Desarrollo de un cliente web de emails seguros; Secretify.

Memoria de Proyecto Final de Máster

Máster en Aplicaciones Multimedia

Autor: Esaú Suárez Ramos

Consultor: Sergio Schvarstein Liuboschetz

Fecha de entrega: 18/06/2014

Créditos/Copyright

Cita

“The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards” – Gene Spafford.

Agradecimientos

A Urko Martínez Círez, por tener una de sus ideas revolucionarias de la cual se ha podido derivar este TFM.

A Sergio Schvarstein Liuboschetz, por su paciencia y buenos consejos de cara al desarrollo del proyecto.

Abstract

Email security has become a trending topic due to the way in which email providers deal with information. Classic email users are starting to get tired of the opacity and ways companies deal with their personal and sensitive information.

The platform developed during this Project tries to give an innovative solution to this problem. The solution is not about creating a substitutive product but instead about adding a security layer on top of existing services.

In order to achieve its security goals, Secretify makes extensive use of a public key infrastructure, and client-to-client encryption using symmetric cryptography standards.

Secretify's final goal is to become a platform in which PGP security principles can be applied to the emails the user sends. This platform will need to guarantee the best user experience through good design and well-known usability patterns in order to become a serious product for security-concerned users.

Keywords

Secure communication, email web client, PKI, asymmetric cryptography, PGP, symmetric cryptography, hash functions.

Resumen

La seguridad en el email es un tema de actualidad cuyo interés está justificado debido a la opacidad con la que se tratan los datos personales del usuario por parte de los servicios más populares de correo electrónico.

La plataforma desarrollada en este proyecto, aborda el problema desde un punto de vista innovador, en el que la solución no pasa por tratar de crear un sustitutivo a los productos actuales, sino en complementarlos con una capa extra de seguridad.

Para conseguir sus objetivos de seguridad, Secretify hace uso extensivo de una infraestructura de clave pública y de cifrado cliente a cliente utilizando estándares de criptografía simétrica.

El propósito final de Secretify es reunir, en una plataforma única, una infraestructura que permita aplicar los estándares de seguridad PGP a los emails que envía el usuario. Esta plataforma deberá garantizar la mejor experiencia de usuario a través de un buen diseño y la aplicación de patrones de usabilidad conocidos, para poder así postularse como un producto serio para usuarios preocupados por la seguridad.

Palabras clave

Comunicación segura, cliente de correo web, PKI, criptografía asimétrica, PGP, criptografía simétrica, funciones hash.

Índice

Figuras y tablas

Índice de figuras

Índice de tablas

Capítulo 1: Introducción

1. Introducción

En la actualidad, existen varios temas polémicos en cuanto a comunicación y seguridad.

Comenzando por el ejemplo más sonado, que es el del espionaje de la NSA [7] parece claro que los usuarios cada vez son más conscientes de las distintas vulnerabilidades en cuanto a privacidad a las que se enfrentan día a día. Si una agencia encargada de garantizar la seguridad de los ciudadanos está envuelta en casos de espionaje y de utilización indebida de información personal, entonces ¿qué puede esperarse de aquellos individuos que tengan como finalidad única y exclusivamente dañar al usuario?

Otro caso destacado sobre el que se oyen nuevas noticias día a día [12] es el de la inseguridad en las conversaciones de la aplicación móvil de mensajería más popular; Whatsapp [21]. Los desarrolladores se han desentendido por completo de un tema tan importante como es el de la privacidad de los usuarios y además se valen de políticas de uso ambiguas para evitar dar a conocer el uso que hace de la información personal de los usuarios.

En relación más directa con este Trabajo de Fin de Máster se encuentra un asunto también controversial; la opacidad de empresas de gran relevancia en el mundo de las comunicaciones respecto al tratamiento de datos personales. Es el caso de Google (cita), que recientemente ha sido sancionado por la Agencia Española de Protección de Datos (AEPD) incumplir leyes referentes a la protección de datos debido a que *"El organismo público ha «constatado» que la compañía recoge y trata ilegítimamente información personal, que no proporciona al usuario información adecuada de qué tipo de datos y con qué fines se recogen"* [39].

En base a lo expuesto, pueden sintetizarse como tres las motivaciones principales para el desarrollo de este Trabajo de Fin de Máster:

1. Seguridad y privacidad son temas de actualidad que requieren una gestión especial y a los que se está dando cada vez más y más importancia como característica intrínseca de diversos productos.
2. Las compañías a las que la mayoría de gente recurre para almacenar y compartir información personal ofrecen pocas garantías de que esta valiosa información no se utilizará en beneficio propio.
3. Existe un conjunto de usuarios potenciales que podría beneficiarse de la utilización de *Secretify*, ya que gracias a sus características exclusivas, es un producto bastante diferenciado de la competencia.

2. Descripción

Como ya se ha explicado, este proyecto surge como respuesta a una necesidad palpable en la sociedad actual: la necesidad de seguridad y privacidad en la información sensible del usuario.

Secretify surge no sólo como la solución a un problema, sino también como una oportunidad de negocio, orientada tanto a empresas como a individuales preocupados por la seguridad.

Se trata, asimismo, de un proyecto innovador, ya que no existen alternativas Open Source [38] o de pago capaces de proveer de todos los servicios que Secretify pretende ofrecer.

Las características que definen al producto que se derivará del desarrollo de este TFM son las siguientes:

1. Cliente de correo web, capaz de sincronizarse con múltiples cuentas de correo de un mismo usuario. Esta aplicación podrá ser utilizada en cualquier dispositivo con un navegador sin necesidad de instalación previa. Solamente será necesario crear una cuenta en la misma con la que generar las claves de cifrado del usuario.
2. Cifrado de mensajes entre emisor y receptor. El mensaje nunca pasará por los servidores de Secretify en un formato legible y no existirá ningún método para descifrar el mensaje por un tercero que no sea el propio destinatario.
3. Diseño e interfaz simples, amigables y altamente usables que garanticen una gran experiencia de usuario. El éxito de cualquier producto orientado a usuarios depende, en gran medida, de la percepción de simplicidad o dificultad de uso que el usuario tenga de él; si es difícil de usar, el comportamiento habitual consiste en buscar alternativas.

3. Objetivos y alcance

Debido a la característica temporal y la duración específica de este Trabajo de Fin de Máster, es necesario acotar qué funcionalidades podrán ser implementadas con éxito para conseguir los objetivos que se proponen.

Teniendo en cuenta la finalidad de Secretify, se puede definir tres grandes objetivos que sirvan como guía para el desarrollo del proyecto. Cada uno de estos objetivos diferenciados se ve como una pieza de lo que terminará convirtiéndose en la plataforma web Secretify. Estos objetivos son:

1. Desarrollo del algoritmo para el proceso de autentificación, almacenamiento de credenciales y cifrado de correos.
2. Implementación del módulo de autentificación de usuarios.
3. Implementación del prototipo de cliente de correos seguros.

Como ya se adelantaba, las limitaciones temporales influirán en el alcance de este proyecto. Para delimitar qué es lo que puede esperarse de este proyecto, debe destacarse que:

1. El algoritmo será implementado en su totalidad para obtener un Producto Mínimo Viable de Secretify.
2. Hoy en día existen múltiples entornos integrados de autentificación (OAuth [37]), sobre todo a través de redes sociales. Para esta prueba de concepto no se incluirán dichas posibilidades en el módulo de autentificación.
3. Un cliente de correos posee una gran cantidad de funcionalidades. Este trabajo se centrará en:
    a. Envío de correos. En este caso, de forma cifrada.
    b. Recepción de correos cifrados.

El prototipo de cliente web no incluirá como funcionalidad el envío de ficheros adjuntos.

4. Metodología y proceso de trabajo

Dado que la mayor parte de la dedicación de este proyecto está orientada a desarrollo de un producto software, en este apartado se describirá la metodología utilizada para la producción de código. Adicionalmente, se describirá la metodología utilizada para el refinamiento de la interfaz y la experiencia de usuario.

4.1. Metodología de desarrollo

Con el fin de garantizar la calidad del producto, se ha optado por el desarrollo dirigido por pruebas (Test Driven Development [43]). Esta metodología consiste en desarrollar pruebas, tanto unitarias como funcionales, del código desarrollado con las que validar que cumple su finalidad de forma correcta.

La peculiaridad adicional de esta metodología es que primero deben desarrollarse las pruebas, para después implementar el código que satisfará los requisitos especificados en las mismas. Suele utilizarse, además, un paso adicional de refinado del código para mejorar la calidad del producto elaborado.

Este proceso se conoce como RGR (Red, Green, Refactor) [41] y es una garantía a la hora de reducir la deuda técnica derivada del desarrollo tradicional.

4.2. Metodología de diseño y usabilidad

El proceso de diseño se ha basado en una simplificación de diseño centrado en el usuario (user centered design) [46]. Las actividades principales que han sido desarrolladas son:

1. Elaboración de prototipos de baja fidelidad; concretamente prototipos en papel.
2. Validación de prototipos con usuarios de diversos grupos demográficos.
3. Estudio de la usabilidad de los elementos presentes en el prototipo.
4. Implementación del diseño e iteración.

5. Planificación

La planificación principal de este proyecto se ha realizado dividiendo las tareas de implementación en unidades lógicas más sencillas que permitiesen monitorizar en todo momento el avance del desarrollo.

Para ello, y puesto que se ha seguido una metodología de desarrollo dirigida por tests, cada módulo de desarrollo se divide en dos partes: una parte de implementación de pruebas y otro de implementación de código. Además, ha sido necesario añadir una serie de tareas de diseño y organización del entorno de trabajo, previas al comienzo del desarrollo del prototipo en sí.

Los grupos principales que aglutinan todos estos módulos son los siguientes:

1. Desarrollo del algoritmo para el proceso de autentificación, almacenamiento de credenciales y cifrado de correos.
2. Iniciación del proyecto de desarrollo.
3. Implementación del módulo de autentificación de usuarios.
4. Implementación del prototipo de cliente de correos seguros.

Las fechas clave en el desarrollo del proyecto serán aquellas que coincidan con los días fijados para realizar entregables y aquellas en las que se cumplan los hitos. Con el objeto de reunir las fechas clave se utiliza la tabla siguiente:

Evento	Tipo	Fecha
Desarrollo del algoritmo para el proceso de autentificación, almacenamiento de credenciales y cifrado de correos	Hito	15/03/2014
PEC 1	Entregable	17/03/2014
Iniciación del proyecto de desarrollo.	Hito	23/03/2014
PEC 2	Entregable	31/03/2014
PEC 3	Entregable	28/04/2014
PEC 4	Entregable	26/05/2014
Implementación del prototipo de cliente de correos seguros	Hito	24/06/2014
PEC 5	Entregable	16/06/2014

Tabla 1. Fechas clave del proyecto

Para ilustrar la planificación del Trabajo de Fin de Master se ha elaborado un diagrama de Gantt.

Id de tarea	Nombre de tarea	Duración	Inicio	Fin	Predecesoras
1	**Desarrollo del algoritmo para el proceso de autentificación, almacenamiento de credenciales y cifrado de correos**	**30 hrs**	**sáb 01/03/14**	**sáb 15/03/14**	
2	Diseño del algoritmo de seguridad para autentificación y almacenamiento de credenciales de usuario	5 días	sáb 01/03/14	mié 05/03/14	
3	Diseño del algoritmo de envío y recepción de correo cifrado.	5 días	jue 06/03/14	lun 10/03/14	2
4	Diseño de la arquitectura cliente-servidor: webapp + API	5 días	mar 11/03/14	sáb 15/03/14	3
5	**Iniciación del proyecto de desarrollo**	**16 hrs**	**dom 16/03/14**	**dom 23/03/14**	
6	Creación del espacio de trabajo e integración con control de versiones	4 días	dom 16/03/14	mié 19/03/14	4
7	Creación de un proyecto sandbox en una plataforma PaaS (Heroku) para poder comprobar los avances en el desarrollo	4 días	jue 20/03/14	dom 23/03/14	6
8	**Implementación del módulo de autentificación de usuarios**	**48 hrs**	**lun 24/03/14**	**mié 16/04/14**	
9	Tests para módulo de autentificación en la API	4 días	lun 24/03/14	jue 27/03/14	7
10	Implementación de métodos para el módulo de autentificación en la API	10 días	vie 28/03/14	dom 06/04/14	9
11	Implementación de autentificación en la webapp	10 días	lun 07/04/14	mié 16/04/14	10
12	**Implementación del prototipo de cliente de correos seguros**	**116 hrs**	**jue 17/04/14**	**vie 13/06/14**	
13	Tests para el envío de correo en la API	4 días	jue 17/04/14	dom 20/04/14	11
14	Implementación de métodos para el envío de correo en la API	6 días	lun 21/04/14	sáb 26/04/14	13
15	Implementación de envío de correo cifrado en la webapp	16 días	dom 27/04/14	lun 12/05/14	14

16	Tests para recuperación de correos del usuario en la API	4 días	mar 13/05/14	vie 16/05/14	15
17	Implementación de recuperación de correos del usuario en la API	12 días	sáb 17/05/14	mié 28/05/14	16
18	Implementación de recuperación de correos en la webapp	16 días	jue 29/05/14	vie 13/06/14	17

Tabla 2. Tareas: duración, dependencias y fechas de inicio y finalización de las mismas.

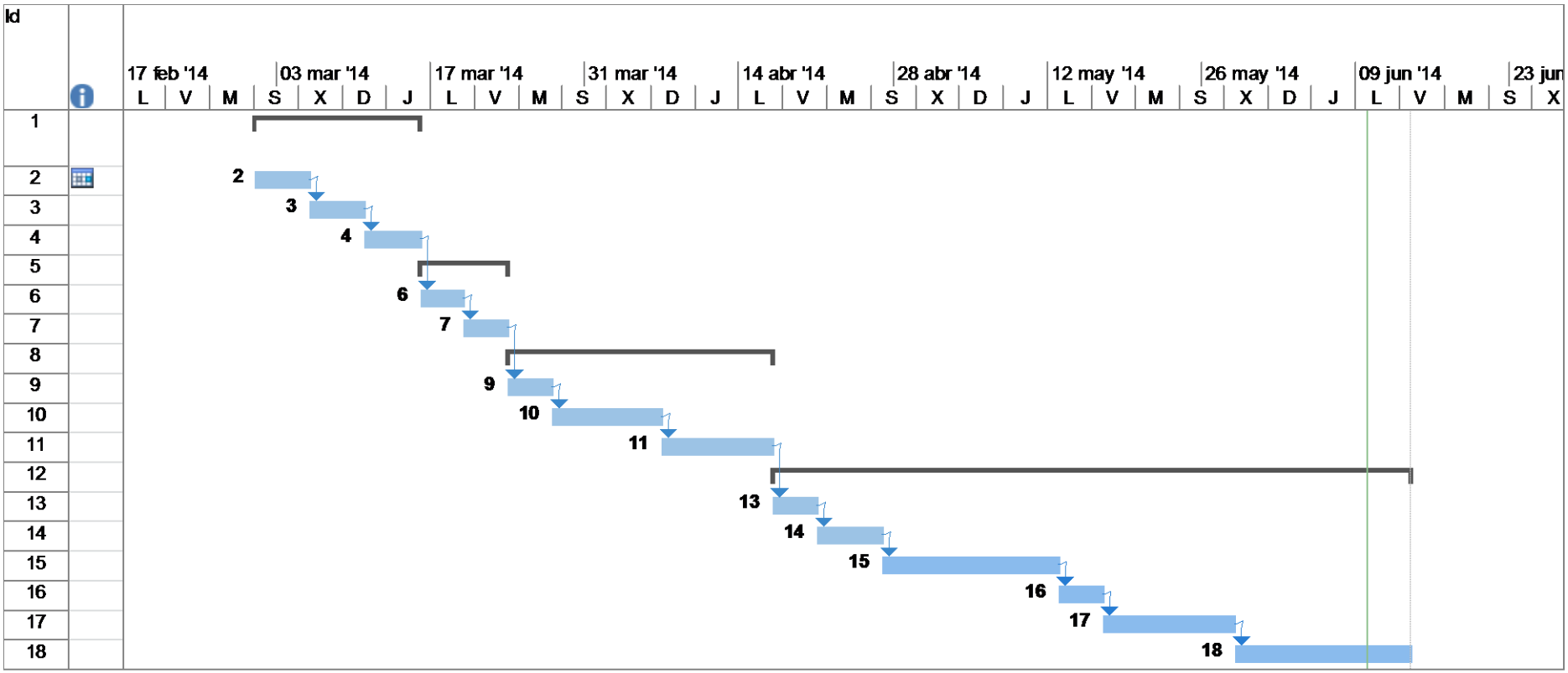


Figura 1. Diagrama de Gantt a día 29 de marzo de 2014

6. Presupuesto

El coste estimado para el proyecto a fecha de 16 de junio de 2014 se desglosa como sigue:

Objeto	Coste Unitario (€)	Unidades	Coste
Elaboración de documentos entregables y memoria final	25€ / hora	10h * 4 = 40h 60h * 1 = 60h Total: 100h	2500 €
Diseño y desarrollo del prototipo	25€ / hora	200h	5000 €
Licencia de IntelliJ IDEA	180 €	1	180 €
		Total	7680 €

Tabla 3. Desglose de presupuesto

El precio total del desarrollo del prototipo y los distintos documentos intermedios producidos asciende a 7680 €.

7. Estructura del resto del documento

Breve descripción de los otros capítulos de la Memoria.

Explicación de los contenidos de cada capítulo y su relación con el trabajo en global.

A continuación, la memoria se divide en cinco secciones relacionadas cada una con procesos distintos en el transcurso del trabajo:

- Capítulo 2: Estado del arte. Evaluación previa de aplicaciones ya existentes; ventajas y desventajas. Obtención de conclusiones al respecto.
- Capítulo 3: Plan de Negocio. Propuesta inicial de plan de negocio para Secretify.
- Capítulo 4: Diseño e implementación. Metodología, principios y patrones seguidos en el diseño y desarrollo de la aplicación.
- Capítulo 5: Demostración. Ejemplo de uso de la aplicación.
- Capítulo 6: Conclusiones. Conclusiones y líneas futuras de trabajo.

Capítulo 2: Estado del arte

1. Introducción

En la actualidad no existe ningún servicio operativo que reúna todas las funcionalidades principales de Secretify. Debido a esto, el proceso de evaluación de estado del arte se centrará en valorar, para una serie de aplicaciones relacionadas con la gestión del correo y la seguridad, sus características; puntos fuertes y puntos débiles. En concreto, se evalúan los siguientes servicios:

1. Horde Webmail [23].
2. SquirrelMail [42].
3. Microsoft Outlook [30].
4. Mozilla Thunderbird [35].
5. Lockify [27].
6. Mailpile [29].

Para elegir esta lista de servicios se han escogido dos servicios de cliente de correo íntegramente web, dos servicios de cliente de correo de escritorio de uso mayoritario y dos servicios en fases tempranas de desarrollo.

2. Alternativas analizadas

A continuación, se analizan una a una las características de los servicios listados en el apartado anterior.

2.1. Horde Webmail

Horde Groupware Webmail Edition es una suite de comunicaciones de navegador preparada para empresas totalmente gratuita. Utilizándola los usuarios pueden leer, enviar y organizar emails y organizar y compartir calendarios, contactos, tareas, etc., con otras aplicaciones que componen Horde.

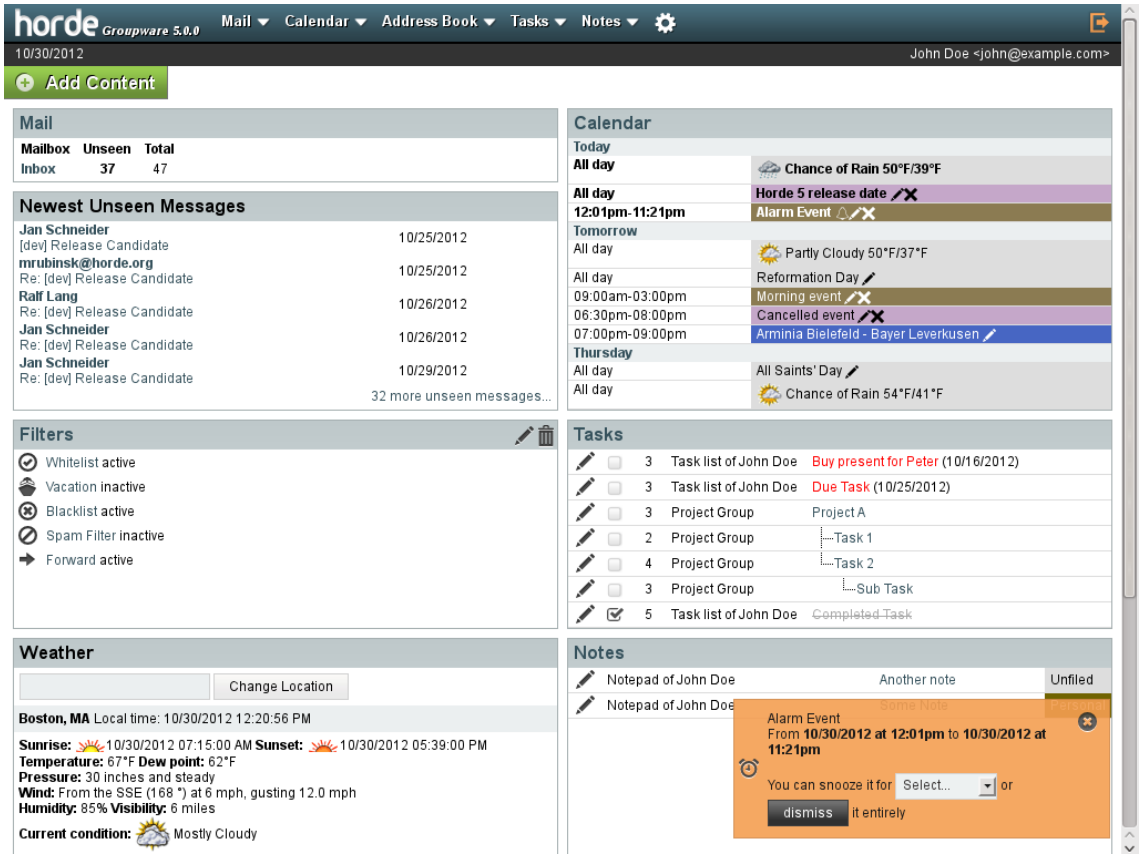


Figura 2. Pantalla de inicio del servicio Horde Webmail

En la figura 1 puede observarse el diseño de la plataforma. A continuación, se analizarán sus puntos fuertes y puntos débiles:

Pros:

- Ofrece multitud de funcionalidades; organización de tareas, notas, alarmas de eventos, etc.
- Permite realizar todas las operaciones básicas relacionadas con el correo electrónico.

Contras:

- Diseño desfasado, poco atractivo.
- Demasiadas opciones que acaban saturando al usuario.
- No ofrece ninguna capa extra de cifrado.

A pesar de que no se trata de una aplicación con la que gestionar el correo del usuario, sino de una aplicación web de dominio específico con la que gestionar buzones corporativos, es una muestra de qué puede esperarse en la actualidad de un cliente web.

2.2. SquirrelMail

SquirrelMail es una aplicación webmail creada por Nathan y Luke Ehresman y escrita en PHP. Puede ser instalado en la mayoría de servidores web siempre y cuando éste soporte PHP y el servidor web tenga acceso a un servidor IMAP y a otro SMTP.

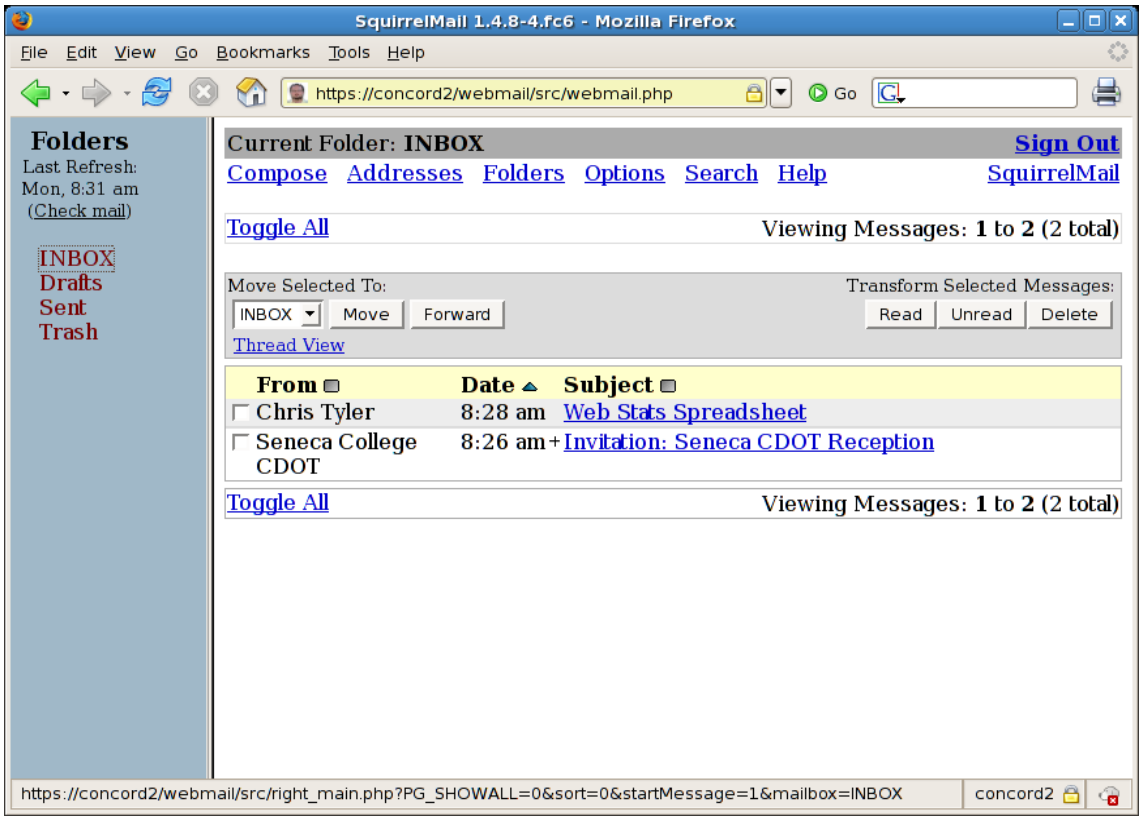


Figura 3. Pantalla de bandeja de entrada de SquirrelMail.

La figura 2 ilustra cual es el aspecto de este cliente web. Respecto a sus ventajas e inconvenientes, puede destacarse lo siguiente:

Ventajas:

- Sencillez, se limita a la gestión de emails.
- Completo a pesar de ser un cliente básico.

Inconvenientes:

- Se echan en falta opciones como etiquetado o gestión avanzada del correo.
- Diseño anticuado y poco amigable para el usuario.

Si bien es cierto que se trata de un cliente sencillo que cumple con su objetivo, y que, al igual que Horde está orientado a la gestión de correos corporativos, es otro ejemplo de que no hay alternativas aceptables desde el punto de vista del usuario para realizar la gestión de su correo desde el navegador.

2.3. Microsoft Outlook

Microsoft Outlook es el cliente de correos de Microsoft, ofrecido dentro del paquete de aplicaciones Microsoft Office.

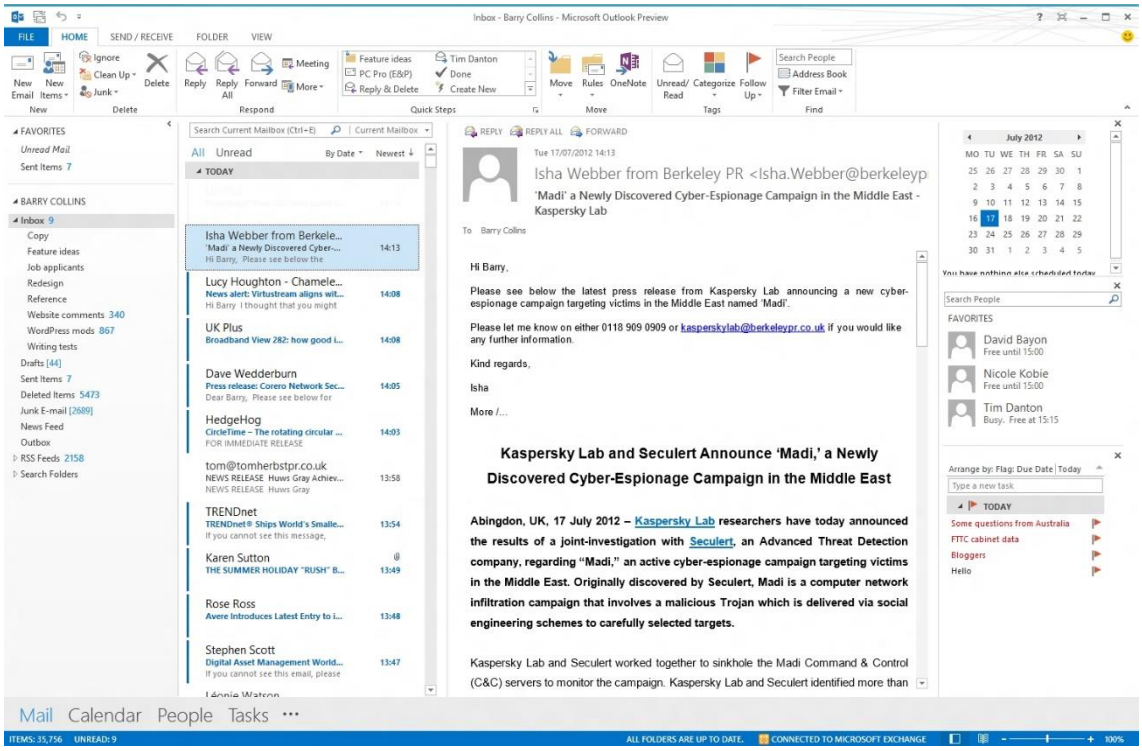


Figura 4. Pantallazo de Microsoft Outlook.

La interfaz de Outlook se ha actualizado a la interfaz Metro propuesta por Microsoft a partir de Windows 8. En concreto, ahora su interfaz imita a la de los otros programas de la suite ofimática, como por ejemplo Word o Excel.

En este caso, hay que destacar como ventajas:

- Opciones que ofrece.
- Posibilidad de sincronizar varias cuentas y unificar la gestión.
- Integración con otras partes de la suite ofimática.

Y como desventajas:

- Se trata de software propietario por el que es necesario pagar una licencia para ser utilizado.
- No ofrece soluciones seguras de correos cifrados para el usuario.

Outlook es uno de los clientes de correo más utilizados, con más del 10% de share en el sector de los clientes de correo electrónico [10]

2.4. Mozilla Thunderbird

Extraído de la página oficial [35]: “Thunderbird es una aplicación de correo gratuita fácil de configurar y personalizar, ¡y con muchas características geniales!”. Se trata de un cliente de correo gratuito creado por la fundación Mozilla que puede ser descargado e instalado de forma rápida y sencilla.

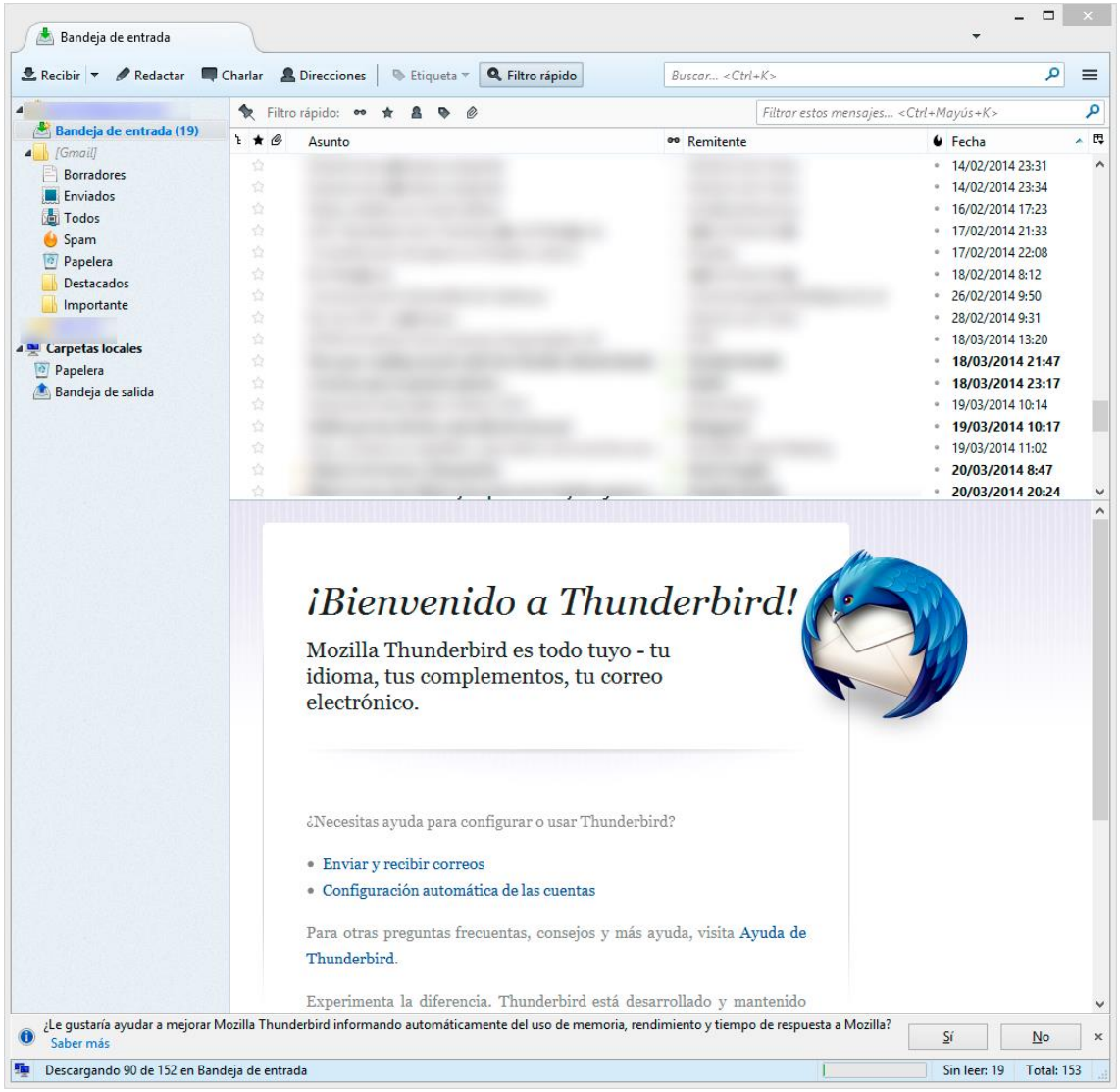


Figura 5. Pantallazo de Mozilla Thunderbird.

Como detalles característicos de Thunderbird pueden nombrarse:

- Cliente de correo gratuito.
- Open Source.
- Interfaz amigable.
- Gestión de múltiples cuentas de correo.

Sus puntos débiles son:

- Requiere instalación.

- No existe versión móvil.

2.5. Lockify

Lockify es un proyecto en desarrollo que se encuentra actualmente en beta privada. Podrá ser utilizado a través del navegador, bien como extensión de Google Chrome o como aplicación web.

La principal diferencia con la propuesta de Secretify radica en su objetivo; no desea convertirse en un cliente web de email, sino en una solución para la seguridad de los correos en sí; Secretify apuesta por la utilización de la seguridad como elemento diferenciador respecto a otros productos. Sus principales características son:

- Correo cifrado utilizando el estándar AES-256.
- Experiencia de usuario agradable a través de la simplicidad.
- No requiere registro ni acceso a la plataforma.

Se trata de un proyecto muy reciente que aún no ha comenzado su andadura en el mercado. Sin embargo, por sus características, resultará en un competidor de Secretify.

2.6. Mailpile

Mailpile es, al igual que Lockify, un proyecto actualmente en desarrollo que se encuentra en fase alpha. Se trata de una solución gratuita de código abierto, que pretende convertirse en un cliente de correo seguro, caracterizado además por una interfaz de usuario simple, usable y amigable.

En este caso, sí es un cliente de correo electrónico seguro. Las principales características de este producto son:

- Implementación de OpenPGP para conseguir privacidad en el correo.
- Etiquetado y categorización de correos.
- Sincronización con varias cuentas.
- Instalación local.

La principal diferencia entre Secretify y Mailpile reside en su ubicación; mientras que Mailpile requiere una instalación local, Secretify se alojará en la web y será siempre accesible para el usuario.

3. Conclusiones de la comparativa

Las conclusiones principales a extraer de este proceso de benchmarking son:

1. Los servicios web de correo electrónico tienen una interfaz anticuada y poco amigable para el usuario común. No están extendidos ni realizan acciones en la línea de extender su cuota de mercado.
2. Empresas como Microsoft y Mozilla ofrecen un gran servicio a través de clientes de escritorio, que tienen el inconveniente principal de requerir una instalación y no tener versiones móviles que permitan la gestión cómoda del correo.

3. La creación de una nueva capa de seguridad sobre el correo para garantizar su privacidad se ve, cada vez más, como una necesidad. A consecuencia de ello están surgiendo nuevas iniciativas que persiguen una finalidad similar a Secretify.

Capítulo 3. Plan de Negocio

1. Resumen Ejecutivo

1.1. Idea de negocio

Por un lado, una aplicación web de cliente de correo electrónico multicuenta; por otro, la posibilidad de enviar y recibir correo electrónico cifrado en cualquiera de las cuentas que hayan sido registradas en el sistema.

El principal valor añadido que se obtiene con esta propuesta es:

- Por un lado, la centralización de los usuarios, permitiendo una comunicación más sencilla y directa.
- Por otro lado, la implantación de forma efectiva de los mecanismos necesarios de seguridad de la información. Por último, la implantación de todo ello sin necesidad de instalar nada, simplemente accediendo al servicio desde un navegador web con conexión a Internet.

1.2. Mercado, crecimiento esperado y competidores

En esencia, el público objetivo será todo aquél usuario individual preocupado por su privacidad y toda empresa que maneje datos sensibles que merezcan protección.

Actualmente, hay más de 2.500 millones de usuarios de correo electrónico, y se espera un crecimiento anual mantenido del 6%.

En cuanto a los competidores, existen diferentes alternativas, pero ninguna dispone de todas las ventajas que sí presenta esta solución.

1.3. Estado del desarrollo

El prototipo se encuentra cercano a su finalización y se pretende sacar al mercado, al menos una versión beta, en septiembre de 2014.

1.4. Inversión

Para la puesta en marcha del proyecto, se estima en 80.000 € una primera ronda de financiación.

1.5. Objetivos a medio y largo plazo

El objetivo principal a medio plazo es el de llegar a un gran número de usuarios de correo electrónico conscientes de la precariedad de la seguridad de su información, así como llegar al mayor número de empresas posible.

A largo plazo, se busca implantar este sistema como un estándar de facto para las comunicaciones seguras dentro del correo electrónico.

2. Producto de valor

2.1. Contexto

Es innegable el mundo actual está globalizado, y que cada segundo que pasa se genera una cantidad ingente de información de todo tipo.

Mucha de esta información se puede catalogar como personal, puesto que hace referencia a la vida privada de las personas y está protegida por las distintas leyes de privacidad de cada país. Como ejemplos se tiene desde el nombre o la dirección de una persona, hasta su nómina o sus datos médicos. Sin embargo, el uso y gestión que generalmente se hace de esta información personal no es siempre la correcta o deseable.

Por otro lado, se tiene información que, aun no siendo personal, sí es sensible y cabe esperar cierta protección por parte de las organizaciones que la poseen, ya sea porque hacen referencia a aspectos clave de un negocio, recursos específicos, secretos o, simplemente, porque esa información se considera confidencial.

La respuesta a por qué hoy en día se hace una consideración tan cuidadosa de la información es muy clara: la información es poder.

Y es por ello por lo que cualquier organización pública o privada, incluyendo desde grandes empresas hasta meros individuos, hace acopio y protege todo tipo de información que considera valiosa para sí mismo.

No obstante, la información, para ser útil, es necesario que se comparta o se transfiera entre distintas entidades. La clave para no perder el control sobre esta información, en general sensible, está en realizar la comunicación de forma segura, garantizando la confidencialidad, integridad, autenticación y disponibilidad. Es decir, garantizando la seguridad de la información.

La forma más común de comunicarse y compartir información actualmente es a través del correo electrónico. Sin embargo, generalmente, se hace de forma insegura, a través de proveedores de correo electrónico que lo ofrecen de forma gratuita a costa de perder el control y la confidencialidad de la información. En el mejor de lo casos, las organizaciones utilizan servidores de correo propios, los cuales son vulnerables a ataques internos.

¿Cuál es la alternativa?

2.2. Alternativas

La alternativa pasa por utilizar un servicio de correo electrónico que implemente los mecanismos necesarios para mantener la seguridad de la información. En especial, mantener la privacidad en las comunicaciones, cifrando los mensajes y procurando los mecanismos necesarios para que sólo el emisor y el receptor del mensaje sean capaces de leerlo.

2.3. Valor añadido

La idea principal es poder enviar y recibir correo electrónico de forma segura, a través de las cuentas que ya existen, añadiéndoles una capa de seguridad. Es decir, sin que sea necesario crear una nueva cuenta de correo electrónico.

Así pues, cada usuario podrá dar de alta una serie de cuentas de correo electrónico, desde las cuales podrá enviar correos a las cuentas de otros usuarios que también estén registrados en el sistema.

Además, todo esto se realiza a través de una interfaz web, sin necesidad de instalación alguna, utilizando el navegador. El usuario interactúa con un cliente de correo electrónico (similar a Mozilla Thunderbird, Microsoft Outlook o Apple Mail), totalmente web. El sistema se encarga de permitir el envío de mensajes cifrados a aquellos usuarios que estén registrados. Existe la opción de enviarlos sin cifrar en caso de que se desee hacerlo de este modo.

En resumen, tenemos dos partes distintas. Por un lado, una aplicación web de cliente de correo electrónico multicuenta; por otro, la posibilidad de enviar y recibir correo electrónico cifrado en cualquiera de las cuentas que hayan sido registradas en el sistema.

3. Mercado Potencial

3.1. Mercado

A finales de 2013, según Radicati Group [11], había alrededor de 2.500 millones de usuarios de correo electrónico en el mundo, moviendo más de 196 mil millones de mensajes al día, de los cuales, más de la mitad (108 mil millones), son de ámbito empresarial.

El mercado del correo electrónico es inmensamente grande y tiene un crecimiento esperado del 6%. Cabe destacar que cada usuario tiene, de media, 1,6 cuentas de correo electrónico.

3.2. Público objetivo

Dentro de esta ingente masa de usuarios, hay muchas personas interesadas y preocupadas por su seguridad; más aún después de las últimas publicaciones en los medios sobre la falta de privacidad que tienen los usuarios de algunos grandes proveedores de servicio, que monitorizan los correos electrónicos, vendiendo o cediendo esa información a terceras partes.

El problema con el sector de los usuarios individuales en general, es que están acostumbrados a los servicios gratuitos, sin darse cuenta de que realmente están perdiendo privilegios y derechos. Así pues, debe tenerse en cuenta que al usuario común no se le puede pedir un precio alto por el servicio.

Por otro lado, se tiene a las empresas, que son más conscientes de la importancia y del valor de la información que manejan y transmiten, y están dispuestas a pagar un mayor precio.

En esencia, el público objetivo será todo aquél usuario individual preocupado por su privacidad y toda empresa que maneje datos sensibles que merezcan protección.

4. Competencia

4.1. Antecedentes

Desde hace milenios, la humanidad es consciente de la importancia de la información. En el siglo I antes de Cristo, el político y militar romano Cayo Julio César utilizaba mensajes cifrados para transmitir órdenes y noticias entre sus aliados. Con ello conseguía que, si algún mensaje era interceptado, el atacante no pudiera conocer el contenido del mensaje.

Este concepto es la base de la criptografía y, como puede observarse, no es algo novedoso. Si ya hace más de 2.000 años, muchas civilizaciones se dieron cuenta de la importancia de salvaguardar sus comunicaciones, en los tiempos que corren, esa importancia se ha incrementado a la enésima potencia.

Han surgido distintas soluciones para hacer del correo electrónico algo mucho más seguro.

Entre los intentos más importantes, destaca Lavabit. Actualmente se encuentra con el servicio suspendido, presumiblemente por presiones gubernamentales. Se puede cuantificar la repercusión que tuvo este servicio de correo electrónico seguro viendo el número de usuarios: antes de ser suspendido, alcanzó la cifra de 410.000 registrados.

Entre las principales diferencias, destaca que era necesario crearse una cuenta de correo nueva (usuario@labavit.com), usaba una clave maestra y estaba sujeto a las leyes de Estados Unidos.

4.2. Competidores actuales

Entre los competidores más importantes destacan dos servicios.

Por un lado, está Mailpile. Provee de seguridad en el email, pero es necesario instalarlo en el propio servidor. Además, no contempla una centralización de las claves públicas, por lo que es necesario conocer a priori la clave pública del receptor del mensaje. Está en fase alpha y es open source.

Por otro lado, cabe destacar Hushmail. Es un servicio de pago (la licencia individual ronda los 35 $ al año), y obliga a la creación de una nueva cuenta con dominio Hushmail.

Tenemos así, distintas alternativas. Sin embargo, ninguna de ellas engloba todo lo que esta solución propone: multicuenta, sin necesidad de crear una nueva, asequible y totalmente seguro e independiente.

5. Modelo de negocio y plan financiero

5.1. Modelo de negocio

Como se ha comentado anteriormente, la propuesta de valor es clara: correo electrónico seguro junto con un cliente de correo web multicuenta.

Los potenciales clientes a los que se dirige son los usuarios individuales concienciados con la seguridad de la información y las empresas que manejan datos sensibles y quieren proteger sus comunicaciones a través de correo electrónico. Además, formará parte de la política de empresa

ofrecer un servicio asequible a cualquier persona. Teniendo en cuenta que existen dos perfiles muy diferenciados de clientes, es lógico tener una oferta distinta para cada uno: a los usuarios individuales se les ofrece una licencia individual por 0,89 € al año, mientras que a las empresas, la tarifa de uso es de 1.000 € anuales (con opción de 100 € al mes). Éstas, dispondrán de una serie de funcionalidades extra, orientadas al ámbito empresarial.

Los usuarios individuales tendrán acceso a la funcionalidad básica del sistema. Esto es:

- Enviar y recibir correos electrónicos cifrados a través de cuentas de correo existente.
- Configurar distintas cuentas de correo electrónico por usuario (máximo 5).
- Claves de 128 bits para cifrado simétrico y de 512 para cifrado asimétrico (clave privada).

Los usuarios empresariales dispondrán de una serie de funcionalidades añadidas. Entre ellas:

- Envío del mismo mensaje a distintos receptores.
- Revocación de mensajes (haciendo inaccesible un mensaje para algún receptor).
- Claves de 256 bits para cifrado simétrico y de 2048 para cifrado asimétrico (clave privada).
- Soporte de configuración de servidores de correo privados.
- Herramientas para comunicación segura con individuos u organizaciones.

Dependiendo del uso, presumiblemente, se marcará un número máximo de mensajes por motivos técnicos.

5.2. Plan financiero

Tabla 4. Gastos iniciales

Concepto	**Unidades**	**Precio unitario (€)**	**Total (€)**
Gastos de gestión y notaría de constitución de la sociedad	1	1.000	1.000
Capital constitución de la sociedad	1	3.000	3.000
Servidores	2	2.000	4.000
Certificaciones y auditorías	1	10.000	10.000
		Total (€):	**18.000**

Tabla 5. Costes de operación

Concepto	**Coste (€)**
Sueldos empleados	52.114,56
Housing	5.200
Luz e internet	1.800
Total (€):	**59.114,56**

Tabla 6. Estimación de clientes

	Caso esperado		**Caso optimista**		**Caso pesimista**	
Año	Empresas	Usuarios	Empresas	Usuarios	Empresas	Usuarios
1	**100**	**10.000**	200	20.000	20	2.000
2	**150**	**30.000**	300	50.000	50	8.000
3	**200**	**50.000**	500	100.000	100	15.000
4	**300**	**70.000**	800	200.000	150	20.000
5	**400**	**100.000**	1.000	500.000	200	30.000

Tabla 7. Estimación de ingresos (€)

Año	**Caso esperado**	**Caso optimista**	**Caso pesimista**
1	**108.900**	217.800	21.780
2	**176.700**	344.500	57.120
3	**244.500**	589.000	113.350
4	**362.300**	978.000	167.800
5	**489.000**	1.445.000	226.700

Tenemos así que el primer año, en el caso esperado:

Tabla 8. Caso esperado de ingresos

Año	Gastos (€)	Ingresos (€)	Beneficio (€)
1	77.114,56	108.900	31.785,44
2	59.114,56	176.700	117.585,44

Si se mantiene la tendencia de crecimiento del caso esperado, a partir del segundo año será necesario adquirir nuevos equipos para dar soporte al número de usuarios. Así mismo, se podrán subir los sueldos, contratar más personal y alquilar un local.

Tabla 9. Caso pesimista de ingresos

Año	Gastos (€)	Ingresos (€)	Beneficio (€)	Total (€)
1	77.114,56	21.780	-55.334,56	-55.334,56
2	59.114,56	57.120	-1.994,56	-57.329,12
3	59.114,56	113.350	54.235,44	-3.093,68
4	59.114,56	167.800	108.685,44	105.591,76
5	59.114,56	226.700	167.585,44	273.177,2

Tendría que esperarse hasta el cuarto año para obtener unos beneficios que permitan aumentar sueldos y contratar personal. A partir del tercer año, se empezaría a obtener beneficio positivo y se podría alquilar un local de cara al público.

Las estimaciones han sido realizadas en base a los usuarios de distintas plataformas que ofrecen servicios parecidos.

Por tanto, para una primera ronda de financiación con la que poner en marcha el proyecto, se necesitaría de unos 80.000 €.

6. Organización

6.1. Fundadores

Nombre del fundador A. Graduado en Ingeniería Informática. Máster Universitario en Ingeniería Informática. Realizará las funciones CEO en la empresa, encargándose en un principio de las gestiones comerciales y de marketing.

Nombre del fundador B. Graduado en Ingeniería Informática. Máster Universitario en Aplicaciones Multimedia. Realizará las funciones de CTO en la empresa, siendo el máximo responsable del desarrollo de software.

6.2. Personal a incorporar

Para lograr un desarrollo de negocio satisfactorio, se precisa de un perfil más empresarial, aunque interesado en las Tecnologías de la Información y las Comunicaciones.

6.3. Salarios

En un primer momento, cada uno de los tres primeros empleados percibirán un sueldo de 1.000 € netos al mes (haciendo un total de 17.371,52 € al año, por persona), y 14 pagas. En el caso de estar alguno dedicándose a media jornada, percibirá exactamente la mitad de dicha retribución.

7. Plan de Implantación

7.1. Estado y previsión de puesta en marcha

Actualmente, el prototipo se encuentra en plena fase de desarrollo, y se estima su finalización hacia mediados de julio de 2014.

Durante el verano de 2014 (desde mediados de junio hasta finales de agosto), se procederá a la puesta a punto del prototipo para convertirlo en una versión, si no final, muy cercana al servicio final que se quiere ofrecer. Se tiene el mes de septiembre como objetivo para la creación de la empresa y la salida al mercado del servicio.

7.2. Control

Cada tres meses, desde la puesta en marcha del proyecto, se realizará un informe para conocer el estado actual del proyecto, y analizar así las necesidades, debilidades, oportunidades, amenazas y fortalezas.

8. Alianzas, Marketing y ventas

8.1. Alianzas

Desde las más tempranas etapas de despliegue, se pretende buscar alianzas con el mayor número posible de personalidades y organizaciones relacionadas con la seguridad de la información en Internet. Se presentará el proyecto y se buscará apoyo en estos círculos.

8.2. Estrategia de marketing

La dificultad de implantar una nueva solución y la resistencia al cambio de hoy en día son factores importantes a considerar. Para tratar de superarlos, se realizarán campañas de concienciación sobre la falta de seguridad que existe actualmente. Además, se promocionarán los 1.000 o 2.000 primeros registros con un descuento del 100% para los usuarios individuales. En cuanto a las empresas, se les

facilitará una prueba de uno o dos meses, para que se familiaricen con el servicio y se percaten de sus ventajas.

En cuanto a la publicidad, se promocionará el servicio a través de distintos medios en Internet, dentro de las posibilidades de financiación existentes.

9. Riesgos y estrategia de salida

9.1. Riesgos

El mayor riesgo al que se enfrenta el proyecto es el de la falta de encaje entre el servicio y las necesidades que cubra del público objetivo.

Esto se puede deber a múltiples causas. Por un lado, puede ser que los usuarios de correo electrónico no estén realmente concienciados con la seguridad de la información o que no estén preparados para un nuevo sistema integrador y centralizador. Por otro lado, puede que manejar la desconfianza del usuario para con nuestro servicio represente una barrera, no fiándose de su seguridad o no dando de alta las cuentas de correo electrónico personales.

Para tratar de mitigar estos riesgos, se publicarán los esquemas de seguridad utilizados, así como una serie de términos y compromisos de la empresa con sus usuarios, garantizando, en la mayor medida posible, la seguridad de los datos de éstos.

En caso de una menor aceptación o unos costes más elevados de los previstos, el crecimiento de la empresa se producirá de forma más lenta.

9.2. Estrategias de contingencia

En el caso de llegar a un punto en el que la continuidad del negocio sea imposible, se aplicarán las estrategias de contingencia necesarias.

En primer lugar, se buscarán empresas u organizaciones que puedan estar interesadas en adquirir parte del negocio, transfiriéndoles los derechos de uso y explotación de la plataforma.

En paralelo, se analizarán las causas que han producido esa situación. Se revisarán los aspectos del negocio que no ayuden a mejorar el nivel de negocio hasta un mínimo aceptable, desde las tarifas o licencias de uso, pasando por recurrir a otras formas de ingreso (como publicidad en la web), la modificación del segmento de clientes objetivo o el cambio del servicio que se ofrece.

En caso de que ninguna de estas medidas de contingencia surtiese efecto, se pasaría a liquidar el proyecto en su conjunto.

Capítulo 4: Diseño e implementación

1. Arquitectura de la aplicación

Secretify constará de dos partes bien diferenciadas, un lado de cliente y un lado de servidor.

1.1. Servidor

En concreto, la parte de servidor consiste en una base de datos con la información necesaria relativa a cada usuario y una API [4] a través de la cual poder realizar operaciones de manejo de datos (crear, leer, modificar y borrar) sobre dicha base de datos y utilizar los servicios de los proveedores de email; sincronización con IMAP y envío de correos a través de SMTP.

1.2. Cliente

Respecto a la parte de cliente, está constituida por una aplicación web encargada de mostrar al usuario la información referente a los correos y de generar localmente la clave privada del mismo. Con esta capa de funcionalidad implementada en el lado del cliente, se garantiza al usuario que será el único con la posibilidad de descifrar los emails cifrados que reciba.

2. Algoritmo de cifrado

Secretify emplea un algoritmo de cifrado y descifrado totalmente seguro y desarrollado a nivel de cliente. La función del servidor en esta plataforma es la de actuar como medio de transporte y proveer información referente a la infraestructura de clave pública.

Los algoritmos de cifrado que se utilizan en Secretify son:

- AES-256 [3]
- RSA [2]

El proceso de cifrado es el que se ilustra en el siguiente esquema (*Figura 6*).

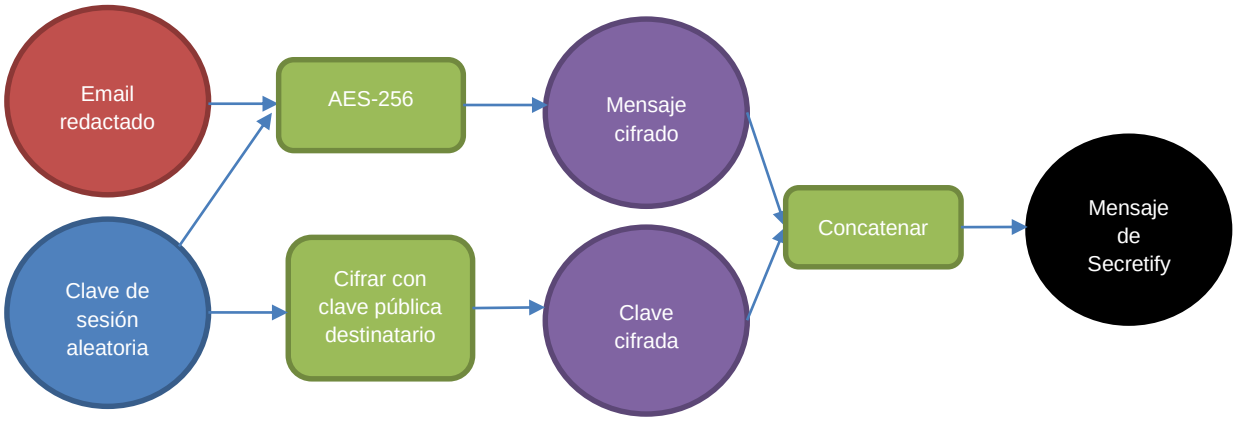


Figura 6. Esquema de cifrado de Secretify

A fin de demostrar las interacciones que se dan entre cliente y servidor para las acciones relacionadas con el envío de emails cifrados, puede consultarse el esquema siguiente (*Figura 7*).

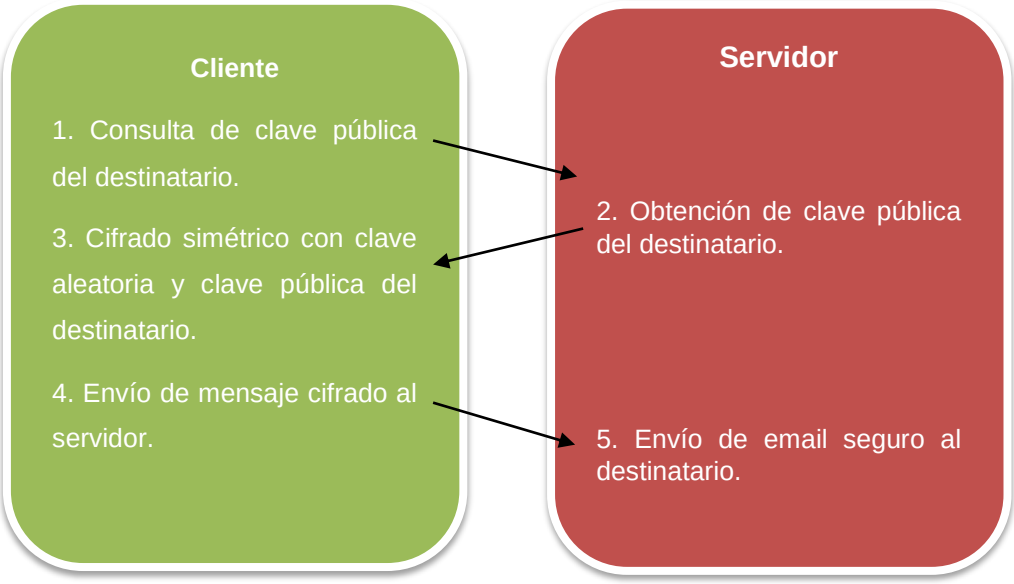


Figura 7. Esquema de interacción cliente-servidor para envío de emails seguros

El proceso de descifrado de emails es el que se ilustra en el esquema siguiente (*Figura 8*).

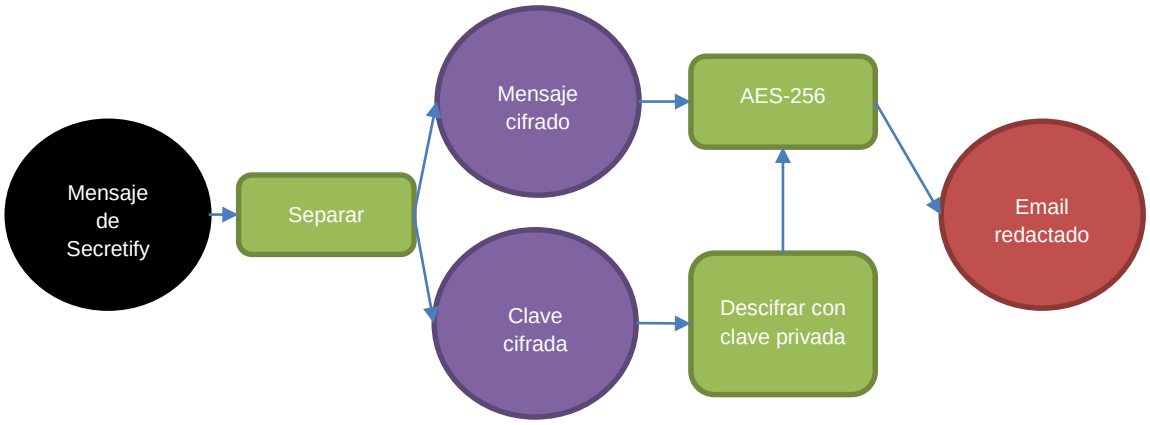


Figura 8. Esquema de descifrado de Secretify

Respecto a la recepción de emails, consúltese el siguiente esquema para comprobar la interacción cliente-servidor (*Figura 9*).

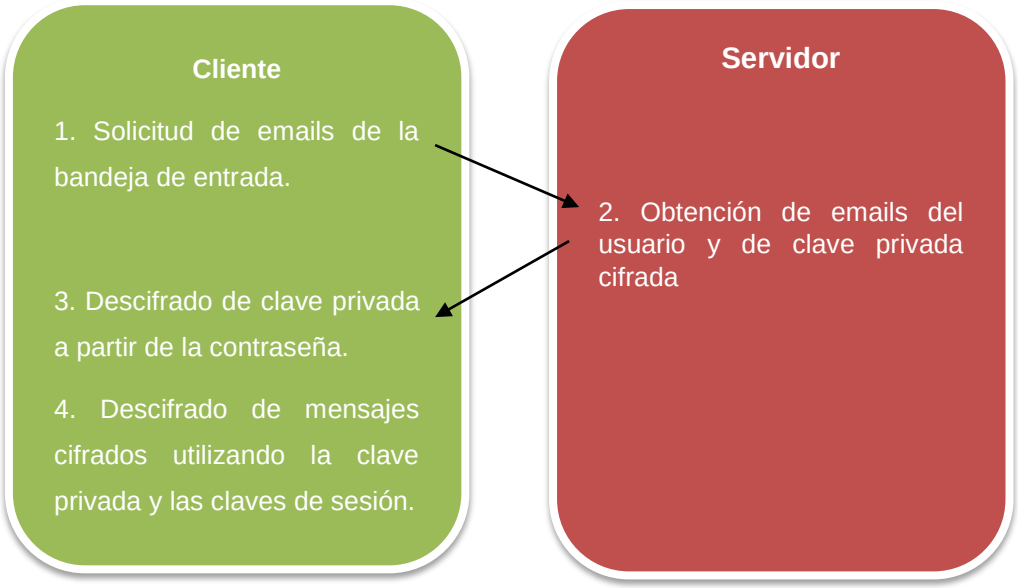


Figura 9. Esquema de interacción cliente-servidor en recepción de emails seguros.

Con el fin de conseguir una comunicación segura entre cliente y servidor, será imprescindible que la plataforma funcione sobre un dominio HTTPS [40] que evite la intercepción de información sensible en la comunicación entre ambos extremos mediante.

3. Arquitectura software

La aplicación de Secretify tiene cuatro componentes principales:

1. Base de datos. Almacena los datos cifrados de usuarios y sus credenciales de correo.
2. API. Ofrece una serie de métodos con los que trabajar sobre los datos almacenados en la base de datos de Secretify y funcionalidad necesaria para el envío y sincronización de correos.
3. Servidor web. Se encarga de esperar por peticiones del cliente y actuar en consecuencia. Estas peticiones pueden ser:
    a. Peticiones web: el servidor web responde con el contenido de las páginas estáticas de información o la página principal de la aplicación web.
    b. Peticiones para la API: el servidor web pasa la petición a la API.
4. Aplicación web. Aplicación con la que el cliente utiliza las funcionalidades de Secretify. Se comunica con la API a través del servidor web.

La jerarquía de interoperabilidad de las distintas partes se ilustra en el esquema siguiente (*figura 10*).

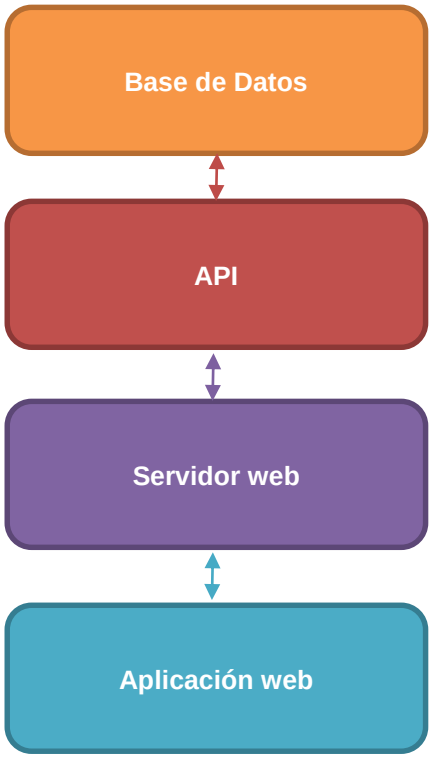


Figura 10. Esquema de interacción de componentes.

Tanto para el diseño de la parte de servidor como para la de la parte de cliente, se ha optado por la utilización de una variación del patrón clásico de modelo, vista y controlador (MVC) [32]. A continuación, se describe cada componente en más detalle.

3.1. Base de datos

En la base de datos se almacenan datos cifrados del usuario, de modo que no tenga que introducir sus credenciales cada vez que acceda a la aplicación. También se ha creado un campo ampliable de atributos de estado con el que almacenar información estadística sobre el usuario.

Dadas las características de los datos que serán almacenados, se ha optado por la utilización de una base de datos no relacional basada en documentos, en lugar de una base de datos clásica SQL.

Para ilustrar la estructura y los campos que son almacenados, se muestra un ejemplo de documento de usuario del entorno de pruebas de Secretify (en formato JSON [26]):


```
{
  "__v": 4,
  "_id": {"$oid": "539369284b25f4a817898fe5"},
  "confirmed": false,
  "email": "test@test.com",
  "emailCredentials": [
  {
    "_id": {"$oid": "53938baf1bd967081d18d4db"},
    "imapUrl": "imap.gmail.com",
    "smtpUrl": "smtp.gmail.com",
    "password": "~®é|6»—²·®?‡ª\u000f\u0012î",
    "email": "secretify.test@gmail.com",
    "tlsRequired": true,
    "imapPort": 993,
    "smtpPort": 465
  }
  ],
  "password": "$2a$10$xldkrLoXwqpLQL0yL7JuFOpT/sn1i1FTb5TMB5xFfVWmWytn9xq4.",
  "privateKey": "J]'0Æ\u0011X□Ê‡\u001cƒëb\u0004n\tÈ‡/Ï•IàÏÀæ\u0006\u001e™{\u0017ÃÈ
\u0007.□èm•Ÿÿ/®Æ\u0012Å”ý-¿‰J ð,tí/,,‡JŸ□ÛY;‰ÅFä,ì_ä—…b¨\f´)□éƒwu¤¯Mæ\u0014x-÷Æœ(
½T¹□»BŸÛ}□Ã'Ép¯\u000b$3ÆÌ\u0018¯?¡gi”\u0019Ð¨£†™ïŒè\"÷§ÚØs\u000f<XKX\tJ\u0014ù›~k\
u0012•uû O\u0002ËH4Ïrô\u0005\r’~\u0000\"¨.¡_—Î»^W\u0016\u0002rà\u0019U5^^¥^7çÈAAO\
u0014).%Wì¡|\u000fàF6)47aGúŸu\u0013ò'G]\tsN`Î¢5\u0001G©AC¡Vhú)\u001dÉ\f,•òyÙnºrËèh
\u001d\u0007J¹?-â“?º_Í«\u0016□aûÉ¶2Ú•ÆnEK\fñ¢P\u0011˜Éé·xÚ><□ËÆ˜Og9¹YcÏKû\u0018Eð<
YJÙz\b!ÑÚpr&—¸œ~KGv0M\u0013¤‰aZÛ5ê\f‰#8UÛ¸B’n\u0010xae¼OŠºñ{ùQ\u0010U±GÝ\nÚ¢oð/¶Ÿ+
¬ý(\u0017æ°Ô¦§AÔèsÿIxdÔZ\u0019Ac\u0002•Wy…ßºªÍ\u001f’aE.¥Ù\u001aœº\u0013}¦îü²¬\u00
13R\bpÀ%’FÙ½<½ópâlh\\¶ Òœ³õ’öÌ\u000e&{\u0016Ô;Ö,<<.ƒ›Ó•O±\f9<ûÛòø\u0002\u0001—ëb§\
u000eÈ]|f¿□Ò<□Ü„ñŸ³–\u0001M¥-‰ë»\u0016\u001cÀ(D%^‡-Þ²ØS\u0000ÿ©\u0005Ó\u001f\u0004
HhûoFÏíÙý¤ª\u0006û¶ê’\u001eõ\u001bÑËÆ\u0003ã¨6□³êÍÌÑs‰ŠSSàuÍæ\u001a’’\u0011öz–\u00
11*\u001a³wšm\u0004†ií\"µT>PµøÛÑoùcÓ±□D™i·qŒlÂÎÓ/¦Ä˜ád\u001171›zì¡P gGœ•ì xq+\u001
7ÂÄÒ\":&ð\\í‰o\u001e-ÈPœt2¼k\u001fµ\tr□'y?\u0007)Q\"S\u001d¬\u000f,L9AX%\u0016¾·?i
\t\rX\f§K*Æ°d›¦•~úXÆ—••`Z¢X¯î\u0004¬n+ÇjL†\u0006²Åt1cÑË\u001eÌ\u0013{ç&¿\u001f»HR‡
```

```
éÞ9%çä£1)_ð%(K·H’ÀÞÀµROL\u0017‘Ö([¡Q¦_®ÀD^?pJ\u0019gÛº\r=\\ë~b%©nA\bõ)\u001d¤²u¾Ã
µ²ä¥1,ƒZc`/©Ìw«\u000eèÇr(ND©8\u0013*§Ú÷\u001d`äj’\u0018}Ð\fwtb•éìï\u0002g#pS§kÐ‘
\fŸ.^q¦¹å©©&{×á\u001eð\u0003+„‰Ó\u000f\\ƒŸŸ}vÎ«D}|„k\u0006\u0012,®¦tWu\u0016¿Tô4
È\u0013~„Üò÷Ù‹2$[*ÎËUfáš7¤ç‰é¨•U\u0003 $\u0011.úõæ\u001c\u0001ú\u0001\\jÔñ}\u000eù
µ3°]Åu³\u0011\u001aø,¦FT\u0003›Þ¢~•È\u0001£4u$Ã|¦WÀ½'ðæWRø¡\u0014\u0014¼Q~MÙ”ŸaÍkz
Nkaú,D_%Zèm‡µ’\u0006³Ø¥kuYŠ956}\nN\u0003T—gˆá¨4xhÐ\u0004ýaÏw]àV{jCÌj.¯èMû]BÐŒ¦ã¥¥H
Øˆð\u001ac\u0012cs„4]•a-&(,#Û™i;†$ã\u0018ŒÛ¹¬Ä\u0011ÝÄœ\u0017Cß\u0012Ö\u00166tC<Á
\f%JtÉuœ‘^~¹ëRu=‰\u0014š\u00049\u0010˜Û¹©s°Å\tå1èÒø¥ÈÄ.Ï\u001bú«ñPú ð®Ì•[½e\u0014Ã
L~\u000e,…Eù¸Çk™.+]¬TÌ\u0010n\\\u0013›¨qŒB–a¹‹þð¸µk\u0002wã‰2Ápê\u0019\u0017\u00
01fb‰PIå:s,ÚqrºJöáå´\u0002uÃÞnoÒ\bæœßÞ,®µƒ\"{ŠÇ’áëœêÝ¢\u0017FMû^-0•ÙË•\\\u000fä
ìØvî¢^\u0018vKuzâ}?w³ùnU»w´˜LiV•Ë¡WNC ÛÐÙLíÀ\u001c4ï\u0000ß£1sG\u0018/º)6ü8¦\u0010
ˆÇî&×sµ÷\téoR\f=\\þ 2mãw®«ÆT©•ô¿Ê-ô);#ø\u0019j`vv\u0012øÊú\u001f\u0016–ÃÉ$ÉÇÝÕº
\bGZyR rÃgX@b?˜¸øùLè3\u000f°83{ö·®¡/ì=ð¹6\u001cmh,éÒ²¡[#tiû\u00074ÿÒ0^ ”Â–¨¢¥Ú@è
ˆÿSEoô„,ÎÂ;\u000f´Y,Š\u001b¾äK¤%@\u000fHyÆAb%e/¾+\u001aèÙÛ'n³\u0001Ö/\u0000ÏñŸL!¿Ç
0À\u0018±{a\u0005&ÉW/É¯å£À$Ì¾\u0005f$<aÃÍ9´\\¼\n^ØóÄïéð±ý¾£%U:„’‘½>Ø.W{ÂoÚšXeñG»
\u001f]\u001f=ÆTÍ\tß•¸mð«uJ**î•3¾ö\u0011²c\u0017O¹Ü\u0015ôM“ì™‡±ƒ£ŸEéNL=ðùzt&\u001
3%y)š\u0012ÞÁÄoî•\bqYLx÷\n\u001bôù\u001a²ãr3XôØSûÁñ•¾ÏÂ{õ'WÂLá\u000eð¢øã†Þ\u000fyÿ
æœ\u001bÇÔ\u0004ä×2\f„j •ÅÅ&Y\u001eûÂ\rÇ\u0007Á1",

  "publicKey": "-----BEGIN PUBLIC KEY-----\r\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAjCYk1ZM7n6OHCaRV8Yuy\r\n7RoE0ThJ1YqNjNtNEl2gDEFgnCtc88w3IOUktcgQcVedc8tET7
qz4n8Es2fkLqVO\r\nm4doTJl8XhX0rbEvk22yl0W83PZLMcooY87tEQ2MuzRjNC9VRM7kdfgY8ng5qtRA
\r\nCSyCRb25eTzMNnqAuDU6iZx9jdoVVTl9bRZ36funLOosotxmbtSGYAcnbDbC/+ub\r\nmhU+Vlfc09
rqRcatcvk2vmOnanRDAPVlXN9oSE26X5S6hd8Nr8AaqnG8PKreyyru\r\nxas74UP+sWJKV285eEr0v9u0
GJhSwd7dPNlpHcanM7bZhJq2t1QUI5ImYQQqq65T\r\n1QIDAQAB\r\n-----END PUBLIC KEY-----\r
\n",

  "salt": "±\u0005ædÛˆ\u0007â†|Y,cŸô,ÂìÏæÀ` MÜ„Ù\n_·¬",

  "stats": {

    "created": "2014-06-07 16:34:00 UTC"

  },

  "username": "test"

}
```


3.2. API

En cuanto a la API, su diseño se deriva de una evolución del patrón MVC en el que no hay vistas, puesto que la API actúa como una interfaz que ofrece una serie de métodos al cliente y es el cliente el encargado de actualizar el estado de las vistas en cada momento.

La forma concreta en que está diseñada esta API puede ilustrarse como sigue (*figura 11*):

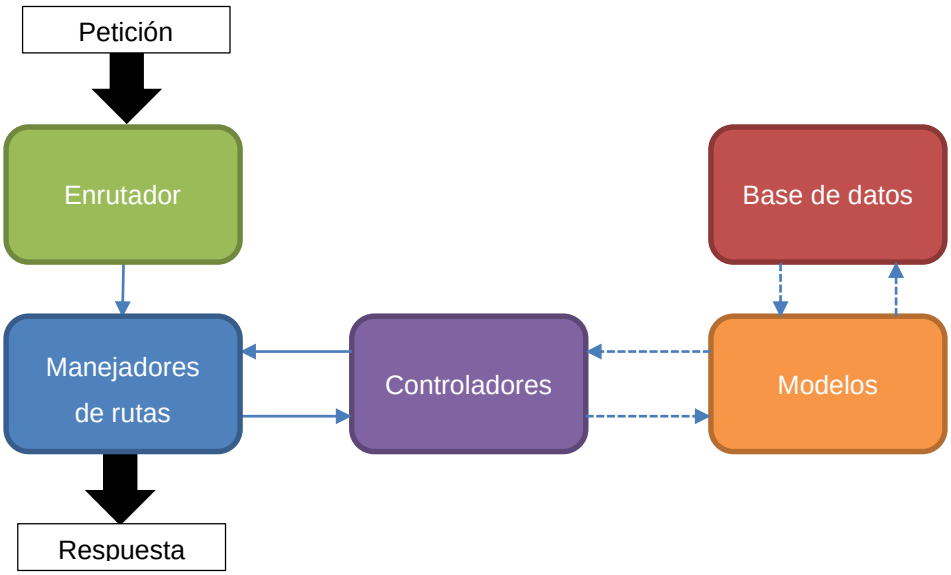


Figura 11. Diseño de la API

El proceso se describe como sigue:

1. Un cliente realiza una petición a la API. Esta petición llega al enrutador que se encarga de enviársela al manejador de rutas adecuado.
2. El manejador de rutas conoce los controladores existentes en la API y se encarga de traducir los parámetros de la petición a los parámetros requeridos por el controlador y llamarlo.
3. El controlador ejecuta tareas concretas utilizando instancias de los modelos, que no son más que representaciones de los datos a nivel de lenguaje de programación y envía resultados de vuelta al manejador de rutas.
4. El manejador de rutas se encarga de generar una respuesta web y enviársela al cliente.

3.3. Servidor web

El servidor web es el encargado de recibir todas las peticiones web, tanto de API como de páginas web.

Desempeña, pues, dos tareas sencillas:

1. Si una petición va dirigida a la API, tiene que hacerla llegar a la misma y será esta última quien se encargue de generar una respuesta.
2. Debe también servir el contenido web correspondiente con las páginas estáticas de información y, además, la aplicación web en sí. La aplicación web no es más que una página web HTML que incluye ficheros Javascript en los que se define toda la lógica de la misma.

3.4. Aplicación web

La aplicación web también está diseñada utilizando una variación del esquema MVC, conocida como MV* (model-view-asterisk) [1]. Este patrón de diseño puede ilustrarse con el siguiente esquema (*figura 12*):

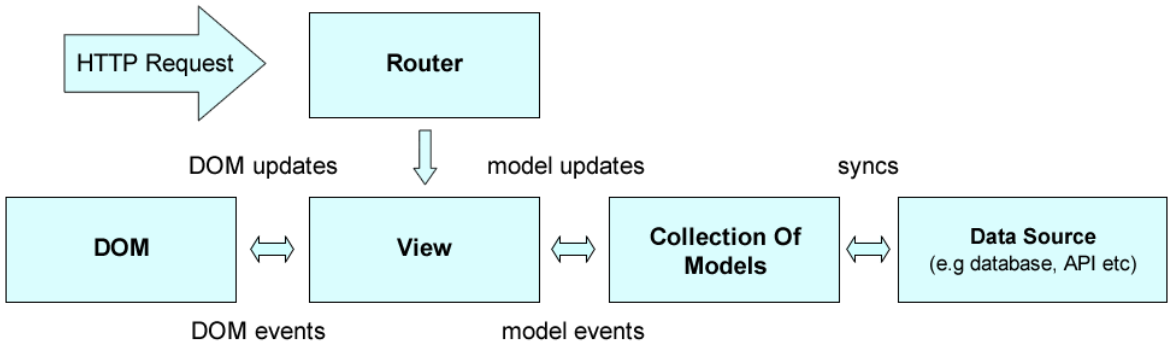


Figura 12. MV* en Backbone. Esquema extraído de [1]

Como puede observarse en el esquema, el elemento central es la vista (View). Las vistas se encargan de:

- Trabajar directamente sobre los modelos, que no son más que representaciones estándar de los datos que la aplicación recibe del servidor.
- Escuchar a eventos que suceden en el DOM [9] y lanzar los manejadores adecuados.
- Actualizar la página cuando se producen peticiones HTTP (navegación).

La labor principal del enrutador (Router) es la de notificar a las vistas cuándo se ha producido navegación en la página.

En cuanto a los modelos, además de representar la información, ofrecen una serie de métodos con los que actualizar sus contrapartidas en el lado de servidor cuando se producen cambios en los mismos.

4. Diseño y experiencia de usuario

Los principios básicos de diseño que se han utilizado para desarrollar la interfaz de Secretify son los siguientes:

1. Minimalismo. El diseño debe ser simple y no ofrecer al usuario demasiadas opciones a primera vista.
2. Convenciones. Se utilizan iconos estándar para la representación de acciones (búsqueda, sincronización, etc.).
3. Usabilidad. Las opciones que ofrece la aplicación deben ser claras y no llevar al usuario a dudas sobre para qué sirve cada elemento de la interfaz.

Bocetos, croquis, modelos, etc., creados durante el proceso de trabajo, incluyendo especialmente:

4.1. Disposición del contenido

El siguiente esquema ilustra cuál es el reparto de espacio de las distintas componentes de la aplicación (*figura 13*):

Figura 13. Disposición del contenido.

En esta disposición, cada elemento tiene una función:

- La barra de navegación superior permite al usuario acceder a opciones de su cuenta y volver al inicio haciendo clic sobre el logo de la aplicación.
- La barra de navegación lateral ofrece al usuario las opciones de navegación de la aplicación y muestra dónde se encuentra actualmente.
- El contenido varía según la sección que se esté mostrando en ese momento.

4.2. Recursos visuales

El diseño de Secretify se compone de tres piezas fundamentales:

1. Bootstrap [6] como base; framework CSS y Javascript.
2. Font Awesome [14]. Librería CSS que incluye multitud de iconos reescalables representados como una fuente.
3. Google Fonts [18]. API de Google con la que utilizar tipografías públicamente disponibles en el diseño.

4.3. Iteraciones sobre el diseño

De cara a maximizar la velocidad de desarrollo, en el transcurso de este proyecto se ha optado por la implementación de prototipos HTML utilizando los componentes que se creaban durante el proceso de desarrollo. A continuación, se ha iterado sobre los bloques básicos de la interfaz para refinar el diseño y obtener mejores resultados.

En el Anexo B se pueden observar los cambios producidos entre las dos iteraciones ejecutadas durante el desarrollo del proyecto.

4.4. Experiencia de usuario

Las buenas prácticas que se han empleado para proporcionar la mejor experiencia de usuario posible son las siguientes:

1. Utilización del color y los contrastes para transmitir mensajes al usuario, como por ejemplo en el resaltado de la sección activa (*figura 14*).

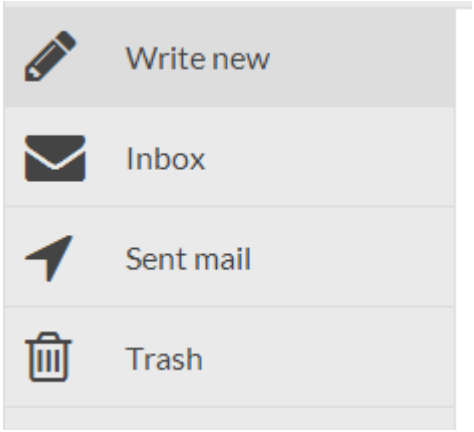


Figura 14. Resaltado de sección activa.

2. Uso de iconos estandarizados para acciones; mensajes borrados corresponde con papelera, escribir nuevo mensaje con un lápiz, etc. (ver *figura 14*)
3. Muestra de “spinners” (iconos de carga mientras se está realizando alguna acción), como por ejemplo, mientras se sincroniza el correo del usuario (*figura 15)*.

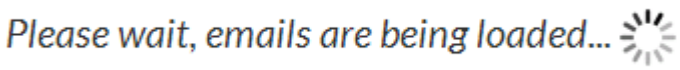


Figura 15. Ejemplo de "spinner".

4. URLs descriptivas en cada página y navegación por URL, además de haciendo clic en los botones de la barra de navegación.

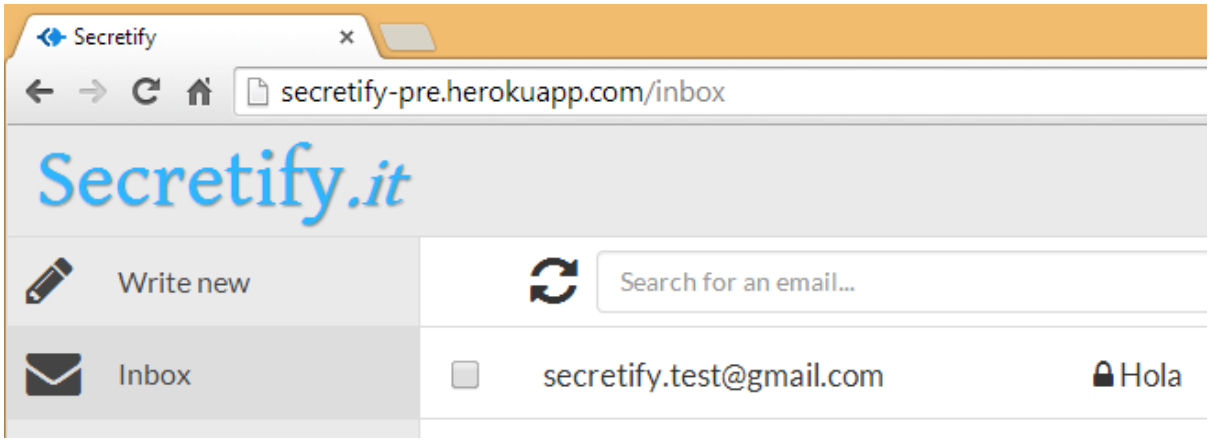


Figura 16. Ejemplo de URL para la sección "Inbox"

5. Disposición del contenido coherente entre secciones. Las barras de navegación se mantienen inmutables entre distintas secciones, sólo varía la sección destacada en caso de que sea necesario.

5. Detalles de implementación

La solución se ha implementado utilizando tecnología Javascript “full-stack”, es decir, tanto el lado del servidor como el del cliente utilizan frameworks y tecnologías basadas en Javascript. Esta elección se ha debido, principalmente, a dos razones:

1. Experiencia previa en el desarrollo de aplicaciones utilizando Javascript como lenguaje de programación principal.
2. Escalabilidad del sistema gracias a utilizar Node.JS [36] como lenguaje de servidor.

A continuación, se especifican las opciones utilizadas tanto a nivel de servidor como de cliente.

Servidor

Secretify cuenta con una base de datos no relacional MongoDB [33].

El servidor o backend de la aplicación está implementado utilizando Node.JS como tecnología principal. Los frameworks y librerías más importantes de los que hace uso son:

1. Express [13]. Framework inspirado en Sinatra que provee una serie de características para la construcción de aplicaciones web en Node.js.
2. Mongoose [34]. Librería de modelado de objetos MongoDB en Node.js. Es el encargado de gestionar el acceso a los datos y su creación, actualización y borrado.

Cliente

En cuanto a la aplicación web, se trata de una aplicación single-page implementada utilizando Javascript. Las librerías de las que hace uso son:

1. Backbone.js [5]. Backbone.js es una librería que provee a las aplicaciones web de una forma de estructurarse basada en el patrón de diseño MVC.
2. jQuery [25]. Librería de utilidades para manejo del DOM.
3. Lodash [28]. Librería de funciones de utilidad para manejar colecciones y objetos.
4. Handlebars [20]. Lenguaje de plantillas que permite pre-compilar plantillas para utilizar en tiempo de ejecución.
5. Forge [15]. Librería de métodos de cifrado (simétrico y asimétrico), funciones de hash y otras utilidades.

6. Tests

Como se introdujo en la sección “Metodología y proceso de trabajo” del capítulo 1, el desarrollo de Secretify ha seguido el proceso de Test Driven Development. En este apartado se describe cómo se han desarrollado dichos tests.

Debido a las limitaciones temporales, es necesario establecer un compromiso entre cobertura de los tests y velocidad de desarrollo, que permita garantizar la calidad del producto pero no repercuta de forma drástica en el tiempo de desarrollo. Por ello, las pruebas implementadas se han orientado a:

1. Testeo de modelos y datos en el servidor; tests unitarios.
2. Testeo de métodos que operan sobre los modelos en el servidor; tests funcionales.
3. Testeo de rutas de la API.
4. Testeo de métodos de cifrado en el cliente.
5. Testeo módulos particulares en el cliente.

En cuanto a los detalles de implementación, las tecnologías empleadas para los tests han sido las siguientes:

1. Framework de test: Mocha [31].
2. Librería de asserts: Chai.js [8].
3. Automatización de tests: IntelliJ IDEA y Grunt.

Tipo de test	Número de tests
Tests unitarios en la API	14
Tests funcionales en la API	24
Tests de rutas en la API	23
Tests unitarios de métodos de cifrado en el cliente	17
Tests unitarios de otros módulos de cliente	24

Tabla 10. Desglose de tests por sección.

7. Herramientas

7.1. Trello

Para gestionar de forma cómoda las tareas a ejecutar, se ha optado por la utilización de Trello [45]; una herramienta que permite representar tareas a modo de tablón de Kanban, en el que puede diferenciarse entre tareas pendientes, en proceso y terminadas.

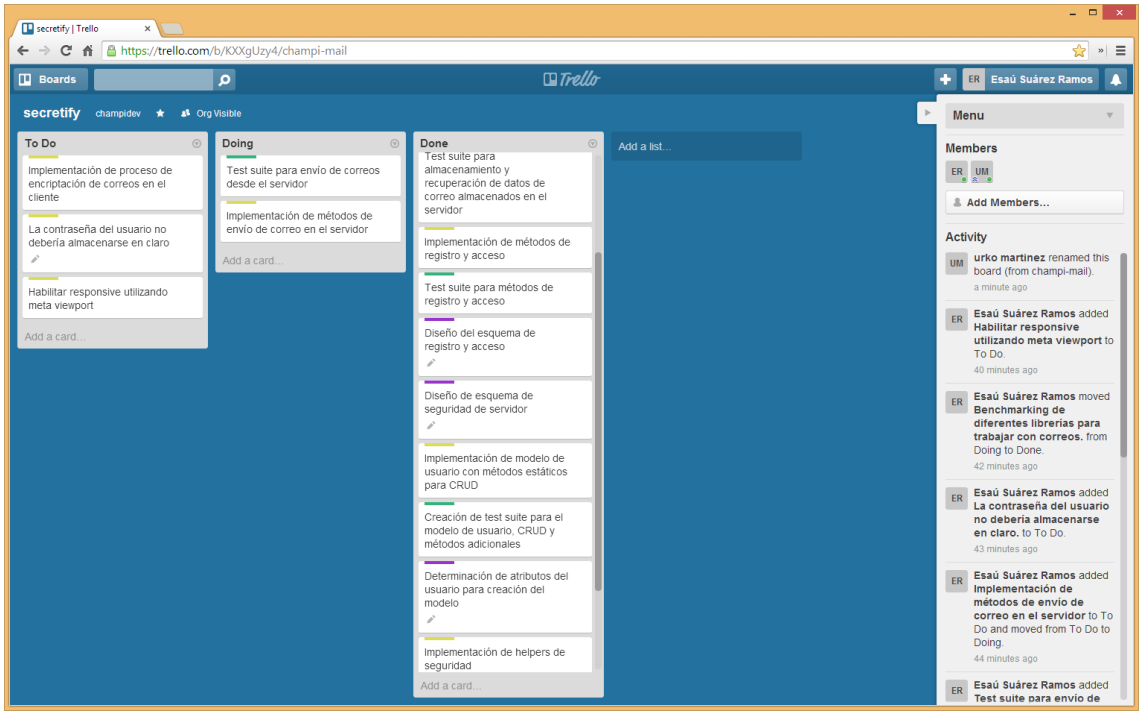


Figura 17. Tareas en Trello.

7.2. Github

El sistema de gestión de versiones empleado para almacenar el código es Git [16] y la plataforma elegida para su alojamiento en la nube Github [17].

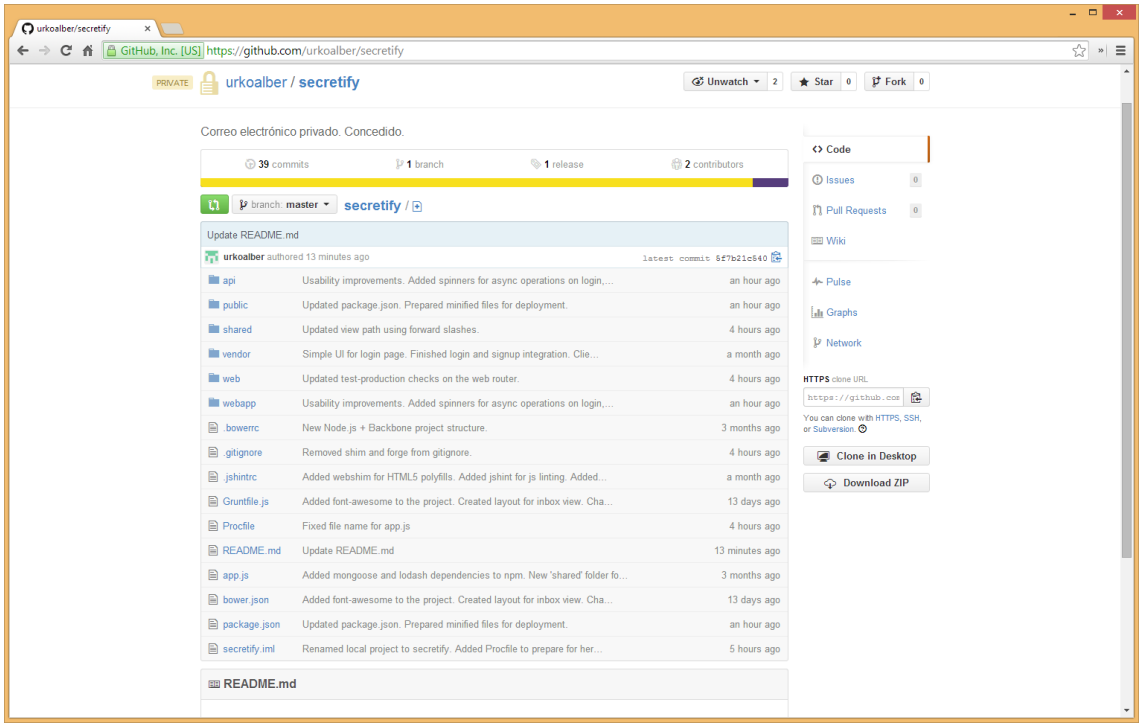


Figura 18. Código fuente en GitHub.

7.3. IntelliJ IDEA

Como herramienta indispensable para la productividad, se utiliza el entorno integrado de desarrollo IntelliJ IDEA [24].

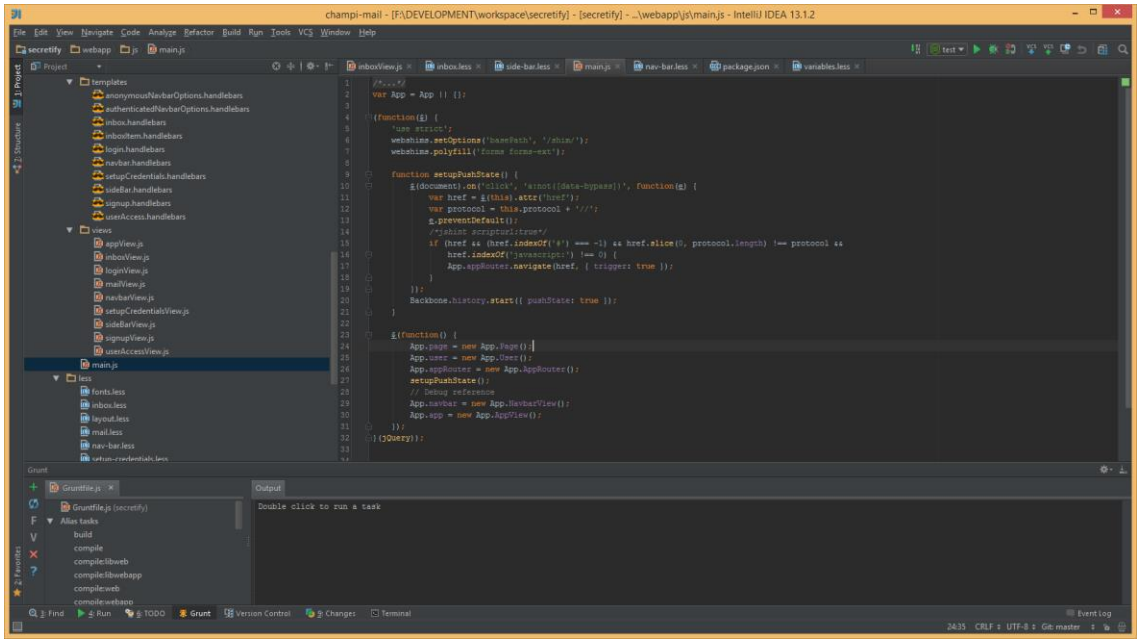


Figura 19. Proyecto en IntelliJ IDEA.

7.4. Chrome DevTools

Las herramientas de desarrollador de Chrome permiten depurar el código Javascript en tiempo de ejecución.

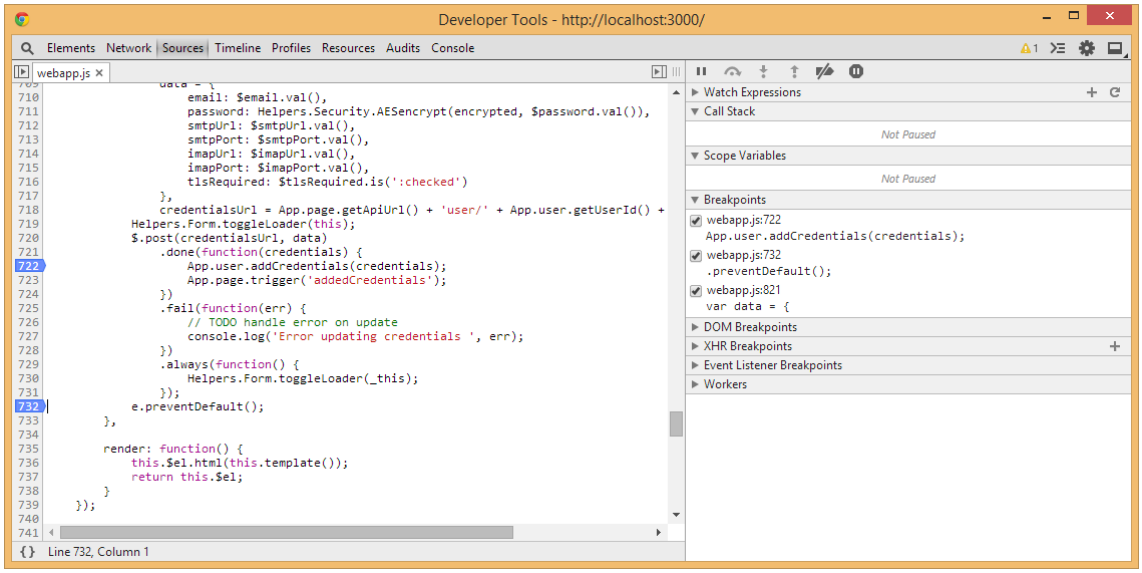


Figura 20. Depuración con Chrome DevTools.

7.5. TooloMongo

Para gestionar las bases de datos locales en MongoDB se utiliza TooloMongo [44], una aplicación simple que permite editar y eliminar documentos de las distintas colecciones existentes en una base de datos no relacional MongoDB.

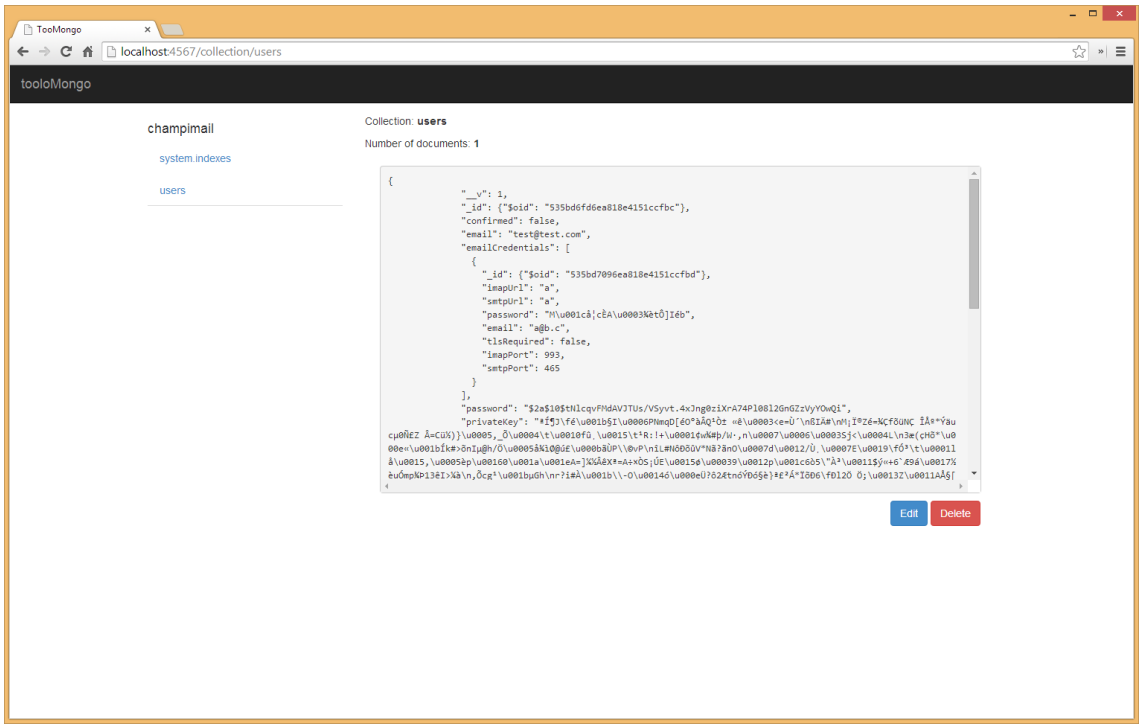


Figura 21. Muestra de datos en TooloMongo.

7.6. Heroku

Los resultados del desarrollo son accesibles en la URL http://secretify-pre.herokuapp.com/. La aplicación ha sido desplegada como una aplicación de Heroku [22], ubicada en Europa y que utiliza una cuenta sandbox del proveedor de bases de datos MongoHQ.

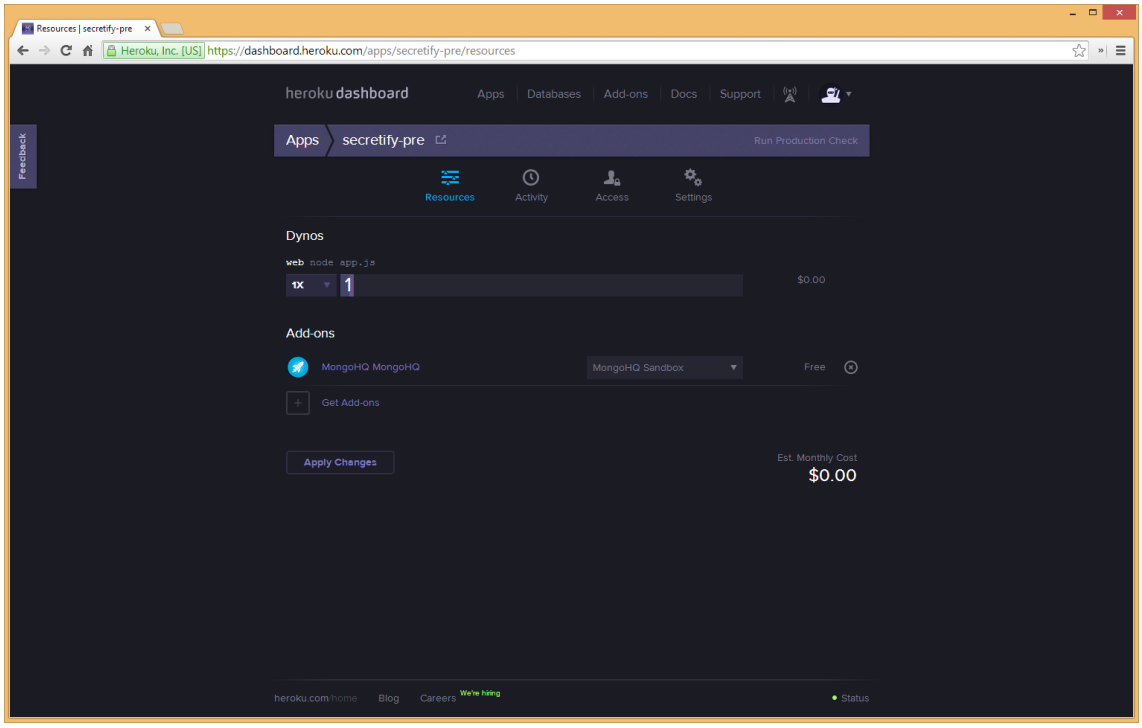


Figura 22. Aplicación en Heroku Europe.

7.7. Grunt

- Se ha utilizado Grunt.js [19] como herramienta de automatización de tareas repetitivas. En concreto, gracias a Grunt se han automatizado las tareas de:
- Compilación de LESS.
- Compilación de plantillas Handlebars.
- Concatenación de librerías Javascript.
- Concatenación de ficheros fuente de Javascript.
- Minificación de código Javascript para producción.
- Minificación de código CSS.
- “Watch”: monitorización de ficheros; si hay cambios se produce una compilación para reflejarlos en el navegador.
- Ejecución de tests.

Capítulo 5: Demostración

1. Despliegue en entorno de pruebas

De cara a ofrecer a los posibles interesados en el proyecto una muestra de las funcionalidades implementadas hasta el momento, se ha optado por el despliegue en un entorno de pre-producción del prototipo implementado en el transcurso de este proyecto.

La plataforma elegida es la descrita en el apartado 6 del capítulo anterior; Heroku. La URL de acceso a esta versión de prueba de Secretify es http://secretify-pre.herokuapp.com.

2. Cuentas de prueba

Con el objetivo de mostrar las funcionalidades implementadas en este prototipo, se ha optado por crear una serie de cuentas de usuario de prueba y una guía paso a paso en PowerPoint. Siguiendo este proceso se podrán comprobar las características actuales del producto.

Por una parte, se han creado dos cuentas de usuario en Gmail para disponer de cuentas de correo electrónico ordinarias. Estas cuentas son:

- Dirección: secretify.test@gmail.com

  Contraseña: test.secretify

- Dirección: secretify2.test@gmail.com

  Contraseña: test.secretify

Por otra parte, se han creado dos cuentas de usuario en Secretify y se han vinculado con estas cuentas de correo electrónico ordinario. Las credenciales son las siguientes:

- Nombre de usuario: sandbox1

  Dirección de correo: sandbox1@secretify.com

  Contraseña: sandbox

- Nombre de usuario: sandbox2

  Dirección de correo: sandbox2@secretify.com

  Contraseña: sandbox

3. Guía en formato presentación

En el directorio de anexos incluido junto a esta memoria, puede consultarse el documento PowerPoint GUIA_SECRETIFY.pptx en el que se explica cómo utilizar la aplicación (también se ha adjuntado una versión PDF por temas de compatibilidad y software). Por favor, remitirse a él para comprobar las instrucciones de uso y ejecutar las verificaciones pertinentes.

Capítulo 6: Conclusiones y líneas de futuro

1. Conclusiones

Este trabajo de fin de máster ha supuesto un gran desafío, tanto a nivel personal, puesto que ha sido necesario compaginar un trabajo a jornada completa con el desempeño académico del segundo semestre completo, como a nivel profesional, ya que ha implicado la formación en aspectos de seguridad informática y el descubrimiento y utilización de nuevas librerías de soporte en el desarrollo.

Como motivación y aliciente especial para el desarrollo de este proyecto, he podido sentar las bases para la creación de una plataforma que cubre una necesidad real y que, en mi opinión, podrá ofrecer muchas posibilidades de negocio en el corto y medio plazo.

Puesto que mi perfil profesional está más orientado al desarrollo que al diseño, las experiencias más enriquecedoras de este trabajo han sido: por una parte, el diseño de un algoritmo de cifrado cliente a cliente sin puntos vulnerables a ataques en el transporte y, por otra, el diseño de una interfaz usable que garantice una buena experiencia de usuario.

Los objetivos iniciales expuestos al comienzo de esta memoria han podido cumplirse y se ha obtenido un prototipo estable que sirve como demostración del potencial de la aplicación Secretify. Con este prototipo se ilustra, de forma bastante realista, cómo funciona el algoritmo de cifrado de Secretify y por qué representa una herramienta valiosa a la hora de proteger la información personal. Asimismo, deja entrever cuál será el estilo a seguir por la aplicación definitiva en cuanto a diseño y experiencia de usuario.

Respecto a la planificación, el proceso de estimación de tareas ha demostrado ser bastante preciso y se ha podido completar las tareas recogidas en el diagrama de Gantt en un tiempo inferior al propuesto inicialmente, por lo que el tiempo adicional del que se disponía pudo dedicarse a la redacción de un plan de negocios, vital para dar a conocer el producto entre posibles inversores y/o interesados en la plataforma.

Finalmente, y puesto que, al fin y al cabo, este proyecto ha consistido principalmente en el desarrollo de software, me gustaría recalcar que, gracias al trabajo constante y al desarrollo utilizando una metodología orientada a tests, el prototipo final consta de una buena calidad de código y la deuda técnica derivada de su creación se encuentra en un nivel aceptable y manejable.

2. Líneas de futuro

Los siguientes pasos a realizar para completar Secretify pasan por la implementación de funcionalidades estándar de clientes de correo, así como una serie de acciones que agreguen valor a la plataforma. Las principales ramas a seguir son las siguientes:

1. Añadir funcionalidades de cliente de correo:
    a. Eliminación de correos.
    b. Revisión de mensajes enviados.
    c. Sincronización de más cuentas de correo electrónico.
    d. Actualización de contraseña.
    e. Etiquetado y organización.
2. Crear un dominio de correos @secretify para los usuarios de la aplicación, de modo que cuando se registren dispongan de una cuenta con la que hacer pruebas en la plataforma. El objetivo de esta acción es reducir la desconfianza del usuario a proveer sus credenciales de correo.
3. Añadir distintos niveles de seguridad que puedan ser comercializados como un extra o un plan pro del producto.
4. Redactar un plan de marketing y ejecutar acciones concretas de promoción de la plataforma una vez se cuente con un producto estable.

Bibliografía

[1] Addy Osmani, Developing Backbone.js Applications, O'Reilly, 2013.

[2] A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. R.L. Rivest, A. Shamir, and L. Adleman. http://people.csail.mit.edu/rivest/Rsapaper.pdf, consultado 31/01/2014.

[3] Announcing the Advanced Encryption Standard (AES). http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, consultado 31/01/2014.

[4] Application Programming Interface (API): http://www.3scale.net/wp-content/uploads/2012/06/What-is-an-API-1.0.pdf, consultado 01/02/2014.

[5] Backbone.js. http://backbonejs.org/, consultado 05/02/2014.

[6] Bootstrap. http://getbootstrap.com, consultado 05/02/2014.

[7] Caso Ciberespionaje EE.UU., El País. http://elpais.com/tag/caso_ciberespionaje_eeuu/a, consultado 31/01/2014.

[8] Chai.js. http://chaijs.com/, consultado 07/02/2014.

[9] DOM. http://www.w3.org/DOM/, consultado 05/02/2014.

[10] Email Client Market Share: http://emailclientmarketshare.com/, consultado 02/02/2014.

[11] Email Statistics Report, 2014-2018. Sara Radicati, The Radicati Group, INC. http://www.radicati.com/wp/wp-content/uploads/2014/04/Email-Statistics-Report-2014-2018-Executive-Summary.pdf, consultado 23/05/2014.

[12] ¿Es seguro hablar por Whatsapp? Armando Hueso. http://www.socialunderground.co/se-lo-que-dices-por-whatsapp/, consultado 12/02/2014.

[13] Express. http://expressjs.com/, consultado 12/02/2014.

[14] Font Awesome. http://fortawesome.github.io/Font-Awesome/, consultado 15/03/2014.

[15] Forge. https://github.com/digitalbazaar/forge, consultado 16/02/2014.

[16] Git. http://git-scm.com/, consultado 05/02/2014.

[17] GitHub. https://github.com/, consultado 05/02/2014.

[18] Google Fonts. https://www.google.com/fonts, consultado 15/03/2014.

[19] Grunt: http://gruntjs.com/, consultado 20/02/2014.

[20] Handlebars. http://handlebarsjs.com/, consultado 20/02/2014.

[21] Here Are Some Realistic Revenue Numbers For Whatsapp. Jim Edwards. http://www.businessinsider.com/here-are-some-realistic-revenue-numbers-for-whatsapp-2014-2, consultado 31/01/2014.

[22] Heroku: https://www.heroku.com/, consultado 20/03/2014.

[23] Horde Webmail: http://www.horde.org/apps/webmail, consultado 25/02/2014.

[24] IntelliJ IDEA. http://www.jetbrains.com/idea/, consultado 12/02/2014.

[25] jQuery. http://jquery.com/, consultado 10/02/2014.

[26] JSON. http://json.org/, consultado 10/02/2014.

[27] Lockify: https://lockify.com/, consultado 25/02/2014.

[28] Lodash. http://lodash.com/, consultado 25/02/2014.

[29] Mailpile: https://www.mailpile.is/, consultado 25/02/2014.

[30] Microsoft Outlook: http://office.microsoft.com/es-es/outlook-software-de-correo-electronico-y-calendario-FX010048775.aspx, consultado 25/02/2014.

[31] Mocha. http://visionmedia.github.io/mocha/, consultado 07/02/2014.

[32] Model-View-Controller: A Design Pattern for Software. https://ist.berkeley.edu/as-ag/pub/pdf/mvc-seminar.pdf, consultado 15/02/2014.

[33] MongoDB. http://www.mongodb.org/, consultado 15/02/2014.

[34] Mongoose. http://mongoosejs.com/, consultado 15/02/2014.

[35] Mozilla Thunderbird: http://www.mozilla.org/es-ES/thunderbird/, consultado 25/02/2014.

[36] Node.js. http://nodejs.org/, consultado 12/02/2014.

[37] OAUTH. http://oauth.net/, consultado 12/02/2014.

[38] OpenSource Definition, Open Source. http://opensource.org/osd, consultado 12/02/2014.

[39] Protección de Datos multa a Google con 900.000€ por infracciones graves, J.M.Sánchez, Madrid, 8/01/2014. http://www.abc.es/tecnologia/redes/20131219/abci-google-multa-aepd-201312191217.html, consultado 12/02/2014.

[40] RFC 2818, HTTP over TLS. https://tools.ietf.org/html/rfc2818, consultado 20/02/2014.

[41] Red Green Refactor cycle, Goyello Blog. http://blog.goyello.com/2011/09/13/red-green-refactor-cycle/, consultado 27/02/2014.

[42] SquirrelMail: http://squirrelmail.org/, consultado 25/02/2014.

[43] Test Driven Development, Agile Data. http://www.agiledata.org/essays/tdd.html, consultado 25/02/2014.

[44] TooloMongo. https://github.com/Toolo/tooloMongo, consultado 15/02/2014.

[45] Trello. https://trello.com/, consultado 31/01/2014.

[46] What is User Centered Design, Usability Professionals. http://www.usabilityprofessionals.org/usability_resources/about_usability/what_is_ucd.html, consultado 15/03/2014.

Anexos

Anexo A: Entregables del proyecto

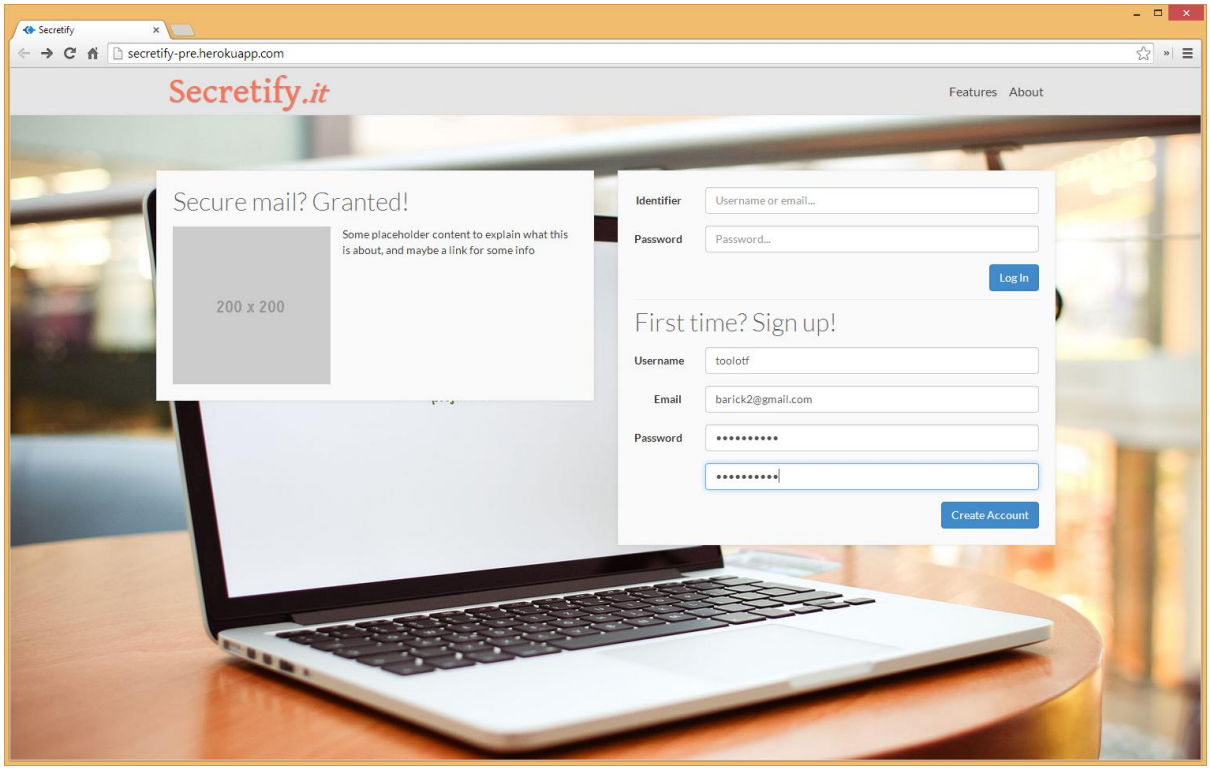
Lista de archivos entregados:

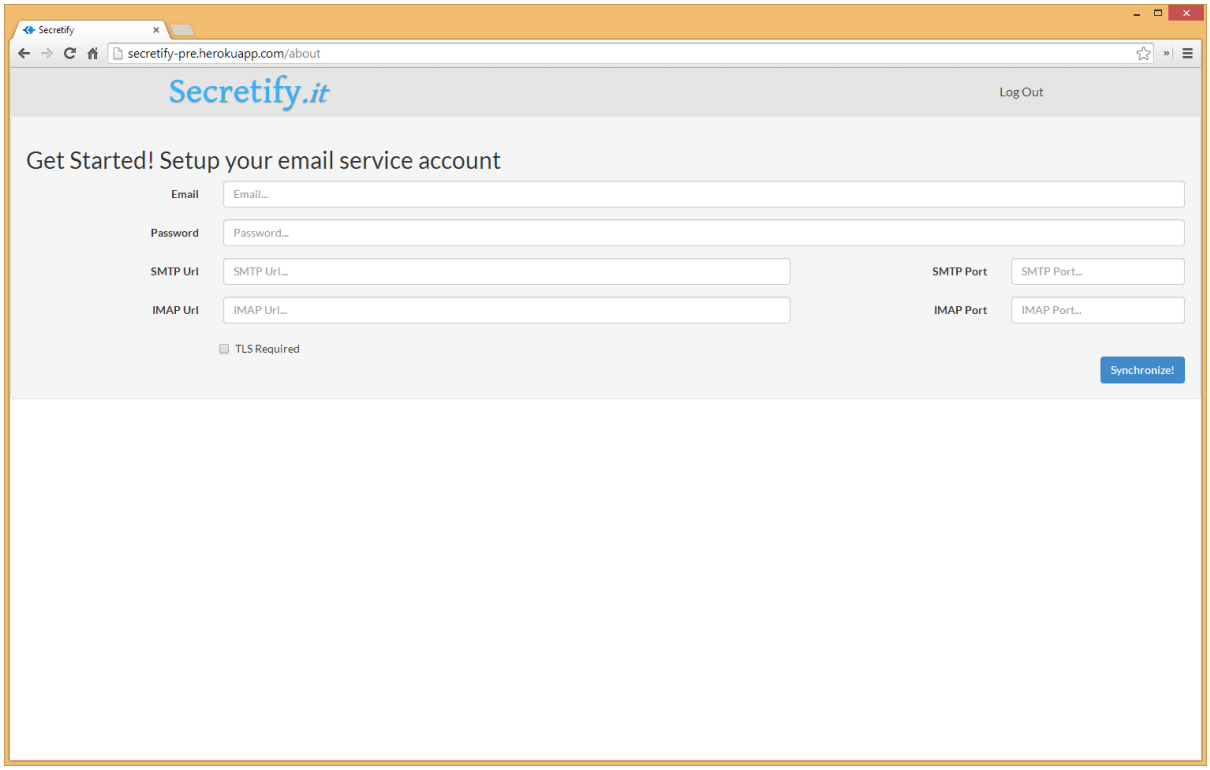
1. GUIA SECRETIFY.pptx. Presentación en formato Power Point con instrucciones de uso para el prototipo funcional de Secretify alojado en Heroku.
2. GUIA SECRETIFY.pdf. Versión de compatibilidad de las instrucciones de uso.
3. PLAN DE NEGOCIO.pdf. Versión independiente del plan de negocio incluido en esta memoria.
4. SECRETIFY.zip. Código del proyecto, incluye código fuente y archivos preparados para despliegue.

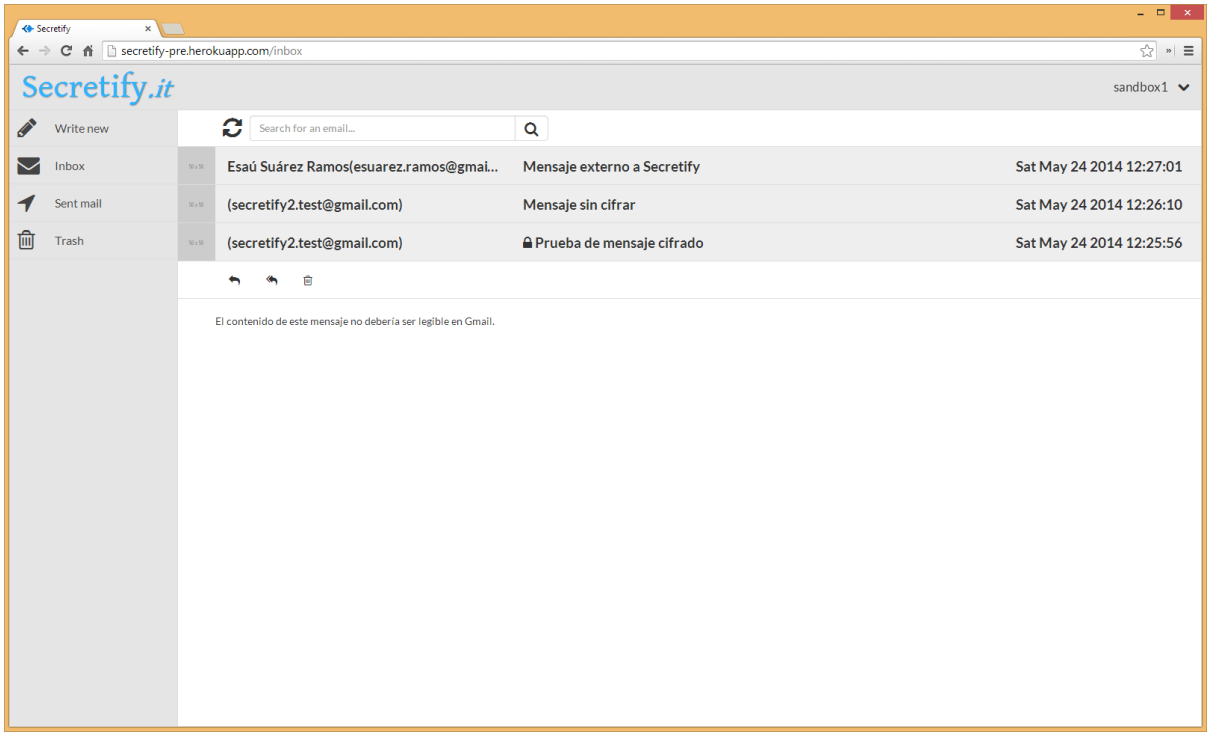
Anexo B: Capturas de pantalla

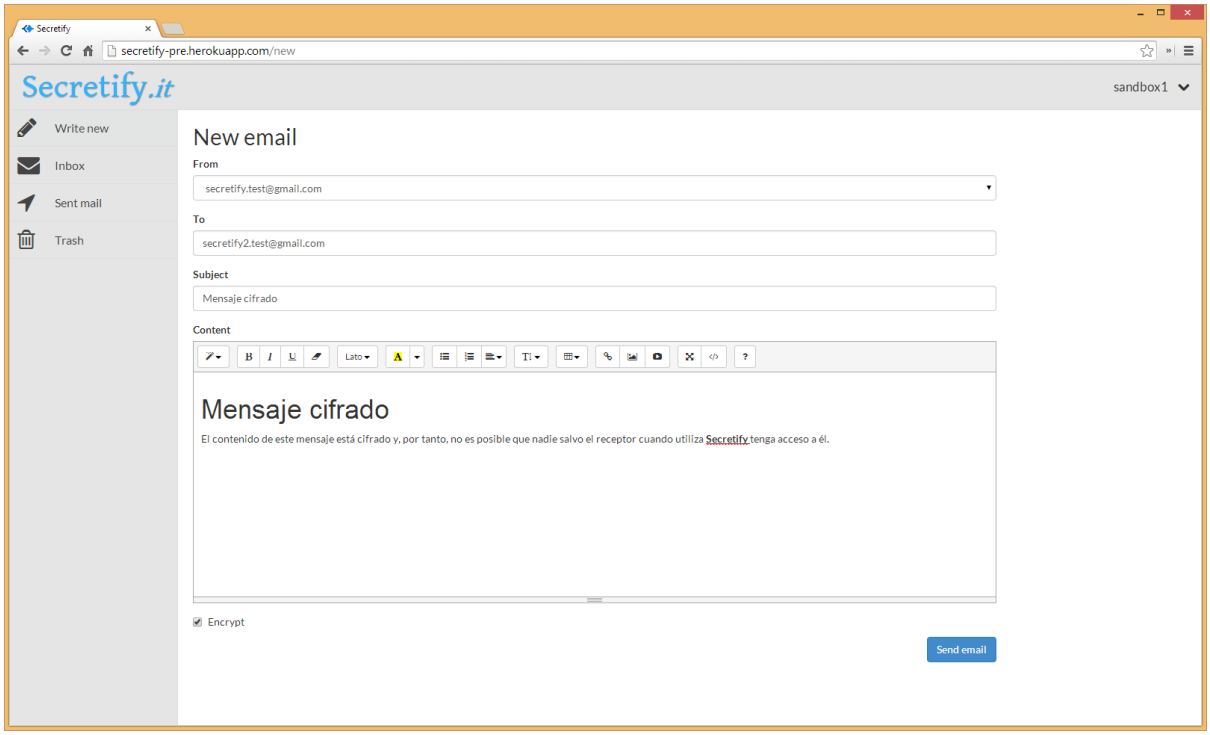
En este anexo se muestran capturas de pantalla de las dos iteraciones que se han realizado sobre el diseño.

Primera iteración

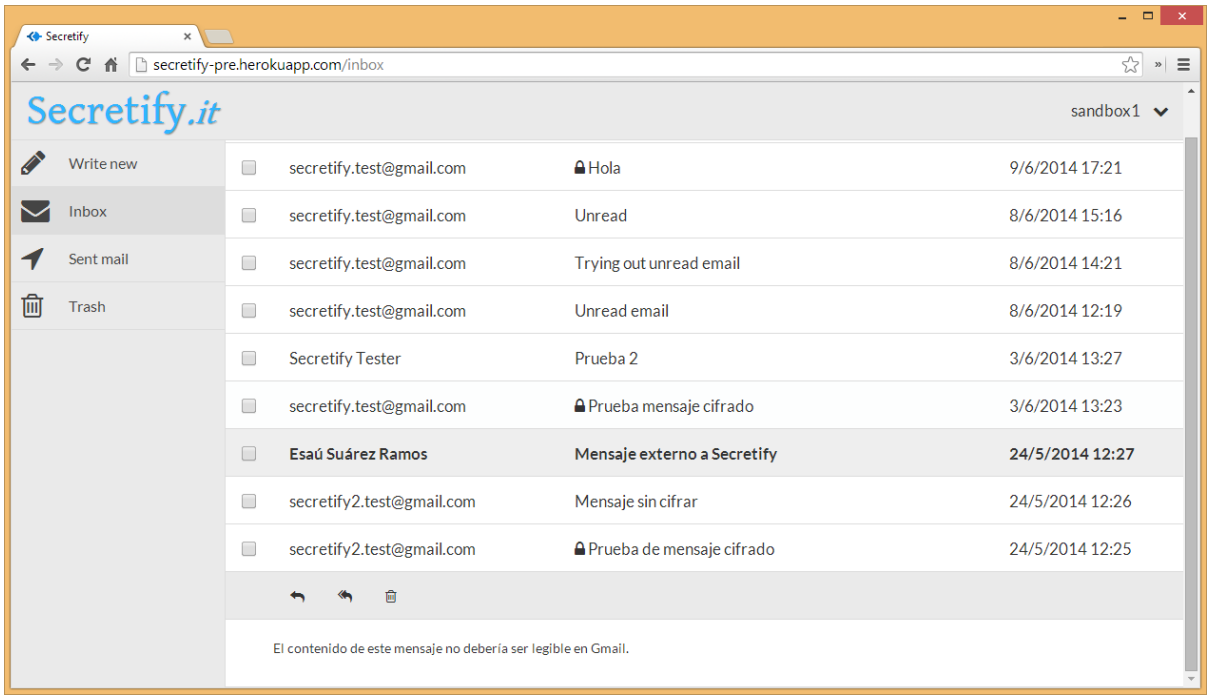


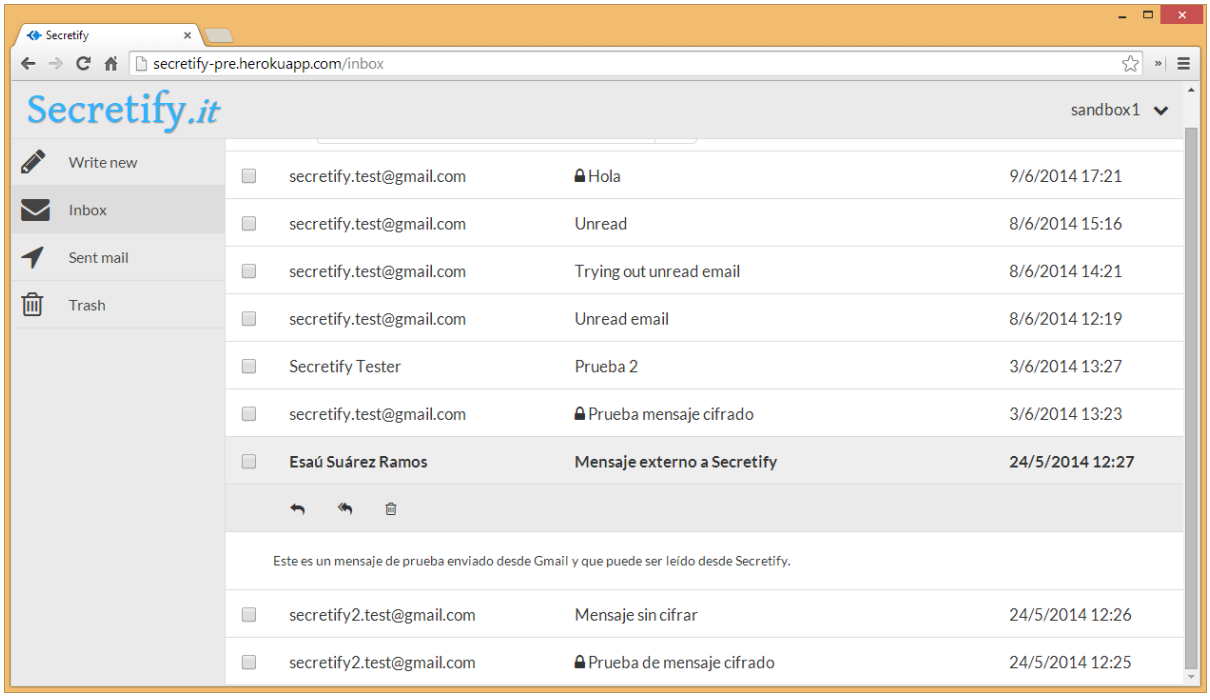


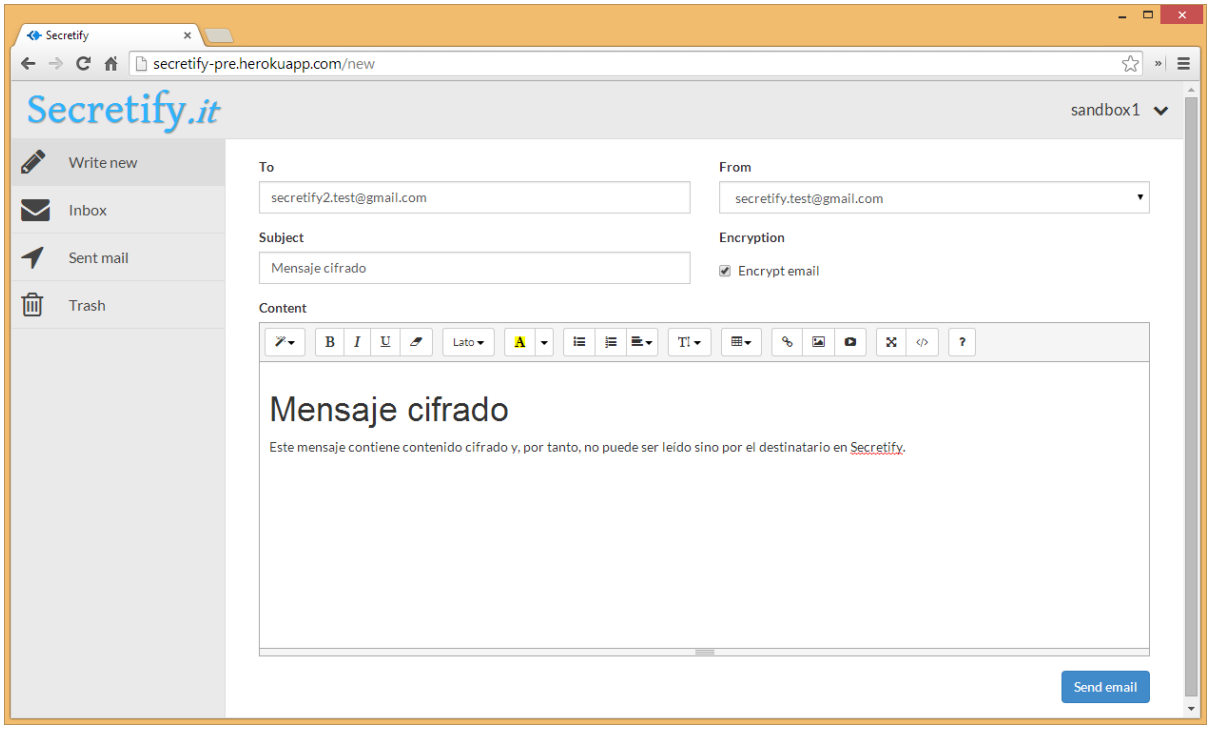

Secretify
secretify-pre.herokuapp.com/inbox
Secretify.it
sandbox1
Write new
Inbox
Sent mail
Trash
Search for an email...
Esaú Suárez Ramos(esuarez.ramos@gmai...
Mensaje externo a Secretify
Sat May 24 2014 12:27:01
(secretify2.test@gmail.com)
Mensaje sin cifrar
Sat May 24 2014 12:26:10
(secretify2.test@gmail.com)
Prueba de mensaje cifrado
Sat May 24 2014 12:25:56
El contenido de este mensaje no debería ser legible en Gmail.


Secretify
secretify-pre.herokuapp.com/new
Secretify.it
sandbox1
Write new
Inbox
Sent mail
Trash
New email
From
secretify.test@gmail.com
To
secretify2.test@gmail.com
Subject
Mensaje cifrado
Content
Mensaje cifrado
El contenido de este mensaje está cifrado y, por tanto, no es posible que nadie salvo el receptor cuando utiliza Secretify tenga acceso a él.
Encrypt
Send email

Segunda iteración






Secretify
secretify-pre.herokuapp.com/new
Secretify.it
sandbox1
Write new
Inbox
Sent mail
Trash
To
secretify2.test@gmail.com
From
secretify.test@gmail.com
Subject
Mensaje cifrado
Encryption
Encrypt email
Content
Lato
Mensaje cifrado
Este mensaje contiene contenido cifrado y, por tanto, no puede ser leído sino por el destinatario en Secretify.
Send email

Anexo C: Currículum Vitae

Esaú Suárez Ramos

Tenerife, 1991

Graduado en ingeniería informática en el año 2013 por la Universidad de La Laguna, obteniendo el premio extraordinario de fin de grado al mejor expediente académico.

Mis características definitorias son la curiosidad y la proactividad, necesito sentirme productivo en todo momento y, por ello, siempre estoy aprendiendo nuevas técnicas y lenguajes de desarrollo. Mi perfil profesional se centra en el desarrollo de software, pero dispongo de otras capacidades complementarias que me permiten desempeñar labores como diseño y marketing.

Actualmente, estoy empleado en una empresa de base tecnológica ubicada en Tenerife, Langproving, y mi trabajo consiste principalmente en el desarrollo web utilizando tecnologías modernas e innovadoras y en el desarrollo móvil nativo, aunque también desempeño labores de diseño y marketing online.