se destinaban demasiados recursos a una tarea repetitiva y que podría ser automatizada mediante un bot.

Mediante esta automatización, se minimizan los errores en la inserción de incidencias, ya que está demostrado que la efectividad de un humano, desarrollando una tarea disminuye con el paso del tiempo. Además del ya comentado ahorro de recursos, ya que la persona que se encargaba de estas tareas, puede dedicarse a otras con mayor relevancia, dentro de la empresa.

Capítulo 2

Estado del arte

Actualmente la plataforma Jira [5] de la empresa Atlassian es una de las más empleadas por las empresas del sector de la informática [6] [7] que desarrollan una metodología ágil [8].

Jira permite gestionar las incidencias y las diferentes tareas dentro de cada uno de los proyectos de la empresa. Pudiéndose asignar trabajadores a las diferentes incidencias así como las horas que estos destinan a cada tarea de cada proyecto.

En la actualidad, los clientes de la empresa envían sus incidencias mediante correo electrónico, para que la empresa que va a prestar ese servicio las gestione.

Al recibir dichas incidencias por email, debe haber una persona designada por parte de la empresa, para que lea dichos correos e inserte de forma manual, las incidencias recibidas en la plataforma y que de esta manera queden registradas para su posterior tratamiento y resolución por parte de la empresa.

Lo que se pretende con la solución planteada es la creación de manera automatizada de las diferentes incidencias y tareas en Jira.



Capítulo 3

Planificación y desarrollo del proyecto

3.1. Metodología empleada: Metodología Scrum

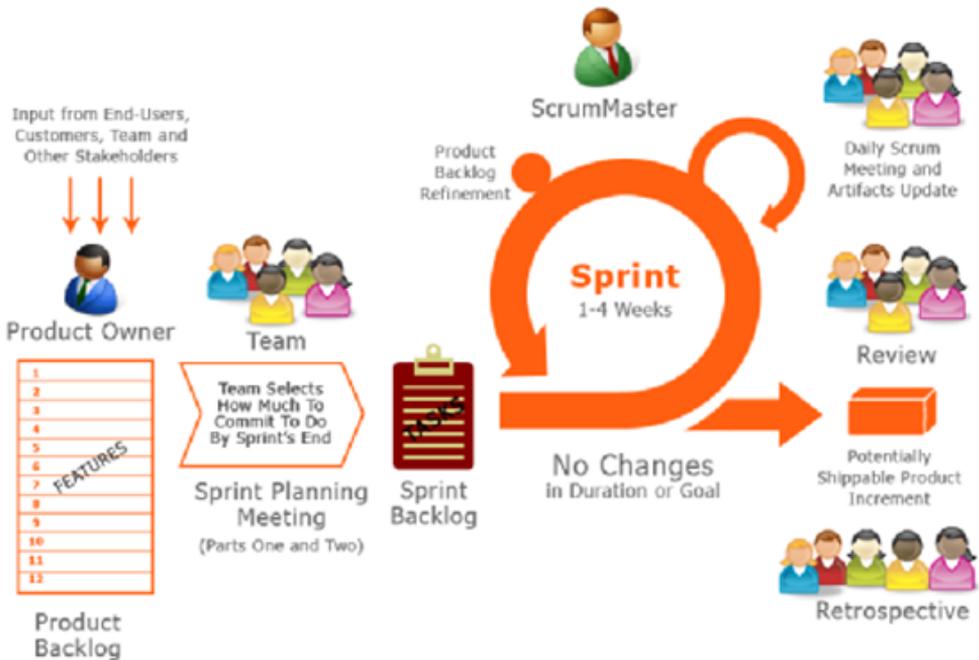


Figura 3.1: Metodología Scrum. Fuente [9]

Como se aprecia en la figura 3.1, la metodología a usar seguirá un marco ágil, en nuestro caso SCRUM. SCRUM se encuentra dentro de los procesos ágiles que según el *Manifiesto ágil* [13] se caracterizan por:

- Software funcionando por encima de documentación extensiva.
- Respuesta ante el cambio por encima de seguir un plan.
- Interacciones e individuos por encima de herramientas y procesos.
- Colaboración con el cliente por encima de negociación de contrato.

SCRUM se define como un marco de referencia en el que un equipo único y de dimensiones reducidas desarrolla, entrega y mantiene productos complejos a través de un enfoque iterativo e incremental en bloques temporales cortos y fijos, gestionando en lo posible la incertidumbre y por tanto el riesgo asociado a la misma. El proceso se basa en tres pilares: [14]

- Transparencia
- Inspección
- Adaptación

Es por ello que SCRUM es adecuado para proyectos con un carácter exploratorio y/o con una alta incertidumbre, es decir, que pueden ser propensos a sufrir muchos cambios, y que requieren de continua retroalimentación por tratarse de proyectos complejos. Estos proyectos suelen implicar también equipos multifuncionales altamente cualificados, un entorno de desarrollo hipercompetitivo y que se esté trabajando con productos tecnológicos de vanguardia.

3.1.1. Eventos

En el marco de trabajo de SCRUM existen diversos eventos [9], que tienen lugar con el objetivo de cumplir con los tres pilares principales explicados anteriormente. Estos eventos, de periodicidad y duración definida, tratan de minimizar la necesidad de reuniones adicionales no definidas en el marco original. Son los siguientes:

- **Sprint:** Constituye el evento fundamental del marco SCRUM. Es un bloque de tiempo, con una duración de entre 1 y 4 semanas, en la cual se realiza un incremento utilizable y potencialmente desplegable. Lo ideal es mantener una duración fija a lo largo del proceso de desarrollo. Cada sprint comienza inmediatamente después de la finalización del anterior. Mantiene unos requisitos y objetivos claros y estáticos a lo largo de su duración.

- **Sprint planning:** Es el evento que da comienzo a un sprint y en el que se planifica cuál será el contenido y los objetivos del sprint. En esta reunión participan todos los componentes del equipo SCRUM, y se seleccionan los ítems del Product Backlog ya priorizados, que se pretenden realizar en el sprint que se está planificando. Para ello, a las distintas tareas se les asigna una estimación basada en puntos de historia, que a su vez se traduce en tiempo estimado de trabajo. El resultado de este evento es el Sprint Backlog.
- **Daily SCRUM:** Es una reunión de unos 15 minutos en la que el equipo de Desarrollo pone en común las actividades desarrolladas durante las 24 horas anteriores y fija las que se realizarán en las siguientes. El objetivo es evaluar el progreso del equipo y sincronizar a todos sus miembros, alineándolos en la consecución del Sprint Goal. Para este trabajo no es necesario una reunión diaria ya que se aplica SCRUM for one.
- **Sprint Goal:** Es una meta que se establece en el Sprint Planning, desglosada mediante las historias de usuario del backlog que pertenecen al sprint. El equipo de Desarrollo mantiene en mente este objetivo, implementando de la forma más adecuada la funcionalidad para llegar a satisfacerlo.
- **Sprint review:** Es una reunión que ocurre al final del sprint, en la que el equipo SCRUM al completo y los clientes discuten sobre los objetivos que se han cumplido en el sprint, para poder decidir qué hacer en el siguiente. Si es necesario, puede haber ajustes en el *Product Backlog*.
- **Sprint Retrospective:** Constituye una oportunidad para el Equipo SCRUM de evaluar esfuerzos y problemas y abordar soluciones o mejoras que puedan optimizar el trabajo en el desarrollo de próximos sprints. Este tipo de reuniones se ha decidido agruparlas junto con las sprint planning del siguiente sprint.

3.1.2. Artefactos

El resultado de las distintas actividades en SCRUM se materializa en artefactos [9], que representan trabajo o valor. El objetivo de los mismos es maximizar la transparencia de la información clave para todas las partes involucradas. Los artefactos considerados en SCRUM son:

- **Product Backlog:** Es una lista ordenada de requisitos del producto, denominados historias de usuario. Es importante destacar que esta lista se encuentra ordenada por prioridad. En un primer momento, se realiza una detección y definición previa de los mismos, pudiendo variar a lo largo de todo el proceso de desarrollo. Por ello, el *Product Backlog* es cambiante, sujeto a todo tipo de variaciones. En él se incluyen todas las mejoras, correcciones, evoluciones y funcionalidades de las que se debe dotar al producto. Sus elementos se examinan y se modifican en función de los requisitos que se pretendan acometer o abordar, pudiendo surgir otros nuevos o modificaciones sobre los ya existentes en cualquier momento.

- **Sprint Backlog:** Está constituido por la lista de requisitos e historias de usuario del *Product Backlog* que serán abordados a lo largo del sprint actual.

Es estático, no pudiendo ser modificado mientras el sprint se encuentra en curso, a fin de mantener una adecuada cohesión y coordinación entre los objetivos a implementar y los miembros del equipo, favoreciendo así la optimización del trabajo. Hace visible la lista de tareas necesarias para alcanzar el *Sprint Goal*. Contiene el suficiente detalle como para poder observar el avance realizado en el *Daily Scrum*. Pertenece únicamente al Equipo de Desarrollo.

- **Incremento:** Es la suma de todos los elementos completados del *Product Backlog* hasta la fecha. Cada incremento debe estar en condiciones de ser utilizado

3.2. Descripción detallada del Sistema

En esta sección se va a realizar una descripción lo más detalladamente posible del sistema objeto de este trabajo de fin de grado [2].

El sistema fruto de este trabajo es un sistema de RPA (*Robotic Process Automation*) [10]. Para su desarrollo se ha empleado la herramienta de desarrollo RPA llamada Automation Anywhere [1] en su versión gratuita [12].

La idea del desarrollo de este trabajo surge tras la realización de las prácticas en empresa por parte del alumno, en las que se observa que se destina un gran volumen de recursos a la tarea de leer las incidencias que llegan a través de un buzón de correo [4] y que se han de introducir de manera manual en el sistema gestor de incidencias empleado por la empresa (Jira) [5]. Hecho susceptible a errores humanos en la introducción de los datos.

Para tratar de liberar estos recursos tanto materiales como humanos y minimizar los posibles errores humanos se propone la solución de crear un bot RPA que automáticamente lea los correos recibidos en el buzón de correo electrónico, descarte los que no tienen en su contenido incidencias, procese los que si las contienen, extrayendo la información necesaria, para finalmente, crear dicha incidencia de manera automática en Jira, el sistema gestor de incidencias.

El sistema simplemente cuenta con la interacción humana, cuando se inicia su ejecución. Tras haberse iniciado, se ejecuta la tarea de lectura del archivo *configFile.xml* en dicho archivo se encuentran los diferentes valores de las variables necesarias para la ejecución del bot. Tras leer y mapear estos valores, se ejecuta la tarea de lectura de los correos. Para ello el sistema realiza una lectura de los correos marcados como no leídos, descartando aquellos que no se ajusten al formato fijado y procesando la información referente a las incidencias de aquellos que si cumplan el formato. Una vez encontrada una incidencia y extraída su información se ejecuta la tarea de creación de incidencias, en la que se crea una llamada a una Api REST [28] en la que se envía un JSON [30] con la información para la creación de la incidencia en el sistema gestor. Si se hubieran detectado más incidencias en el mismo correo se volvería a ejecutar la tarea de creación de incidencias creando las llamadas correspondientes.

El sistema cuenta con un sistema de registro log en el que se reflejan los principales eventos y errores que suceden durante cada una de las ejecuciones del bot. Indicando su fecha y hora [26] para así tener una trazabilidad completa de los errores y poder actuar en consecuencia.

3.3. Planificación

Cada uno de los sprints establecidos tendrá una duración de una semana, con un total de 8 sprints. Está compuesto de las siguientes historias épicas:

- Elicitación de requisitos
- Análisis funcional
- Diseño funcional
- Implementación
- Pruebas
- Fase de producción del sistema

Cada una de ellas se desglosará en historias de usuario que estarán repartidas en los diferentes sprints.

3.3.1. HE01 Elicitación de requisitos:

Cada una de las expectativas y necesidades detectadas en la empresa en la que el alumno ha desarrollado las prácticas se han convertido en requisitos, en este caso en historias de usuario que serán desarrolladas durante este proyecto.

- **REQ01** Identificar el contenido de los emails recibidos
- **REQ02** Determinar el origen de los datos de los emails
- **REQ03** Definir el formato de los emails que van a ser recibidos
- **REQ04** Procesar y extraer la información necesaria para la creación de incidencias en el sistema gestor
- **REQ05** Realizar la conexión del RPA con Jira
- **REQ06** Crear la incidencia con los datos extraídos del correo

3.3.2. HE02 Análisis funcional:

- AN01 Identificar el contenido de los emails recibidos
- AN02 Determinar el origen de los datos de los emails
- AN03 Definir el formato de los emails que van a ser recibidos
- AN04 Procesar y extraer la información necesaria para la creación de incidencias
- AN05 Realizar la conexión del RPA con Jira
- AN06 Crear la incidencia con los datos extraídos del correo
- AN07 Enviar el correo resumen con las incidencias que han sido creadas

3.3.3. HE03 Diseño:

- DIS01 Identificar el contenido de los emails recibidos
- DIS02 Determinar el origen de los datos de los emails
- DIS03 Definir el formato de los emails que van a ser recibidos
- DIS04 Procesar y extraer la información necesaria para la creación de incidencias
- DIS05 Realizar la conexión del RPA con Jira
- DIS06 Crear la incidencia con los datos extraídos del correo
- DIS07 Enviar el correo resumen con las incidencias que han sido creadas

3.3.4. HE04 Implementación:

- IMP01 Identificar el contenido de los emails recibidos
- IMP02 Determinar el origen de los datos de los emails
- IMP03 Definir el formato de los emails que van a ser recibidos
- IMP04 Procesar y extraer la información necesaria para la creación de incidencias
- IMP05 Realizar la conexión del RPA con Jira
- IMP06 Crear la incidencia con los datos extraídos del correo
- IMP07 Enviar el correo resumen con las incidencias que han sido creadas

3.3.5. HE05 Pruebas:

- PRU01 Desarrollar los casos de prueba y probar el segundo sprint
- PRU02 Probar el tercer sprint

A parte de las historias épicas mencionadas, el product backlog contiene inicialmente las siguientes historias de usuario (Columna *por hacer* de la imagen 3.2 :

- PROY01 Hacer la introducción del TFG
- PROY02 Describir el estado del arte
- PROY03 Realizar la planificación (este apartado)
- PROY04 Hacer la presentación PPT de defensa
- PROY05 Revisar la documentación por parte de los tutores

3.3.6. Sprint Backlog

Sprint 1 Planificación y preparación del trabajo

Se realizan las siguientes historias:

PROY01 Hacer la introducción del TFG

PROY02 Describir el estado del arte

PROY03 Realizar la planificación (este apartado)

Sprint 2

REQ01 Identificar el contenido de los emails recibidos

REQ02 Determinar el origen de los datos de los emails

REQ03 Definir el formato de los emails que van a ser recibidos

Sprint 3

REQ04 Procesar y extraer la información necesaria para la creación de incidencias

REQ05 Realizar la conexión del RPA con Jira

REQ06 Crear la incidencia con los datos extraídos del correo

REQ07 Enviar el correo resumen con las incidencias que han sido

AN01 Identificar el contenido de los emails recibidos

AN02 Determinar el origen de los datos de los emails

AN03 Definir el formato de los emails que van a ser recibidos

Sprint 4

AN04 Procesar y extraer la información necesaria para la creación de incidencias

AN05 Realizar la conexión del RPA con Jira

AN06 Crear la incidencia con los datos extraídos del correo

AN07 Enviar el correo resumen con las incidencias que han sido creadas

DIS01 Identificar el contenido de los emails recibidos

DIS02 Determinar el origen de los datos de los emails

DIS03 Definir el formato de los emails que van a ser recibidos

Sprint 5

DIS04 Procesar y extraer la información necesaria para la creación de incidencias

DIS05 Realizar la conexión del RPA con Jira

DIS06 Crear la incidencia con los datos extraídos del correo

DIS07 Enviar el correo resumen con las incidencias que han sido creadas

IMP01 Identificar el contenido de los emails recibidos

IMP02 Determinar el origen de los datos de los emails

IMP03 Definir el formato de los emails que van a ser recibidos

Sprint 6

IMP04 Procesar y extraer la información necesaria para la creación de incidencias

IMP05 Realizar la conexión del RPA con Jira

IMP06 Crear la incidencia con los datos extraídos del correo

IMP07 Enviar el correo resumen con las incidencias que han sido creadas

PRU01 Probar el segundo sprint

Sprint 7

PRU02 Probar el tercer sprint

PROY04 Hacer la presentación PPT de defensa

Sprint 8

PROY05 Revisar la documentación por los tutores.

También se realiza la entrega de la memoria y creación de línea base de productos de trabajo (al ser una buena práctica del Área de Práctica Gestión de Configuración del modelo CMMI para Desarrollo de Software). [15]

3.3. PLANIFICACIÓN

La herramienta elegida para el seguimiento del proyecto, dentro de un marco de trabajo ágil, es un tablero Kanban [16], para lo cual hemos utilizado Trello de la compañía Atlassian. Como se muestra en la figura 3.2

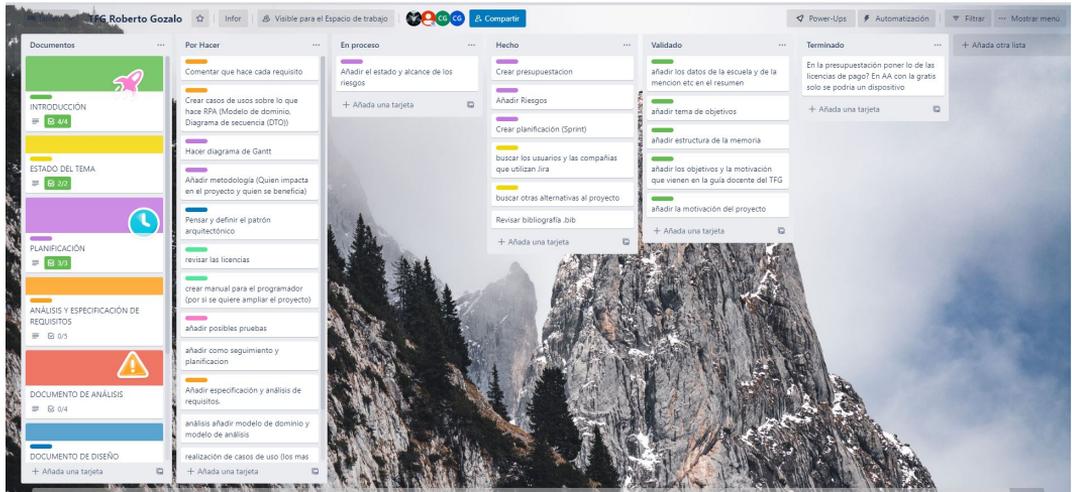


Figura 3.2: Captura del Trello empleado en el proyecto

Capítulo 4

Tecnologías empleadas

A continuación se va a proceder a describir cada una de las tecnologías y herramientas que se han empleado durante el desarrollo de este proyecto:

4.1. Automation Anywhere

Esta herramienta de desarrollo RPA[10] es la que se ha empleado para el desarrollo de nuestro bot. Existen otras herramientas en el mercado como pueden ser Uipath [11] o el desarrollo de sistemas de automatización mediante el lenguaje de programación Python, pero se ha optado por emplear esta tecnología debido a que ya se había realizado un aprendizaje previo mediante los cursos de los que dispone esta herramienta en su web.

Automation Anywhere (también conocida como AA) cuenta con dos versiones [12], una gratuita (la empleada en este proyecto) que cuenta con la misma funcionalidad que la versión de pago, pero en la que se limitan algunas opciones de gestión e importación de los bots creados. Hecho que para el desarrollo de este trabajo no es relevante.

4.2. Jira

Se ha empleado el sistema gestor de incidencias de la compañía Atlassian llamado Jira [5], porque como ya se ha comentado, es uno de los más empleados en la actualidad por las empresas referentes en el sector y porque ya se disponía de cierta familiaridad con el sistema.

Otros sistemas, no disponían de APIs públicas para poder realizar las operaciones pertinentes o no constituían un porcentaje relevante de uso dentro del sector de las empresas que proveen soluciones informáticas.

4.3. Trello

Se ha utilizado esta herramienta, también de la compañía Atlassian, para emplear un tablero Kanban [16] con el que poder realizar un seguimiento del proyecto. Esta herramienta se ha empleado en conjunto con los tutores para ir revisando las diferentes historias pertenecientes a los diferentes sprints del proyecto.

4.4. Overleaf

Se ha optado por este editor colaborativo de textos en LaTeX [17] por su gratuidad y porque permite compartir los documentos con otros usuarios. De esta manera, se ha compartido esta memoria del proyecto con los tutores para que pudiesen realizar un seguimiento íntegro del desarrollo de este. Tanto los profesores como el alumno conocían previamente las bondades de esta herramienta, por lo que no se consideró optar por otros procesadores de textos.

4.5. Microsoft Excel

Se ha empleado Excel para realizar aquellos cálculos necesarios para realizar la presupuestación del proyecto así como para algunos aspectos de la gestión de riesgos. Pese a ser una herramienta del paquete Microsoft Office [19] se ha decidido emplearla en este proyecto, ya que la Universidad de Valladolid provee de una licencia gratuita a sus estudiantes.

4.6. Astah

Se ha empleado Astah [24] como herramienta para realizar los diferentes diagramas UML [25] correspondientes a las fases de análisis y diseño del proyecto. Se ha optado por esta herramienta ya que ha sido la empleada en las diferentes asignaturas del grado referentes al análisis diseño y modelado del software. Para ello se ha empleado la licencia de la Escuela de Ingeniería Informática de Valladolid.

Capítulo 5

Plan de riesgos y estimación de costes del proyecto

5.1. Planificación de riesgos

Como se define en el libro que se usó como referencia en la asignatura de planificación y gestión de plataformas informáticas [20], un riesgo es cualquier evento o condición probable cuya aparición causa cierto efecto en los objetivos de un proyecto. Los riesgos pueden dividirse en:

- **Riesgos de negocio:** Son los relacionados con temas de mercado y de negocio, como puedan ser aquellos relacionados con los gastos e ingresos del proyecto.
- **Riesgos de proyecto:** Son aquellos que están relacionados con la no consecución de los objetivos del proyecto que han sido marcados por el jefe del proyecto o por el equipo de desarrollo.

5.1.1. Pautas para la elaboración del plan de riesgos

Para realizar un correcto plan de riesgos para este proyecto se realizarán los siguientes pasos:

1. Identificación del riesgo.
2. Análisis y establecimiento de su prioridad.
3. Planificación del riesgo.
4. Monitorización y control de dicho riesgo.

5.1.2. Riesgos y oportunidades del proyecto

A continuación se va a proceder a la enumeración de los diferentes riesgos y oportunidades que se han ido encontrando durante la elaboración de este trabajo. Para ello se dotará a cada uno de los riesgos de un título, una descripción, una probabilidad de que sucedan con la clasificación de muy baja, baja, media, alta y muy alta; su impacto sobre el proyecto con la clasificación de muy bajo, bajo, medio, alto y crítico; el impacto, el estado en el que se encuentra el riesgo, un plan de mitigación para reducir la probabilidad de aparición de dicho riesgo y un plan de contingencia que será el que se llevará a cabo en caso de que ese riesgo se materialice. [21]

Título	Retraso en las tareas
Descripción	Debido al retraso de las tareas planificadas en los diferentes sprints pueden llevar más tiempo del esperado, pudiendo retrasar la entrega de este trabajo
Probabilidad	Baja
Impacto	Medio
Nivel de riesgo	Bajo
Estado	Cerrado
Plan de mitigación	Todas aquellas tareas retrasadas se introducirán en el siguiente sprint
Plan de contingencia	Retrasar otras tareas menos prioritarias al siguiente sprint

Cuadro 5.1: Riesgo de retraso

Título	Ausencia de información
Descripción	Debido a que el personal destinado a dicha tarea no dispone de la suficiente información para realizarla, la tarea puede verse retrasada
Probabilidad	Baja
Impacto	Medio
Nivel de riesgo	Bajo
Estado	Cerrado
Plan de mitigación	Se evaluará si se dispone de toda la información. Se planificará una tarea específica para la recolección de la información, si fuera necesario
Plan de contingencia	Se retrasará el resto de tareas dependientes hasta completar esta

Cuadro 5.2: Riesgo de ausencia de información

Título	Enfermedad de un desarrollador
Descripción	Debido a una enfermedad que incapacite al proyectante puede retrasarse la consecución de las tareas que componen proyecto
Probabilidad	Baja
Impacto	Alto
Nivel de riesgo	Medio
Estado	Cerrado
Plan de mitigación	Se ha previsto de un sprint extra para poder hacer frente a las tareas retrasadas
Plan de contingencia	Replanificar el trabajo restante

Cuadro 5.3: Riesgo de enfermedad

5.1. PLANIFICACIÓN DE RIESGOS

Título	Cambio en las historias de usuario
Descripción	Debido a una modificación de las historias de usuario estas pueden variar la funcionalidad del sistema y retrasar el proyecto
Probabilidad	Media
Impacto	Medio
Nivel de riesgo	Medio
Estado	Cerrado
Plan de mitigación	Como el proyecto emplea una metodología ágil ésta prevé ciertos cambios en las historias. La planificación fue creada con antelación suficiente para hacer frente a imprevistos, para ello se introducen las nuevas historias en el siguiente sprint
Plan de contingencia	Replanificación de los diferentes sprints

Cuadro 5.4: Riesgo de cambio de historias de usuario

Título	Ausencia de Scrum Master/tutor del proyecto
Descripción	Debido a la ausencia del Scrum Master (rol ejercido por el tutor) por incompatibilidad horaria en las diferentes reuniones de las reuniones de las reuniones del desarrollo del proyecto estas pueden verse retrasadas
Probabilidad	Baja
Impacto	bajo
Nivel de riesgo	Bajo
Estado	Cerrado
Plan de mitigación	Las reuniones son fijadas con antelación suficiente y de mutuo acuerdo entre el estudiante y el profesor. Por otra parte, este trabajo esta tutorizado por dos profesores, por ello el nivel de riesgo es bajo
Plan de contingencia	El desarrollador ha de asumir una mayor responsabilidad a la hora de la toma de decisiones

Cuadro 5.5: Riesgo de ausencia del Scrum Master

Título	Trabajar en temas similares en las prácticas en empresa
Descripción	Al haber realizado las prácticas sobre RPA dichas prácticas pueden ser provechosas para el cometido y objetivo de este TFG. Permitiendo un mejor resultado final.
Probabilidad	Alta
Impacto	Muy alto
Nivel de oportunidad	Alto
Estado	Aprovechada
Acciones de aprovechamiento	Solicitar a la organización formación y especialización en RPA. Extender el estudio dedicado en la empresa como formación para la realización de este trabajo. Poder emular el proyecto una validación y/o simulación una vez realizado el proyecto
Plan de contingencia	Retrasar otras tareas menos prioritarias al siguiente sprint

Cuadro 5.6: Oportunidad de reutilización de conocimientos

5.2. PRESUPUESTACIÓN INICIAL

Para hacer la valoración de riesgos y oportunidades se ha empleado la siguiente tabla:

NIVEL DE RIESGO Y OPORTUNIDAD		IMPACTO		
		baja	medio	alta
PROBABILIDAD	bajo	bajo	bajo	medio
	medio	bajo	medio	medio
	alto	medio	medio	alta

Cuadro 5.7: Tabla de clasificación de riesgos

5.2. Presupuestación inicial

A continuación se va a describir la presupuestación inicial de este proyecto. Para ello se han consultado las fuentes [22] y [23] para obtener una estimación del salario de un programador junior en la ciudad en la que se desarrolla este proyecto (Valladolid, España).

CALENDARIO y PRESUPUESTO DE RECURSOS HUMANOS	Sólo las del Alumno					
	Fecha Inicio	Fecha Final	Participantes	Horas Estimadas	Horas reales	Coste Personal
Matriculación e inicio de cuatrimestre	feb-22	feb-22	Roberto Gozalo Andrés	0	0	0,00 €
Reunión de arranque del TFG	abr-22	abr-22	Roberto Gozalo Andrés, César Pablo Gutiérrez	2	2	28,52 €
Reuniones de seguimiento de TFG (2 horas)	abr-22	jun-22	Roberto Gozalo Andrés, César Pablo Gutiérrez, Diego García	24	20	285,20 €
Trabajo inicial de estudio: Formación RPA	abr-22	may-22	Roberto Gozalo Andrés	68	70	998,20 €
Trabajo inicial de estudio: Aprendizaje de programación en RPA con Automation Anywhere	abr-22	may-22	Roberto Gozalo Andrés	60	90	1.283,40 €
Introducción, resumen en memoria y estado del arte	abr-22	abr-22	Roberto Gozalo Andrés	16	20	285,20 €
Definición y gestión de los requisitos	abr-22	abr-22	Roberto Gozalo Andrés	36	40	570,40 €
Análisis funcional	may-22	may-22	Roberto Gozalo Andrés	80	80	1.140,80 €
Diseño funcional	may-22	jun-22	Roberto Gozalo Andrés	80	80	1.140,80 €
Desarrollo de RPA y pruebas	jun-22	jun-22	Roberto Gozalo Andrés	16	16	228,16 €
Elaboración de la memoria	abr-22	jul-22	Roberto Gozalo Andrés	40	40	570,40 €
Presentación de TFG	jun-22	jul-22	Roberto Gozalo Andrés	15	16	228,16 €
				437	474	6.759,24 €

Cuadro 5.8: Calendario y Presupuesto de recursos humanos

En la Tabla 5.8 se puede ver el calendario y presupuesto del proyecto. Como se observa tanto la estimación como las horas realizadas finalmente exceden las 300 horas requeridas para la realización de este TFG.

5.3. ANÁLISIS FINAL DEL COSTE DEL PROYECTO

En la tabla 5.9 se pueden observar los detalles acerca del presupuesto de recursos materiales así como de las diferentes licencias de software necesarias para el desarrollo de este proyecto.

CALENDARIO y PRESUPUESTO DE RECURSOS MATERIAL				
	Fecha Inicio Uso	Fecha Final Uso	Fabricante	Coste Personal
Portátil	feb-22	jul-22	Lenovo	800,00 €
Overleaf (licencia gratis)	abr-22	jul-22	Roberto Gozalo Andrés, César Pablo Gutiérrez	0,00 €
Automation Anywhere (licencia gratis)	mar-22	jul-22	Automation Anywhere	0,00 €
Microsoft Office (incluido Project) estudiantes	mar-22	jul-22	Roberto Gozalo Andrés	149,00 €
Jira/Trello	jun-22	jul-22	Atlassian	35,00 €
				984,00 €

un mes de uso

Cuadro 5.9: Presupuesto de recursos materiales

5.3. Análisis final del coste del proyecto

El presupuesto total del coste del proyecto es el que se describe en la tabla que sigue:

PRESUPUESTO del PROYECTO	Coste Total
Personal	6.759,24 €
Recursos materiales	984,00 €
TOTAL	7.743,24 €

Cuadro 5.10: Presupuesto final del proyecto

En 5.10 se visualiza el presupuesto final del proyecto. El total es la suma de los recursos, más el sueldo por las horas del trabajo realizado por el estudiante. Se ha considerado que el salario que vendría cobrando una persona para la realización de este proyecto es de 14,26 € / hora (Salario medio de un Ingeniero Informático en España).

Capítulo 6

Seguimiento del proyecto

6.1. Introducción

Se va a presentar en las siguientes tablas el seguimiento del proyecto acorde a los 8 sprints que se han definido. Para ello se presenta en forma de tablas con las diferentes historias de usuario que forman este trabajo así como el tiempo que se estimó en cada una de ellas en horas así como las horas que finalmente han sido empleadas en la realización de cada una de ellas. También se presenta el estado de cada historia a la finalización del sprint en curso, el estado puede ser Finalizado/No finalizado.

6.2. Seguimiento de los diferentes sprints

6.2.1. Sprint 1

Sprint 1	Tiempo estimado (horas)	Tiempo empleado (horas)	Estado
PROY01 Hacer la introducción del TFG	4	4	Finalizado
PROY02 Describir el estado del arte	4	4	Finalizado
PROY03 Realizar la planificación (este apartado)	12	12	No Finalizado

Cuadro 6.1: Seguimiento del sprint 1

6.2.2. Sprint 2

Sprint 2	Tiempo estimado (horas)	Tiempo empleado (horas)	Estado
REQ01 Identificar el contenido de los emails recibidos	6	7	Finalizado
REQ02 Determinar el origen de los datos de los emails	6	6	Finalizado
REQ03 Definir el formato de los emails que van a ser recibidos	6	8	Finalizado
PROY03 Realizar la planificación (pendiente de terminar)	0	2	Finalizado

Cuadro 6.2: Seguimiento del sprint 2

6.2.3. Sprint 3

Sprint 3	Tiempo estimado (horas)	Tiempo empleado (horas)	Estado
REQ04 Procesar y extraer la información necesaria para la creación de incidencias en el sistema gestor	6	6	Finalizado
REQ05 Realizar la conexión del RPA con Jira	6	6	Finalizado
REQ06 Crear la incidencia con los datos extraídos del correo	6	6	Finalizado
AN01 Identificar el contenido de los emails recibidos	12	8	Finalizado
AN02 Determinar el origen de los datos de los emails	15	17	Finalizado
AN03 Definir el formato de los emails que van a ser recibidos	11	12	Finalizado

Cuadro 6.3: seguimiento del sprint 3

6.2.4. Sprint 4

Sprint 4	Tiempo estimado (horas)	Tiempo empleado (horas)	Estado
AN04 Procesar y extraer la información necesaria para la creación de incidencias	20	24	Finalizado
AN05 Realizar la conexión del RPA con Jira	5	5	Finalizado
AN06 Crear la incidencia con los datos extraídos del correo	12	15	Finalizado
AN07 Enviar el correo resumen con las incidencias que han sido creadas	10	12	Finalizado
DIS01 Identificar el contenido de los emails recibidos	10	10	Finalizado
DIS02 Determinar el origen de los datos de los emails	15	16	Finalizado
DIS03 Definir el formato de los emails que van a ser recibidos	12	12	Finalizado

Cuadro 6.4: seguimiento del sprint 4

6.2.5. Sprint 5

Sprint 5	Tiempo estimado (horas)	Tiempo empleado (horas)	Estado
DIS04 Procesar y extraer la información necesaria para la creación de incidencias	22	20	Finalizado
DIS05 Realizar la conexión del RPA con Jira	5	6	No Finalizado
DIS06 Crear la incidencia con los datos extraídos del correo	11	11	Finalizado
DIS07 Enviar el correo resumen con las incidencias que han sido creadas	10	10	Finalizado
IMP01 Identificar el contenido de los emails recibidos	15	15	Finalizado
IMP02 Determinar el origen de los datos de los emails	25	25	Finalizado
IMP03 Definir el formato de los emails que van a ser recibidos	35	35	Finalizado

Cuadro 6.5: seguimiento sprint 5

6.2.6. Sprint 6

Sprint 6	Tiempo estimado (horas)	Tiempo empleado (horas)	Estado
IMP04 Procesar y extraer la información necesaria para la creación de incidencias	35	35	No Finalizado
IMP05 Realizar la conexión del RPA con Jira	20	25	Finalizado
IMP06 Crear la incidencia con los datos extraídos del correo	20	14	No Finalizado
IMP07 Enviar el correo resumen con las incidencias que han sido creadas	20	22	Finalizado
PRU01 Desarrollar los casos de prueba y probar el segundo sprint	13	15	Finalizado
DIS05 Realizar la conexión del RPA con Jira (pendiente de terminar)	0	2	Finalizado

Cuadro 6.6: seguimiento sprint 6

6.2.7. Sprint 7

Sprint 7	Tiempo estimado (horas)	Tiempo empleado (horas)	Estado
PRU02 Probar el tercer sprint	13	14	Finalizado
PROY04 Hacer la presentación PPT de defensa	10	10	Finalizado
IMP04 Procesar y extraer la información necesaria para la creación de incidencias	0	6	Finalizado
IMP06 Crear la incidencia con los datos extraídos del correo	0	5	Finalizado

Cuadro 6.7: seguimiento sprint 7

6.2.8. Sprint 8

Sprint 8	Tiempo estimado (horas)	Tiempo empleado (horas)	Estado
PROY05 Revisar la documentación por los tutores	5	12	Finalizado

Cuadro 6.8: seguimiento sprint 8

Capítulo 7

Análisis

7.1. Introducción

A continuación se presenta el documento de análisis correspondiente a este trabajo, el principal objetivo del modelo de análisis es capturar el comportamiento del sistema [33]. Se presentará una descripción de los diferentes casos de uso apoyándose en el diagrama de casos de uso UML [25]. También se presentará el modelo de dominio correspondiente a la solución implementada así como el diagrama de clases aplicando BCE [31] [32] y la realización de los diferentes casos de uso identificados mediante sus correspondientes diagramas de secuencia.

7.2. Casos de Uso

En la figura 7.1 se presentará el diagrama de casos de uso correspondiente a la realización de este proyecto:

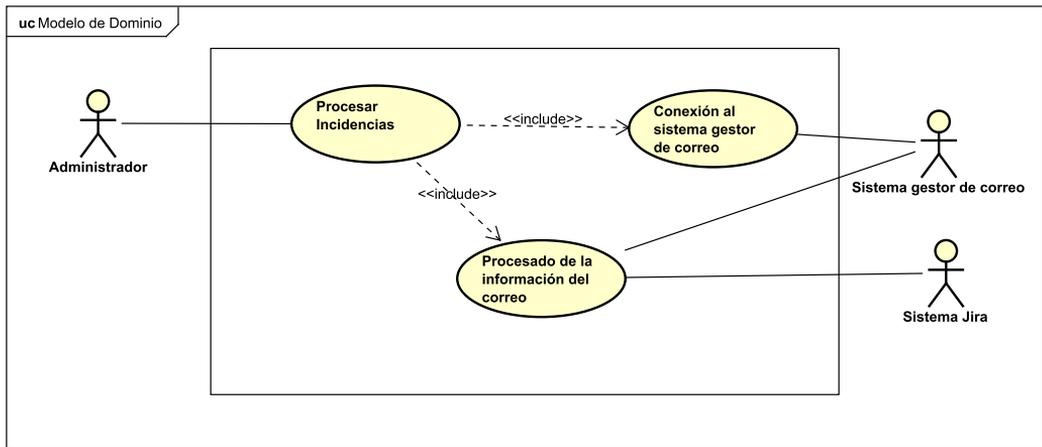


Figura 7.1: Diagrama de casos de uso.

Como se puede apreciar en el diagrama 7.1, los actores *sistema gestor de correo* y *Sistema Jira* son actores secundarios. Este diagrama consta de tres casos de uso. Un caso de uso principal que es el de *Procesar Incidencias* que incluye el caso de uso de *conexión al sistema gestor de correo* y que incluye también al caso de uso *Procesado de la información del correo*.

A continuación, en las diferentes subsecciones, se describen los casos de uso, con la siguiente estructura:

- **Caso de uso:** El nombre del caso de uso.
- **Actor:** Los actores involucrados en el caso de uso.
- **Precondición:** Aquellas condiciones que se deben cumplir antes de realizar el caso de uso.
- **Escenario:** El escenario principal del caso de uso.
- **Flujo alternativo:** Las posibles alternativas que puede tomar el escenario principal del caso de uso.
- **Postcondición:** Las condiciones que cumplirá el caso de uso al finalizar el escenario.
- **Dependencias:** Los requisitos con los que tiene trazabilidad el caso de uso.

7.2.1. Caso de Uso: Procesar Incidencias

Caso de uso	Procesar Incidencias
Actor	Administrador
Precondición	El administrador debe estar logeado en el sistema
Escenario	1. El caso de uso comienza cuando el administrador ejecuta el bot. 2. «Punto de inclusión» Conexión al sistema gestor de correo. 3. «Punto de inclusión» Procesado de la información del correo. 4. Por cada incidencia: Se añade la incidencia al sistema Jira
Flujo alternativo	1. No existe ningún correo con incidencias El sistema marcará como leídos los correos (si los hubiera) y finaliza el caso de uso.
Postcondición	Se crean las incidencias correspondientes en el sistema Jira
Dependencias	REQ05 y REQ06

Cuadro 7.1: Caso de uso Procesar incidencias

7.2.2. Caso de Uso: Conexión al sistema gestor de correo

Caso de uso	Conexión al sistema gestor de correo
Actor	Administrador y Sistema Gestor de Correo
Precondición	Se debe haber iniciado el caso de uso Procesar Incidencias
Escenario	1 El sistema RPA introduce el usuario y la contraseña de la dirección de correo. 2 Se establece la conexión con el sistema de correo electrónico. 3 El caso de uso termina.
Flujo alternativo	1.Si en el paso 1 el usuario o la contraseña son incorrectos. El caso de uso termina.
Postcondición	La sesión debe mantenerse iniciada
Dependencias	REQ01, REQ02 y REQ03

Cuadro 7.2: Caso de uso Conexión al sistema gestor de correo

7.2.3. Caso de Uso: Procesado de la información del correo

Caso de uso	Procesado de la información del correo
Actor	No existen ya que se arranca con la inclusión de 7.1
Precondición	1. Se debe de haber iniciado el caso de uso Procesar Incidencias 2. La sesión de correo electrónico debe estar iniciada
Escenario	Por cada correo No Leído: 1. Se lee la información del email. 2. Se marca el correo como leído. Si el correo cumple el formato establecido. 3. Se introduce la información en cada una de las variables del sistema. 4. Se realiza la llamada de creación de la incidencia. 5. El caso de uso termina.
Flujo alternativo	1. El correo no cumple con el formato establecido. 2. Se marca el correo como leído El caso de uso termina.
Postcondición	Ninguna
Dependencias	REQ04, REQ05 y REQ06

Cuadro 7.3: Caso de uso Procesado de la información del correo

7.3. Modelo de Dominio

Una vez estudiados y analizados los requisitos de este trabajo se presenta el modelo de dominio de la imagen 7.2, correspondiente al sistema implementado como solución:

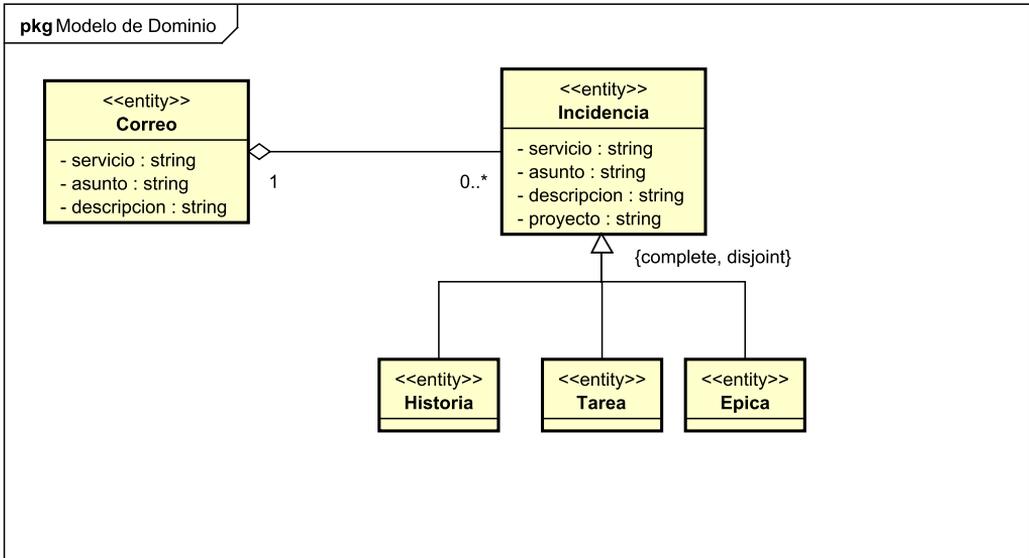


Figura 7.2: Diagrama de clases de dominio.

En el diagrama 7.2, la relación de generalización - especificación es completa, ya que una incidencia no puede ser de ningún otro tipo que no sea historia, tarea o épica y disjunta, ya que una incidencia solamente puede ser de uno de los tres tipos.

También se puede observar que en un correo pueden aparecer cero, una o varias incidencias y que éstas pueden seguir existiendo después de que dicho correo desaparezca del sistema (relación de agregación entre Correo e Incidencia).

7.3.1. Clases de dominio

A continuación se describirán las principales clases que contiene el modelo de dominio del sistema:

- Correo:** Esta clase representa los diferentes correos que recibe el sistema y que contienen en su cuerpo la descripción de las diferentes incidencias que van a ser insertadas en el sistema gestor de incidencias [5]

- **Incidencia:** Esta clase representa las diferentes incidencias que se insertan en el sistema.

Historia: Representa las incidencias de tipo historia

Tarea: Representa las incidencias de tipo tarea

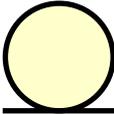
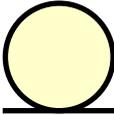
Épica: Representa las incidencias de tipo épica

7.4. Modelo de análisis

7.4.1. Clases de análisis

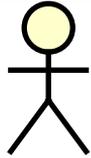
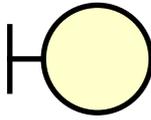
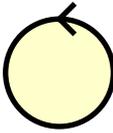
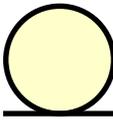
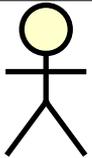
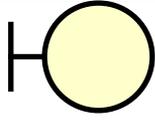
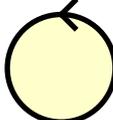
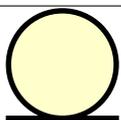
Para la detección de las clases de análisis se ha empleado la técnica de detección de clases de análisis BCE [31] [32] ya que es la que se empleó en la asignatura de Diseño, Integración y Adaptación del Software [33]. Esta técnica define el comportamiento de los casos de uso estableciendo tres tipos de clases:

- **Boundary:** Son aquellas clases encargadas de la interacción con los actores. Trasmiten las acciones del actor a las clases control
- **Control:** Son aquellas clases que realizan, organizan y gestionan el caso de uso. Reciben la información de las clases Boundary y almacenan información en las clases Entity
- **Entity:** Son las clases encargadas de almacenar información persistente. Estas clases corresponden a las diferentes clases de dominio que aparecen en 7.2

Estereotipo	Icono	Semántica
<<boundary>>		Mediadora entre el sistema y su entorno.
<<control>>		Encapsula el comportamiento descrito en los casos de uso
<<entity>>		Modela información persistente (o no).

Cuadro 7.4: Clases BCE. Fuente: [33]

En la imagen 7.5 se muestra un cuadro resumen de las interacciones que pueden tener las diferentes clases BCE

Cuadro 7.5: Interacciones de las clases BCE. Fuente: [33]

Como se aprecia en la figura 7.5 Las únicas clases que pueden relacionarse con los actores son las clases Boundary, éstas a su vez se relacionan con las clases control y estos con las clases Entity que pueden relacionarse con otras clases Entity.

El diagrama de clases BCE resultante se muestra en la figura 7.3 siendo todas las clases del modelo de dominio 7.2 clases de tipo Entity. Además aparece una clase Controller denominada *Controlador RPA* [10] así como tres clases de tipo Boundary, ya que existen tres actores que necesitan interactuar con el sistema.

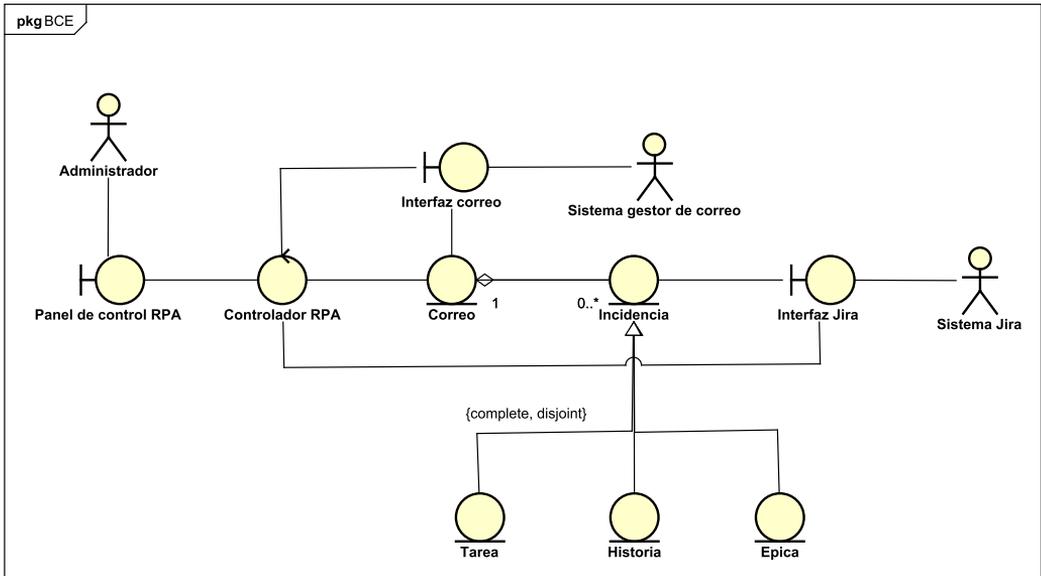


Figura 7.3: Diagrama de clases de análisis.

7.4.2. Realización de casos de uso de análisis

La figura siguiente 7.4 corresponde al desarrollo del diagrama de secuencia del caso de uso correspondiente al procesado de incidencias en el que el sistema RPA interactúa con el sistema Jira para crear en él las diferentes incidencias. Como se indica en 7.1 este caso de uso tiene dos puntos de inclusión que se pueden ver reflejados en esta figura.

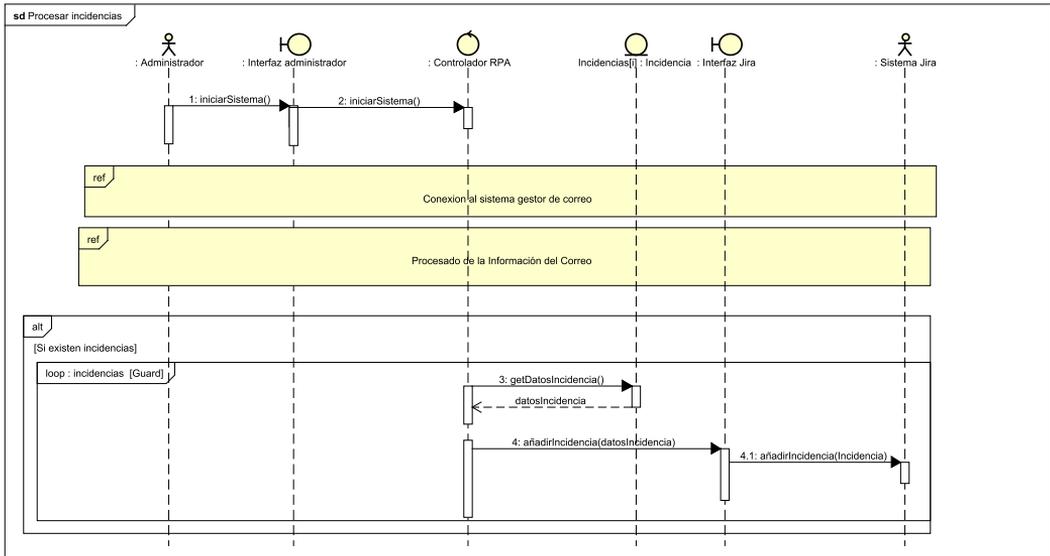


Figura 7.4: Diagrama de secuencia del caso de uso del procesado de incidencias

A continuación en la figura 7.5 se observa el diagrama de secuencia del caso de uso correspondiente al proceso de conexión entre el sistema RPA y el sistema gestor de correo. Cabe destacar que este proceso a diferencia de los otros casos de uso, requiere de la intervención humana (representada en el diagrama por el actor Administrador). También se puede apreciar la ausencia de una clase entidad, este hecho está justificado ya que el login y la contraseña son los pertenecientes al sistema gestor de correo.

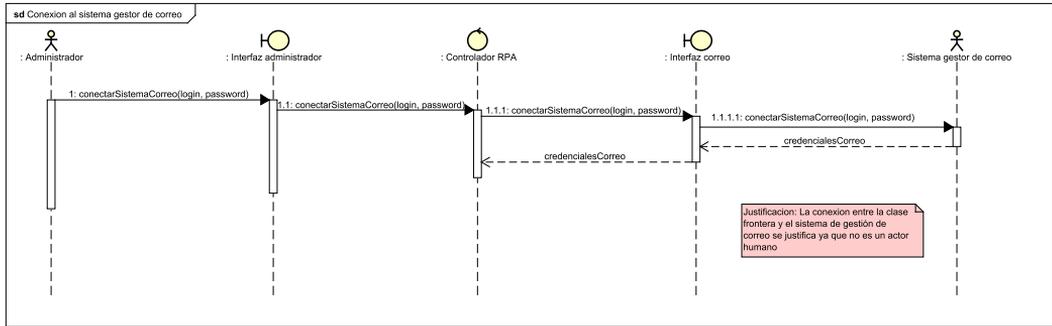


Figura 7.5: Diagrama de secuencia del caso de uso de la conexión al sistema gestor de incidencias

La figura 7.6 muestra el diagrama de secuencia correspondiente al procesado, por parte del sistema RPA, de la información recibida por correo electrónico y la generación de la incidencia correspondiente. Solo se crearán incidencias si se reciben correos con el formato establecido, descartando así el resto como puedan ser correos publicitarios, por ejemplo.

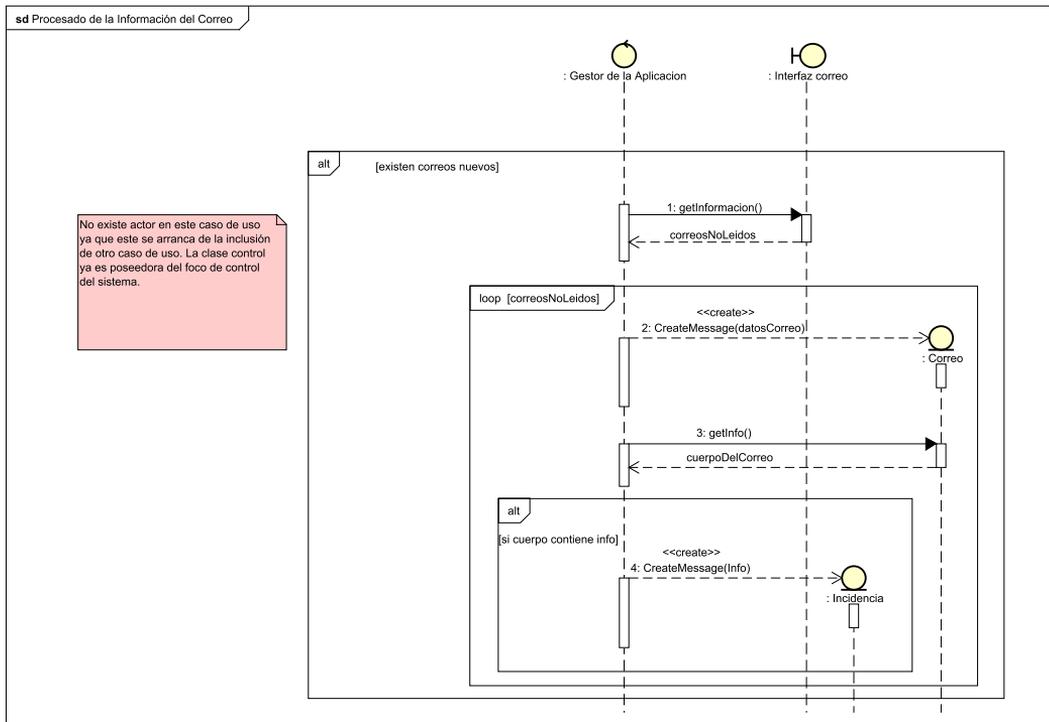


Figura 7.6: Diagrama de secuencia del caso de uso del procesado de la información recibida mediante correo electrónico

Capítulo 8

Diseño

8.1. Introducción

Este documento describe y desarrolla el modelo de diseño, que es un modelo centrado en el dominio de la solución. En él se describe como se implementará la funcionalidad acorde a los requisitos. Para ello se describirán la arquitectura física y lógica del proyecto así como los patrones empleados en su implementación, tanto estructurales como de diseño, apoyándose de los diagramas pertinentes en cada caso. También se describirá el modelo de comportamiento del sistema.

8.2. Arquitectura física

La figura 8.1 muestra el diagrama de despliegue que modela la arquitectura física del sistema; que es un diagrama de despliegue UML [25] que especifica el hardware físico sobre el que el software implementado se ejecutará. Este diagrama especifica como el software se despliega en el sistema, mapea la arquitectura de software creada sobre una arquitectura física del sistema que lo ejecutará, permite tener una visión del hardware del que se dispone donde se desplegará el software desarrollado y permite modelar la topología del hardware [33]

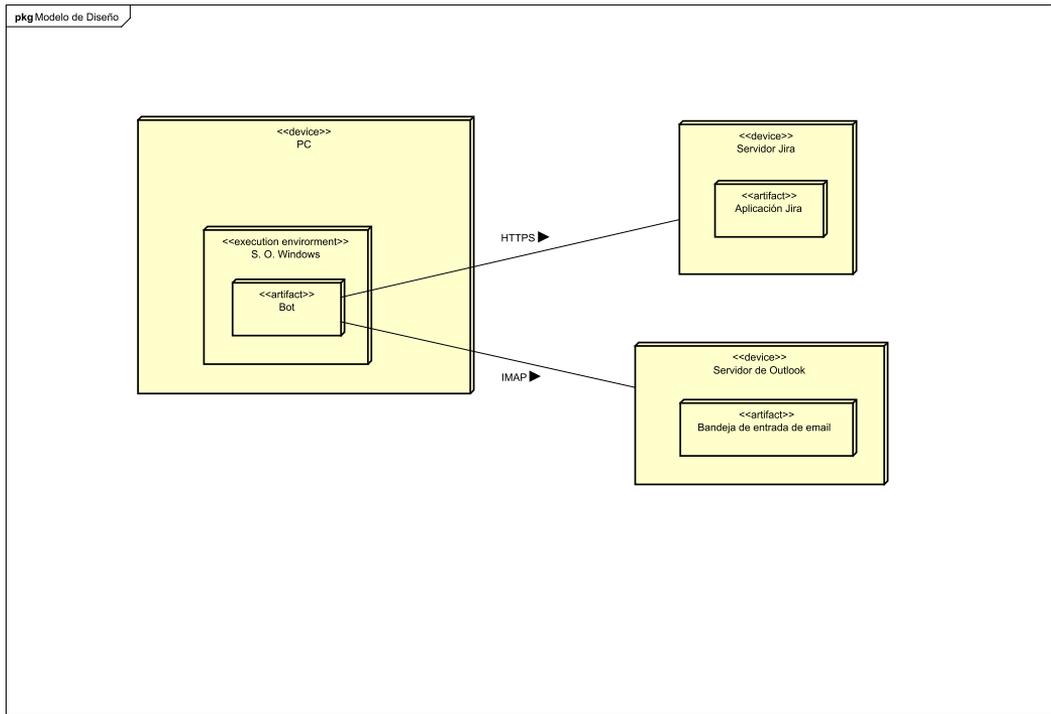


Figura 8.1: Modelo de despliegue de la arquitectura física del sistema.

8.3. Arquitectura lógica: patrón arquitectónico

La fase de diseño es la que se encarga de planear y brindar un modelo de solución al problema expuesto en la fase de análisis, buscando cumplir una serie de objetivos y requisitos, además de estar expuesto a una serie de restricciones, también otorga a los modelos y diagramas un mayor refinamiento que hace que sean más cercanos a la realidad. Permite planificar la futura implementación del sistema software que otorgue la solución al problema que se ha propuesto [33].

Lo común es que en la fase de diseño se utilicen soluciones con patrones que se adapten al sistema software que se está diseñando. Al igual que en la fase de análisis para una mayor claridad se empleó el patrón BCE [31], en la fase de diseño se ha seleccionado el patrón arquitectónico Filtro Tubería, ya que, es el patrón que se ha considerado que mejor se ajusta a la solución planteada [34].

El patrón Filtro Tubería consiste en el establecimiento de dos elementos, filtros y tuberías para representar el flujo de datos del sistema en la ejecución de una tarea que ha sido dividida en subtareas. Cada filtro representa una de las subtareas, y las tuberías la transmisión de datos entre ellas. Suele utilizarse para representar el flujo de los datos durante una tarea de conversión de datos, como puede ser el caso de los compiladores, ya que suele ser el patrón por excelencia en su implementación, como se puede apreciar en la figura 8.2.

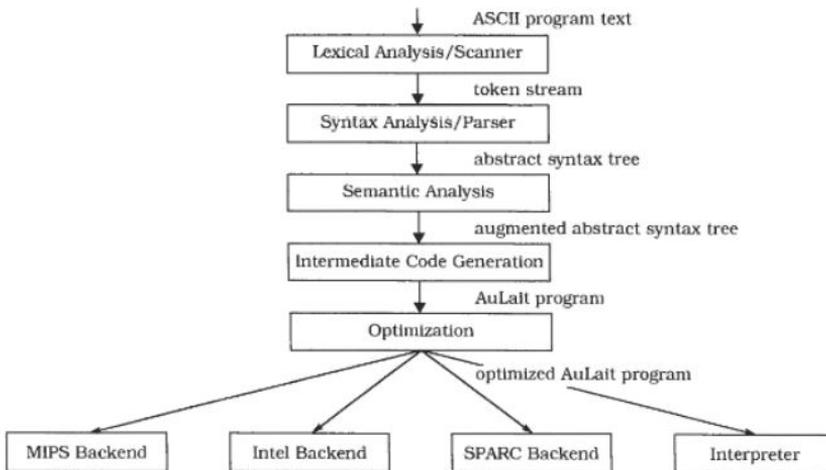


Figura 8.2: Patrón Filtro Tubería aplicado al compilador de C. Fuente [34]

Aplicando dicho patrón al sistema objeto de este trabajo, como se muestra en la figura 8.3. Se establece la primera tubería correspondiente al proceso de inicio de la ejecución del bot. El primer filtro sería el orquestador. Posteriormente, se enviaría la información de ejecución junto con el archivo *configFile.xml* al filtro readConfig este mediante una tubería envía la información del contenido del archivo al filtro readOutlook. A continuación, este envía mediante una tubería la información de la incidencia al filtro createIssue y este finalmente envía la información de la incidencia al sistema gestor de incidencias.

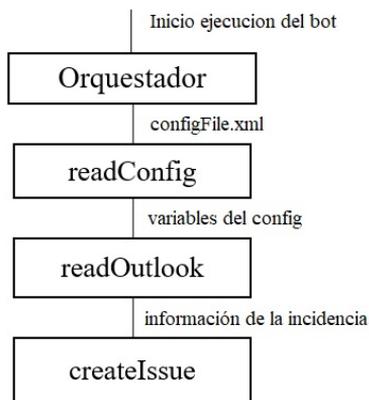


Figura 8.3: Patrón filtro tubería aplicado al sistema. Fuente elaboración propia

8.4. Patrones de diseño

Para el modelado de las diferentes partes que forman la solución se han empleado diferentes patrones de diseño que son aquellos que proporcionan un esquema para refinar los subsistemas de un sistema software o las relaciones entre ellos. Estos patrones describen una estructura de uso recurrente entre subsistemas que resuelve un problema de diseño general dentro de un contexto particular. [33]

8.4.1. Patrones de diseño para la lógica de dominio

Como patrón de diseño para modelar la lógica de dominio o negocio, se ha optado por usar el patrón *Transaction Script* que es aquel patrón que organiza la lógica de negocio como un conjunto de procedimientos donde cada procedimiento maneja una sola solicitud de la capa de presentación, si la hubiese.

En el contexto de la informática, una transacción es una operación, normalmente formada

por varios procesos, que se trata como una operación atómica, es decir, que se completa completamente o no se completa en absoluto y deja el sistema en un estado consistente.

Este patrón está indicado para casos en los que el workflow de Análisis es muy sencillo ya que existe muy poca lógica de dominio, por lo que es muy adecuado para el sistema objeto del trabajo. [33].

Cada transacción será la llamada que se realiza con la información necesaria para crear la incidencia en el sistema gestor.

8.4.2. Patrones de diseño para el acceso a datos

Se ha empleado como patrón de acceso a datos el patrón DAO (Data Access Object). El DAO es un patrón arquitectónico que propone el uso de un objeto (el DAO) que centraliza el acceso y almacenamiento en las fuentes de datos. [33]

Las ventajas de usar el patrón DAO son las siguientes:

- Modificar el API (*Application Program Interface*) [28] de acceso fácilmente. Permite cambiar el acceso a la fuente de datos cómodamente. Para ello simplemente habría que modificar las clases DAO para adaptarlas a la nueva fuente.
- Modificar el repositorio de datos sin realizar grandes cambios en el sistema.
- Posibilidad de implementar *triggers* o *listeners*. Al estar todo el código centralizado en las clases DAO se puede implementar fácilmente políticas de seguridad en el acceso al repositorio de datos.

8.5. Realización de casos de uso en el modelo de diseño

La figura 8.4 muestra el diagrama de secuencia de diseño que realiza uno de los casos de uso más representativos de este sistema como es el del procesado de la información del correo descrito en la tabla 7.3 cuyo diagrama de secuencia es 7.6. Como se ha comentado previamente el DAO Acceso al correo se trata como una fuente de datos y el DTO (*Data Transfer Object*) [36] se trata como texto plano. Cabe destacar también, que en la solución propuesta se han modelado tres DAOs, uno para el sistema de registro log, otro para el archivo XML *config* y un tercero, que es el que se observa en este diagrama y que es el referente a los correos electrónicos recibidos.

8.5. REALIZACIÓN DE CASOS DE USO EN EL MODELO DE DISEÑO

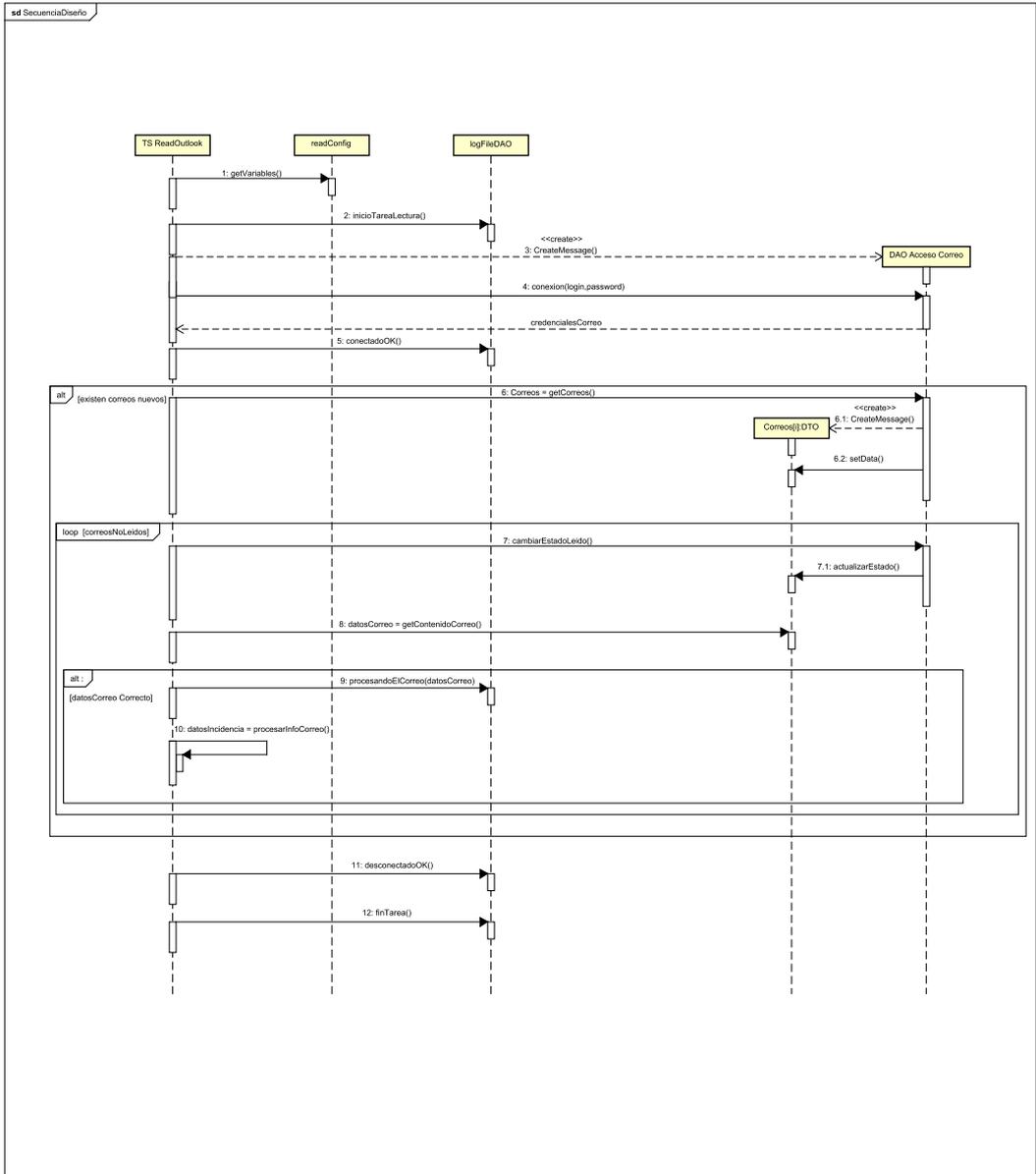


Figura 8.4: Diagrama de secuencia de diseño correspondiente al caso de uso *Procesado de la información del correo 7.3*.

Capítulo 9

Implementación del proyecto

Este documento describe los entornos de desarrollo empleados en este trabajo así como las principales funcionalidades desarrolladas durante su realización.

Por ello para la implementación de este proyecto, como ya se ha comentado a lo largo de la memoria, se ha empleado la herramienta de Automation Anywhere. Se procederá a realizar una descripción detallada de como se ha ido construyendo en esta herramienta el sistema.

9.1. Acciones de Automation Anywhere empleadas

Para comenzar se describen todas las acciones de Automation Anywhere (en adelante AA) que han sido empleadas para el desarrollo de este sistema.

Automation Anywhere (AA) es una empresa líder mundial en automatización robótica de procesos (RPA) e inteligencia artificial (AI). Con su solución de software, ofrecen la plataforma y la comunidad RPA más ampliamente extendida permitiendo a las empresas automatizar los procesos de principio a fin por parte de todos los miembros de la organización.

AA permite mediante su plataforma la implementación y programación de robots (o bots) de forma gráfica e intuitiva mediante una serie de lo que se denomina acciones. Las principales acciones son:

- Acciones del tipo Task Bot
- Acciones de tipo XML
- Acciones de control
- Acciones de Log
- Acciones de Email

- Acciones de String
- Acciones REST

En las siguientes subsecciones se describirán cada una de ellas en el orden en el que han ido apareciendo en el desarrollo de la solución propuesta para la realización de este trabajo de fin de grado.

9.1.1. Task Bot

El Task Bot es la unidad base, es cada una de las partes en las que se divide nuestro sistema y a las que se puede acceder a través de la control room que es como AA denomina a la web desde la que podemos controlar todos nuestros bots.

Task Bot: Run

Runs the selected task bot.

Required bot agent version: 20.11 or above

Task Bot to run

Current Task Bot | Control Room file | Variable

Bots\jiraAutomatization\00_readConf | X | Choose...

Input values ↻

This bot has no input values

Repetition (optional)

Do not repeat ▾

Delay between repetitions

Minutes (optional)

##

Seconds (optional)

##

Upon error continue with next repetition or action in this bot

Save the outcome to a variable (optional)

Multiple variables | Dictionary

Mapped variables ↕

Key	Map to variable	
strEmail	strEmail	⋮
strPasswordEmail	strPasswordEmail	⋮
strEmailReport	strEmailReport	⋮
strJiraAddress	strJiraAddress	⋮
strJiraUser	strJiraUser	⋮

Figura 9.1: Ejecucion del Task Bot. Fuente [1]

9.1.2. XML start session

Esta acción inicia una sesión XML los parámetros que recibe son el nombre de la sesión (en nuestro caso Default) y la fuente de datos que debe abrir, en nuestro caso el archivo config.xml Es necesario iniciar una sesión XML para luego poder leer o escribir en el archivo xml al que deseamos acceder.

XML: Start session

Start xml session
Required bot agent version: 20.11 or above

Session name
 (x)

Data Source
 File

(x)

Required extension: ".xml"

Text

Figura 9.2: Iniciar sesión XML. Fuente [1]

9.1.3. XML get single node

Con esta acción se asigna una de las variables del fichero XML a una variable del sistema RPA. Para ello se debe indicar como parámetros la sesión XML en la que se desea trabajar, la ruta de la variable en el XML, los atributos, si existieran y la variable del entorno RPA a la que se quiere asignar ese valor.

XML: Get single node

Fetch value of a specific node from the xml
Required bot agent version: 20.11 or above

Session name
 (x)

XPath expression
 (x)
e.g. /bookstore/book/title

Attribute (optional)
 (x)

Assign the output to variable
 (x)

Figura 9.3: Obtener datos de XML. Fuente [1]

9.1.4. Try/Catch

Al igual que en otros lenguajes de programación, se emplea esta acción para el control y la gestión de errores. En este caso, en el bloque catch se asignan los tipos de errores que se quieren capturar. La variable donde se debe guardar el mensaje de error y la variable donde almacenar el número de línea donde se ha producido el error o la excepción. Ambas variables son opcionales.

Error handler: Catch

Run a sequence of commands if the commands in Try fail with an exception.
Required bot agent version: 20.11 or above

Exception

Error handler
AllErrors

Indicates a generic bot exception when a bot is run
Code: AA-BotException-111-111

Assign exception message to (optional)

prompt-assignment (x)

Assign line number to (optional)

numErrorLineNumber (x)

Figura 9.4: Bloque Try/Catch. Fuente [1]

9.1.5. Log to File

En este sistema se van a registrar los eventos principales en un log. Para ello se emplea esta acción de AA. En ella se ha de especificar la ruta del archivo en el que se va a escribir, el mensaje que deseamos registrar, si deseamos añadir un timestamp [26] si queremos añadirlo al final del archivo o sobrescribir este y la codificación que deseamos emplear (en este caso UTF8 [27]).

Log to file

Logs any text into a file

Required bot agent version: 20.11 or above

File path

” \$strPathMainLog\$ (x) Browse...

Enter text to log

” -----> [ERROR] \$strErrorLineNumber\$ Error en el l
----->Mensaje de error: \$prompt-assignment\$ (x)

Append timestamp

When logging

Append to existing log file

Overwrite existing log file

Encoding

UTF8

Figura 9.5: Añadir texto al log. Fuente [1]

9.1.6. Message Box

Es la acción que se emplea en AA para lanzar un mensaje a través de una ventana emergente. En ella se permite especificar el nombre de la ventana, el mensaje a mostrar, el número de líneas que se muestran antes de tener que desplazar el texto y si se desea que la ventana se cierre automáticamente después de un tiempo especificado y el valor de dicho tiempo.

Message box

Displays a message box
Required bot agent version: 20.11 or above

Enter the message box window title

Automation Anywhere Enterprise Client (x)

Enter the message to display

ERROR: \$prompt-assignment\$
LINE: \$strErrorLineNumber\$ (x)

Scrollbar after lines

30 (x)

Close message box after

Seconds

5 (x)

Figura 9.6: Ventana emergente. Fuente [1]

9.1.7. Task Bot: Run

Para ordenar la ejecución de un Task Bot se emplea la tarea Task Bot: Run en ella, se debe especificar el bot a ejecutar, si se desea que se repita dicha ejecución, si se quiere que exista un cierto retraso entre las diferentes ejecuciones si se desea que la ejecución de las siguientes veces continúe a pesar de que hayan ocurrido errores y las variables que se quieren mapear provenientes de ese bot.

Task Bot: Run

Runs the selected task bot.

Required bot agent version: 20.11 or above

Task Bot to run

Current Task Bot | Control Room file | Variable

Bots\jiraAutomatization\00_readConf | X | Choose...

Input values ↻

This bot has no input values

Repetition (optional)

Do not repeat ▼

Delay between repetitions

Minutes (optional)

##

Seconds (optional)

##

Upon error continue with next repetition or action in this bot

Save the outcome to a variable (optional)

Multiple variables | Dictionary

Mapped variables ↕

Key	Map to variable	
strEmail	strEmail	⋮
strPasswordEmail	strPasswordEmail	⋮
strEmailReport	strEmailReport	⋮
strJiraAddress	strJiraAddress	⋮
strJiraUser	strJiraUser	⋮

Figura 9.7: Ejecucion de un task bot. Fuente [1]

9.1.8. Email: Connect

Si se desea realizar operaciones con un gestor de correo electrónico se ha de emplear esta acción, en ella se especifica el nombre de la sesión (en este caso EmailSession) a iniciar, así como la bandeja de entrada de Outlook empleada.

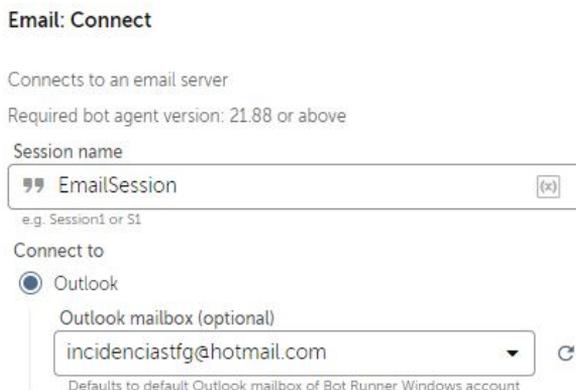


Figura 9.8: Conectar sesion de correo. Fuente [1]

9.1.9. Email: Change Status

Esta acción se emplea para cambiar el estado de los correos procesados de no leídos a leídos. Para ello se indica la sesión de correo en la que se está trabajando y al estado al que se desea cambiar los correos (en este caso leídos).

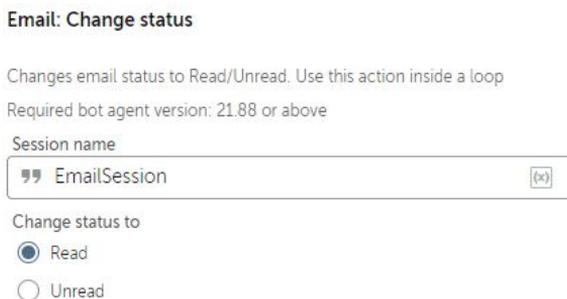


Figura 9.9: Modificar el estado de los correos. Fuente [1]

9.1.10. String: Assign

Es la acción empleada en AA para asignar un valor a un String en ella se debe especificar el valor así como el String de destino.

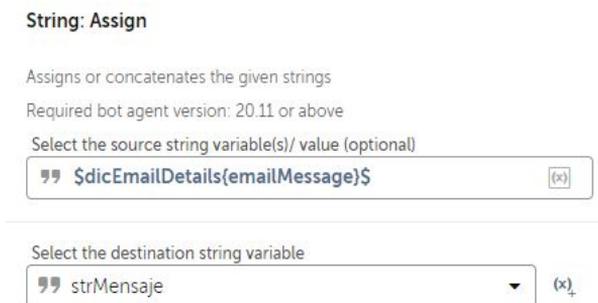


Figura 9.10: Asignar un valor a una variable. Fuente [1]

9.1.11. String: Extract text

Acción utilizada para extraer una porción de texto de un String. En ella se debe especificar el String del que se quiere extraer el texto. Si se quiere extraer el texto antes o antes y después del texto especificado y el número de veces que ha de aparecer este para que se extraiga el texto. Además se puede seleccionar si se desea eliminar los espacios en blanco así como los retornos de carro. Por último se ha de especificar el String donde almacenar el texto extraído.

String: Extract text

After.

Required bot agent version: 20.11 or above

Source string (optional)

☞ \$strMensaje\$ (x)

Get characters

Before

Start after text (optional)

☞

Occurrence

##

Before and/or after

Start after text (optional)

☞ Servicio/App: (x)

Occurrence

1 (x)

OR

End before text (optional)

☞ Asunto (x)

Occurrence

1 (x)

E.g. to extract \$250 from "Price is \$250 today" specify "Start after text" as "Price is" and "End before text" as "today". Tip: Press F2 to enter variables

After

End before text (optional)

☞

Occurrence

##

If no match found, return

Source String

Empty (null) String

Number of characters to get

All

Only

##

Trim the extracted text (remove blank spaces)

Remove Enter from the extracted text

Assign the output to variable

☞ strServicio (x)

Figura 9.11: Extraer una porción de un string. Fuente [1]

9.1.12. Email: Disconnect

Antes de terminar la ejecución del bot hay que desconectarse de la sesión de correo previamente iniciada, para ello se emplea esta acción y en ella, simplemente, se selecciona el nombre de la sesión a desconectar.

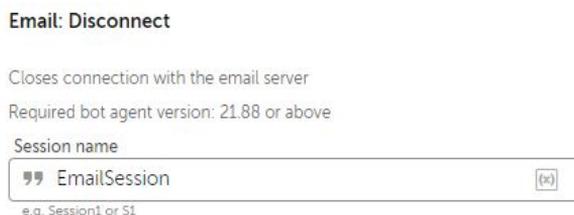


Figura 9.12: Finalizar la sesion de correo. Fuente [1]

9.1.13. String: Lenght

Acción empleada para obtener la longitud de una cadena de texto. En ella debemos especificar el String del que queremos conocer su longitud y la variable en la que deseamos guardar dicha información.



Figura 9.13: Obtener longitud de una cadena de texto. Fuente [1]

9.1.14. String: Substring

Esta acción sirve para extraer una porción de texto de una longitud determinada de una cadena de texto. En ella se especifica el String del que se desea extraer el texto, la posición desde la que se empieza a obtener el substring y la longitud de este, si no se selecciona ninguna se extrae hasta el final del string original. También se debe especificar la variable en la que almacenar el texto extraído.

String: Substring

Extracts a sub-string from a given string.
Required bot agent version: 20.11 or above

Source string (optional)

☞ \$strDescripcion\$ (x)

Start from

1 (x)

Length (optional)

200 (x)

Assign the output to variable

☞ strDescripcion (x)

Figura 9.14: Obtener substring. Fuente [1]

9.1.15. REST Web Services. POST

Para realizar una llamada a una API REST [28] se emplea esta tarea. En este caso, se realiza una llamada POST para crear la incidencia.

Para crear la llamada se debe especificar la URI [29] sobre la que realizar la llamada, el tipo de autenticación que empleada con sus correspondientes credenciales, también, se puede personalizar la cabecera (en este caso no se empleará), el tipo de contenido (en este caso JSON [30]) y por último los parámetros de la llamada.

REST Web Services: Post method

The POST method (Create) adds to the information (in the form of an entity) that is identified by the Request URI.

Required bot agent version: 20.11 or above

URI

Enter the URI

”” \$strJiraAddress\$ [x]

e.g https://domainname.com/resource/search?param1=value1¶m2=value2

Select credential as URI

Credential Variable

Authentication Mode

Basic

Username

Credential Variable Insecure string

”” \$strJiraUser\$ [x]

Password

Credential Variable Insecure string

”” \$strJiraPassword\$ [x]

Header

Custom headers (optional)

Enabled	Name	Value
No headers		

Add header

Content type

JSON (application/json)

Custom parameters

Enter the parameters

```

{
  "fields": {
    "project": {
      "key": "PREMILIO2"
    },
    "summary": "$strAsunto$",
    "description": "$strDescripcion$",
    "issuetype": {
      "name": "Historia"
    },
    "components": [
      {
        "self": "https://jira.luceit.com",
        "id": "16000",
        "name": "componente1"
      }
    ]
  }
}
    
```

Figura 9.15: Realizar llamada POST. Fuente [1]

9.2. Organización de los diferentes Task Bots

A continuación se va a mostrar la organización de los diferentes Task Bots 9.7 que componen el sistema. La herramienta elegida (Automation Anywhere [1]) nos proporciona una vista de "Flow" que aunque no es la que se utiliza durante el desarrollo proporciona una buena aproximación para entender cómo se desarrolla el flujo de nuestro sistema.

En las siguientes subsecciones se comentará los aspectos más destacados de dicho flujo así como la implementación de los subsistemas identificados en la etapa de diseño 8

9.2.1. readConfig

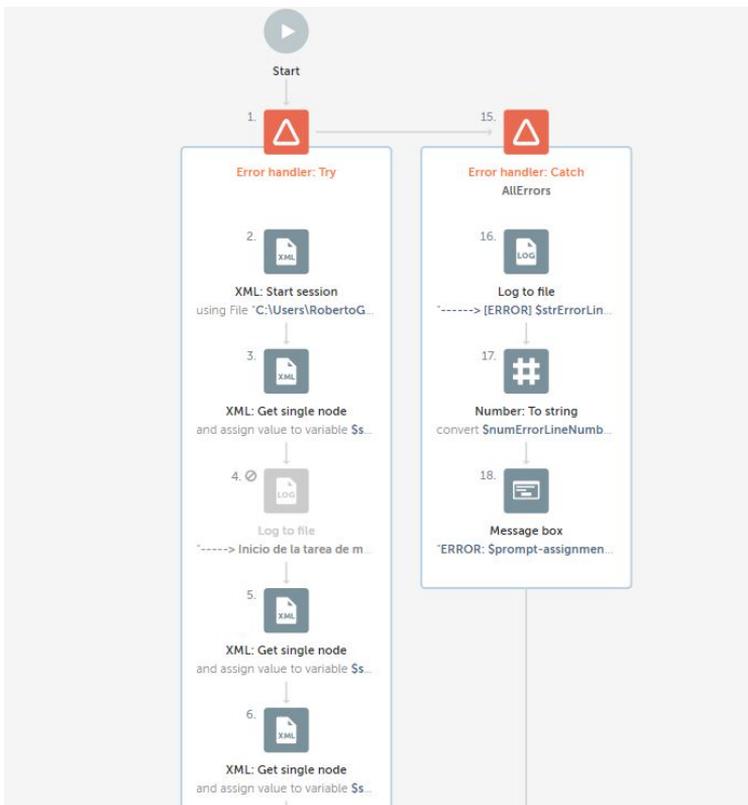


Figura 9.16: Flujo de la lectura del archivo config. Parte 1. Fuente [1]

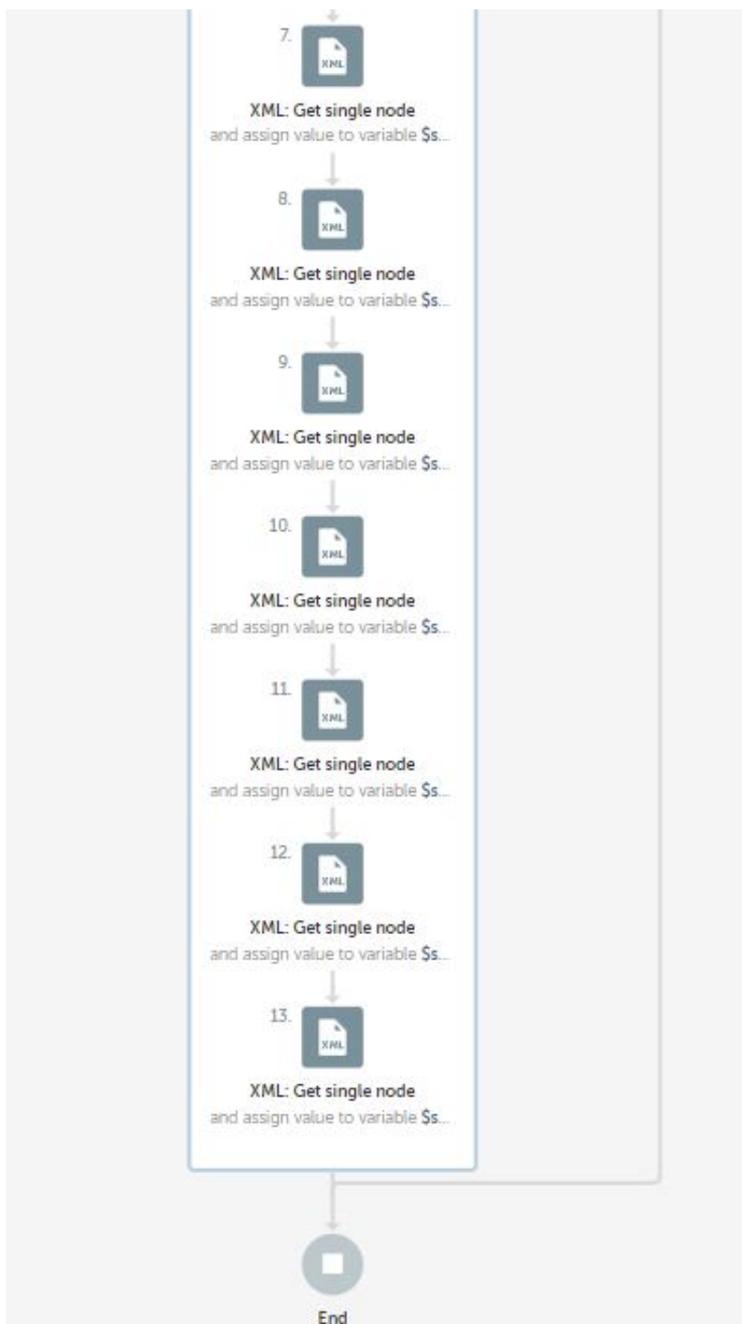


Figura 9.17: Flujo de la lectura del archivo config. Parte 2. Fuente [1]

9.2.2. Orquestador

A continuación se describe el flujo del Orquestador. En este caso solo realiza la llamada al taskBot de lectura del archivo *configFile.xml* y al de la lectura de correos, pero se crea este orquestador pensando en una posible ampliación del proyecto, permitiendo así una escalabilidad del sistema. Creando este orquestador, solo sería necesario ejecutar este Task Bot para iniciar este sistema. Hecho que facilita la programación de las ejecuciones del sistema.



Figura 9.18: Flujo del orquestador. Fuente [1]

9.2.3. readOutlook

A continuación se puede apreciar el flujo del Task Bot encargado de la lectura de los correos:

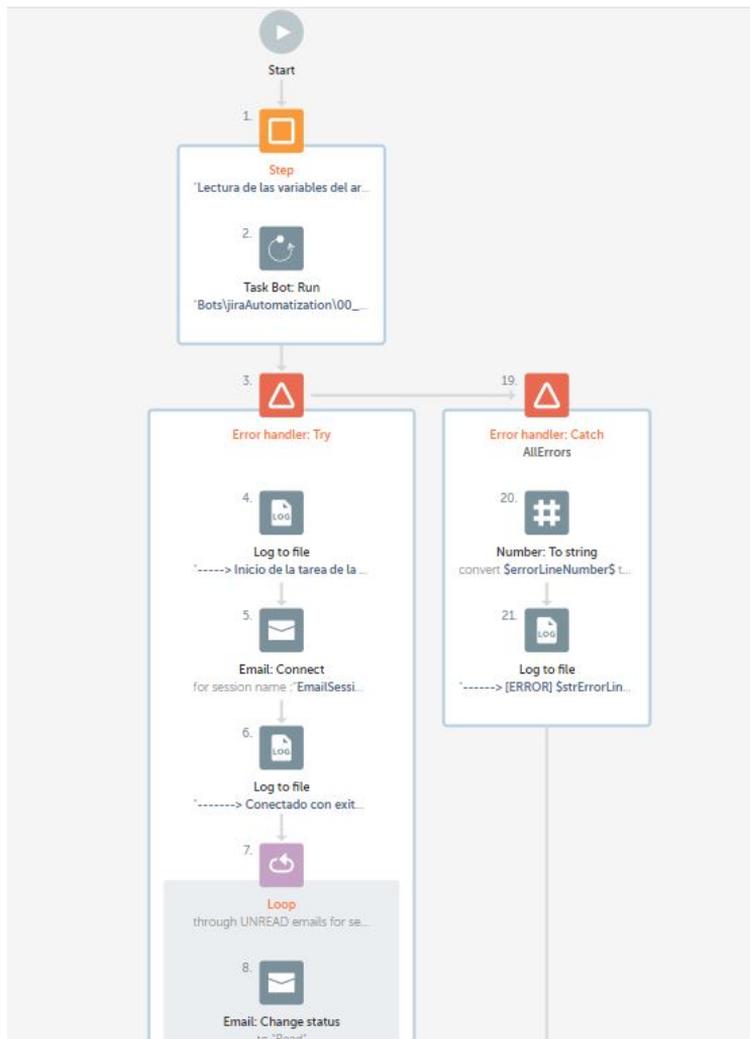


Figura 9.19: Flujo de la lectura de correos.Parte 1. Fuente [1]

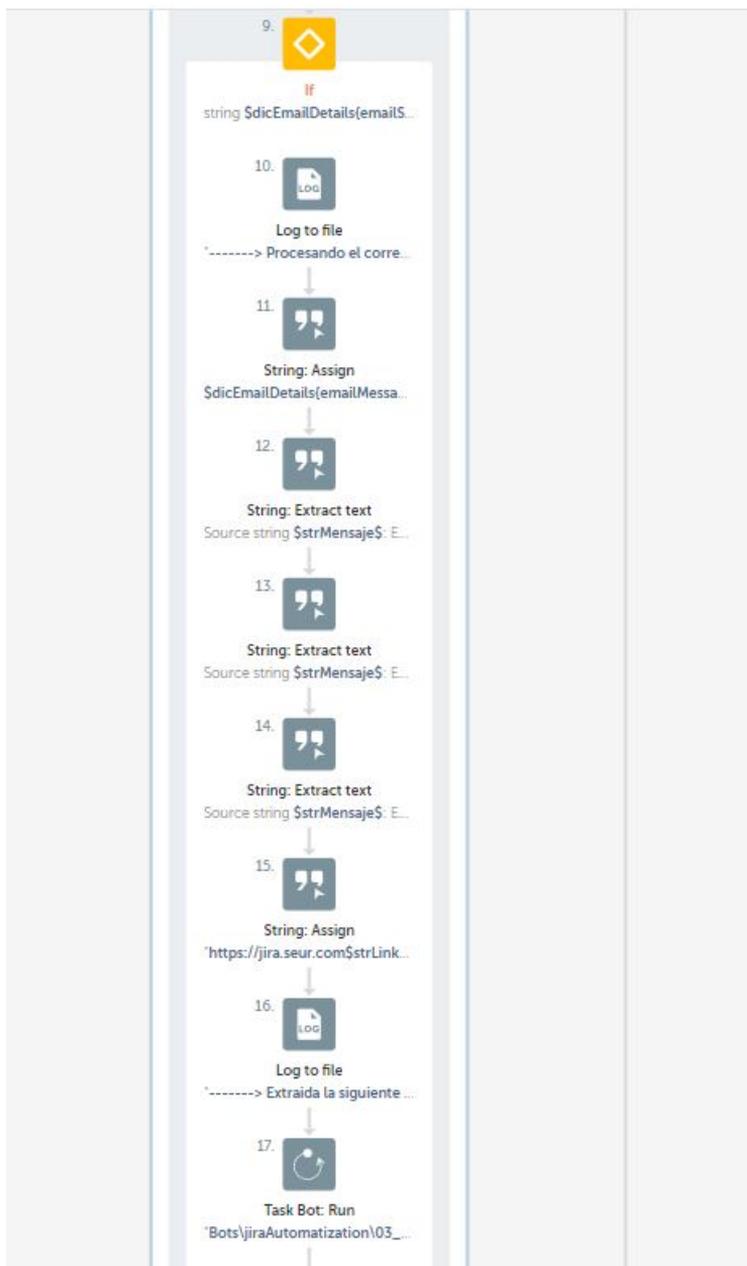


Figura 9.20: Flujo de la lectura de correos.Parte 2. Fuente [1]

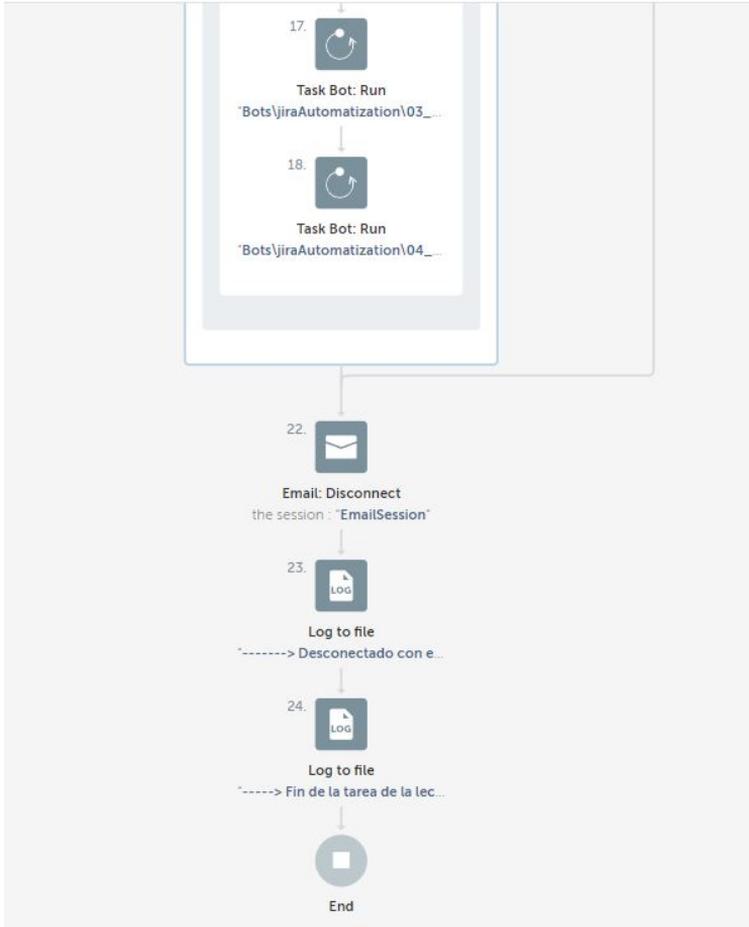


Figura 9.21: Flujo de la lectura de correos.Parte 3. Fuente [1]

9.2.4. createIssue

En esta sección se va a presentar el flujo del task bot encargado de crear y realizar la llamada REST necesaria para crear las incidencias en el sistema gestor de incidencias Jira.

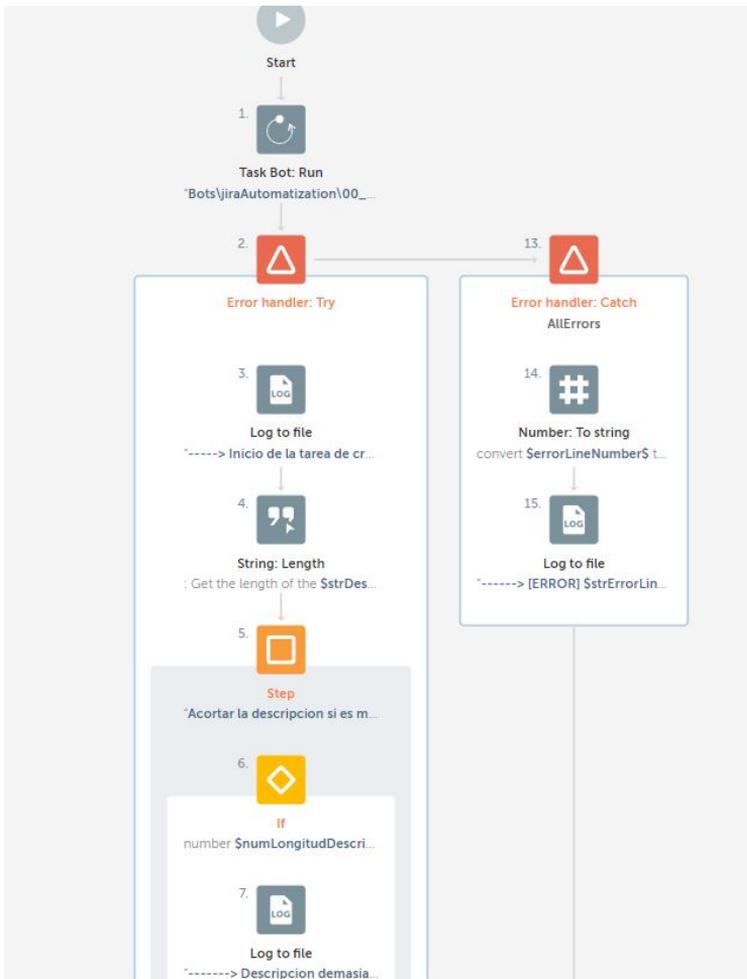


Figura 9.22: Flujo de la creación de incidencias.Parte 1. Fuente [1]

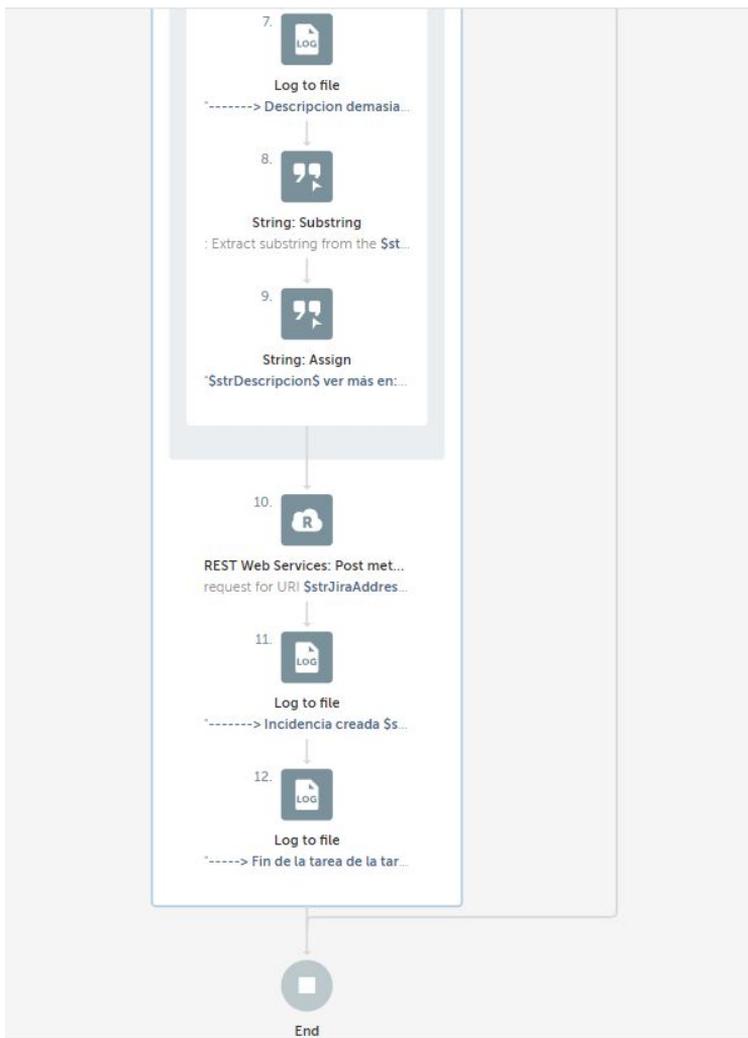


Figura 9.23: Flujo de la creación de incidencias.Parte 2. Fuente [1]

Capítulo 10

Pruebas

10.1. Introducción

La verificación del software es el proceso que consiste en comprobar los requisitos en cada una de las fases de un proyecto. Es decir, es un proceso de evaluación de un sistema o componente para determinar si los productos resultantes de una determinada fase de desarrollo satisfacen las condiciones especificadas al inicio de la fase.

La validación del software consiste en examinar y comprobar que las especificaciones del software cumplen con los requisitos y/o necesidades del usuario. En el proceso de validación y verificación del software hay dos aproximaciones complementarias para la comprobación de sistemas:

- **1. Inspecciones de software:** analizan y comprueban las representaciones del sistema como el documento de requisitos, los modelos de análisis y diseño o el código fuente. Es estático, no se necesita ejecutar el software. Se puede dar en cualquier momento del proceso de desarrollo.
- **2. Pruebas del software:** implican ejecutar una implementación del software con datos de prueba. Son dinámicas. Requieren de al menos un prototipo del sistema.

Existen dos tipos de pruebas:

- **Pruebas de validación:** intentan demostrar que el software es el que el cliente quiere. Incluyen las pruebas de aceptación o las de rendimiento.
- **Pruebas de defectos:** intentan revelar defectos del sistema. Intentan hallar inconsistencias entre un programa y su especificación.

Las pruebas pueden dividirse en cuatro niveles:

- **Pruebas unitarias:** evalúan que las unidades mínimas de software (procedimientos, clases) proporcionan la funcionalidad requerida con independencia del resto de unidades funcionales.
- **Pruebas de integración:** parten de que cada unidad de software funciona correctamente y se centran en probar su interacción. Por lo tanto, prueban que los componentes software ofrecen la funcionalidad deseada cuando actúan en conjunto.
- **Pruebas de sistema:** su objetivo es evaluar que el software completo, con todos sus componentes integrados, satisface los requisitos.
- **Pruebas de aceptación:** se realizan con los usuarios con el objetivo de validar que el software responde realmente a sus necesidades.

10.2. Pruebas desarrolladas

Durante este proyecto se han realizado pruebas unitarias que se han ido realizando a lo largo de su desarrollo pero que no se especificarán en este documento.

En este trabajo de fin de grado sólo se muestran los casos de prueba, que son especificaciones de las entradas para la prueba y la salida esperada del sistema más una afirmación de lo que se está probando [33], correspondientes a las pruebas del sistema.

Los casos de prueba que han sido establecidos durante este trabajo están dirigidos por los diferentes casos de uso 7.1. En ellos se muestran las dependencias de estos con cada requisito, obteniéndose así una trazabilidad completa entre los requisitos, los casos de uso y los casos de prueba.

Identificador	CP01 Procesar Incidencias	
Precondiciones	El bot debe haber sido ejecutado	
Datos de entrada	Incidencia de prueba 01	
Pasos del caso de prueba	PASOS DEL CU01: 7.1 1. El administrador ejecuta el bot 2. Incluye Conexión al sistema gestor de correo 3. Punto de extensión: Procesado de la información del correo 1b No existe ningún correo con incidencias	SALIDA ESPERADA 1. Se realiza correctamente la ejecución 2. La conexión se realiza de manera exitosa 3. La información del correo es mapeada correctamente 1b El sistema marca como leídos los correos (si los hubiera)
Resultados de la prueba	EJECUCIÓN: 11/07/2022 Se ejecuta el caso de prueba	RESULTADO: El caso de prueba ha sido realizado con éxito
Defectos encontrados	No se han encontrado defectos	
Dependencias	REQ05 y REQ06	

Cuadro 10.1: Caso de prueba 01

10.2. PRUEBAS DESARROLLADAS

Identificador	CP02 Conexión al sistema gestor de correo	
Precondiciones	El bot debe haber sido ejecutado Se debe haber iniciado el CU01	
Datos de entrada	Ninguna	
Pasos del caso de prueba	PASOS DEL CU02: 7.2 1. El caso de uso comienza cuando se ha arrancado el sistema 2. El bot introduce las credenciales de la cuenta de correo 3. Se establece la conexión con el sistema de correo 1b Las credenciales son incorrectos	SALIDA ESPERADA 1. Se realiza correctamente la ejecución 2. Se establece la conexión 3. La conexión se realiza de manera exitosa 1b El sistema registra esto en el log y el CP termina
Resultados de la prueba	EJECUCIÓN: 11/07/2022 Se ejecuta el caso de prueba	RESULTADO: El caso de prueba ha sido realizado con éxito
Defectos encontrados	No se han encontrado defectos	No aplica
Dependencias	REQ01, REQ02 y REQ03	

Cuadro 10.2: Caso de prueba 02

Identificador	CP03 Procesado de la información del correo	
Precondiciones	El bot debe haber sido ejecutado Se debe haber iniciado el CU01 La sesión de correo electrónico debe estar iniciada	
Datos de entrada	Ninguna	
Pasos del caso de prueba	PASOS DEL CU03: 7.3 Por cada correo no leído: 1. Se lee la información del email 2. Se marca el correo como leído 3. Se introduce la información en cada una de las variables 4. Se realiza la llamada de creación de la incidencia 5. El Caso de prueba termina	SALIDA ESPERADA 1. Se realiza correctamente la lectura 2. El correo pasa a leído 3. Las variables tienen su valor correspondiente 4 El sistema crea la incidencia 5 El CP termina
Resultados de la prueba	EJECUCIÓN: 11/07/2022 Se crea la incidencia en el sistema gestor de incidencias	RESULTADO: El caso de prueba ha sido realizado con éxito
Defectos encontrados	No se han encontrado defectos	No aplica
Dependencias	REQ04, REQ05 Y REQ06	

Cuadro 10.3: Caso de prueba 03

Capítulo 11

Conclusiones y líneas futuras

11.1. Conclusiones

Para finalizar este trabajo de fin de grado se van a exponer diferentes conclusiones derivadas del proceso de desarrollo software.

Tras la elaboración de este trabajo se puede concluir que se han cubierto todos los requisitos establecidos durante la planificación establecida. Se ha creado para tal efecto un sistema capaz de identificar el contenido de los emails recibidos, determinar el origen de los datos de estos.

Además es capaz de procesar y extraer la información necesaria para la creación de incidencias en el sistema gestor conectarse con Jira y crear la historia perteneciente a los datos extraídos del correo. Cumpliendo así con todos los requisitos establecidos.

Se ha construido un sistema que resuelve una necesidad real detectada en la empresa donde el alumno realizó las prácticas del grado. Con la solución propuesta se provee de un sistema que reducirá notablemente el uso de recursos destinados a esta tarea. Tanto humanos, ya que es un sistema autónomo, como económicos, ya que reduce los costes de realizar esta tarea.

Este trabajo cumple con todas las fases del desarrollo de un proyecto de ingeniería, hecho que sirve para afianzar los conceptos explicados durante diversas asignaturas del grado.

11.2. Líneas futuras

Las líneas futuras de este proyecto podría ser la ampliación del sistema para el procesado de una mayor información recibida en los correos.

También se podría realizar una ampliación para guardar un registro de las incidencias creadas en algún tipo de archivo y que este fuese enviado a los miembros del equipo deseados.

Además, otra posible ampliación podría ser la creación de una aplicación mediante un lenguaje de programación que provea al usuario de una interfaz para la creación de las incidencias y que esta se realice de una forma más amigable.

Apéndice A

Manual de usuario

El manual de usuario de este sistema es un manual breve, puesto que se trata de un sistema automático (por ello se emplea RPA).

El usuario que desee emplear este sistema simplemente deberá entrar en Automation Anywhere en la *control room* donde se encuentre alojado este sistema y ejecutar el orquestador creado para tal efecto, como se ha explicado en otras secciones de esta memoria.

Cabe destacar que se deben cumplir los requisitos mínimos para la ejecución y los pasos previos, descritos en el apéndice B

Apéndice B

Manual de despliegue y ejecución

B.1. Requisitos mínimos para el despliegue y ejecución del bot

Para poder desplegar y ejecutar el sistema se ha de disponer de un equipo con conexión a internet, ya que es necesario tener acceso a una cuenta de Automation Anywhere [1]. El equipo donde se va a desplegar y ejecutar el bot deberá tener un sistema operativo Windows en su versión 7 o posteriores y deberá tener instalado la aplicación de Outlook (versión *Consumer Version 2009* o posterior) con la cuenta de correo configurada debidamente. El ordenador también deberá tener al menos 1 Gb de almacenamiento libre y una memoria RAM de al menos 4Gb.

B.2. Pasos previos a la ejecución del bot

Para poder ejecutar el bot previamente se ha de descargar el archivo denominado *AutomationAnywhereBotAgent.exe*, una vez instalado, desde AA debemos marcar la opción de conectar dispositivo. Esta opción desplegará un asistente que nos irá guiando durante el proceso, que básicamente consta de habilitar la extensión de AA para el navegador y esperar a que el software se conecte con el dispositivo, como se muestra en la figura B.1.

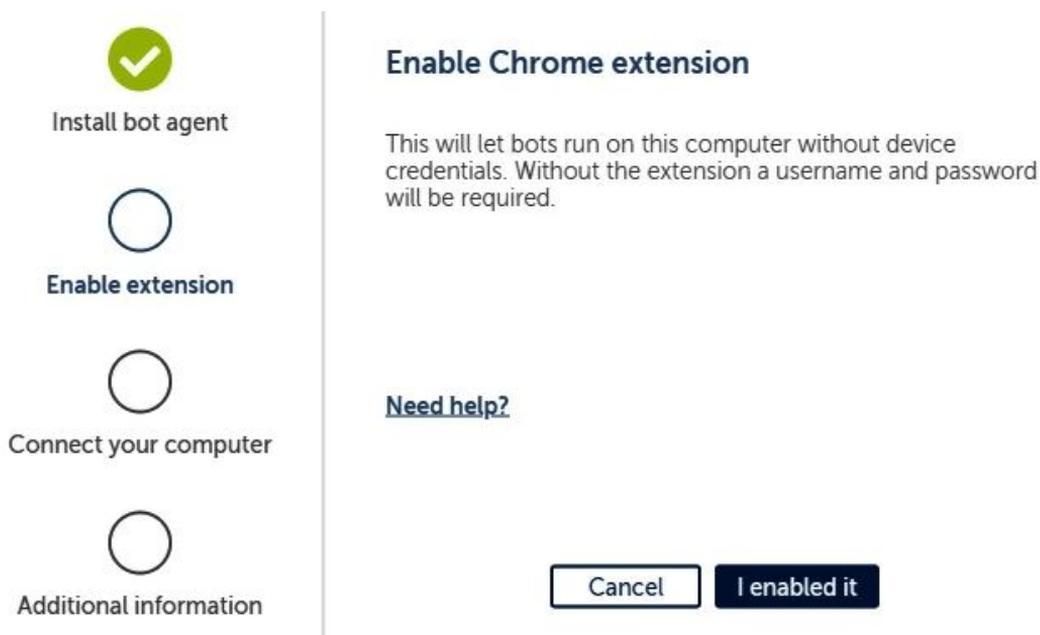


Figura B.1: Instalación del software para ejecución del bot. Fuente [1]

Se deberá crear la siguiente estructura de carpetas y archivos:

- En el disco local C dentro de Users y el usuario correspondiente en la carpeta de Documentos se deberá crear la carpeta *documentosBotJira*.
- En dicho directorio se deberán crear las carpetas *logs* y *recursos*.
- En la carpeta *logs* se deberá crear el archivo *mainLog.txt*.
- En la carpeta *recursos* se deberá crear el directorio *config*.
- En la carpeta *config* se ha de crear el archivo *configFile.xml*

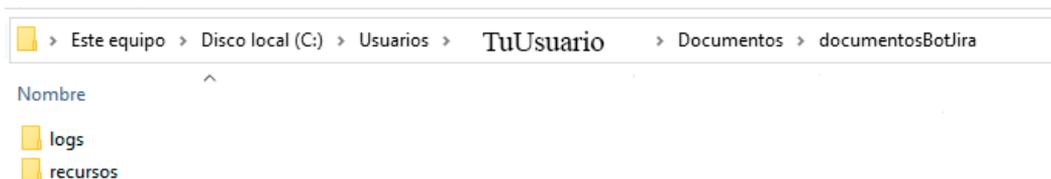


Figura B.2: Árbol de directorios necesarios para la ejecución del sistema

B.3. Ejecución del bot

Una de las ventajas de la versión de pago de AA es que permite la programación de las ejecuciones del bot de forma desatendida. Para este sistema se ha considerado que dicha característica no es relevante y que se ejecutará el bot de forma manual.

Para la ejecución del bot simplemente habrá que pulsar el botón de *Run Task Bot* correspondiente al Orquestador como muestra la figura B.3. De esta manera se ejecutará el bot en la máquina asociada.



Figura B.3: Ejecución del bot. Fuente [1]

Bibliografía

- [1] *Automation Anywhere*,
[HTTPS://WWW.AUTOMATIONANYWHERE.COM/LA/](https://www.automationanywhere.com/la/).
Recuperado el 11/05/2022,
de <https://www.automationanywhere.com/la/>

- [2] *Guía docente del TFG*,
[HTTPS://WWW.INF.UVA.ES/WP-CONTENT/UPLOADS/2016/06/G46976.PDF](https://www.inf.uva.es/wp-content/uploads/2016/06/G46976.PDF).
Recuperado el 11/05/2022,
de <https://www.inf.uva.es/wp-content/uploads/2016/06/G46976.pdf>

- [3] *Scrum*,
[HTTPS://WWW.SCRUM.ORG/RESOURCES/WHAT-IS-SCRUM](https://www.scrum.org/resources/what-is-scrum).
Recuperado el 13/05/2022,
de <https://www.scrum.org/resources/what-is-scrum>

- [4] *Outlook*,
[HTTPS://WWW.MICROSOFT.COM/EN-WW/MICROSOFT-365/OUTLOOK/EMAIL-AND-CALENDAR-SOFTWARE-MICROSOFT-OUTLOOK](https://www.microsoft.com/en-ww/microsoft-365/outlook/email-and-calendar-software-microsoft-outlook).
Recuperado el 13/05/2022,
de <https://www.microsoft.com/en-ww/microsoft-365/outlook/email-and-calendar-software-microsoft-outlook>

- [5] *Jira*,
[HTTPS://WWW.ATLASSIAN.COM/SOFTWARE/JIRA](https://www.atlassian.com/software/jira).
Recuperado el 20/05/2022,
de <https://www.atlassian.com/software/jira>

- [6] *Jira Marketplace*,
[HTTPS://WWW.SIMILARTECH.COM/TECHNOLOGIES/JIRA](https://www.similartech.com/technologies/jira).
Recuperado el 20/05/2022,
de <https://www.similartech.com/technologies/jira>

- [7] *Usuarios Jira*,
[HTTPS://WWW.ATLASSIAN.COM/ES/SOFTWARE/JIRA/GUIDES/USE-CASES/WHO-USES-JIRA#HOW-DEVOPS-TEAMS-USE-JIRA](https://www.atlassian.com/es/software/jira/guides/use-cases/who-uses-jira#how-devops-teams-use-jira).
Recuperado el 20/05/2022,
de <https://www.atlassian.com/es/software/jira/guides/use-cases/who-uses-jira#how-devops-teams-use-jira>
- [8] *Agile*,
[HTTPS://WWW.REDHAT.COM/ES/DEVOPS/WHAT-IS-AGILE-METHODOLOGY](https://www.redhat.com/es/devops/what-is-agile-methodology).
Recuperado el 20/05/2022,
de <https://www.redhat.com/es/devops/what-is-agile-methodology>
- [9] *Metodología Scrum*,
[HTTPS://WWW.EUROPEANSCRUM.ORG/CERTIFICACION-SCRUM-FOUNDATIONS.HTML](https://www.europeanscrum.org/certificacion-scrum-foundations.html).
Recuperado el 23/05/2022,
de <https://www.europeanscrum.org/certificacion-scrum-foundations.html>
- [10] *¿Qué hace un desarrollador RPA?*,
[HTTPS://RECLUIT.COM/QUE-HACE-UN-DESARROLLADOR-RPA/L](https://recluit.com/que-hace-un-desarrollador-rpa/l).
Recuperado el 23/05/2022,
de <https://recluit.com/que-hace-un-desarrollador-rpa/>
- [11] *UiPath*,
[HTTPS://WWW.UIPATH.COM/](https://www.uipath.com/).
Recuperado el 23/05/2022,
de <https://www.uipath.com/>
- [12] *Licencias AA*,
[HTTPS://ACORTAR.LINK/BMGOAV](https://acortar.link/bMgOAv).
Recuperado el 23/05/2022,
de <https://acortar.link/bMgOAv>
- [13] *Manifiesto for Agile Software Developement*,
KEN SCHWABER Y 15 AUTORES MÁS.
Recuperado el 26/05/2022,
de <https://agilemanifiesto.org/>
- [14] *The 2020 Scrum Guide*,
KEN SCHWABER Y JEFF SUTHERLAND.
Recuperado el 26/05/2022,
de <https://scrumguides.org/scrum-guide.html>

- [15] *What is CMMI?*,
SARAH K. WHITE.
Recuperado el 26/05/2022,
de <https://www.cio.com/article/274530/process-improvement-capability-maturity-model-integration-cmmi-definition-and-solutions.html>
- [16] *Tablero Kanban*,
<HTTPS://KANBANIZE.COM/ES/RECURSOS-DE-KANBAN/PRIMEROS-PASOS/QUE-ES-TABLERO-KANBAN>.
Recuperado el 23/05/2022,
de <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-tablero-kanban>
- [17] *LaTeX*,
<HTTP://DESARROLLOWEB.DLSI.UA.ES/CURSOS/2015/HERRAMIENTAS-INVESTIGACION/QUE-ES-LATEX>.
Recuperado el 23/05/2022,
de <http://desarrolloweb.dlsi.ua.es/cursos/2015/herramientas-investigacion/que-es-latex>
- [18] *Microsoft Project*,
<HTTPS://WWW.MICROSOFT.COM/EN-US/MICROSOFT-365/PROJECT/PROJECT-MANAGEMENT-SOFTWARE>.
Recuperado el 23/05/2022,
de <https://www.microsoft.com/en-us/microsoft-365/project/project-management-software>
- [19] *Microsoft Excel*,
<HTTPS://WWW.MICROSOFT.COM/EN-US/MICROSOFT-365/EXCEL>.
Recuperado el 23/05/2022,
de <https://www.microsoft.com/en-us/microsoft-365/excel>
- [20] *Software Project Management*,
BOB HUGHES AND MIKE COTTERELL., 5th ed. Europa, Oriente Medio y Africa:
McGraw-Hill Education, 2009, pp 163-166.
Consultado el 24/05/2022
- [21] *Análisis de riesgos Incibe*,
<HTTPS://WWW.INCIBE.ES/PROTEGE-TU-EMPRESA/BLOG/ANALISIS-RIESGOS-PASOS-SENCILLO>.
Recuperado el 24/05/2022,
de <https://www.incibe.es/protege-tu-empresa/blog/analisis-riesgos-pasos-sencillo>
- [22] *Convenio colectivo estatal de empresas de consultoría, y estudios de mercado y de la opinión pública*,

MINISTERIO DE EMPLEO Y SEGURIDAD SOCIAL, GOBIERNO DE ESPAÑA.

Recuperado el 25/05/2022,

de <https://www.boe.es/boe/dias/2018/03/06/pdfs/BOE-A-2018-3156.pdf>

[23] *¿Cuánto cuesta contratar a un trabajador?*,

JAVIER SANTOS PASCUALENA.

Recuperado el 25/05/2022,

de <https://www.infoautonomos.com/blog/cuanto-cuesta-contratar-un-trabajador/>

[24] *Astah*,

<HTTPS://ASTAH.NET/>.

Recuperado el 26/05/2022,

de <https://astah.net/>

[25] *What is UML?*,

<UML.ORG/WHAT-IS-UML.HTM>.

Recuperado el 26/05/2022,

de uml.org/what-is-uml.htm

[26] *¿Qué es Timestamp?*,

<HTTPS://ES.COINTELEGRAPH.COM/EXPLAINED/WHAT-IS-TIMESTAMP-TIMING-THE-OPERATION-OF-THE-BLOCKCHAIN>

Recuperado el 27/05/2022,

de <https://es.cointelegraph.com/explained/what-is-timestamp-timing-the-operation-of-the-blockchain>

[27] *¿Qué son los caracteres UTF8?*,

<HTTPS://WWW.ARSYS.ES/BLOG/UTF8>

Recuperado el 27/05/2022,

de <https://www.arsys.es/blog/utf8>

[28] *¿Qué es una API REST?*,

<HTTPS://WWW.IBM.COM/ES-ES/CLOUD/LEARN/REST-APIS>

Recuperado el 27/05/2022,

de <https://www.ibm.com/es-es/cloud/learn/rest-apis>

[29] *¿Qué es una URI?*,

<HTTPS://WWW.ABOUTESPANOL.COM/QUE-ES-URI-157626>

Recuperado el 27/05/2022,

de <https://www.aboutespanol.com/que-es-uri-157626>

- [30] *¿Qué es JSON?*,
[HTTPS://WWW.HOSTINGER.ES/TUTORIALES/QUE-ES-JSON](https://www.hostinger.es/tutoriales/que-es-json)
Recuperado el 27/05/2022,
de <https://www.hostinger.es/tutoriales/que-es-json>
- [31] *What Is BCE?*,
[HTTPS://QUICK-ADVISORS.COM/WHAT-IS-ENTITY-CONTROL-BOUNDARY-PATTERN/#WHAT_IS_BCE_MODEL](https://quick-advisors.com/what-is-entity-control-boundary-pattern/#What_is_BCE_model)
Recuperado el 01/06/2022,
de https://quick-advisors.com/what-is-entity-control-boundary-pattern/#What_is_BCE_model
- [32] *UML 2 / Jim Arlow, Ila Neustadt*,
ARLOW, JIM, AND ILA NEUSTADT..Madrid: Anaya Multimedia, 2006. Print.
Consultado el 01/06/2022
- [33] *Diapositivas de la asignatura Diseño, Integración y Adaptación del Software*,
DIEGO GARCÍA ÁLVAREZ.
Consultado el 01/06/2022
- [34] *Pattern-oriented software architecture: A system of patterns.*,
F. BUSHMANN, R. MEUNIER, H. ROHNERT, P. SOMMERLAD, AND M. STAL..Volume
1. Wiley, 1996
Consultado el 01/06/2022
- [35] *Patron DAO*,
[HTTPS://ACORTAR.LINK/TJR2MI](https://acortar.link/tjr2MI) Recuperado el 01/06/2022,
de <https://acortar.link/tjr2MI>
- [36] *Data Transfer Object*,
[HTTPS://ACORTAR.LINK/8DSGV D](https://acortar.link/8dsgvD) Recuperado el 01/06/2022,
de <https://acortar.link/8dsgvD>