Universidad
Carlos III de Madrid

PROYECTO FIN DE CARRERA
INGENIERÍA INFORMÁTICA

APLICACIÓN PARA EL ANÁLISIS DE LA UTILIZACIÓN DEL CORREO ELECTRÓNICO

Autor: Eduardo Muñoz Peña
Tutor: José María de Fuentes García-Romero de Tejada
Co-tutora: Lorena González Manzano
Leganés, abril de 2014

Título: Aplicación para el análisis de la utilización del correo electrónico

Autor: Eduardo Muñoz Peña

Director: José María de Fuentes García-Romero de Tejada

Co-Directora: Lorena González Manzano

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 29 de abril de 2014 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

Quiero dar las gracias en primer lugar a mis padres y a mis hermanos que han sido testigos de toda mi evolución a lo largo de todos estos años de trabajo y lucha por lograr llegar a ser un Ingeniero Informático.

En segundo lugar quiero hacer una mención especial a mis abuelos, tanto los que están como los que desgraciadamente ya no me acompañan porque estoy seguro de que se alegran mucho de este día.

Además quiero hacer una especial mención a los siguientes compañeros de clase por orden alfabético: Antonio, Eduardo, Iñigo y Pedro. Me dejo en el camino muchos nombres de compañeros que me han ayudado a llegar donde estoy, pero con un listado mayor este apartado de agradecimientos perdería parte de su significado.

Por último querría agradecer a José María su labor como tutor en este proyecto y por confiar en mí a pesar de las difíciles circunstancias en las que se ha desarrollado el mismo.

Resumen

Desde la llegada de Internet hasta nuestros días el uso del correo electrónico ha ido incrementando notablemente hasta incluso casi sustituir por completo al servicio físico de mensajería en algunos sectores empresariales.

Las razones principales que le hacen más práctico que el modelo tradicional son su menor precio, mayor velocidad y escalabilidad. Además de haber sido fomentado por las empresas, también los ciudadanos han optado por el uso del mismo. La ventaja principal para los usuarios, además de las ya citadas, es que el email se puede integrar dentro de los numerosos dispositivos electrónicos que hoy en día están disponibles en el mercado.

La posibilidad de poder hacer un estudio exhaustivo de las diferencias de uso del correo tradicional entre diferentes grupos de personas es una tarea casi imposible al no poder contar con una infraestructura adecuada que pudiese analizar el contenido de los mensajes físicos. Sin embargo, este análisis sí que es posible en el formato electrónico, pero hasta hoy, no se han desarrollado estudios de peso sobre el uso que hace una persona de su cuenta de correo electrónico.

Es por ello, que el objetivo de este proyecto es desarrollar una plataforma que permita hacer mediciones estadísticas sobre el uso de cuentas de correo electrónico de usuarios para su posterior análisis. Se podrán tener en cuenta desde factores culturales del usuario, como su nacionalidad o edad, hasta mediciones más específicas basadas en el contenido o tipología de los emails analizados.

Palabras clave: correo electrónico; estadística; análisis

Abstract

Since Internet´s arrival, use of email has increased dramatically to almost completely replace post in some business sectors.

The main reasons that make it more practical than the traditional model are its lower price, higher speed and scalability. Besides being promoted by companies, citizens also have chosen its use. The main advantage for users in addition to those already mentioned is that email can be integrated into many electronic devices that are available today in the market.

The possibility of doing a comprehensive study of the differences in use of traditional post between different groups of people is almost impossible due to the lack of adequate infrastructure that could analyze the content of the physical task messages. However, this analysis is possible in electronic format, but until now, no studies have been conducted on this direction.

Therefore, the objective of this project is developing a platform that would enable statistical measures on the use of e-mail accounts of users for further analysis. They will include cultural factors from the user, such as his nationality or age, and also more specific measures based on the content of the emails.

Key words: email; statistics; analyze

Índice de contenido

Índice de figuras

Índice de tablas

Capítulo 1

Introducción y objetivos

En este capítulo se va a hacer una breve introducción sobre el proyecto, su motivación y objetivos.

1.1 Introducción

Según datos de un estudio de 2013 a la población española de entre 16 y 73 años del INE (Instituto Nacional de Estadística) [1], más del 75% de los encuestados utilizaban Internet al menos 5 veces por semana y sólo un 8% menos de una vez por semana.

Una de las herramientas más utilizadas dentro del uso de Internet es el correo electrónico ya que se ha convertido en el medio de transmisión de información más importante a través de esta plataforma.

Al estar tan extendido el uso de Internet y en particular el del correo electrónico, se han formado numerosos perfiles de usuarios en base al uso que hacen de estas herramientas. Por ejemplo, un usuario de poca edad y que aún no ha empezado con su vida laboral no hace un uso de su bandeja de correo similar al que hace una persona que está actualmente trabajando y cuenta con una cuenta de correo exclusiva para el trabajo.

A pesar de estas diferencias y formas de utilizar el email, no sólo en base a la edad sino a factores culturales como el país de residencia o el sector laboral, no existen realmente estudios estadísticos de peso, que muestren estas diferencias de uso en cifras palpables y que permitan elaborar un esquema de los tipos de usuarios con los que cuenta hoy en día el correo electrónico.

Las características más destacadas entre las que se podría diferenciar a los usuarios serían las siguientes:

- País de residencia
- Sector laboral
- Edad
- Sexo

La decisión de diferenciar en base al país de residencia y no a la nacionalidad, es porque es más probable que el uso de la cuenta de email esté más relacionado con las facilidades con las que cuente el usuario en el país en el que reside que con su nacionalidad.

Por otro lado, para este estudio sería interesante medir datos como la regularidad con la que se contesta a los emails, el volumen diario de los mismos o incluso el número de líneas con las que cuenta de media cada email.

Se debería hacer una división entre los correos electrónicos dentro del ámbito laboral y el personal, ya que probablemente un mismo usuario haga dos usos muy distintos de su bandeja de entrada en función de si son de dominio laboral o por el contrario, se trata de un email personal.

En la figura 1 se puede ver de forma esquematizada las subdivisiones que se podrían hacer para el estudio estadístico:

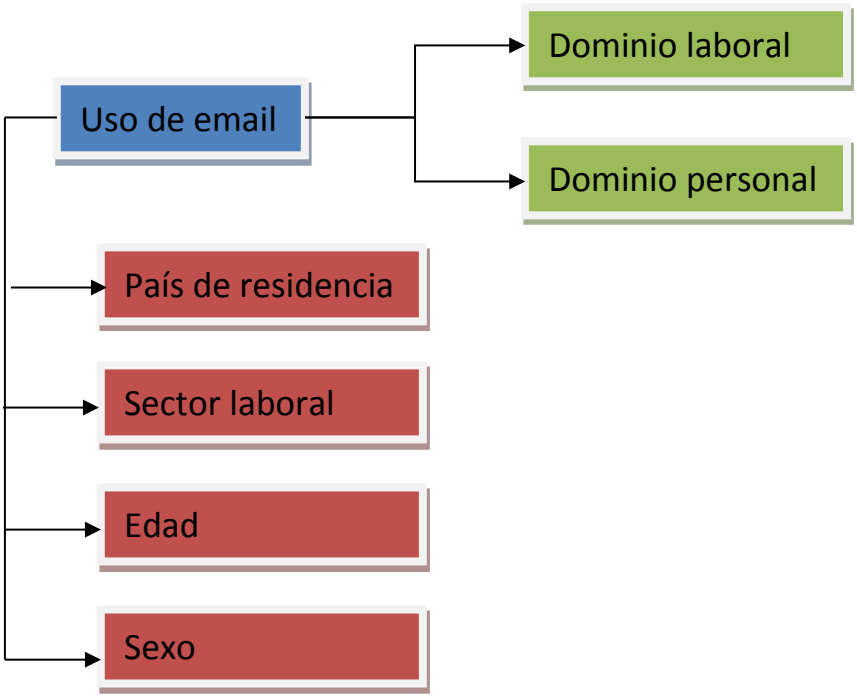


Figura 1 Subdivisiones del estudio estadístico

1.2 Motivación

Contar con un estudio estadístico de peso, sobre el uso que hacen los usuarios de su cuenta de correo en base a diferentes factores culturales, sería muy interesante no sólo desde el punto de vista académico, por las investigaciones que se podrían generar del mismo para entender de dónde nacen esas diferencias, sino también desde una perspectiva comercial, ya que si se contase con un esquema de usuarios en base a factores culturales y a dominios de aplicación, se podrían crear herramientas más personalizadas que se ajustasen mejor a las necesidades de los usuarios.

Con unos datos lo suficientemente amplios, por ejemplo se podría descubrir que un porcentaje significativo de los usuarios del correo electrónico, escriben emails muy cortos y lo que verdaderamente les interesa es poder filtrar un gran número de volumen emails que les llega regularmente.

Con esta información se podría llegar a la conclusión de que sería interesante crear una herramienta que se centrase menos en la edición del contenido de los emails y más en una forma más cómodo de filtrar la bandeja de entrada. Es decir, una nueva oportunidad de negocio.

Otra posibilidad sería la creación de mecanismos de seguridad en base a patrones de uso conocidos. En el caso de un uso fuera de lo normal, sería muy sencillo detectar datos anómalos fruto de un uso fraudulento.

Como este ejemplo pueden surgir muchos más, pero hasta contar con datos palpables no se pueden hacer más que meras suposiciones. De ahí nade la necesidad de contar con un medio que nos permita contar con la información necesaria para poder hacer estas estadísticas y mediciones en base a los factores ya explicados en el apartado anterior.

1.3 Objetivo

El objetivo de este proyecto es crear una plataforma que permita extraer toda la información mencionada en apartados anteriores para un futuro estudio estadístico del uso del email por parte de la totalidad de los usuarios del mismo.

Es necesario poder conocer datos específicos del usuario para el muestreo posterior a la totalidad, como su edad o sexo, pero también las mediciones específicas sobre el uso que hace sobre su cuenta de email.

Para lograr este objetivo, se requiere de un análisis exhaustivo de las mediciones que se quieren hacer y de cómo se pueden hacer. Es importante que se tenga en cuenta que debe ser una solución escalable y además, que es posible que en el futuro se quiera ampliar la información que se quiere consultar.

Por tanto, en base a todos estos principios se pueden concretar los siguientes objetivos a cubrir:

- Diseñar e implementar una aplicación denominada E-RETO que se conecte con la bandeja de entrada de una cuenta de email y haga mediciones estadísticas sobre el contenido de la misma

- Diseñar e implementar una forma de agrupar y procesar los resultados obtenidos en la aplicación creada en el primer objetivo.

1.4 Organización del presente documento

Al tratarse de un documento bastante extenso, en esta sección se hace una breve mención sobre el contenido de cada apartado:

Capítulo 1. Introducción y objetivos: En este capítulo se hace una breve introducción sobre el alcance actual del uso del correo electrónico y la necesidad de realizar un estudio estadístico sobre el uso del mismo.

Capítulo 2. Análisis: En este capítulo se lista y explica qué requisitos debe cubrir una aplicación que quiera cumplir los objetivos descritos en apartados anteriores. Además se analizan las posibles tecnologías a utilizar para realizar este sistema y las posibles pruebas con las que se mida si cumple o no con estos requisitos.

Capítulo 3. Diseño detallado: En este capítulo se detalla la implementación de la aplicación en base a las decisiones tomadas en el capítulo de análisis. Para ello se hace uso de diagramas de clase UML (Lenguaje Unificado de Modelado) [2] y de secuencia [3]

Capítulo 4. Implementación y pruebas: En este capítulo se describen las decisiones de implementación tomadas tras la etapa de diseño y los resultados de las pruebas de aceptación de la etapa de análisis.

Capítulo 5. Conclusiones y líneas futuras: Este último capítulo se explican las conclusiones a las que se ha llegado tras el desarrollo de esta aplicación y futuras mejoras o ampliaciones que se pudiesen hacer del sistema.

Anexo 1. Gestión del proyecto: En este anexo se muestra la planificación y seguimiento del proyecto, incluyendo su presupuesto y las desviaciones del mismo tras su desarrollo.

Anexo 2. Manual de usuario: Este anexo facilita al usuario de la aplicación el uso de la misma.

Anexo 3. Plantillas: En este anexo se adjuntan todas las plantillas utilizadas para la realización de este documento.

Capítulo 2

Análisis

En este capítulo se hará un análisis de todas las partes del sistema, sus requisitos y casos de uso.

2.1 Perspectiva general del sistema

En este bloque de la memoria se procederá a ahondar en el diseño de la herramienta desarrollada en este proyecto fin de carrera para cubrir las necesidades ya especificadas en el apartado de introducción de este documento.

E-RETO es una aplicación software de escritorio que necesita de una conexión a internet para poder funcionar. Requiere de esta conectividad tanto para poder acceder a la cuenta de email del usuario de la herramienta como para poder enviar los resultados extraídos del análisis de la misma.

Para poder contar con un entorno controlado dentro de la aplicación, se ha decidido limitar el acceso a cuentas de email de Gmail [12], Yahoo! [13] y Hotmail [14]. En cualquier caso, el diseño de esta herramienta ha sido basado en la idea de poder ampliar el número de servidores de correo soportados de una forma sencilla y sin impacto en su estructura interna.

Para permitir la tramitación de los resultados derivados de los cálculos que hace e-RETO, la Universidad Carlos III de Madrid ha cedido el uso de uno de sus servidores para este proyecto. En citado servidor se guardan todos los resultados enviados por la herramienta para su posterior uso y estudio.

E-RETO debe ser diseñada con el fin de permitir su integración dentro de un paquete de herramientas que compartan dominio de estudio para la Universidad, de forma que su implementación debe ser lo suficientemente abstracta como para poder cubrir este requisito de integración.

2.2 Panorámica de herramientas existentes

Este apartado de la memoria va a hacer mención de las principales herramientas sobre el manejo de email que se encuentran en el mercado. Es importante destacar desde el principio que no existe ninguna herramienta que cubra los objetivos que se ha marcado E-RETO.

La mayoría de las herramientas de análisis de emails se utilizan dentro de los programas para Informática Forense [4]. Su utilidad en este caso es la de detectar fraudes, suplantación de identidad y delitos de similar gravedad. Una de las herramientas por excelencia es la denominada Aid4Mail [5], que permite más de 40 formatos distintos de email y es compatible con todos los servidores comerciales más utilizados. Otra de las herramientas más populares pero también de pago en este ámbito es la denominada Paraben [6].

En el caso de herramientas libres como E-RETO, en la página web de Forensic Control [7], se puede encontrar en el apartado de analizadores de email una lista de las opciones más populares actualmente en el mercado. En ningún caso ninguna de ellas cubre nada del dominio de E-RETO.

Al margen de las herramientas de análisis ya citadas, existe una aplicación libre en versión beta subida a la web CodePlex [8], con el nombre de *Email Behavior Analyzer* [9]. Esta aplicación parte con una idea parecida a la de E-RETO, pero cubre sólo a qué personas se les manda un email y en qué cantidad.

Existe una tesis en la Universidad de *Strathclyde* [10] en Glasgow titulada *Email Analyzer Software* [11] que apoya la idea de la necesidad de ampliar las investigaciones en este área, pero la herramienta que diseña se queda muy lejos de los que quiere cubrir E-RETO, al hacer sólo un análisis de la estructuración en carpetas que se hace de una cuenta de correo electrónico.

2.3 Casos de uso

En esta sección se muestra el diagrama de casos de uso de la aplicación junto con la definición textual de cada uno de los mismos. El estudio de los casos de uso descritos en esta sección, facilitará la extracción de requisitos en fases posteriores del proyecto.

2.3.1 Diagrama de casos de uso

En el siguiente diagrama de la figura 2, se puede apreciar que se han detectado 3 casos de uso en los que participa un único actor que es el usuario de la interfaz de usuario. En el primer caso de uso se representa la acción en el que el usuario inicia sesión en E-RETO y decide cargar sus emails. En el segundo caso de uso el usuario decide clasificar sus emails entre los que son del entorno laboral y los que no. En el último caso de uso, el usuario decide visualizar los resultados obtenidos en el estudio realizado por E-RETO:

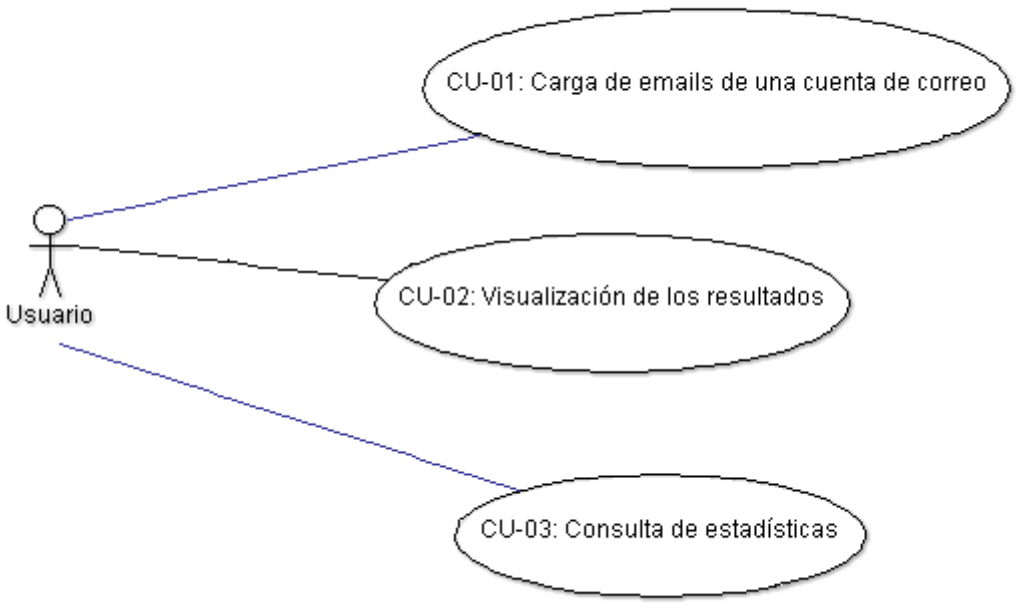


Figura 2 Casos de uso

2.3.2 Definición textual de los casos de uso

En este apartado se muestra información detallada de los distintos casos de uso identificados en el diagrama de casos de uso del apartado anterior. Para mostrar esta información se seguirá el formato definido en la plantilla 1 del Anexo 3.

La Tabla 1 muestra el primer caso de uso identificado que consiste en el inicio de sesión del usuario en un servidor de correo externo y la carga de todos sus emails. El usuario podrá elegir entre una lista de servidores de correo, rellenar su usuario y contraseña y deberá decidir si quiere o no contribuir con el estudio para el que se ha creado este software.

Identificador: CU-01	
Nombre	Analizar una cuenta de correo
Descripción Inicio de sesión del usuario en un servidor de correo externo, carga de todos sus emails y posterior filtro y envío de los mismos.	
Actores Usuario de la aplicación	
Precondiciones La aplicación ha sido arrancada.	
Pos-condiciones La aplicación carga todos los emails de la cuenta especificada por el usuario o lanza un mensaje de error, filtra todos los emails en base a las condiciones del usuario y envía los resultados generados.	

Tabla 1 CU-01

Identificador: CU-01	
Nombre	Analizar una cuenta de correo
Flujo normal 1. Usuario escribe el nombre de usuario de la cuenta de email 2. Usuario escribe la contraseña de la citada cuenta 3. Usuario selecciona el correspondiente servidor de correo de esa cuenta 4. Usuario acepta las condiciones legales del software 5. Usuario india al sistema que inicie la conexión *Sistema conecta con el servidor de correo* 6. Usuario rellena sus datos personales *Sistema guarda datos personales del usuario* 7. Usuario indica al sistema que continúe *Sistema despliega información al usuario* 8. Usuario confirma que quiere cargar todos sus emails *Sistema carga los emails* 9. Usuario selecciona buzones de la lista 10. Usuario añade buzones a la lista *Sistema añade buzones* 11. Usuario introduce un dominio laboral *Sistema añade dominio a la lista* 12. Usuario indica al sistema que continúe *Sistema filtra los emails de la lista, calcula las estadísticas y envía los resultados al servidor*	
Flujo alternativo Si antes de paso 5 se añade: 4.2 Usuario decide deseleccionar la opción de contribuir al estudio para el que ha sido diseñado el software. Entonces directamente de paso 5 se pasa a paso 8.	

Tabla 1 CU-01

La Tabla 2 muestra el segundo caso de uso identificado que consiste en la visualización de los resultados obtenidos por el estudio estadístico.

Identificador: CU-02	
Nombre	Visualización de los resultados
Descripción Visualización de los resultados obtenidos por el estudio estadístico.	
Actores Usuario de la aplicación	
Precondiciones La aplicación ha sido arrancada. Inicio de sesión del usuario en un servidor de correo externo y carga de todos sus emails. Clasificación de buzones de email que son sólo para fines laborales con envío de los resultados al servidor.	
Pos-condiciones La aplicación muestra los resultados obtenidos del análisis	
Flujo normal 1. Usuario selecciona qué gráfica desea visualizar *Sistema genera gráfica con los resultados*	
Flujo alternativo Usuario pulsa cerrar aplicación y no se muestran los resultados.	

Tabla 2 CU-02

La Tabla 3 muestra el caso de uso que representa la visualización de las estadísticas almacenadas en el servidor.

Identificador: CU-03	
Nombre	Consulta de estadísticas
Descripción Consulta de las estadísticas almacenadas hasta la fecha.	
Actores Investigador	
Precondiciones E-RETO ya ha sido utilizada por al menos un usuario y se han enviado datos estadísticos de manera satisfactoria.	
Pos-condiciones El servidor muestra los datos estadísticos por pantalla.	
Flujo normal 1. Investigador escribe correspondiente url en navegador para visualizar los datos. *Sistema muestra los resultados en el navegador.*	
Flujo alternativo Usuario cierra navegador sin llamar al *script.*	

Tabla 3 CU-03

2.4 Requisitos de software

En esta sección se presentan los requisitos de software identificados que deberán ser cubiertos para cumplir con los objetivos especificados para este proyecto. Las distintas subsecciones que componen esta sección detallan los requisitos de los distintos tipos definidos por la metodología de la ESA (Agencia Espacial Europea) [39]. La lista completa se puede ver en las Tablas 4 y 5.

2.4.1 Requisitos funcionales

Identificador	Nombre	Descripción	Estabilidad	Prioridad
RF-01	Log in en la cuenta de email	El sistema debe permitir introducir credenciales para hacer log in de una cuenta de email: usuario, contraseña y servidor de correo.	Alta	Alta
RF-02	Decidir contribución al estudio de investigación	El sistema debe contribuir al estudio de investigación asociado a esta aplicación.	Alta	Media
RF-03.01	Decidir si aceptar política de privacidad de la aplicación	El sistema debe permitir aceptar la política de privacidad y responsabilidades asumidas por la aplicación.	Alta	Alta
RF-03.02	Decidir si aceptar política de privacidad de la aplicación	El sistema debe permitir rechazar la política de privacidad y responsabilidades asumidas por la aplicación.	Alta	Alta
RF-04	Conectarse al servidor de correo	El sistema debe permitir conectarse a un servidor de correo con los credenciales introducidos.	Media	Alta

Identificador	Nombre	Descripción	Estabilidad	Prioridad
RF-05	Seleccionar un idioma	El sistema debe permitir seleccionar el idioma de la aplicación.	Media	Media
RF-06	Seleccionar nacionalidad	El sistema debe permitir al usuario seleccionar país de procedencia.	Alta	Media
RF-07	Seleccionar sector laboral	El sistema debe permitir al usuario seleccionar sector laboral al que se dedica.	Alta	Media
RF-08	Seleccionar edad del usuario	El sistema debe permitir al usuario seleccionar su edad.	Alta	Media
RF-09	Seleccionar género	El sistema debe permitir al usuario seleccionar el género del usuario que realiza el estudio.	Alta	Media
RF-10	Cargar emails	El sistema debe permitir al usuario cargar todos los emails de la cuenta de usuario a la que se ha conectado.	Media	Alta
RF-11	Mostrar información de carga de emails	El sistema debe mostrar información durante la carga de los emails.	Alta	Media

Identificador	Nombre	Descripción	Estabilidad	Prioridad
RF-12	Mostrar información sobre autores	El sistema debe mostrar nombre y apellidos del autor y los responsables del proyecto de investigación.	Alta	Alta
RF-13	Mostrar información sobre lo que realiza la aplicación	El sistema debe informar al usuario de lo que realiza con sus emails: si guarda datos o los envía.	Alta	Alta
RF-14	Salir de la aplicación	El sistema debe permitir al usuario salir de la aplicación en todo momento.	Alta	Alta
RF-15	Filtrar los emails	El sistema debe permitir al usuario filtrar los emails de la bandeja de entrada para fines estadísticos en base a si son de entorno laboral o no.	Media	Alta
RF-16	Filtrar los dominios	El sistema debe permitir al usuario filtrar los dominios de la bandeja de entrada para fines estadísticos en base a si son de entorno laboral o no.	Alta	Alta
RF-17	Generar resultados estadísticos	El sistema debe generar resultados estadísticos para el estudio en base al apartado 2.4.3 de este documento.	Alta	Alta

Identificador	Nombre	Descripción	Estabilidad	Prioridad
RF-18	Enviar resultados	El sistema debe enviar resultados estadísticos al servidor de la Universidad.	Media	Alta

Tabla 4 Requisitos Funcionales

2.4.2 Requisitos no funcionales

Identificador	Nombre	Descripción	Estabilidad	Prioridad
RNF-01	Comunicación con E-RETO	Las funcionalidades de E-RETO se ejecutan mediante eventos de ratón.	Alta	Alta
RNF-02	Idioma de la aplicación	La aplicación muestra la información en múltiples idiomas: español, inglés y alemán.	Media	Media
RNF-03	Validación de datos	La aplicación valida los datos introducidos por el usuario: nombre de usuario, contraseña y edad.	Alta	Alta
RNF-04	Mensajes de error	La aplicación muestra mensajes de error en caso de producirse.	Alta	Media

Identificador	Nombre	Descripción	Estabilidad	Prioridad
RNF-05	Listado de países	El listado de países debe extraído del ISO (International Organization for Standardization) 3166 [40]	Alta	Alta
RNF-06	Listado de sectores laborales	El listado de sectores laborales debe ser el otorgado por la red profesional LinkedIn [41]	Media	Alta

Tabla 5 Requisitos No Funcionales

2.4.3 Datos estadísticos generados

En la introducción de este documento se hace un desglose de los parámetros fundamentales con los que se pueden caracterizar a un usuario de una cuenta de email, como se puede ver en la Figura 1 del apartado de 1.1. En base esta estructura, se cuenta con la siguiente información que debe estar tratada por la aplicación:

Country: país de residencia del usuario.
Sector: sector laboral del usuario.
Gender: género del usuario.
Age: edad del usuario.

Tanto para el ámbito laboral como el personal se deben medir las siguientes variables, haciendo una diferenciación para los boletines de noticias:

avNumberEmails: número medio de emails por hilo (incluyendo emails nunca contestados).
dvNumberEmails: desviación estándar de la variable anterior.
avLengEmailsSent: longitud media en número de líneas por email enviado.
dvLengEmailsSent*:* desviación estándar de la variable anterior.
avLengEmailsRecv*:* longitud media en número de líneas por email recibido.
dvLengEmailsRecv*:* desviación estándar de la variable anterior.
avLengConv : número medio de emails por hilo (sin incluir emails en solitario)
dvLengConv*:* desviación estándar de la variable anterior.
avTimeConv: tiempo medio de respuesta entre emails de un hilo en segundos.
dvTimeConv*:* desviación estándar de la variable anterior.
avNumberPeopleSent*:* número de medio de personas a las que se envía en un email.
dvNumberPeopleSent*:* desviación estándar de la variable anterior.
avNumberPeopleRecv*:* número de medio de personas en un email que aparecen como emisoras del mismo.
dvNumberPeopleRecv*:* desviación estándar de la variable anterior.

2.5 Arquitectura del sistema

Como se ha descrito en la introducción de este bloque, E-RETO debe ser una herramienta que permita fácilmente ampliar posteriormente su funcionalidad. Por esa razón es muy importante la creación de una buena arquitectura del sistema basada en módulos completamente independientes.

Se debe hacer una clara distinción en primer lugar entre lo que es la herramienta de escritorio utilizada por el usuario final y el servidor en el que se guardan y tramitan posteriormente los datos generados por E-RETO. La comunicación entre ambos debe hacerse bajo un estándar tecnológico que permita la posible migración a otro servidor en caso de ser necesario, sin impactar de modo alguno en el funcionamiento de la herramienta de escritorio. Una vez hecha esa clara separación de entornos debemos profundizar más en la estructura interna de esta aplicación.

Como se puede apreciar en la figura 3, E-RETO está diseñada bajo el patrón Modelo Vista Controlador [15].

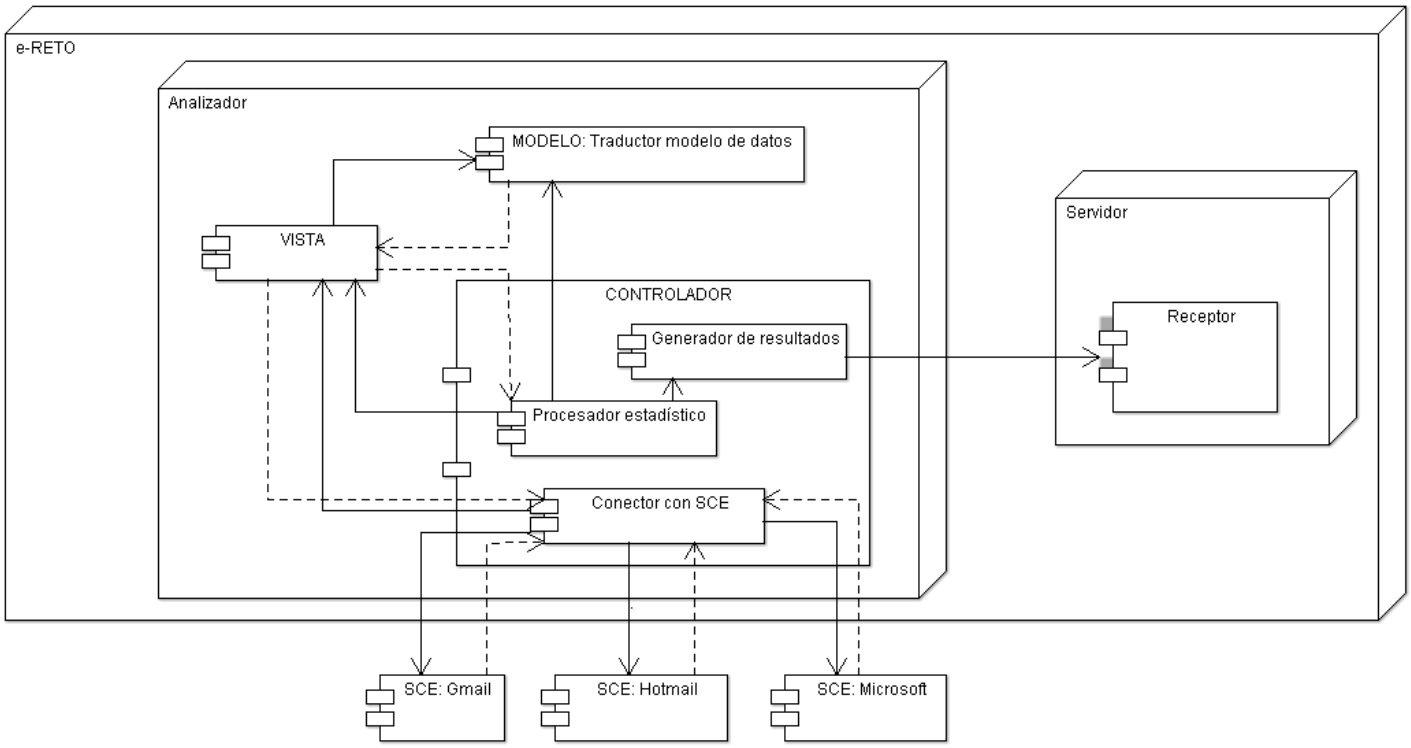


Figura 3 Arquitectura

Dentro del componente analizador, se pueden diferenciar los siguientes componentes:

- Vista: debe ser un módulo totalmente independiente del resto de componentes, de forma que si se quisiera migrar la interfaz gráfica a otra tecnología más amigable al usuario, no se produjese impacto alguno en el funcionamiento del resto de componentes.
- Modelo/Traductor modelo de datos: para el cálculo estadístico de toda la aplicación es necesario contar con entidades de datos manejables por la aplicación, ya que la información proporcionada por los servidores de correo externos es demasiado grande y difícil de manejar. De ahí nace la idea de contar con un componente que se encargue de extraer sólo la información que necesita E-RETO e introducirla en su propio modelo de datos dedicado para sus fines estadísticos.
- Controlador: este componente se encarga de hacer de intermediario entre los dos componentes descritos anteriormente: la vista y el modelo. Está subdividido en varios subcomponentes descritos a continuación:

    - Conector con servidores de correo externos: se debe contar con un módulo específico para la conectividad con servidores de correo externos, de forma que la salida de este componente sea estándar para todos y la ampliación con nuevas cuentas dentro de este componente no produzca la necesidad de modificar la lógica del resto de módulos.
    - Procesador estadístico: este módulo cuenta con una serie de APIs creadas con el fin de calcular todas las estadísticas que conciernen al uso de una cuneta de email. Este sería el componente básico de la herramienta donde se concentra toda la funcionalidad real de la misma. Cualquier ampliación en el alcance que se quiera hacer del estudio para el que se utiliza E-RETO se verá reflejado en este módulo raíz.

    - Generador de resultados: este bloque agrupa la funcionalidad encargada de generar los resultados obtenidos del procesador estadístico de forma que guarden un formato estándar. Además es

el responsable de enviar los datos de una manera que el servidor pueda entender. En el caso de querer modificar la forma conexión con el servidor, este sería el único componente que habría que modificar.

Al margen de la aplicación de escritorio, E-RETO cuenta con un servidor cedido por la Universidad Carlos III de Madrid, en el que se debe permitir recibir los datos generados por el programa para fines estadísticos. De ahí que aquí haya un componente adicional que se encarga de la recepción y procesado de datos generados por E-RETO. Es importante destacar que el alcance de la herramienta no cubre el despliegue de los resultados en el servidor de una forma amigable para el usuario y por ello sólo existe un componente que recibe los resultados y los almacena en el servidor.

2.6 Estudio tecnológico

En esta sección se detalla el estudio de las posibles tecnologías que se pueden aplicar para el desarrollo de los distintos componentes que forman la aplicación. En las siguientes subsecciones se analizarán las tecnologías impuestas junto con las tecnologías aplicables a los distintos componentes definidos en la arquitectura preliminar.

2.6.1 Tecnologías impuestas

Para este proyecto no hay ninguna tecnología impuesta pero si acotaciones derivadas de los requisitos de esta aplicación software.

El lenguaje de programación utilizado para implementar la herramienta debe permitir la creación de una interfaz de usuario amigable para un usuario no experto. Además es requerido que cuente con alguna librería que permita la conexión con servidores de correo externos a través de los protocolos estándar: IMAP [16], POP [17]… y por último debe ser posible su fácil integración en todo tipo de sistemas operativos.

2.6.2 Tecnologías aplicables al componente *Vista*

Para este bloque funcional hay una gama muy amplia de opciones. Una tecnología apropiada en base a los requisitos de esta herramienta es la creación de formularios en HTML [18] con una posible integración con hojas de estilo [19]. Una de las ventajas principales de esta solución es su fácil portabilidad y sobre todo la oportunidad de crear interfaces muy amigables para el usuario. La principal desventaja es que su uso es específico para cubrir las necesidades de este componente y no puede reutilizarse esta tecnología para otros componentes.

En contraste con esta opción, se cuenta con la oportunidad de utilizar algún lenguaje orientado a objetos como Java [20], Grails [21]… que cuentan con librerías extendidas para la creación de interfaces de usuario y además cubren

la desventaja de los formularios HTML en cuanto a su posible utilización en otros componentes de la aplicación.

Al contar con muchas más alternativas de las citadas anteriormente, se ha tenido en cuenta los conocimientos técnicos del programador de la aplicación para filtrar entre sólo algunas tecnologías.

2.6.3 Tecnologías aplicables al componente *Modelo/Traductor a modelo de datos*

Para este apartado se puede hacer uso de múltiples opciones pero en cualquier caso el requisito fundamental es que el lenguaje utilizado pueda manejar el formato de los datos que recibe del componente *Procesador estadístico.* Para el mapeo de objetos se puede utilizar nuevamente numerosas opciones, pero en este documento vamos a centrarnos en Mvel [22] y Smooks [23].

Al ser Java una opción viable en numerosos componentes, es conveniente tener en cuenta Mvel como una opción para esta parte de la aplicación, ya que es un lenguaje de expresión para aplicaciones basadas en Java. Permite hacer un mapeo de objetos con un rendimiento muy superior a cualquier otra librería que se pueda encontrar disponible y además de una forma muy sencilla y amigable para el programador.

Por otro lado se cuenta con la posibilidad del uso de Smooks, un marco de trabajo específico para el procesado de datos en XML (Lenguaje de marcas extensible) [24] u otros formatos (CSV [25], Java, etc.). Es el lenguaje por excelencia para hacer una traducción entre distintos tipos de modelos de datos y además está basado en aplicaciones Java como Mvel. Su principal desventaja es que su uso para el programador no es tan amigable como el de Mvel.

2.6.4 Tecnologías aplicables al componente *Conexión a servidores de correo externos*

En este componente principalmente se realiza la conexión de la aplicación software con servidores de correo externos a través de protocolos estándar como POP3 [17], IMAP [16], etc.

Para este bloque se cuenta con numerosos lenguajes de programación que cuentan con librerías ya creadas para este propósito, pero por continuar filtrando por conocimientos del desarrollador, vamos a hablar de dos de esos lenguajes: Java [20] y Grails [21].

En el caso de Grails, se cuenta con un *plug-in* proporcionado por la propia plataforma grails.org [21] que permite el envío de emails por SMTP [26] pero no la lectura de la bandeja de entrada. Al tratarse de un lenguaje bastante moderno, hay que utilizar alguna librería creada por algún usuario experimentado o en su defecto lo más recomendado es integrar alguna de las librerías estándar de Java. Una de las ventajas de Grails es su facilidad de integrar con Java y el uso de sus librerías.

Si se analiza a continuación las posibilidades que ofrece Java, contamos con un API (Interfaz de programación de aplicaciones) ya integrado dentro del lenguaje, llamado JavaMail [27]. Esta librería cubre el envío y recepción de emails a través de todos los protocolos estándar y es una herramienta muy potente que además cuenta con una fuerte documentación proporcionada por Oracle [28].

2.6.5 Tecnologías aplicables al *Procesador estadístico*

Este componente se encarga del cálculo estadístico y manejo del modelo de datos que se ha creado para esta aplicación software. Se puede hacer uso de múltiples lenguajes pero se ha centrado la atención en uno específico del dominio llamado *R* [29] y un lenguaje orientado a objetos ya citado en otros componentes: Java.

R proporciona un amplio abanico de posibilidades, desde modelos lineales y no lineales hasta análisis de series temporales y algoritmos de clasificación. Es un lenguaje de programación que permite que los usuarios lo extiendan definiendo sus propias funciones aunque para algoritmos computacionalmente exigentes es posible desarrollar bibliotecas en C [30] o Fortran [31] que se cargan dinámicamente.

En el caso de Java, se cuentan con numerosas librerías externas y también es posible crear de forma sencilla tus propias funciones. Una de las múltiples opciones es el proyecto JDistlib [32], que en principio ha pasado la misma batería de pruebas que el lenguaje ya citado en esta sección R [29].

2.6.6 Tecnologías aplicables al componente *Generador de resultados*

Este componente se encarga principalmente de adecuar los resultados obtenidos del componente lógico descrito en el apartado anterior, a un formato que pueda ser enviado a un servidor externo por un protocolo estándar de red.

Como en otras ocasiones, se cuenta con diversas opciones pero se centra la atención en Grails y Java.
Para el envío del fichero de resultados se podía optar por protocolos estándar como HTTP (Protocolo de transferencia de hipertexto) [33] o SMTP ((Protocolo

para la transferencia simple de correo electrónico) [26]. Por facilidad de uso se va a centrar la atención en HTTP y por ello en Grails se puede observar que existe dentro de sus librerías HTTPBuilder [34], con el que se puede hacer un envío al servidor como es requerido en este apartado.

Por otro lado, en el caso de Java se cuenta con la librería Apache HTTP [35] que permite cubrir todas las funcionalidades requeridas para este componente y muchas más.

2.6.7 Tecnologías aplicables al componente *Receptor*

Este componente es el único que se sitúa al margen de la aplicación de escritorio. Su función principal es la de servir de receptor de los datos enviados por el componente *Generador de Resultados*, descrito en el apartado anterior. Es importante destacar que el alcance de este proyecto no cubre la presentación de los resultados en el servidor de forma amigable para el usuario y por ello sólo existe un componente de recepción de los mismos.

Las alternativas tecnológicas para este componente son muy diversas pero siempre dentro del rango de lenguajes que permiten escribir *scripts* dentro de un servidor. Las alternativas que se barajan basándose en los conocimientos del programador son Java y PHP [36].

La razón principal es que son dos de los lenguajes más utilizados hoy en día en el mercado por lo que se presume una facilidad mayor en la implementación de este componente haciendo uso de uno de ellos.

Para la funcionalidad que debe cubrir este componente, tanto Java como PHP pueden ser totalmente válidos y no se puede aportar una ventaja de uno sobre el otro. Quizás desde el punto de vista de integración, si se utiliza Java en otros componentes, tendría bastante sentido el hecho de utilizar Java también en este componente, pero sin embargo es cierto que para los requisitos del sistema, en base a la experiencia en PHP del autor del proyecto se necesitan

menos líneas de código que en Java para cubrir estas necesidades, por lo que ambas soluciones se equiparan a la hora de decidir cuál es más interesante.

2.7 Selección de tecnologías no impuestas

Este software cuenta con numerosos componentes muy independientes entre sí, pero que se retroalimentan. Es por ello que al analizar las múltiples posibles soluciones con las que se cuenta en cada componente, se ha tomado la decisión de utilizar el mismo leguaje para todos los componentes de la aplicación de escritorio, de forma que la depuración del código y sobre todo la sencillez a la hora de integrarlos entre ellos fuese muy alta. Como nota de color se ha tomado la decisión de utilizar PHP en el lado del servidor, ya que el número de líneas de código necesarias se reducía en contraposición al uso de Java o Grails.

Tanto Java como Grails son dos lenguajes válidos para el desarrollo de la aplicación, pero Java es un lenguaje con más edad y documentación por lo que se ha optado por apostar por la seguridad que proporciona Java.

En el componen Vista, se podrá hacer uso de la librería Swing [37] de Java, que si bien no permite una personalización tan compleja como la mezcla de HTML con hojas de estilo, es un componente de fácil portabilidad y conocido por múltiples usuarios.

En el caso de la conexión con los servidores de correo, como ya se explicó en el apartado dedicado a este componente, Java cuenta con una librería denominada JavaMail, que cubre satisfactoriamente el requisito de acceso a un servidor de correo externo por protocolos estándar como POP3 [17],IMAP[16]…

Para el componente Traductor a modelo de datos, ambas opciones citadas en el apartado son destinadas a aplicaciones Java, pero al utilizar el mismo lenguaje en todos los componentes, no es necesario el uso de otro lenguaje

adicional para adaptar la salida del componente que conecta con los servidores de correo y el procesador estadístico.

En el caso del ya citado procesador estadístico, tras analizar en profundidad que sólo es necesario el cálculo de sencillas funciones, se ha optado por utilizar una interfaz creada por el programador que haga los cálculos deseados.

En el caso de la generación y envío de datos, se ha optado por crear un fichero XML [24] que puede ser enviado mediante el API Apache citado en el apartado correspondiente a este componente. Para la generación del fichero XML se ha decidido crear un API dedicado usando la librería Java JAXP [38].

2.8 Diseño de plan de pruebas de aceptación

Una vez que se han definido los requisitos que debe cubrir esta aplicación, se debe definir un plan de pruebas de aceptación que verifiquen que se han cumplido todas las necesidades descritas en los requisitos ya citados.
En esta sección se van a detallar todas las pruebas mínimas que deben ser superadas y además los resultados esperados de las mismas para la verificación de los resultados. La Tabla 6 detalla las pruebas de aceptación definidas de acuerdo al formato de la plantilla 2 del Anexo 3:

Identificador	Requisitos cubiertos	Pasos a ejecutar	Salida
PA-01		1.Creación de una cuenta válida de Gmail: 1.1 Usuario -> *usuario* 1.2 Contraseña -> *contraseña* 2. Abrir E-RETO 3. Indicar al sistema que comience 4. Seleccionar como usuario: *usuario* 5. Seleccionar como contraseña: *contraseña* 6. Seleccionar como servidor de correo: gmail 7. Aceptar contribución al estudio 8. Aceptar bases legales 9. Leo qué hace E-RETO con mis datos 10. Leo qué NO hace E-RETO con mis datos 11. Indico al sistema que se conecte	El sistema debería conectarse al servidor de correo con esos credenciales y mostrar información al usuario para que rellene sus datos personales para el estudio.

Identificador	Requisitos cubiertos	Pasos a ejecutar	Salida
PA-02		1.Creación de una cuenta válida de Gmail: 1.1 Usuario -> *usuario* 1.2 Contraseña -> *contraseña* 2. Abrir E-RETO 3. Indicar al sistema que comience 4. Seleccionar como usuario: *usuario* 5. Seleccionar como contraseña: *Otracontraseña* 6. Seleccionar como servidor de correo: gmail 7. Aceptar contribución al estudio 8. Aceptar bases legales 9. Leer qué hace E-RETO con mis datos 10. Leer qué NO hace E-RETO con mis datos 11. Indicar al sistema que se conecte	El sistema debería conectarse al servidor de correo con esos credenciales y al no ser los correctos, mostrar un mensaje de error de autenticación de usuario.
PA-03		1. Conectarse al servidor de gmail con usuario *usuario* y contraseña *contraseña.* 2. Crear buzón de correo: *buzon 1* 3. Crear buzón de correo: *buzon 3* 4. Desconectarse del servidor de gmail 5. Ejecutar prueba de aceptación PA-01 6. Rellenar nacionalidad con: *España* 7. Rellenar edad con *18* 8. Rellenar sector con *Agricultura* 9. Rellenar sexo con *Hombre* 10. Indicar al sistema que cargue los emails	El sistema debería mostrar que ha cargado *buzón 1, buzón 2* y el Inbox por defecto de una cuenta de email.

Identificador	Requisitos cubiertos	Pasos a ejecutar	Salida
PA-04		1. Realizar prueba de aceptación PA-03 2. Indicar al sistema que se desea continuar 3. Seleccionar *buzon 1* y añadirlo como laboral 4. Cambiar al idioma inglés. 5. Indicar al sistema que debe continuar con la ejecución	El sistema debe mostrar la interfaz de resultados en el idioma inglés. En el servidor debe haber un fichero XML cuyos valores están todos vacíos al no haber emails en los buzones y los datos personales rellenos con los siguientes valores: Country: ES Sector: 63 Gender: MAN Age: 18
PA-05		1. El usuario abre cuenta de email creada en PA-01. 2. Envía 10 emails a 10 personas distintas. 3. Realizar prueba de aceptación PA-04 4. Seleccionar top 10 de los últimos 6 meses 5. Pulsar para generar gráfico.	El sistema debe mostrar un gráfico de sectores en el que muestre los 10 emails a los que se ha enviado un email con un porcentaje exactamente igual para todos. (10%)
PA-06		1. Se ejecuta PA-05. 2. Se llama al *script* del servidor que muestra estadísticas actuales almacenadas.	El sistema debe haber rellenado todos los datos estadísticos de dominio personal con los correspondientes al envío de esos 10 emails.

Tabla 6 Pruebas de aceptación

Capítulo 3

Diseño detallado

En este capítulo se hará una profundización en el análisis realizado en el capítulo anterior.

3.1 Diseño Software

En esta sección se va a profundizar en todos los componentes del apartado de análisis. Es posible que citados componentes deban ser subdivididos en partes más pequeñas que serán detalladas en este apartado. Se debe destacar que toda la implementación se ha realizado utilizando el inglés como idioma por lo que los nombres en los diagramas de clase aparecerán en el ya citado idioma.

3.1.1 Componente Vista

Como ya se explicó en apartados anteriores, este componente contiene todas las interfaces gráficas con las que el usuario puede interactuar. Al margen de los manejadores de eventos de los elementos de los formularios, se deben destacar dos métodos en estas interfaces: *checkValidations() y setTextForAll().*

A continuación se muestran todas las clases que forman este componente.

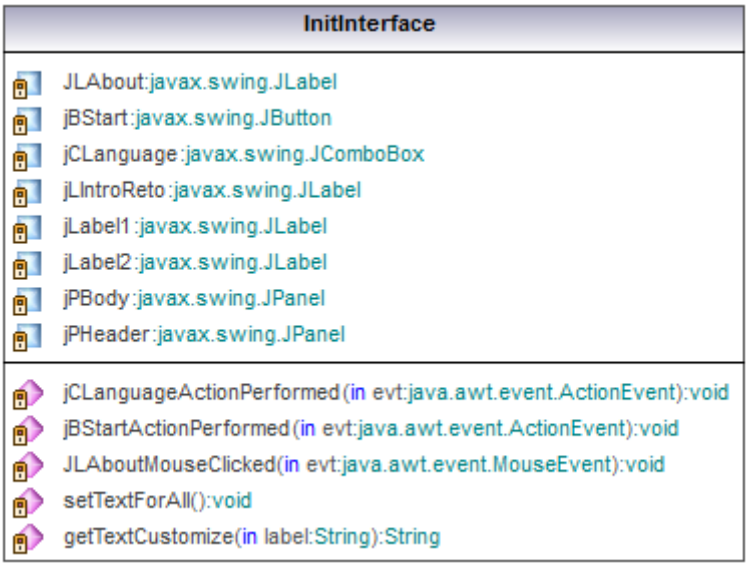


Figura 4 InitInterface

Como se aprecia en la Figura 4 se hace uso del método *setTextForAll()* que se encarga de asignar a todos los elementos de esta Interfaz el valor de su texto correspondiente en función del idioma que esté seleccionado. Para ellos se accede a un fichero label.properties donde están almacenados todos los posibles valores.

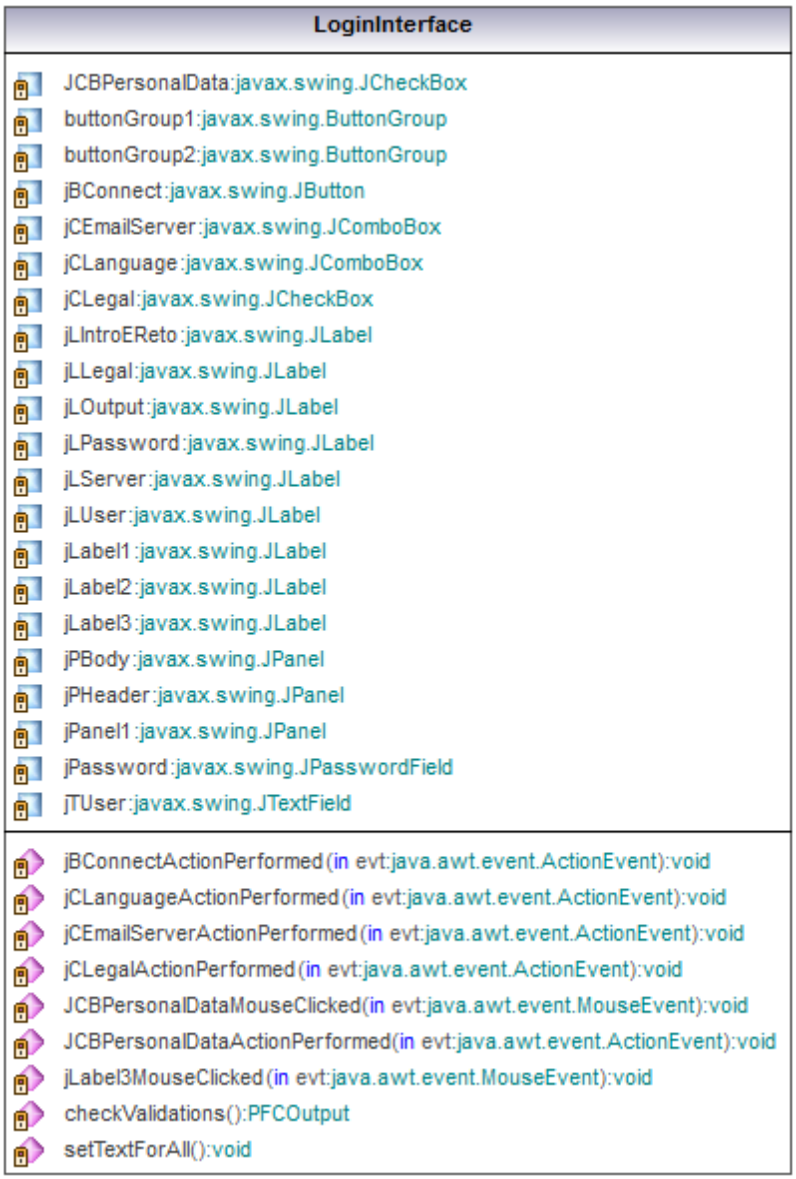
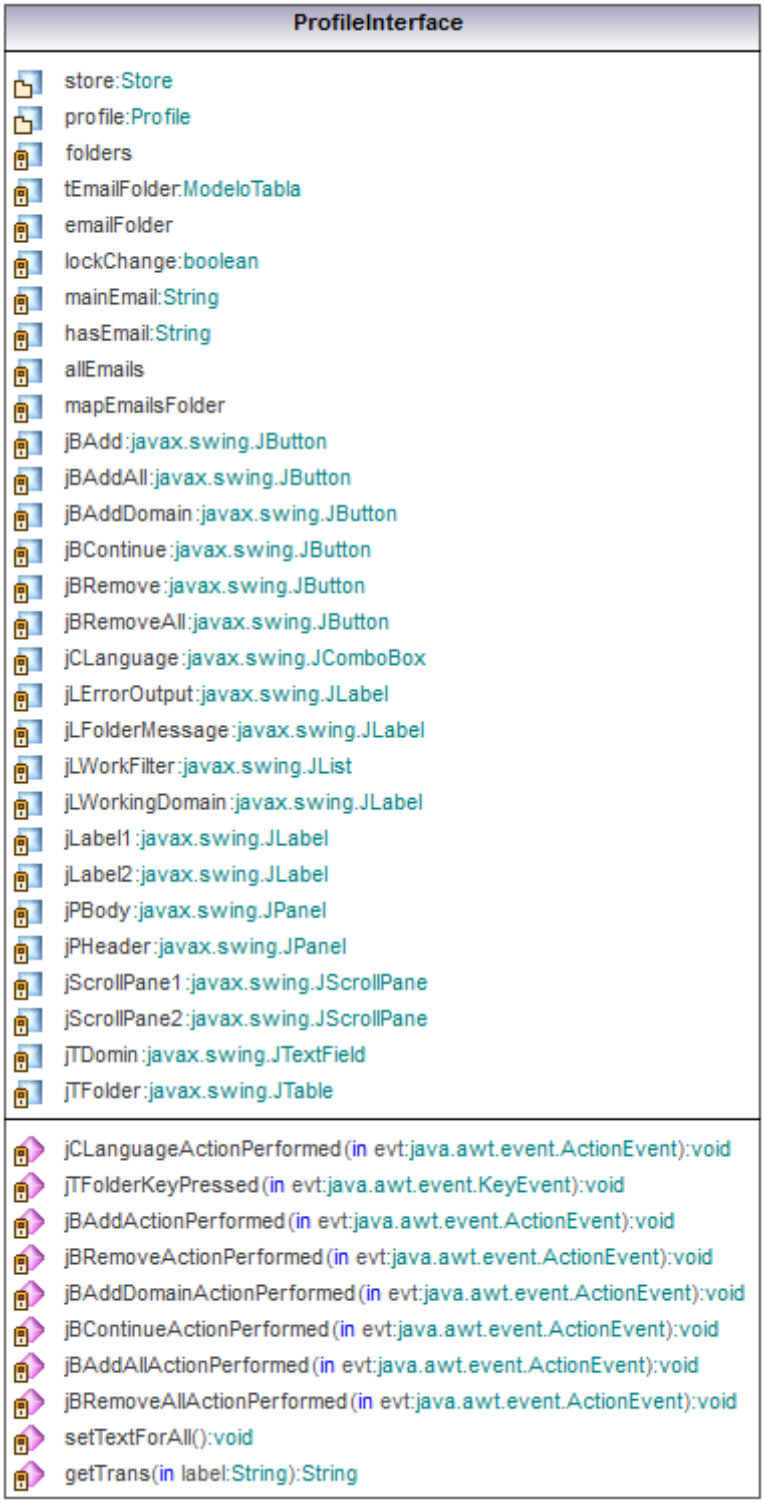


Figura 5 LoginInterface

En la clase mostrada en la Figura 5 se hace uso por ejemplo del método *checkValidations()*, que se encarga de todas las validaciones de los campos introducidos por el usuario. Cabe destacar que si se aprecia en la salida de este método, es un objeto llamado *PFCOutput*. Esta clase se explicará en detalle más

adelante, pero básicamente lo que se encarga es de devolver una salida estándar, ya sea con el error producido en el error en el controlador o su salida deseada.

A continuación se muestra en la Figura 6 el resto de interfaces que en muchos casos hacen uso de los métodos ya citados anteriormente.



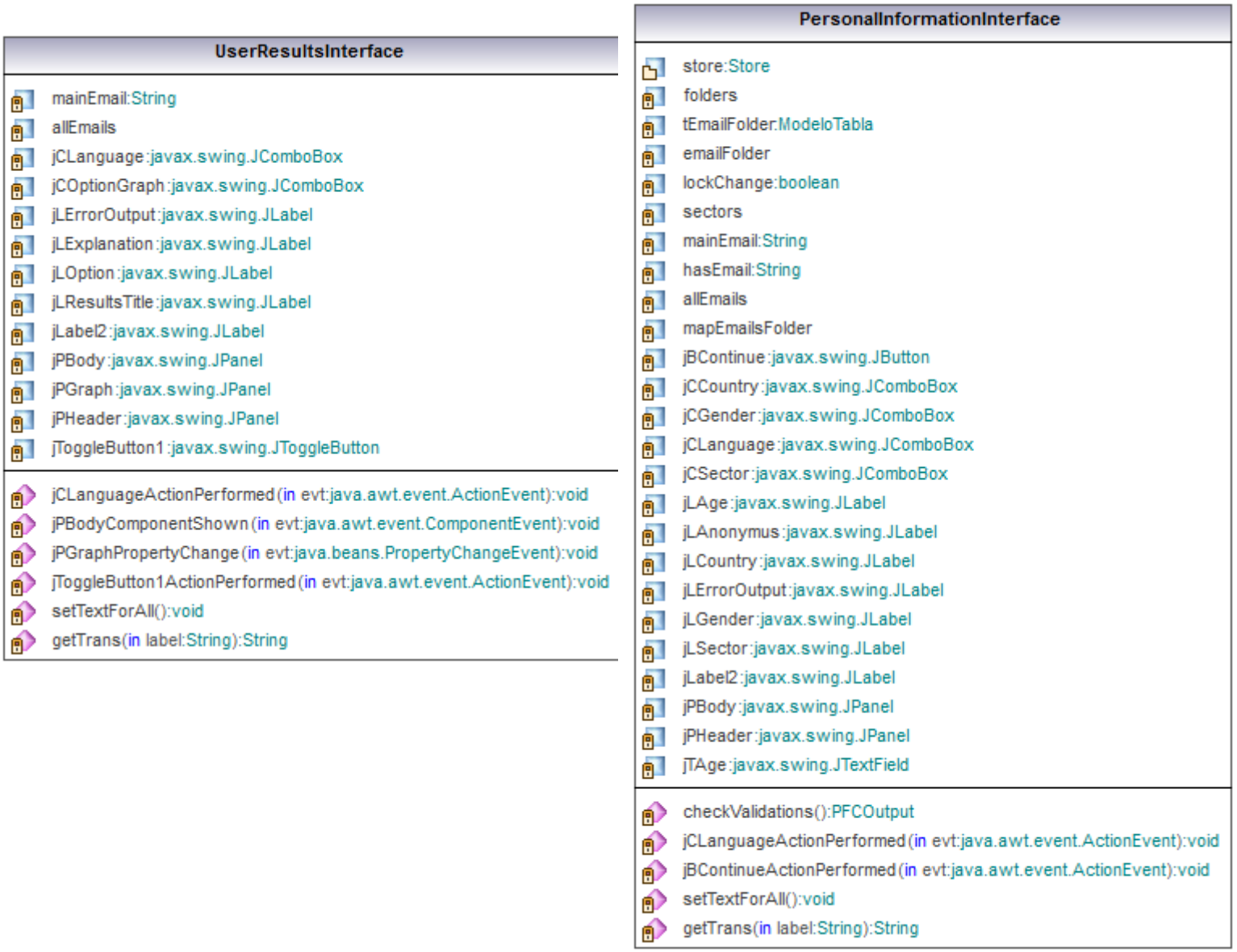


Figura 6 ProfileInterfaces

3.1.2 Componente Modelo/Traductor de modelo de datos

Como ya se explicó en apartados anteriores, este componente se encarga de la modelización de la información con la que opera todo el sistema.

Para la conexión con los distintos servidores externos se ha definido una serie de subClases específicas que heredan de una clase más genérica *Conection*, que contiene todos los atributos comunes a las subclases específicas para cada servidor externo. Se debe destacar que al sólo ser posible la conexión a la vez a un servidor de correo, para aumentar la seguridad se ha optado por utilizar el patrón de diseño Singleton en cada una de las subclases. Esta implementación se puede apreciar en la Figura 7.

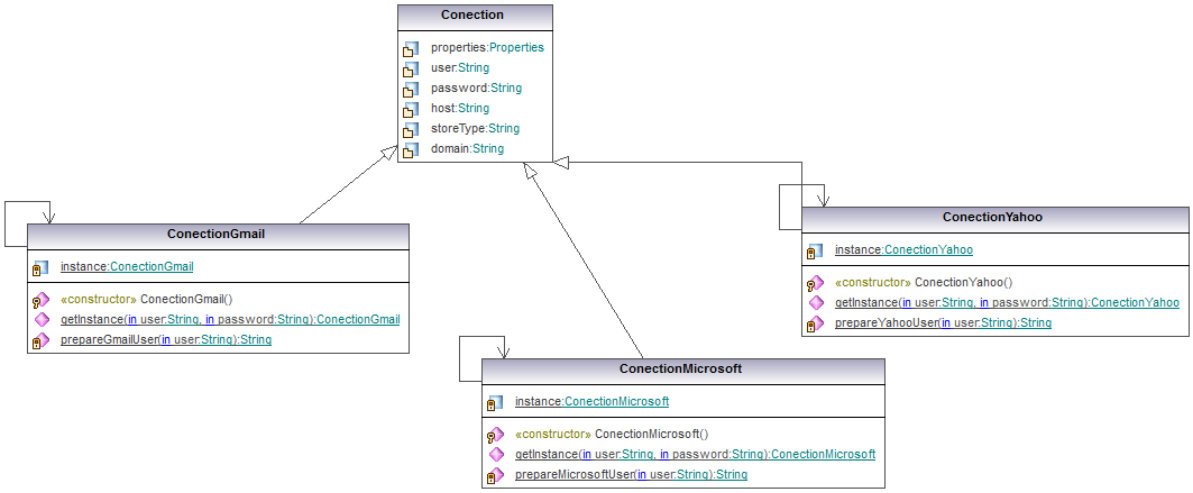


Figura 7 Conection

Con esta distribución, en el caso de querer añadir un nuevo servidor de correo externo, simplemente habría que crear una nueva subclase que heredase de *Conection* donde se especificasen las necesidades propias de la conexión a ese servidor de correo.

Para el manejo de constantes en el componente *Controlador*, el sistema cuenta con dos clases que le proporcionan todos los valores contsnates de los que se hace uso en la aplicación como se aprecia en la Figura 8.

Figura 8 Constantes

El manejo de los emails recibidos por los servidores externos es intratable. Cada mensaje recibido de los servidores cuenta con una cantidad de datos inmensa de la que E-RETO no va a hacer uso y por ello se hace uso de un modelo de datos propio con la información exclusiva a manejar, mostrado en la Figura 9.

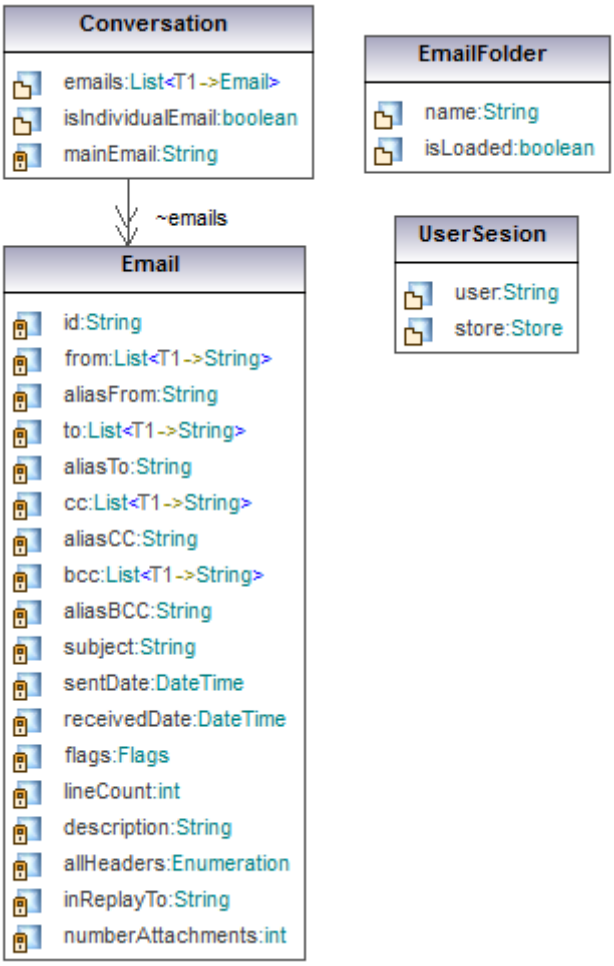


Figura 9 Email

De estas clases cabe destacar la clase *Conversation*, que es un concepto creado con el fin de agrupar emails que guardan una relación muy estrecha. En una clase *Conversation* se agrupan emails que por ejemplo han sido un hilo entre ellos con el mismo asunto, pero también se incluyen emails que nos llegan regularmente a nuestra cuenta con un formato estándar para ofrecernos publicidad, noticias…

Para el manejo de las salidas esperadas por la interfaz, se hace uso de la clase ya citada en otros apartados PFCOutput, mostrada en la Figura 10.

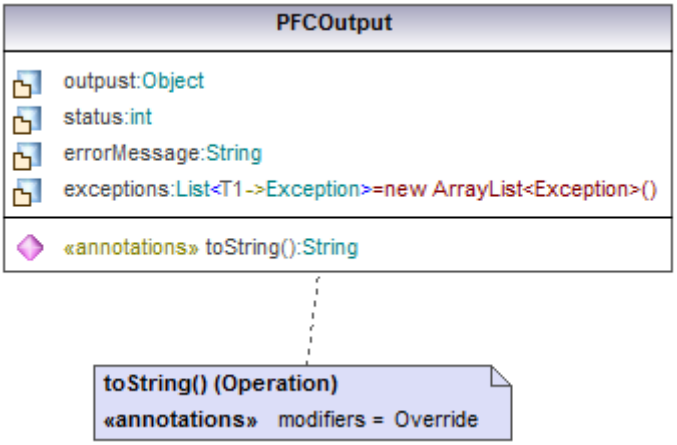


Figura 10 PFCOutput

En esta clase se puede observar que se pueden almacenar los distintos errores generados en el Controlador y se pueden mostrar por pantalla directamente con el método sobrescrito *toString()*.

Como en esta aplicación se va hacer uso de numerosas comparaciones entre objetos, fechas, etc. para las estadísticas, ha sido necesaria la creación de varios comparadores adaptados para su uso en E-RETO, mostrados en la Figura 11:

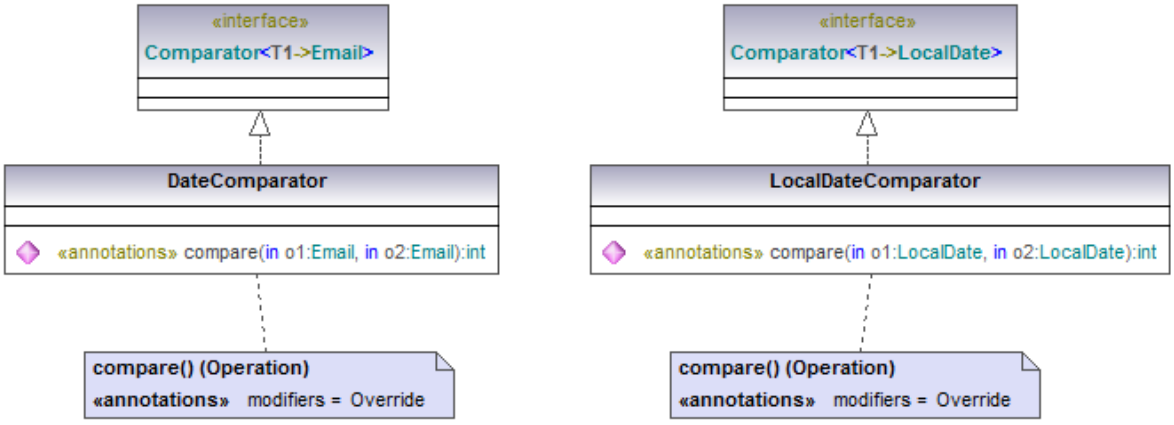


Figura 11 Comparadores

Finalmente, dentro de este componente se debe hacer mención de las clases que ayudan a modelizar los resultados obtenidos en el componente *Controlador*. Su implementación se detalla en la Figura 12.

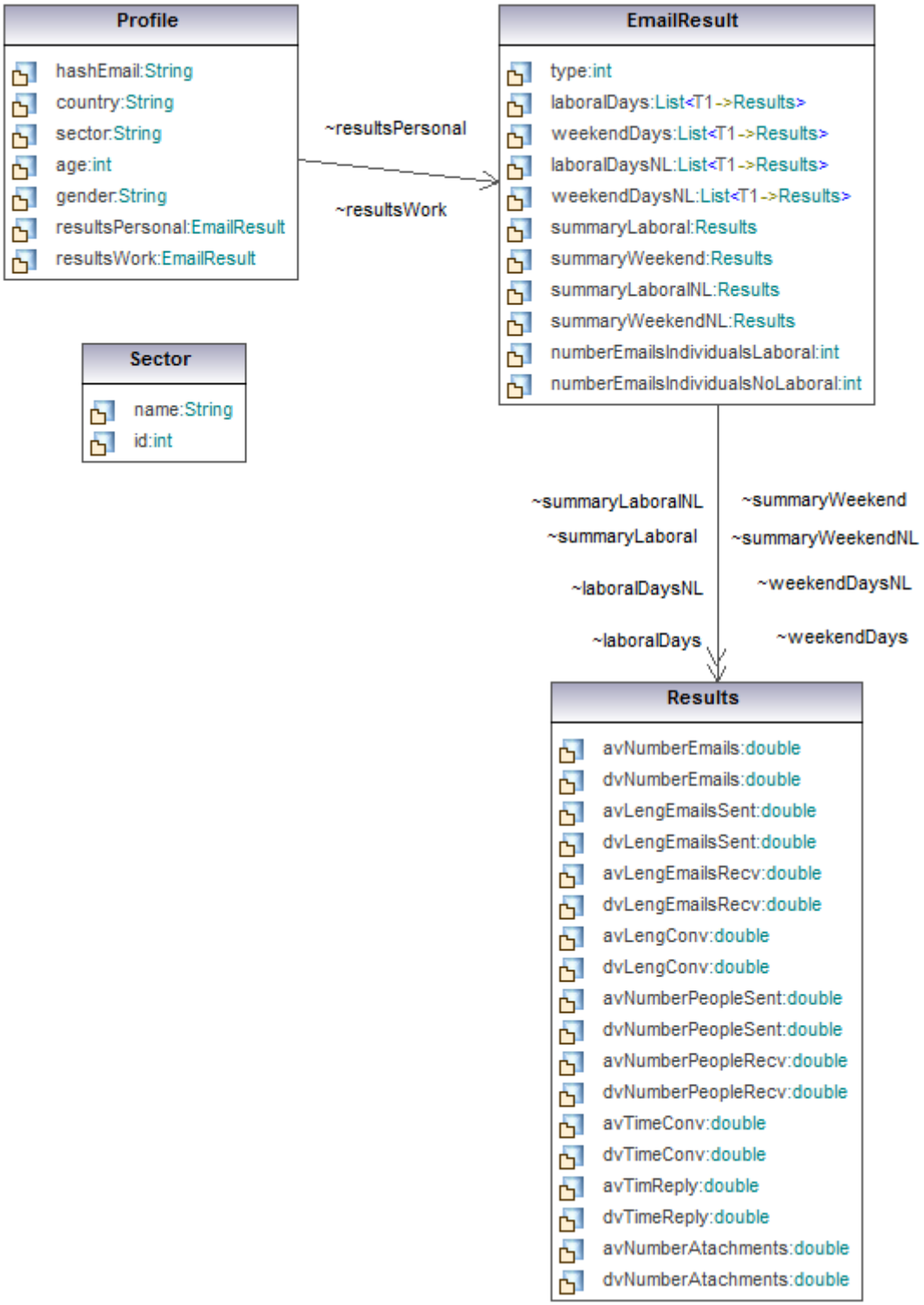


Figura 12 Results

3.1.3 Componente Controlador

Desde el punto de vista de la arquitectura, se ha hecho una división interna de este componente en base a sus componentes funcionales, pero en el caso del código, la correspondencia no es tan directa al agrupar métodos de distintos subcomponentes en la misma clase.

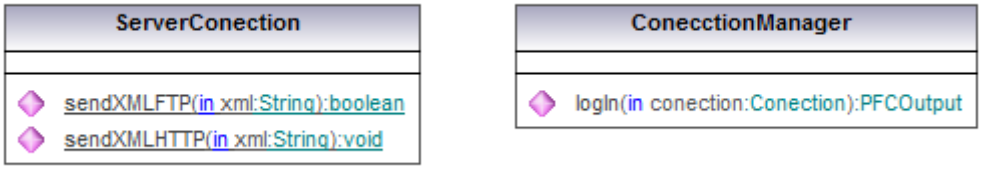


Figura 13 Conexiones con el servidor

Como se aprecia en la figura 13, para la conexión con los servidores externos se hace uso de la clase *ConectionManager* que cuenta con un único método para iniciar sesión en una cuenta de email. Como se puede apreciar en el diagrama, el parámetro de entrada es un objeto *Conection* ya visto en apartados anteriores. La salida es el objeto *PFCOutput,* que permite el poder hacer manejo de los errores que puedan surgir al realizar el logIn.

Por otro lado, en el subcomponente *Generador de resultados*, se hace uso de la clase *ServerConection,* para poder enviar al servidor el fichero XML de resultados por HTTP [33] o FTP [42]. En la implementación actual se hace uso de HTTP, pero el método ha sido creado por si fuese necesaria su utilización en el futuro.

Para el manejo de excepciones, estilos de las etiquetas y los idiomas, se cuenta con unas clases específicas mostradas en la Figura 14.

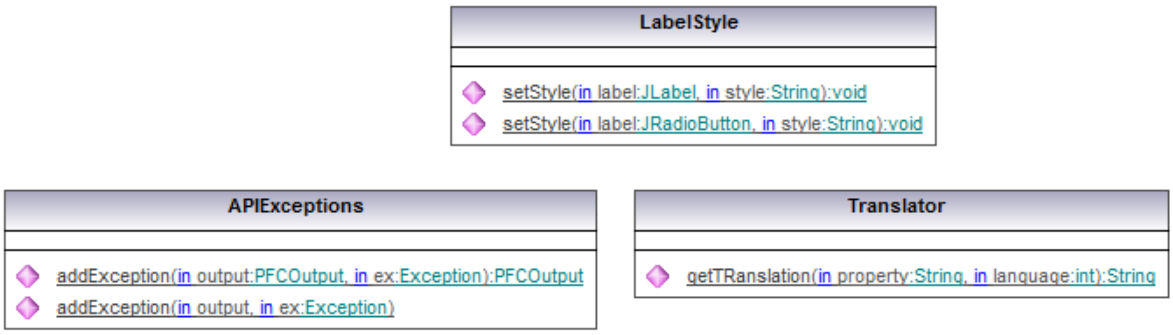


Figura 14 Etiquetas y estilos

A continuación se describe en la Figura 15 la clase fundamental que es utilizada en todos los subcomponentes del *Controlador*.

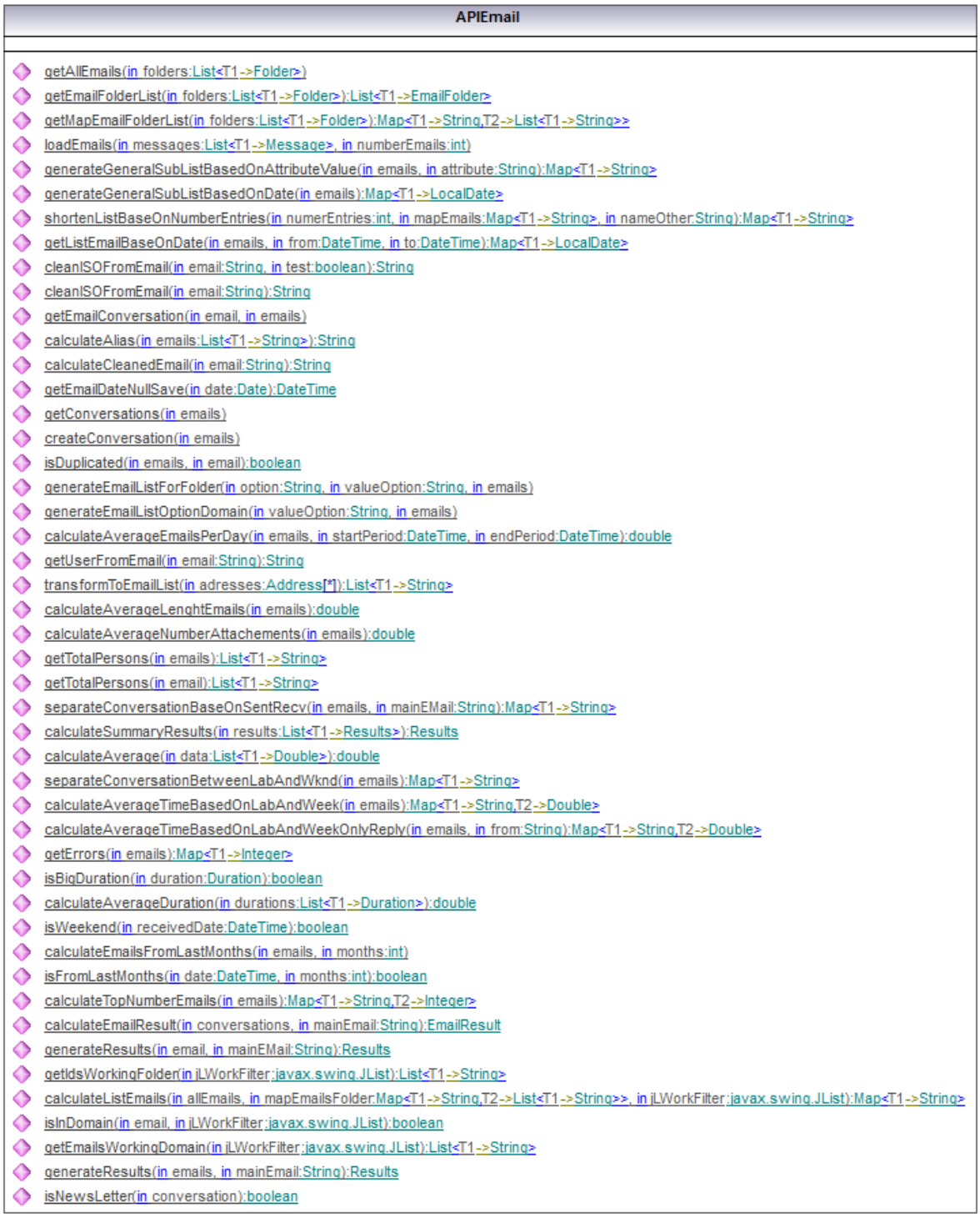


Figura 15 Manejo de Email

Los métodos más importantes de esta clase son los siguientes:

getAllEmails*():* este método hace uso de la librería JavaMail API[27] para hacer la carga de todos los emails de una lista de buzones.

El problema que surge al utilizar directamente el método de carga de mensajes proporcionado por este API, es que su rendimiento no está optimizado y hacer una carga de veinte a treinta mil emails en menos de cinco minutos es una tarea imposible.

Para poder lograr este objetivo es necesario hacer una preselección en el servidor de correo de forma que cuando se llame al ya citado método, la carga no exceda de los cuatro o cinco minutos. Esta preselección se hace gracias a una clase proporcionada por este mismo API llamada *javax.mail.FetchProfile* [27].

Uno de los problemas que plantea el uso de los buzones de los emails, es que los servidores de correo electrónicos almacenan sus emails de una forma indizada en la que cuando un email se sitúa en más de un buzón y se trata de extraer los emails de ambos buzones, citado email aparecerá duplicado.

Para solventar este problema, se ha optado por utilizar un java.util.*HashMap* [43] de Java, que permite organizar objetos en función de una llave única. En este caso esa citada llave única es una propiedad de los mensajes que no admite duplicados denominada *message-id*. Con ayuda de este HashMap es posible almacenar todos los emails sin duplicados.

loadEmails(): este método es invocado por el ya citado *getAllEmails* anteriormente y es el que realmente extrae la información de cada mensaje y la transforma en un objeto del modelo de datos propio de E-RETO. Cabe destacar que los métodos proporcionados por JavaMail no devuelven listas de emails en el caso de por ejemplo querer acceder al emisor de un email, sino que devuelve un texto con toda la lista. Es por ello que se requiere de la función auxiliar *transformToEmailList(),* creada por el programador, para poder adaptarlo al modelo de datos propio de E-RETO.

La forma de recibir los emails de los servidores externos de correo es variada y a veces compleja. Para poder contar con unos datos fiables y que se puedan comparar entre sí, es necesario hacer una estandarización de los mismos con métodos como *cleanISOFromEmail(),* que en este caso por ejemplo elimina las anotaciones ISO y se queda con el email en sí almacenado entre los símbolos „<>“.

Uno de los puntos críticos en cuanto a dificultad en la implementación de E-RETO se encuentra en la forma de definir si un email pertenece a un hilo ya existente. El problema se basa en que no existe una implementación eficaz 100%, ya que muchos servidores de correo no rellenan correctamente las cabeceras de los mensajes.

La opción por la que se ha optado es la de utilizar una serie de certezas que excluyan casos inválidos obvios y no dejen espacio al error. Para empezar, a pesar de no contar con datos oficiales, por intuición se puede afirmar, que la probabilidad de que un asunto de un email sea exactamente el mismo al de otro que no está relacionado es bastante baja y que aumenta exponencialmente con la longitud del asunto.

Pero para evitar posibles errores, se han añadido más validaciones. La primera es la de comprobar que si el mensaje cuenta con la cabecera rellena *in-reply-to*, el contenido de este campo debe pertenecer a alguno de los mensajes del supuesto mismo hilo. En caso contrario se puede afirmar con una certeza del 100% que este email no pertenece a este hilo.

A pesar de que con estas comprobaciones parecía que la implementación no corría el riesgo de cometer errores, durante las pruebas de integración se descubrió que en el caso de asuntos de emails con una longitud muy pequeña, se podía cometer un error si los emails tenían un dominio de actuación similar. De hecho, el caso específico fue el de dos emails de empresa cuyo asunto en ambos casos era *HC1*. El problema era que había un hilo de varios emails y pasado más de 1 año, otro email que no contestaba a este hilo pero con el mismo asunto y con destinatarios comunes a los del hilo. Fue un caso de entre más de treinta mil emails, pero para prevenir estos errores, se ha presupuesto que si la diferencia temporal de dos emails es de más de seis meses, no tienen relación entre sí. Incluso aunque un email sea contestado realmente pasado

esos seis meses y E-RETO esté cometiendo un error en su filtro, desde el punto de vista estadístico tampoco tiene sentido hacer esas mediciones por lo que es correcto su tratamiento por separado.

getListEmailBaseOnDate(): este método recibe un periodo de tiempo y es capaz de devolver los emails que hayan sido enviados o recibidos en ese periodo. Es una función auxiliar que se utiliza en varios métodos, pero que deja patente cómo se ha hecho uso de las fechas en esta implementación.

El manejo de periodos temporales en Java es una tarea a veces bastante compleja, pero en este caso se ha hecho uso del API Joda-Time [44]. Además, para poder manejar unas estructuras de datos tan grandes con listas de miles de objetos, se han creado como se mostró en apartados anteriores de la memoria, comparadores de fechas como *LocalDateComparator,* que sobrescriben los métodos de la interfaz *java.util.Comparator* [45] de Java, con fechas del API Joda-Time.

calculateEmailResult(): este método es uno de los más importantes en la elaboración de los resultados que se enviarán al servidor, al contar con la lógica de todo el proceso de subdivisión de las listas de emails.

Crea una serie de listas para almacenar por separado los emails de ámbito laboral, personal, etc. Y posteriormente llama a funciones auxiliares para que hagan los cálculos. Cabe destacar que se han creado funciones separan la totalidad de los emails en función de perfiles y después funciones que reciben una lista de emails y hacen unos cálculos determinados. De esta forma es posible reutilizar todos los métodos y no tener que crear duplicados innecesarios.

Para el manejo de los XML se hace uso de la clase de la Figura 16.

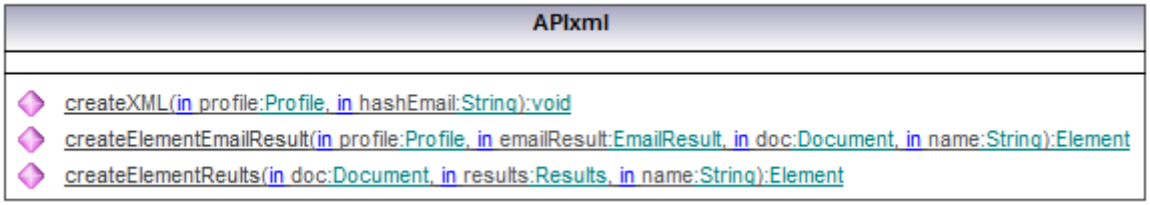


Figura 16 Manejo de xml

Como se aprecia en la figura anterior, estos métodos hacen uso del modelado de datos proporcionado por el componente *Modelo*.

Para mostrar los gráficos el sistema utiliza la clase de la Figura 17:

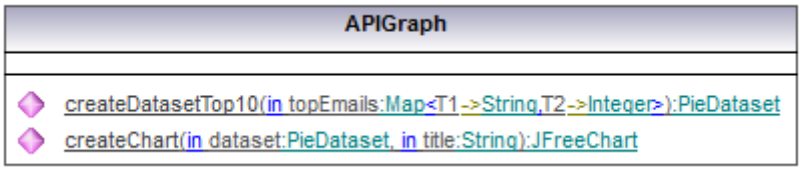


Figura 17 Gráficos

Esta clase hace uso de la librería Java JfreeChart [46], que permite de una forma bastante sencilla la creación de gráficos totalmente personalizados.

Para el cálculo estadístico que no está cubierto por funciones simples en Java, se hace uso de una clase creada para este fin que es la mostrada en la Figura 18.

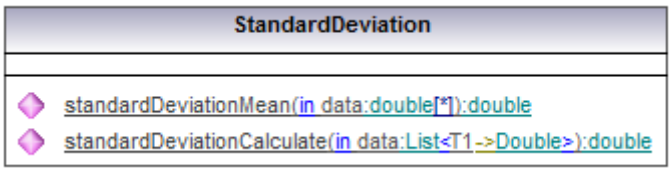


Figura 18 Desviación estándar

Con esta clase se puede hacer un cálculo de la desviación estándar media de una lista de valores.

E-RETO ha optado por una implementación *multihilo* durante la carga de los emails, de forma que sea posible informar al usuario durante el proceso. Se ha decidido mostrar en la memoria la implementación de esta funcionalidad por separado, al tratarse de algo un poco más particular frente al resto de clases. Se puede apreciar dicha implementación en la Figura 19.

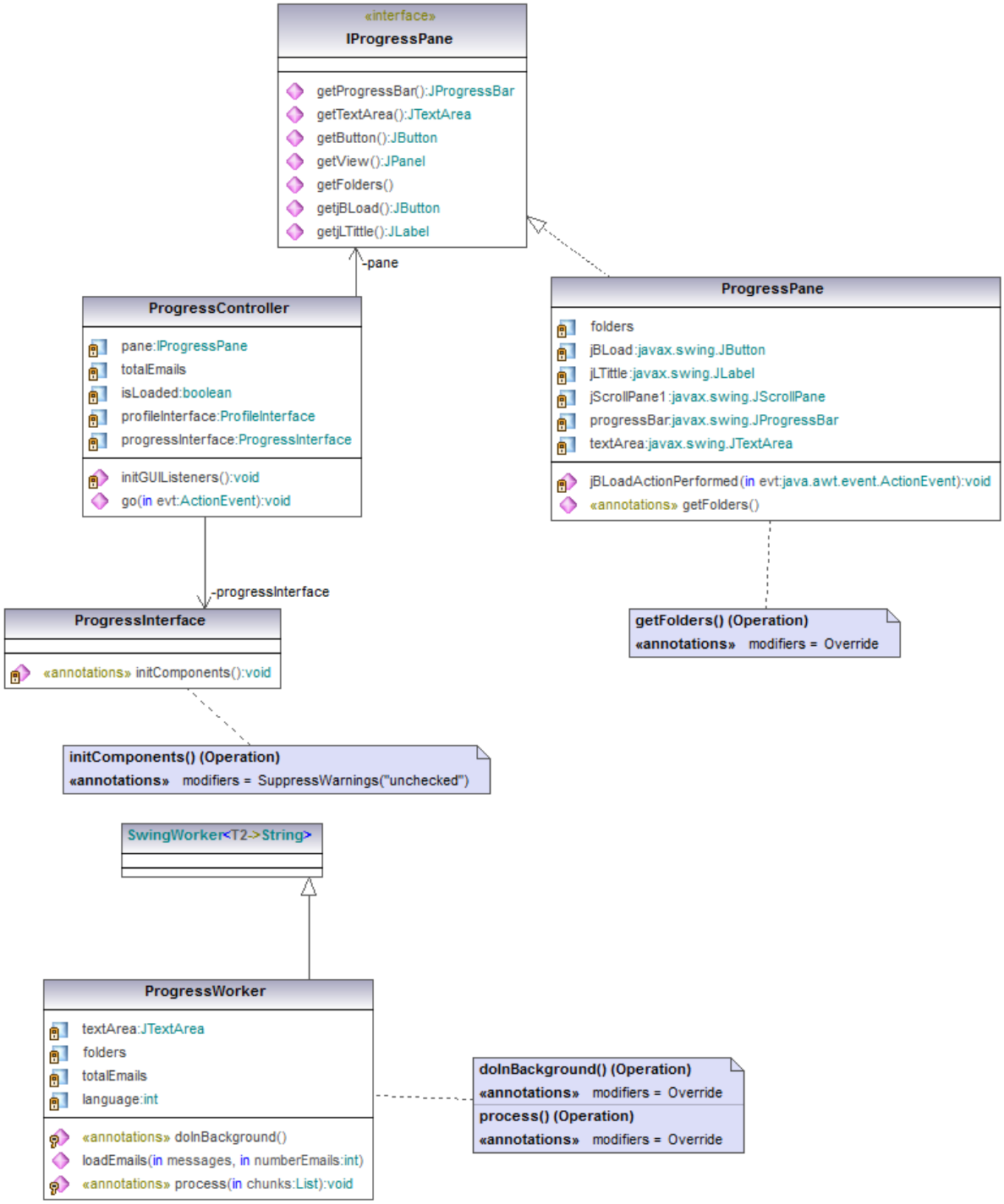


Figura 19 Barra de progreso

3.2 Diagramas de secuencia

En este apartado se van a mostrar distintos diagramas de secuencia asociados a cada caso de uso. En el caso del CU-01, al ser tan grande no es manejable mostrar todos los diagramas de secuencia asociados al mismo y se ha optado por mostrar el que corresponde a la visualización de los emails cargados.

En la Figura 20 se puede apreciar cómo el controlador hace una carga de todos los componentes gráficos de la aplicación en caso de no haber sido ya cargados *!isLoaded.*

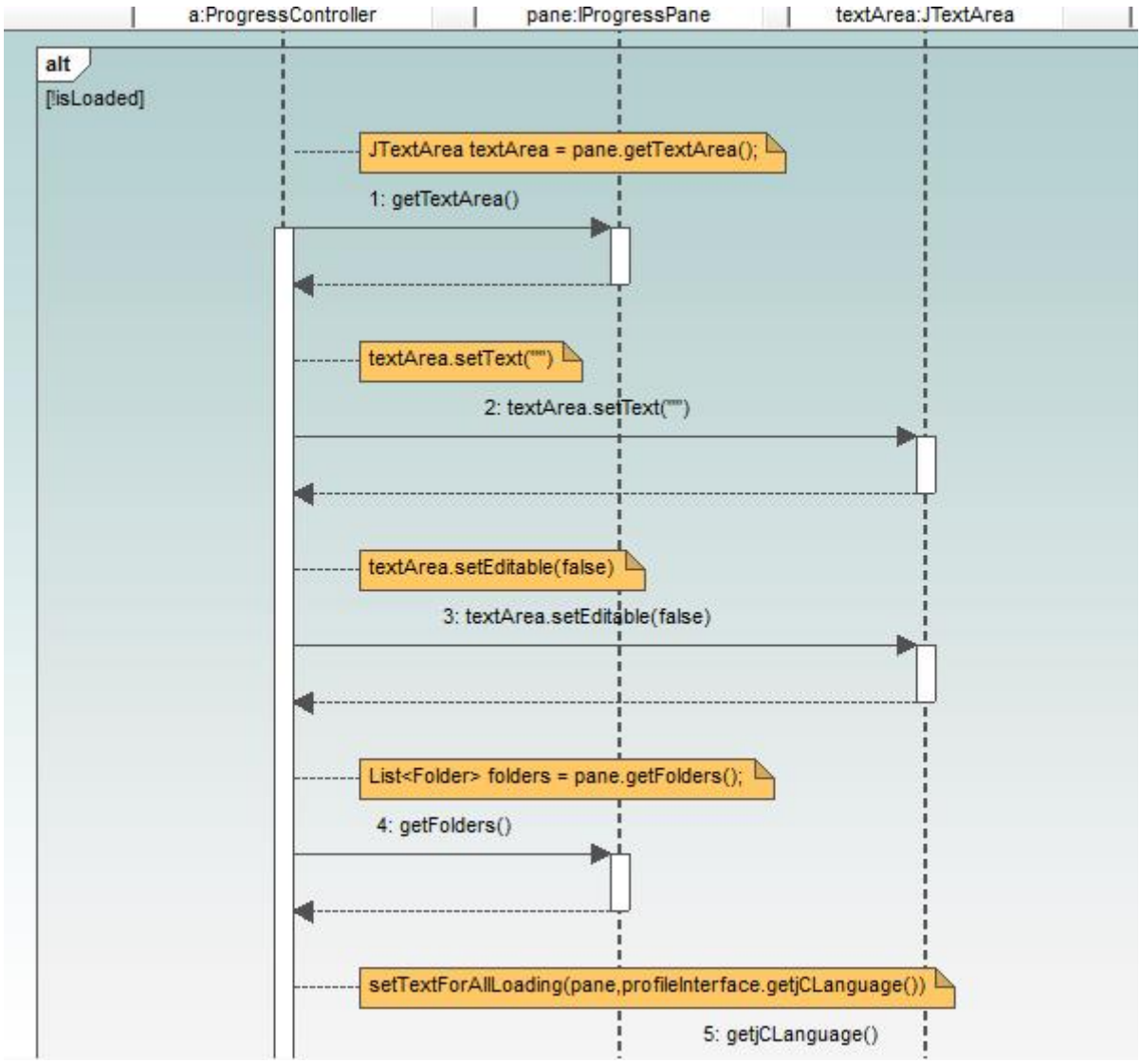


Figura 20 Diagrama de secuencia loadEmails

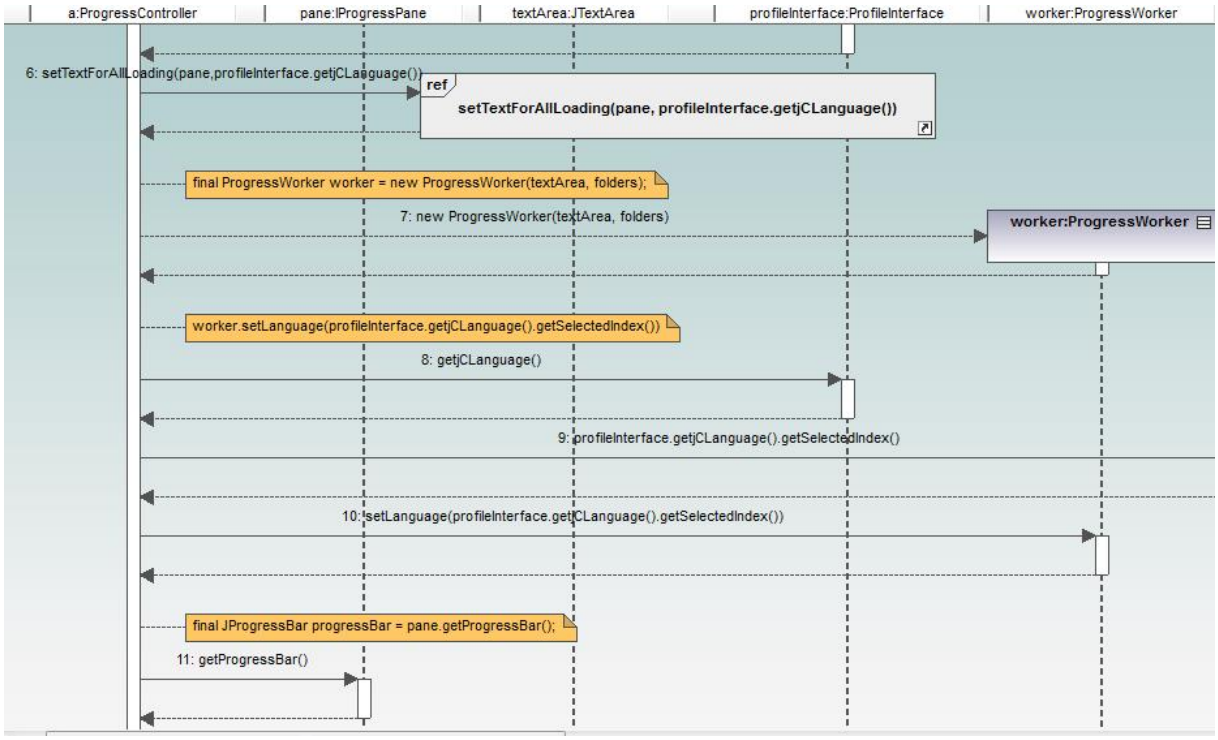


Figura 21 Carga emails

Esta Figura 21 sigue mostrando el proceso de carga de todos los componentes gráficos del sistema y cabe destacar la llamada al método s*etTextForAllLoading,* que se encarga de de completar las etiquetas de todos los componentes de la aplicación.

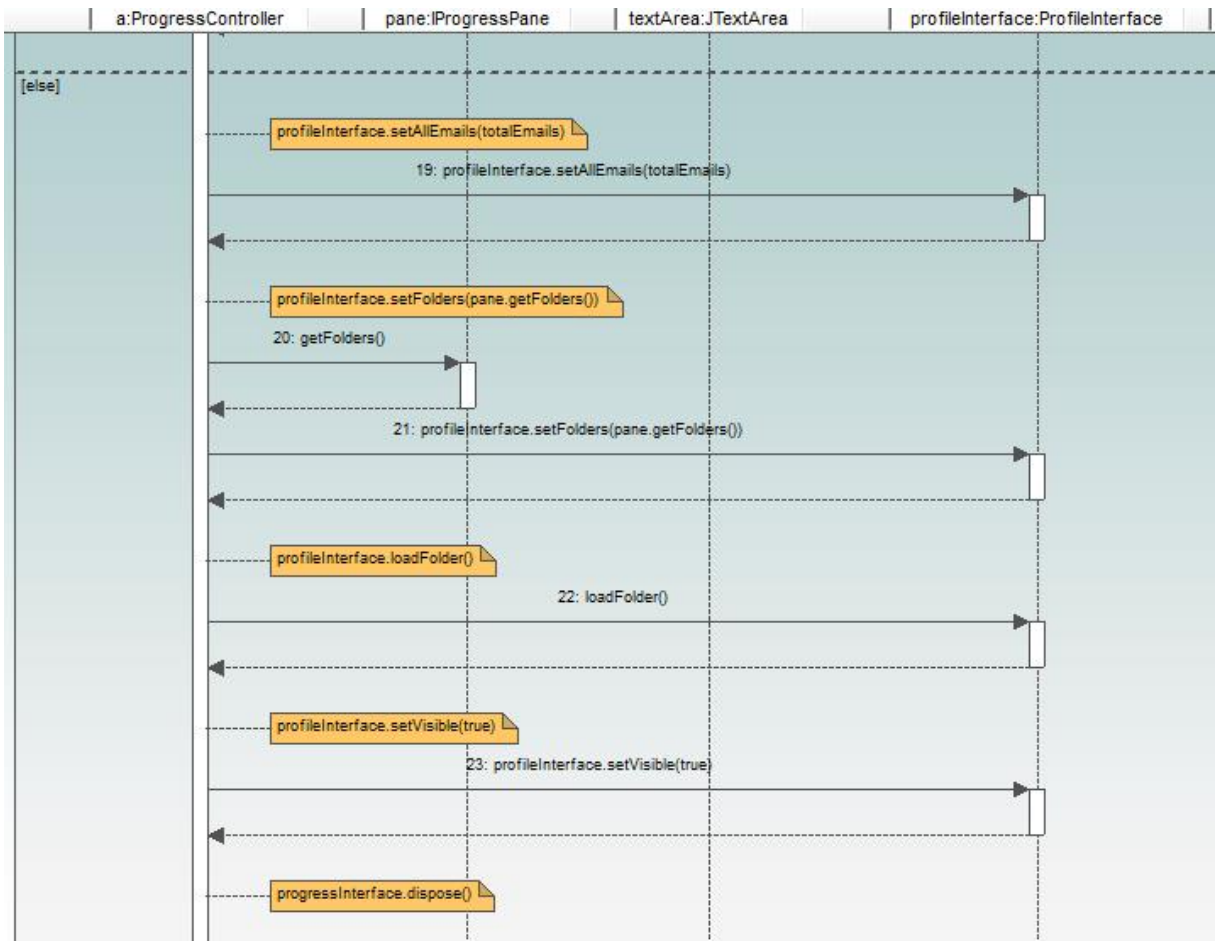


Figura 22 Componentes gráficos

En la figura 22 se muestra el bloque ejecutado en caso de que los componentes gráficos hayan sido ya pre-cargados. En el paso 19 se ve cómo el controlador ya cuenta con la lista de emails cargados y hace llamadas a métodos para obtener la lista de buzones seleccionados. Una vez seleccionados todos los buzones, se hace una llamada al método de carga de buzones para mostrarlos por pantalla y cerrar la aplicación de progreso de la interfaz.

El caso de uso CU-02, en el que se muestran resultados estadísticos al usuario en forma de gráfico, queda representada la visualización de los mismos.

En la figura 23, se puede observar cómo se hacen las primeras instancias de los objetos a utilizar para la creación de los gráficos. Se parte de un *dataset* con los resultados y el título del gráfico.

Se hace uso de la factoría *CharFacory* para crear una instancia del objeto JFreeChart con el que se representará el gráfico.

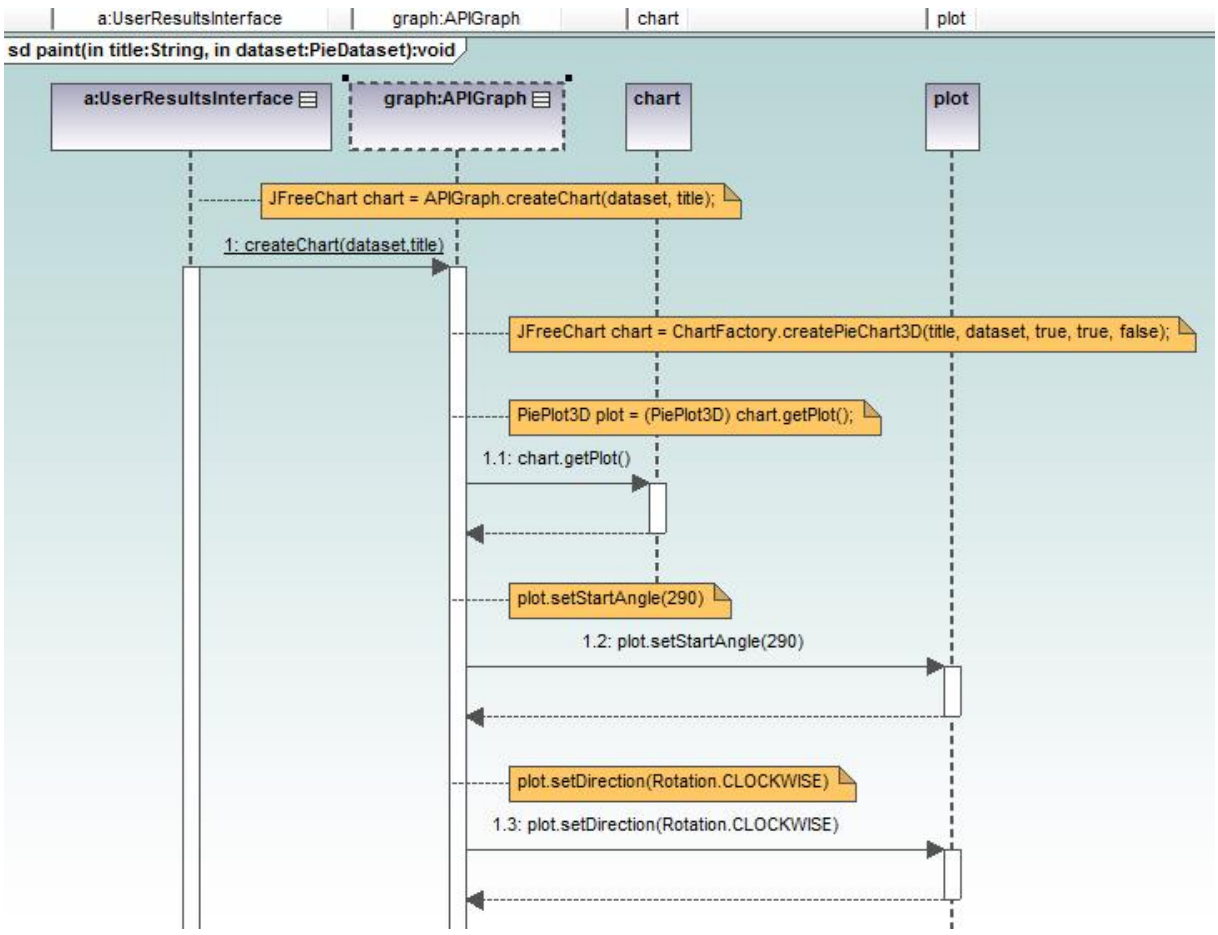


Figura 23 CharFactory

En la figura 24 de las siguientes páginas se seguirá con la instanciación de objetos con la creación de los formatos decimales utilizados para mostrar los gráficos.

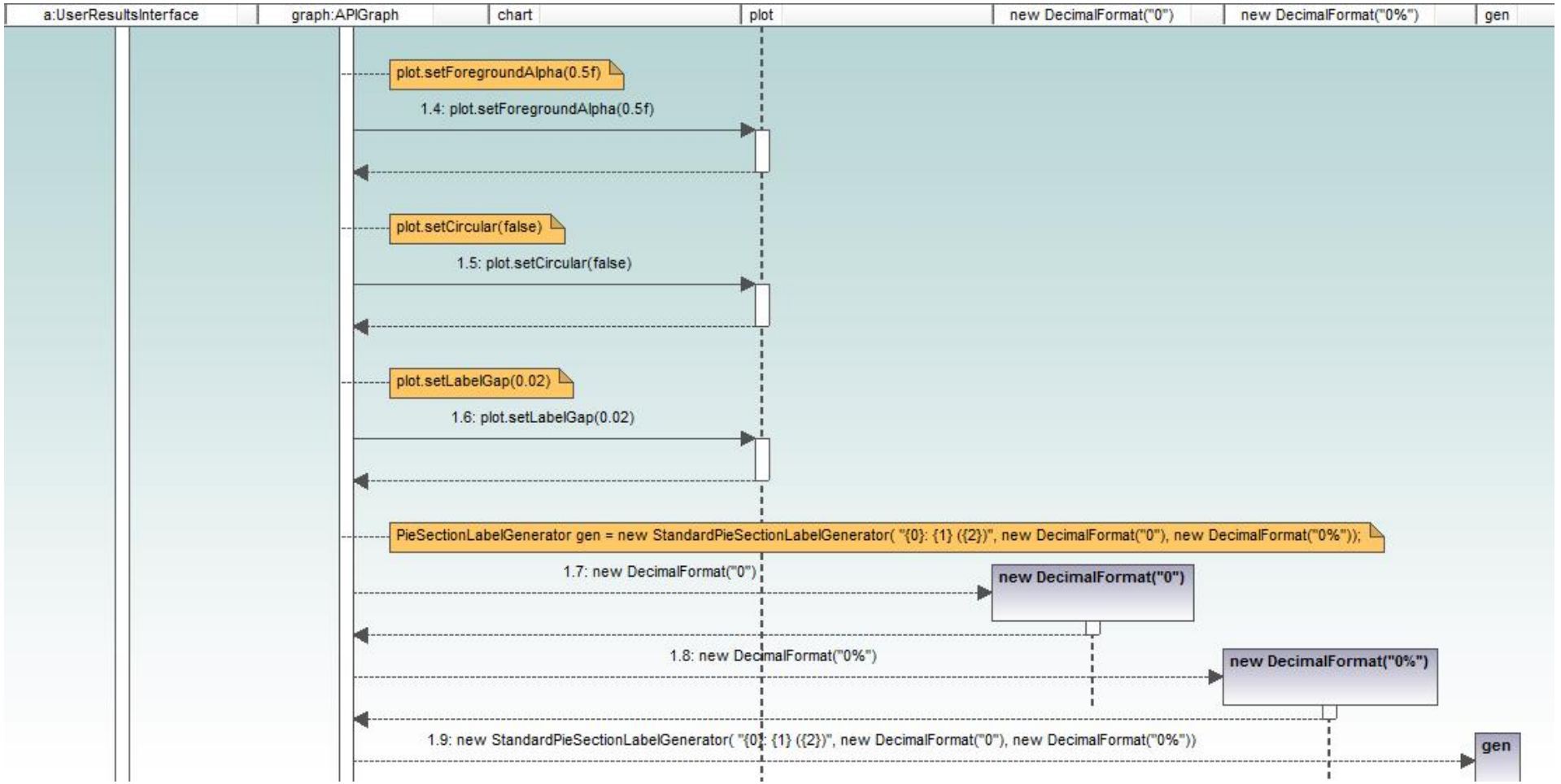


Figura 24 Definición de parámetros

Cabe destacar que se ha requerido de la creación de dos tipos de formatos para la representación numérica en las gráficas.

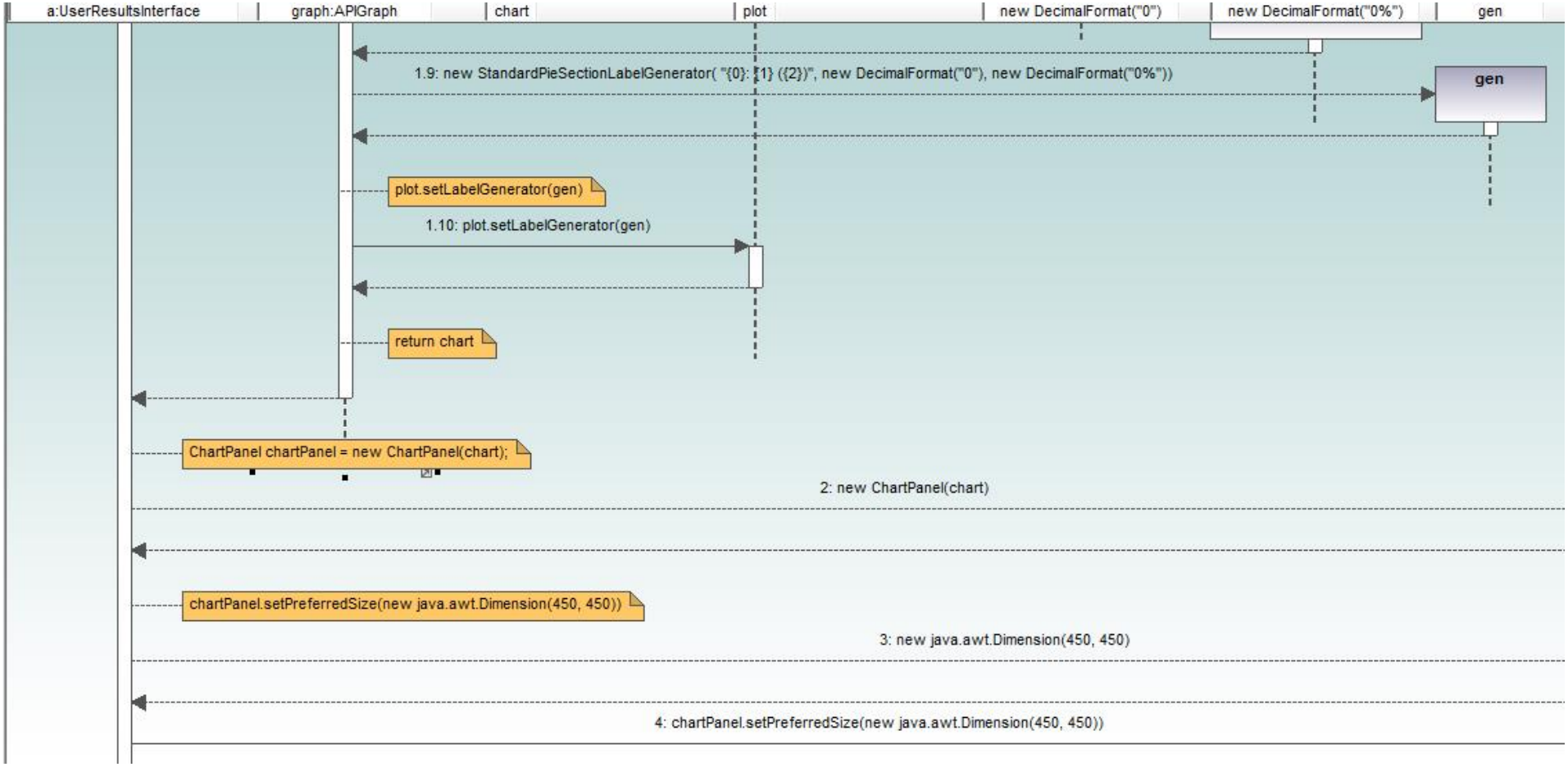


Figura 24 Definición de parámetros

En esta figura 24 se coloca el gráfico creado en un panel en el que se seleccionan las dimensiones deseadas por defecto.

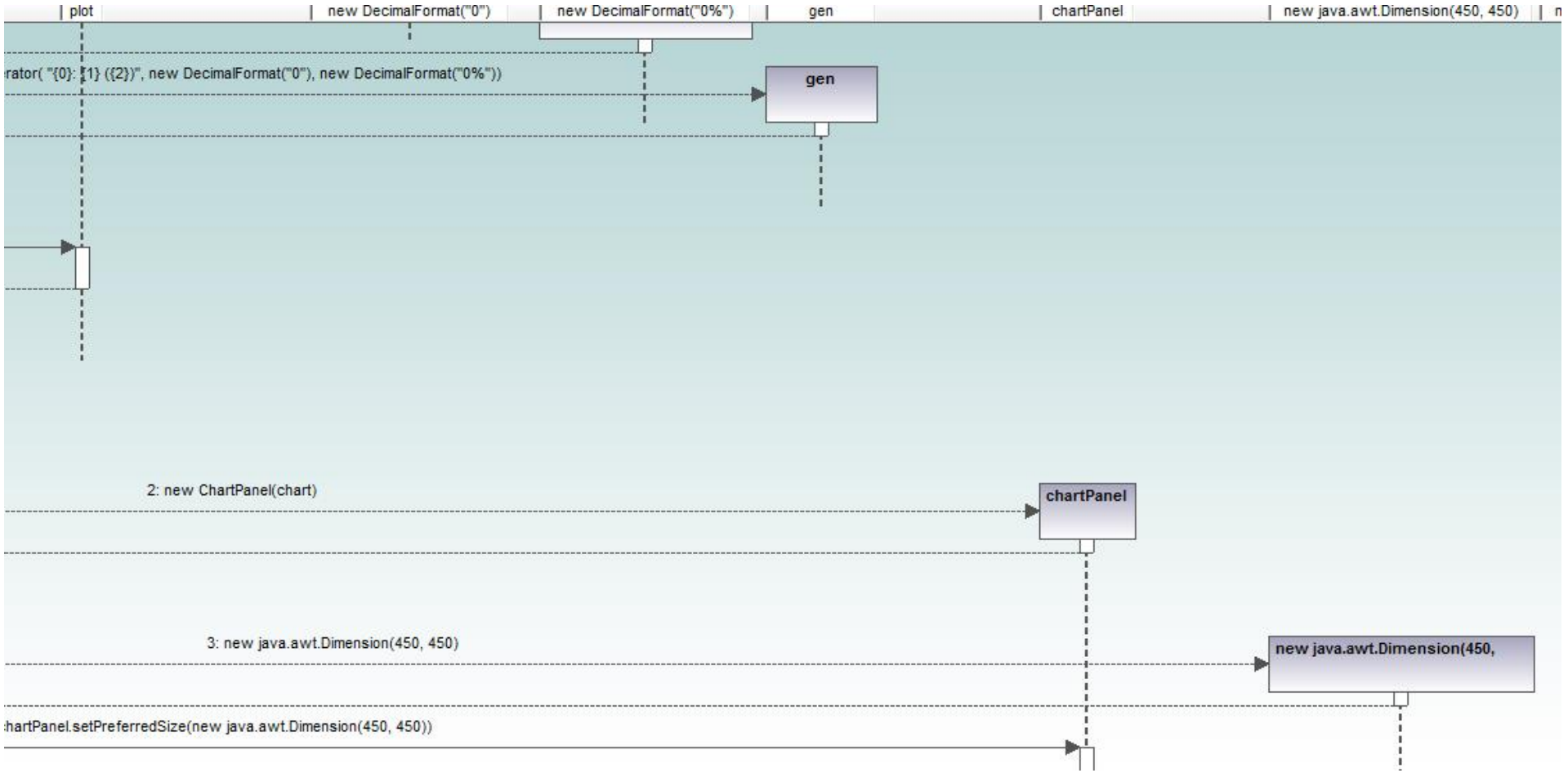


Figura 25 Zoom de las dimensiones

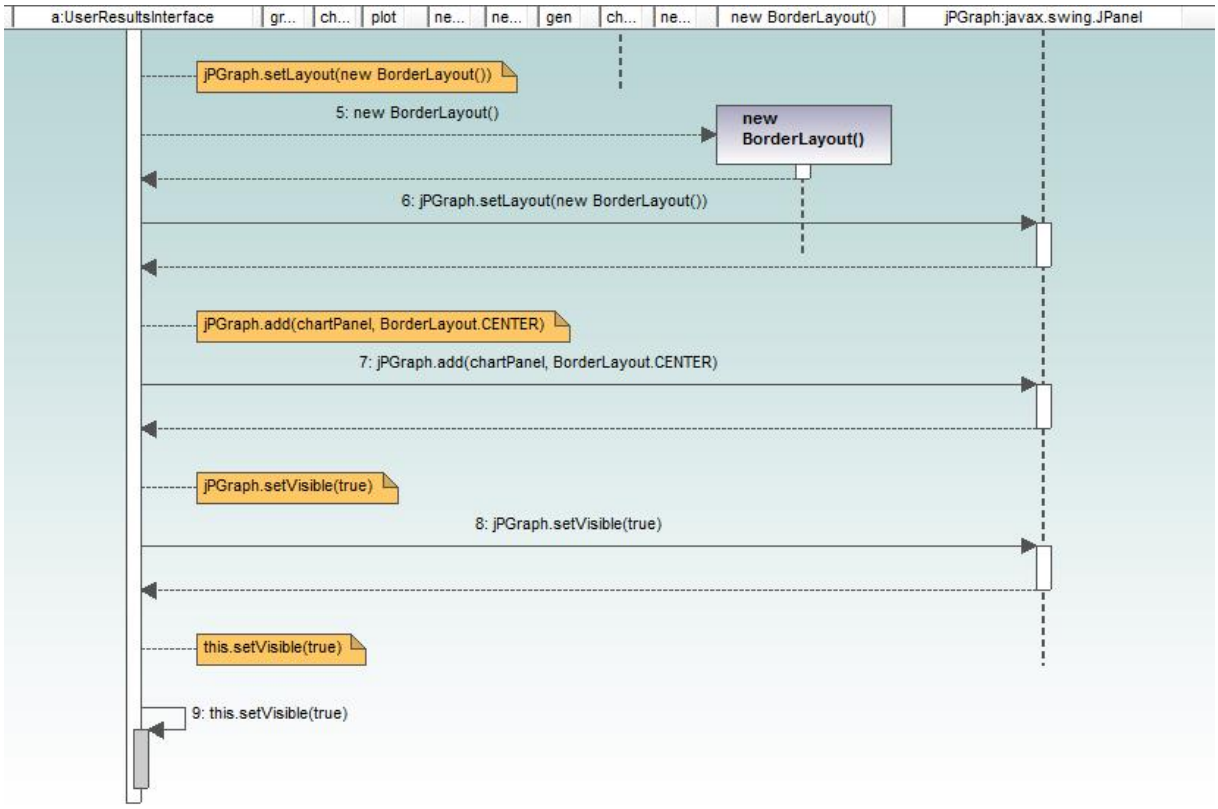


Figura 26 Visibilidad en el panel

Finalmente en esta figura 26, se muestra cómo se añaden las últimas características como el centrado en la pantalla y se hace visible la misma.

Para el caso de uso CU-03, se ha obviado el diagrama al tratarse simplemente de una salida por pantalla de los resultados generados por un script en el servidor.

3.3 Diseño de interfaces de usuario

Para no repetir información de forma innecesaria, el diseño de las interfaces de usuario se puede ver en el Manual de Usuario que se corresponde al anexo 2 de este documento.

Capítulo 4

Implementación del Software

En este capítulo se detallan las decisiones de implementación tomadas a lo largo de la etapa de desarrollo del mismo.

4.1 Decisiones de implementación

4.1.1 Carga de los emails

Existía la posibilidad de cargar los emails en su totalidad al comienzo de la sesión, o por el contrario, manejar sólo los buzones de la cuenta de email y hacer una carga posterior antes de realizar las estadísticas de los mismos.

Se optó por hacer una carga al comienzo del todo de la sesión del usuario por las siguientes razones:

- Tras un periodo de carga de unos minutos, el resto de acciones con los datos son casi instantáneos, incluso la generación de las estadísticas de listas de muchos miles de emails.

- Es más seguro: es mucho más seguro acceder sólo una vez al servidor de correo externo, en lugar de hacerlo varias veces, con la necesidad añadida de guardar durante la sesión la información de autenticación del usuario.

- Es más fiable: al sólo acceder una vez, es mucho más probable que no haya errores añadidos derivados por ejemplo de una caída del servidor al que se está accediendo.

- Interacción con el usuario: desde el punto de vista de la interacción con el usuario al sistema, es mucho más cómodo hacerle esperar sólo una vez unos minutos, a estar constantemente interrumpiéndole unos instantes para hacer cargas más pequeñas.

4.1.2 Botón de retroceso

Proporcionar al usuario un botón de retroceso en las interfaces gráficas es algo muy estandarizado hoy en día en todas las aplicaciones pero sin embargo, en el

caso de E-RETO no tenía mucho sentido, porque es un proceso muy lineal y la posibilidad de volver hacia atrás podría crear confusión al usuario y que finalmente no completase nunca el proceso por completo.

4.2 Resultados de las pruebas de aceptación

En esta sección se analizan si las pruebas de aceptación definidas en el apartado 3.7 son realizadas con éxito por E-RETO, como se aprecia en la Tabla 7.

Identificador	Pasos a ejecutar	Salida esperada
PA-01	1.Creación de una cuenta válida de Gmail: 1.1 Usuario -> *edumupePFC* 1.2 Contraseña -> ******* 2. Abrir E-RETO 3. Indicar al sistema que comience 4. Seleccionar como usuario: *edumupePFC* 5. Seleccionar como contraseña: ******* 6. Seleccionar como servidor de correo: gmail 7. Aceptar contribución al estudio 8. Aceptar bases legales 9. Leo qué hace E-RETO con mis datos 10. Leo qué NO hace E-RETO con mis datos 11. Indico al sistema que se conecte	OK. El sistema muestra la interfaz para introducir los datos personales.

Identificador	Pasos a ejecutar	Salida esperada
PA-02	1.Creación de una cuenta válida de Gmail: 1.1 Usuario -> *edumupePFC* 1.2 Contraseña -> ****** 2. Abrir E-RETO 3. Indicar al sistema que comience 4. Seleccionar como usuario: *edumupePFC* 5. Seleccionar como contraseña: ^** 6. Seleccionar como servidor de correo: gmail 7. Aceptar contribución al estudio 8. Aceptar bases legales 9. Leer qué hace E-RETO con mis datos 10. Leer qué NO hace E-RETO con mis datos 11. Indicar al sistema que se conecte	OK. El sistema muestra un mensaje de error en la autenticación.
PA-03	1. Conectarse al servidor de gmail con usuario *edumupePFC* y contraseña *****. 2. Crear buzón de correo: *buzonPFC1* 3. Crear buzón de correo: *buzonPFC2* 4. Desconectarse del servidor de gmail 5. Ejecutar prueba de aceptación PA-01 6. Rellenar nacionalidad con: *España* 7. Rellenar edad con *18* 8. Rellenar sector con *Agricultura* 9. Rellenar sexo con *Hombre* 10. Indicar al sistema que cargue los emails	OK. El sistema muestra que ha cargado correctamente *buzonPFC1, buzonPFC2 e Inbox*.

Identificador	Pasos a ejecutar	Salida esperada
PA-04	1. Realizar prueba de aceptación PA-03 2. Indicar al sistema que se desea continuar 3. Seleccionar *buzonPFC1* y añadirlo como laboral 4. Cambiar al idioma inglés. 5. Indicar al sistema que debe continuar con la ejecución	OK. El sistema cambia el idioma a inglés y en el servidor el fichero XML tiene todos los valores vacíos excepto: Country: ES Sector: 63 Gender: MAN Age: 18
PA-05	1. El usuario abre cuenta de email creada en PA-01. 2. Envía 10 emails a 10 personas distintas*. 3. Realizar prueba de aceptación PA-04 4. Seleccionar top 10 de los últimos 6 meses 5. Pulsar para generar gráfico.	OK. El sistema ha mostrado un gráfico con 10 emails con un uso del 10% a cada uno.
PA-06	1. Se ejecuta PA-05. 2. Se llama al *script* del servidor que muestra estadísticas actuales almacenadas.	OK. El sistema ha rellenado los datos estadísticos de forma correcta al compararlos manualmente haciendo uso de la calculadora y de la información otorgada por Gmail.

Tabla 7 Resultados pruebas de aceptación

* En las pruebas realizadas se utilizaron 10 emails reales que en este presente documento no se van a anotar, al no contar con la aprobación de los correspondientes dueños para hacerlo.

Capítulo 5

Conclusiones y líneas futuras

En este capítulo se describen las conclusiones a las que se ha llegado tras el desarrollo de este proyecto.

5.1 Conclusiones sobre el proyecto

El objetivo de este proyecto era el de implementar una herramienta que fuese capaz de realizar un análisis estadístico sobre una cuenta de email, generara una serie de resultados y los enviase a un servidor externo propiedad de la Universidad Carlos III de Madrid.

Se puede decir con toda tranquilidad que este objetivo ha sido cumplido sin problemas y se han incluso añadido funcionalidades extras que no estaban en principio pensadas en la propuesta inicial.

5.1.1 Dificultades del proyecto

Dentro del ámbito técnico, la principal dificultad viene dada por la necesidad de tener que manejar los emails a un nivel de cabecera, protocolos de comunicación, etc.

No sólo por su dificultad técnica a la hora de comprender cómo funcionan, sino porque dependes de empresas externas como Gmail, Yahoo o Microsoft, que tienen sus propias implementaciones del manejo de email y a las que tienes que adaptarte.

La otra gran dificultad del proyecto es el manejo de un gran volumen de datos bajo la necesidad de un rendimiento alto en el tiempo de ejecución. Este proyecto hace uso de un volumen de datos bastante alto, con una carga masiva de los mismos procedentes de servidores externos. Una vez hecha esa carga, se debe filtrar la información que se desea utilizar, procesarla y hacer cálculos con la misma, para su posterior envío a un servidor externo. Esto requiere hacer una programación muy meticulosa, en el que todos los métodos estén pensados para manejar estructuras con miles de objetos dentro.

El otro gran hito a conseguir era la creación de un sistema eficiente de filtrado de emails para poder discernir cuáles pertenecían al mismo hilo. Tras muchas horas navegando por Internet y leyendo cómo lo hacían otras grandes empresas de correo, llegué a la conclusión de que la forma más eficaz sería la mezcla de varias estrategias de las utilizadas por otros servidores de correo.

5.1.2 Conclusiones personales

La realización de este proyecto ha sido una travesía dura con numerosos obstáculos de por medio. La mayor dificultad ha sido el hecho de que trabaje en un país extranjero a jornada completa para una empresa de consultoría. Mi jornada laboral no ha sido la más idónea para poder compatibilizarla con el desarrollo de un proyecto de estas magnitudes y sólo gracias a la paciencia de mi tutor y de mis ganas por terminar la carrera ha sido posible llegar a la línea de meta.

Al margen de las dificultades personales, este proyecto ha sido un reto muy grande desde un punto de vista técnico porque me ha obligado a recorrer numerosas parcelas de conocimiento un poco olvidadas de mi etapa como estudiante.

Como anécdota para finalizar, y que sirve de ejemplo de las dificultades encontradas, cuando implementé mi primera carga de los emails de mi cuenta de correo de Gmail con más de cuarenta mil emails, al cabo de más de treinta minutos esperando, tuve que ponerme a averiguar cómo podía disminuir el tiempo de carga a unos valores razonables porque la implementación estándar no era suficiente. Actualmente E-RETO tarda menos de 5 minutos en cargar todos mis emails de esa misma cuenta.

5.2 Líneas futuras

En este apartado se va a explicar las posibles ampliaciones que se podrían hacer a E-RETO.

En primer lugar se podría dar la opción al usuario de poder añadir su propio servidor de correo y no tener sólo una serie de opciones predefinidas en una lista.

La segunda posible mejora sería la introducción de nuevas mediciones estadísticas que no estén cubiertas actualmente. Con el incremento del uso de E-RETO, es posible que se detecte una falta de información en la medición de ciertos valores estadísticos que actualmente no están cubiertos.

Otra posible mejora, sería la incorporación de una mayor variedad de gráficos en los actualmente disponibles al final de la ejecución del sistema para el usuario.

Quizás como principal línea futura, se debería permitir que E-RETO pueda ser además de una aplicación de escritorio, una aplicación en plataformas móviles como Android [47] o iOS [48].

Referencias

[1] Estudio del Instituto Nacional de Estadística
Título: Encuesta sobre equipamiento y uso de tecnologías de la información y comunicación en los hogares
http://www.ine.es/inebmenu/mnu_tic.htm
[2] UML
Lenguaje Unificado de Modelado
http://www.uml.org/
[3] Diagramas de secuencia
http://www.uml.org/
[4] Informática forense:
Autora: Elena Pérez Gómez
http://www.microsoft.com/business/es/Content/Paginas/article.aspx?cbcid=121
[5] Aid4Mail: empresa dedicada a la informática forense
http://www.aid4mail.com/
[6] Paraben: Recurso online:
http://www.paraben.com/
[7] Forensic Control: empresa dedicada a la informática forense
http://forensiccontrol.com/
[8] CodePlex: empresa dedicada a la informática forense
https://www.codeplex.com/
[9] Email Behavior Analyzer: software libre
Autor: Uri Kartoun
http://behavioranalyzer.codeplex.com/
[10]Universidad de Strathclyde
https://www.strath.ac.uk/
[11] Email Analyzer Software: master tesis
Autor: Adnan Mazher Syed
https://personal.cis.strath.ac.uk/mark.dunlop/misc/cit/projects/library/05/Adnan.pdf

[12] Gmail
https://gmail.com/
[13] Yahoo
https://es.yahoo.com/
[14] Hotmail
https://www.hotmail.com/
[15] Modelo Vista Controlador
https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf
[16] Internet Message Access Protocol
http://tools.ietf.org/html/rfc3501
[17] Protocolo de Oficina Postal
http://tools.ietf.org/html/rfc1725
[18] HTML
https://tools.ietf.org/html/rfc1866
[19] Hojas de estilo
http://tools.ietf.org/html/rfc2318
[20] Java
http://www.java.com/es/
[21] Grails
http://grails.org/
[22] Mvel
http://mvel.codehaus.org/
[23] Smooks
http://www.smooks.org/
[24] XML
https://tools.ietf.org/html/rfc4825
[25] CSV
http://tools.ietf.org/html/rfc4180
[26] SMTP
http://www.ietf.org/rfc/rfc0821.txt
[27] JavaMail
http://www.oracle.com/technetwork/java/javamail/index.html
[28] Oracle
http://www.oracle.com/

[29] R
http://www.r-project.org/
[30] C
http://msdn.microsoft.com/en-us/library/fw5abdx6.aspx
[31] Fortran
http://www.fortran.com/
[32] JDistlib
http://jdistlib.sourceforge.net/
[33] http
https://tools.ietf.org/html/rfc2616
[34] HTTPBuilder
http://groovy.codehaus.org/HTTP+Builder
[35] Apache HTTP
http://httpd.apache.org/
[36] PHP
https://wiki.php.net/rfc
[37] Librería Swing
http://docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html
[38] Java JAXP
http://docs.oracle.com/javase/tutorial/jaxp/
[39] ESA
www.esa.int/
[40] ISO 3166
http://www.iso.org/iso/country_codes.htm
[41] LinkdIn
https://www.linkedin.com
[42] FTP
http://www.ietf.org/rfc/rfc959.txt
[43] HashMap
http://docs.oracle.com/javase/7/docs/api/java/util/HashMap.html
[44] API Joda Time#
http://www.joda.org/joda-time/
[45] Comparator
http://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html

[46] JFreeChart
http://www.jfree.org/jfreechart/
[47] Android
http://www.android.com/
[48] iOs
http://www.apple.com/es/ios/
[49] Diagrama de Gantt
http://www.eoi.es/blogs/embatur/2013/08/04/la-gestion-del-tiempo-diagrama-de-gantt/
[50] NetBenas
https://netbeans.org/
[51] Skype
http://www.skype.com/es/
[52] Google Chrome
http://www.google.com/chrome/
[53] Adobe Reader
http://get.adobe.com/es/reader/
[54] Paquete office
http://office.microsoft.com/
[55] Dropbox
http://dropbox.com/
[56] ArgoUml
http://argouml.tigris.org/
[57] Altova
http://www.altova.com/es/

Acrónimos

INE Instituto Nacional de Estadística

UML Lenguaje Unificado de Modelado

IMAP Internet Message Access Protocol

POP Protocolo de Oficina Postal

HTML Lenguaje de marcas de hipertexto

XML Lenguaje de marcas extensible

SMTP Protocolo para la transferencia simple de correo electrónico

API Interfaz de programación de aplicaciones

HTTP Protocolo de transferencia de hipertexto

ESA Agencia Espacial Europea

ISO Organización Internacional para la estandarización

Anexo 1 Gestión del proyecto

1. Planificación del trabajo

En esta sección se detalla la planificación inicial del proyecto y su desviación en caso de existirla al finalizar el proyecto.

1.1 Planificación inicial

Este proyecto se ha dividido en varias tareas para poder hacer una estimación más adecuada y precisa del mismo. Como se aprecia en la Tabla 8, se calcula que la duración total será de 108 días laborales, en los que un programador trabajará un total de 4 horas diarias. Como se aprecia en al siguiente tabla, la etapa que se estima que más tiempo consumirá es la de Implementación, que supone alrededor de un 37% del total.

Nombre de tarea	Duración	Comienzo	Fin
Proyecto fin de carrera	**108 días?**	**mar 03/09/13**	**jue 30/01/14**
Planificación Inicial	2 días?	mar 03/09/13	mié 04/09/13
Estudio del estado del arte	5 días?	jue 05/09/13	mié 11/09/13
Análisis	**12 días?**	**jue 12/09/13**	**vie 27/09/13**
Arquitectura del sistema	3 días?	jue 12/09/13	lun 16/09/13
Estudio tecnológico	2 días?	mar 17/09/13	mié 18/09/13
Definición de casos de uso	2 días?	jue 19/09/13	vie 20/09/13
Definición de requisitos software	5 días	lun 23/09/13	vie 27/09/13
Diseño	**14 días**	**lun 30/09/13**	**jue 17/10/13**
Diseño de software	8 días	lun 30/09/13	mié 09/10/13
Diagramas de secuencia	6 días	jue 10/10/13	jue 17/10/13
Implementación	40 días?	vie 18/10/13	jue 12/12/13
Pruebas de integración	10 días?	vie 13/12/13	jue 26/12/13
Documentación	25 días?	vie 27/12/13	jue 30/01/14

Tabla 8 Previsiones presupuestarias

A continuación en la Figura 27 se puede ver un Diagrama de Gantt [49] de esta planificación.

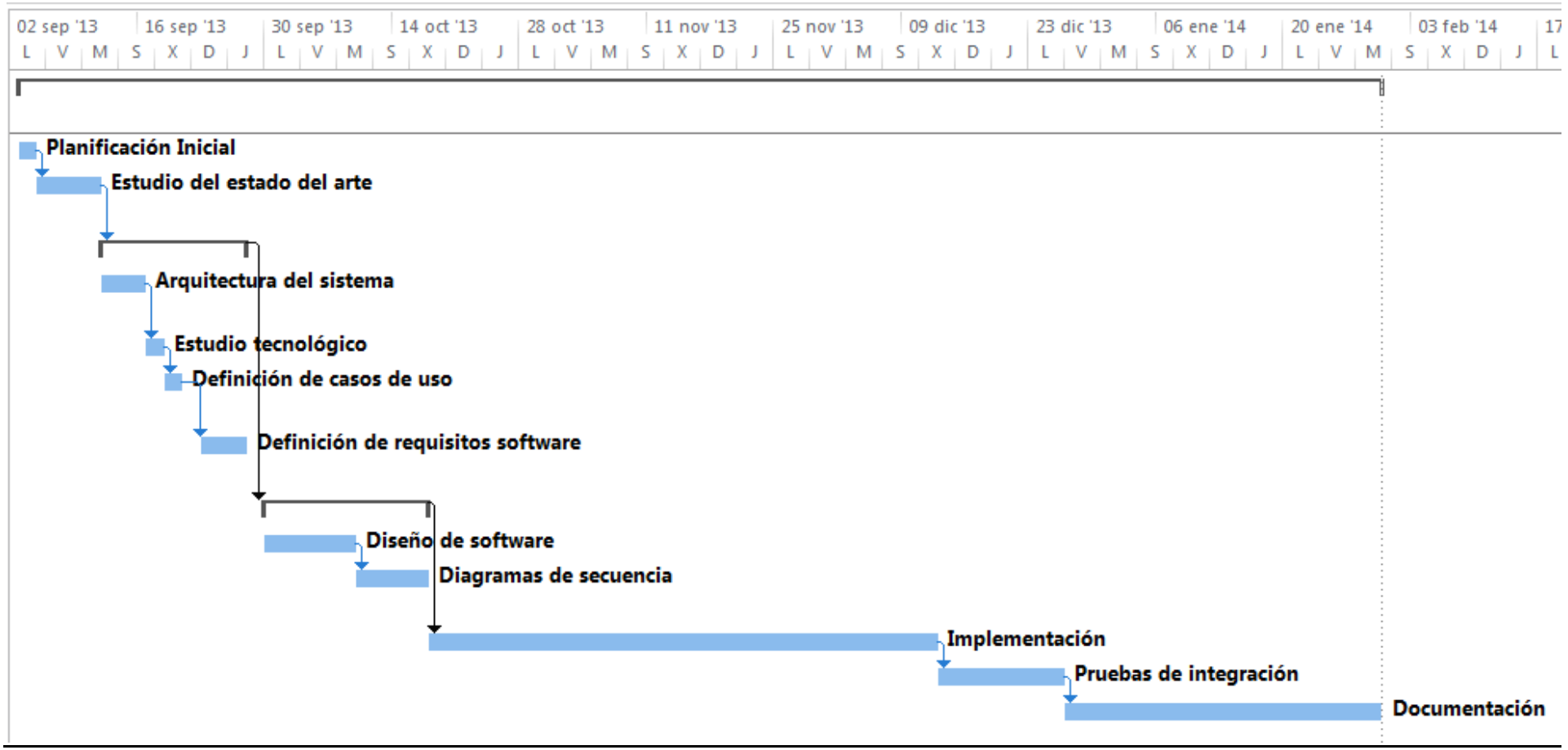


Figura 27 Diagrama de Gantt

1.2 Desarrollo real del proyecto

A continuación en la Tabla 9 se muestra el desarrollo real del proyecto con sus duraciones finales y la desviación en cada etapa.

Nombre de tarea	Duración	Comienzo	Fin	Desviación
Proyecto fin de carrera	**166 días**	**mar 03/09/13**	**mar 22/04/14**	**53,70%**
Planificación Inicial	3 días	mar 03/09/13	jue 05/09/13	50,00%
Estudio del estado del arte	5 días	vie 06/09/13	jue 12/09/13	0,00%
Análisis	**16 días**	**vie 13/09/13**	**vie 04/10/13**	33,33%
Arquitectura del sistema	5 días	vie 13/09/13	jue 19/09/13	66,67%
Estudio tecnológico	2 días	vie 20/09/13	lun 23/09/13	0,00%
Definición de casos de uso	2 días	mar 24/09/13	mié 25/09/13	0,00%
Definición de requisitos software	7 días	jue 26/09/13	vie 04/10/13	40,00%
Diseño	**16 días**	**lun 07/10/13**	**lun 28/10/13**	14,29%
Diseño de software	10 días	lun 07/10/13	vie 18/10/13	25,00%
Diagramas de secuencia	6 días	lun 21/10/13	lun 28/10/13	0,00%
Implementación	66 días	mar 29/10/13	mar 28/01/14	**65,00%**
Pruebas de integración	34 días	mié 29/01/14	lun 17/03/14	**240,00%**
Documentación	26 días	mar 18/03/14	mar 22/04/14	4,00%

Tabla 9 Desviaciones presupuestarias

Cabe destacar que la desviación total del proyecto ha sido de un retraso de un 53%, que es un valor muy alto pero a la vez relativamente lógico. Como ejemplo, el hecho de que durante los meses de invierno, el único recurso humano con el que contaba el proyecto esté enfermo una semana, supone un retraso al total del proyecto de casi un 6,4% y sin embargo, el riesgo de que este hecho se produjese era muy alto.

Por otro lado, la persona asignada para este proyecto no trabajaba a tiempo completo en el mismo, sino sólo 4 horas diarias que en muchos casos eran únicamente 2 o ninguna, debido a sus compromisos laborales con otras entidades.

Al margen de ciertas situaciones excepcionales que siempre pueden afectar en la desviación de un proyecto, se ve una clara mala estimación en la etapa de pruebas, en la que se cometió un error de un 240%. Al realizar las estimaciones

no se tuvo en cuenta de forma acertada la dificultad de estas pruebas y la cantidad de casos especiales que se podían encontrar.

2 Medios técnicos empleados para el proyecto

Para el desarrollo de este proyecto ha sido necesaria la utilización de un ordenador portátil HP Paviilion dv5. Al margen de este hardware, sólo se han necesitado una serie de herramientas software que se listan en la tabla 10:

Software	Uso
Netbeans IDE 7.1 [50]	Programación software
Skype [51]	Comunicación
Google Chrome [52]	Navegación Web
Adobe Reader 10.1 [53]	Lectura de documentación
Paquete Office 2010 [54]	Edición de documentación
Dropbox 2.4.11 [55]	Comunicación
ArgoUML 0.34 [56]	Edición de documentación
Altova 2014 [57]	Edición de documentación

Tabla 10 Software utilizado

3. Análisis económico del proyecto

En esta sección se hacen las estimaciones de los costes del proyecto, presupuesto presentado al cliente y desviaciones finales tras la finalización del proyecto.

3.1 Metodología de estimación de costes

Para la estimación de costes del proyecto se han tenido en cuenta tanto costes directos del proyecto como indirectos.

Dentro de los gastos directos se han incluido todos aquellos que están relacionados de una forma concreta y directa con el desarrollo del proyecto en

sí. En este apartado se incluye la mano de obra, equipos utilizados, licencias software, dietas y gastos de viaje.

Los gastos de mano de obra se estimarán contando que no se es autónomo y se está empleado para una empresa, los precios de los equipos y del software utilizado son los precios de venta al público, el valor de las dietas se fija como pactado entre empleado y empresa y los gastos de viaje en principio en base a la distancia en km.

Los gastos indirectos incluyen el alquiler del inmueble en el que se ha desarrollado el proyecto y su conexión a internet.

3.2 Presupuesto inicial

En esta sección se detalla el presupuesto inicial del proyecto donde se mostrarán todas las estimaciones. Para facilitar los cálculos, no se aplicará el IVA a ninguno de los gastos y se añadirá al total del presupuesto del proyecto.

3.2.1 Gastos de personal

El contratado del encargado de realizar este proyecto está remunerado en base al número de horas trabajadas al mes, de forma que si superan un total de 60 horas mensuales, la remuneración es de 18€ brutos la hora trabajada. De esta forma, al haberse pactado con el empleado una jornada laboral de 4 horas diarias se supera el umbral de las 60 horas y por tanto de debe aplicar esta tarifa.

En la estimación del proyecto se ha cuantificado un total de 108 días trabajados, lo que supone un coste total de 7776€ brutos. Si se aplica un 28,3% a pagar a la Seguridad Social, finalmente los gastos de personal suponen 9757€.

3.2.2 Gastos software y de equipos

El equipo necesario para el desarrollo de este proyecto se trata de un HP Pavillion dv5 cuyo coste de amortización es nulo, debido a que es utilizado en otro proyecto simultáneamente que se hace cargo de este coste.

En cuanto al software utilizado, todas las licencias han sido de programas de software libre, o en el caso del Paquete Office, Altova y Adobe, sus cotes están dentro de una partida de licencias que la empresa paga anualmente y no se pueden añadir adicionalmente como costes de este proyecto.

Se ha incluido una donación de 40€ al Desarrollo del Software Libre.

3.2.3 Dietas y desplazamientos

El acuerdo con la empresa es que por día trabajado el empleado reciba un total de 12€ al día brutos. Eso hace un total de 1296€ en 108 días estimados. En cuanto al coste por desplazamientos, el empleado no requiere de desplazamiento a su puesto de trabajo y realiza su actividad en casa.

3.2.4 Resumen Costes del proyecto

Concepto	Coste en €
Gastos directos	**11093**
Gastos de personal	9757
Gastos software y de equipos	40
Dietas y desplazamientos	1296
Gastos indirectos	**2219**
Gasto total antes del IVA (21%)	13311
Total	**16107 €**

Tabla 11 Presupuesto Final

3.3 Presupuesto para el cliente

Para el cálculo del presupuesto que se va a presentar al cliente, es necesario tener en cuenta dos factores: el riesgo del proyecto y el beneficio que se quiere obtener del mismo. Para ello se van a hacer los cálculos antes de aplicarle el IVA, en base a los cálculos ya presentados en el apartado anterior.

Además del riesgo asociado a todo proyecto por posibles errores en las estimaciones del mismo, este proyecto cuenta con un riesgo añadido muy alto al contar con una sola persona para su desarrollo y que además no tiene una dedicación completa diaria al mismo. Es por ello que se ha decidido aplicarle un riesgo del 24% para cubrir cualquier percance en el desarrollo del mismo. El cliente debe ser consciente de que la única forma de disminuir este riesgo sería la inclusión de otra persona más a la plantilla y eso agrandaría más el presupuesto que la aplicación de este alto porcentaje de cobertura.

Por otro lado, la empresa tiene la política de aplicar un porcentaje menor a los objetivos por beneficios en proyectos con un riesgo superior al 15% y por ello se aplicará sólo un aumento del 25% en concepto de beneficios que se quieren alcanzar. En la Tabla 12 se pueden ver todos estos datos:

Concepto	**Coste en €**
Gastos directos	**11093**
Gastos de personal	9757
Gastos software y de equipos	40
Dietas y desplazamientos	1296
Gastos indirectos	**2219**
Gasto total sin riesgo (15%)	13311
Gasto total sin beneficios (25%)	**15308**
Total sin IVA (21%)	19135 €
Total con IVA	**23154 €**

Tabla 12 Presupuesto para el Cliente

3.4 Coste final y análisis de la desviación

Tras el desarrollo del proyecto ha habido una desviación de 58 días. Esto ha supuesto una serie de costes adicionales que se pueden apreciar en la Tabla 13.

Concepto	Coste en € estimados	Coste en € reales	Desviación
Gastos directos	11093	**15965**	43,92%
Gastos de personal	9757	13933	42,80%
Gastos software y de equipos	40	40	0,00%
Dietas y desplazamientos	1296	1992	53,70%
Gastos indirectos	2219	**3193**	43,89%
Gasto total antes del IVA	13311	**19158**	43,93%
Total	16107 €	**23182 €**	**43,93%**

Tabla 13 Desviación final del proyecto

A la vista de estos resultados, se ve claramente que el gasto se ha liquidado todo el presupuesto que se había asignado para el riesgo asociado al proyecto y lo que es más importante, también el de los beneficios del mismo.

A pesar de no ser una gran noticia el hecho de no haber conseguido beneficios directos tras realizar este proyecto, es importante destacar que con una desviación en el tiempo de un 53%, no acumular pérdidas es una buena noticia, debido a que además se le ha mostrado al cliente que incluso en una situación de extrema gravedad, con un retraso tan grande, se ha sido capaz de mantener el presupuesto inicial y se ha logrado ganar su confianza para futuros proyectos en los que se intentará obtener beneficios tangibles.

Anexo 2

Manual de usuario

1 Introducción

E-RETO es una aplicación de escritorio de libre distribución y código abierto que permite el análisis del uso de una cuenta de correo existente. Permite al usuario acceder a algunas estadísticas de su cuenta de email y además envía una serie de estadísticas anónimas a un servidor externo.

Este manual pretende ayudar al usuario con la interacción con el sistema.

2 Requisitos para su instalación y ejecución

E-RETO sólo necesita un ordenador con un sistema operativo que permita instalar una máquina virtual Java 1.6 o superior. Para su ejecución sólo es necesario pulsar el icono denominado PFC.

3 Funcionamiento de la aplicación

Al hacer clic en el icono de inicio, se iniciará la pantalla de inicio que se muestra en la Figura 28.

Figura 28 e-RETO Inicio

Como se aprecia en la anterior figura, se puede elegir el idioma deseado, pulsando en la lista superior derecha de la interfaz. En caso de querer continuar con el proceso, se puede hacer clic en el botón de comenzar. Si por el contrario se quiere conocer más sobre el estudio de investigación para el que se va a utilizar E-RETO, se puede hacer clic en el enlace subrayado en el texto en azul.

Al pulsar el botón para continuar, se accede a la interfaz mostrada en la Figura 29.

Figura 29 e-RETO Login

Como se aprecia en la figura anterior, se puede rellenar el campo usuario y contraseña deseados y seleccionar el servidor de correo al que pertenecen estos credenciales.

Por defecto aparece seleccionada la opción de contribuir al estudio de investigación, y si se desea más información sobre el mismo, se puede pulsar en el icono de *info* que aparece a la derecha del campo en azul.

Para poder iniciar sesión es obligatorio leer la información legal desplegada en la interfaz y aceptar que se ha leído y comprendido la misma.

Na vez que se han cumplido todos estos requisitos, si los credenciales otorgados son los correctos, E-RETO mostrará la interfaz de la Figura 30.

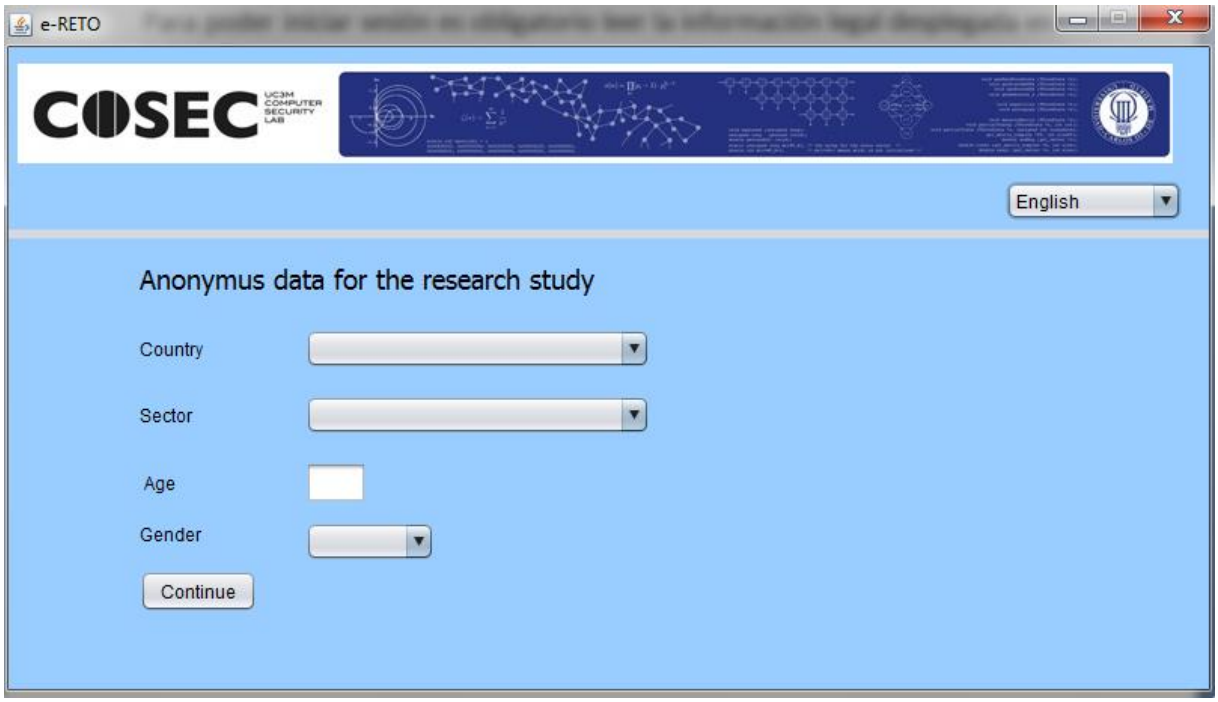


Figura 30 e-RETO Datos Usuario

En esta sección se deben rellenar el campo de edad con un formato numérico, y seleccionar un valor en todas las listas que se despliegan en la figura.

Si se han cumplido todos estos requisitos, al pulsar el botón de continuar aparecerá la interfaz mostrada en la Figura 31.

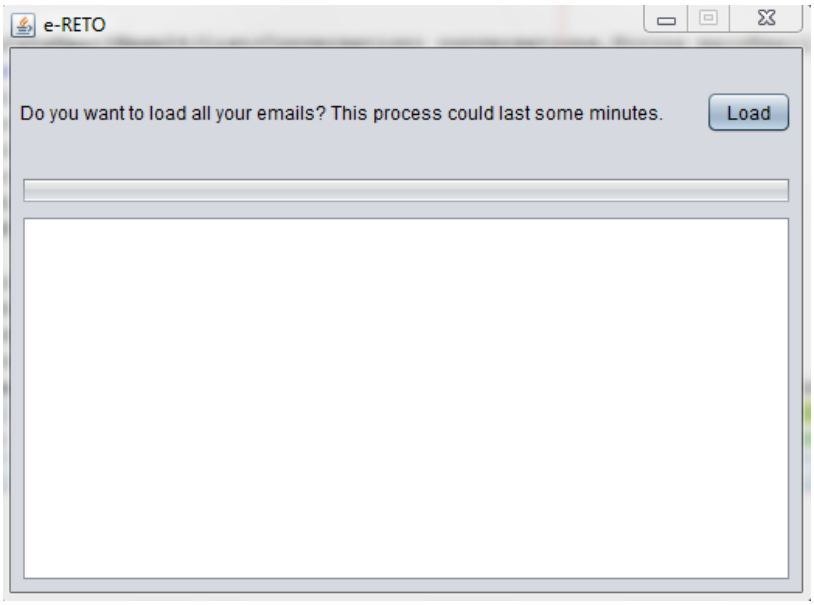


Figura 31 e-RETO Datos de progreso

Si se decide continuar con la carga, E-RETO irá informando al usuario de los buzones que va cargando hasta que termine el proceso, como muestra la Figura 32.

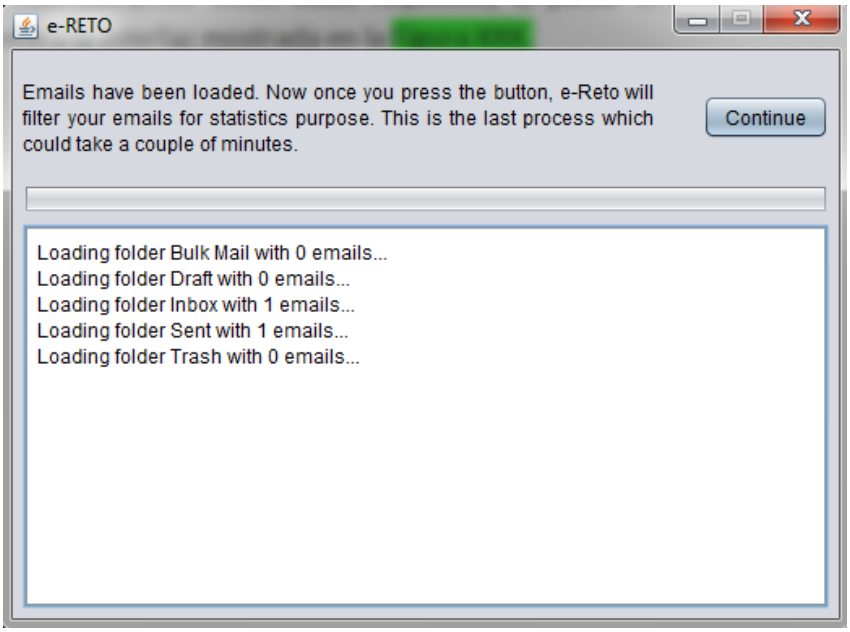


Figura 32 e-RETO Carga de emails

Si se decide pulsar el botón de continuar, E-RETO mostrará la información recogida en la Figura 33.

Figura 33 e-RETO Selección de buzones

En esta interfaz es posible añadir buzones con fines laborales seleccionándolos en la lista de la izquierda z pulsando el botón *Añadir.* En caso de cometer algún error es posible remover los buzones deseados seleccionándolos del listado derecho y haciendo clic en el botón Eliminar.

Si se quiere mover de izquierda a derecha la totalidad de los buzones o viceversa, se puede hacer uso de los botones: *añadir todos* y *eliminar todos.*

Si se desea añadir un dominio de trabajo específico, se puede hacer uso de la caja de texto situada abajo a la izquierda, introduciendo en ella el texto del dominio y pulsando el botón añadir. Si se quiere eliminar un dominio, sólo hay

que seleccionarlo de la lista de la derecha y presionar el botón de *eliminar,* como ocurría en el caso de los buzones.

Una vez que el usuario está conforme con la configuración que ha realizado, puede presionar el botón de continuar y E-RETO mostrará la interfaz de la Figura 34.

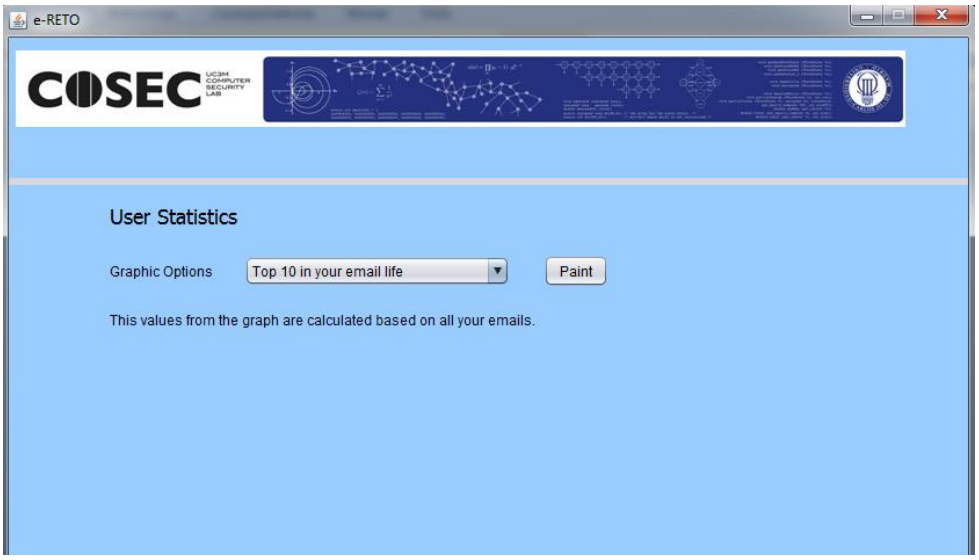


Figura 34 e-RETO Selección de gráfico

Llegados a este punto, E-RETO ya ha enviado la información estadística a su servidor y el usuario puede consultar si lo desea algunas curiosidades sobre su cuenta de email. Para ello puede seleccionar en el listado de opciones de gráfico lo que quiere visualizar en forma de gráfica y pulsar pintar. La interfaz se actualizará y mostrará algo como el ejemplo de la Figura 35.

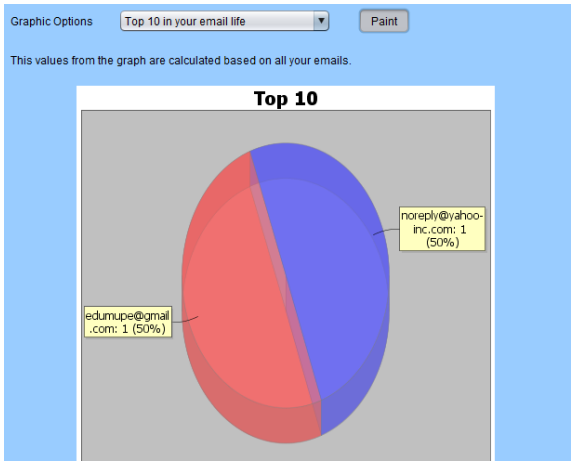


Figura 35 e-RETO Gráficos para el usuario

Anexo 3

Plantillas

1 Definición de casos de uso

Identificador: CU-XX	
Nombre	
Descripción	
Actores	
Precondiciones	
Pos-condiciones	

2 Definición de requisitos

Identificador	**Nombre**	**Descripción**	**Estabilidad**	**Prioridad**
RF-XX.OO Ó RNF-XX			Alta/Media/Baja	Alta/Media/Baja

3 Definición de pruebas de aceptación

Identificador	**Requisitos cubiertos**	**Pasos a ejecutar**	**Salida**
PA-XX			

4. Resultados pruebas de aceptación

Identificador	Pasos a ejecutar	Salida esperada
PA-XX		OK/NOK