Como ya hemos visto en el esquema anterior necesitamos hacer uso de las herramientas adecuadas para poder desarrollar este esquema de cifrado. Las herramientas que analizaremos son las siguientes:

1. Cliente de correo electrónico

Un cliente de correo electrónico es necesario para el desarrollo de este proyecto ya que en él se instalara un complemento que se encargara del cifrado de mensajes, envío de CAPTCHAS y descifrado de los mensajes de correo electrónico. Para ello buscamos un cliente de correo electrónico que cuente con el soporte de los protocolos POP3, SMTP y IMAP; sus licencias son de código libre; soporte la instalación de APIs externas; y tenga soporte en los sistemas operativos Windows, IOS y Linux. Por lo tanto se investigaron los siguientes clientes de correo electrónico que se encuentra en el mercado:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cliente de correo electrónico | Sistema Operativo | Protocolos soportados | Código Libre | Agregar funcionalidad | Extra | Gratuita o de paga |
| eM client | Windows 7, 8 & 10 ; IOS | POP3, SMTP, IMAP, EWS, AirSyn | NO | NO | 100% compatible con gmail y sus APIs | Ambos |
| Postbox | Windows, IOS | POP3, SMTP, IMAP | NO | SI por medio de APIs | Sincronización con Dropbox, OneDrive, Facebook y Twitter | Ambos |
| Zimbra | Windows, IOS & Linux | POP3, SMTP, IMAP | SI | SI por medio de APIs | Una plataforma de nivel empresarial y capas se soportar sincronización con múltiples servicios | Ambos |
| Opera Mail | Windows, IOS & Linux | POP3, SMTP, IMAP | SI | NO | La plataforma para desarrollar en Opera se actualiza cada semana | Gratuito |
| Thunderbird | Windows, IOS & Linux | POP3, SMTP, IMAP | SI | SI por medio de APIs | Cliente de correo versátil y fácilmente escalable y una comunicad de desarrollo bastante amplia | Gratuito |

* El cliente de correo electrónico eM client tiene una sincronización a 100% con las cuentas de gmail y sus APIs, cuenta con una versión gratuita y una versión de paga; puede hacer migración de mensajes de correo electrónico y contactos de diversos clientes de correo electrónico y tiene una compatibilidad con muchos servidores de correo electrónico. [1]

Su desventaja es que su código es cerrado y te permite agregar funcionalidades.

* El cliente de correo electrónico Postbox esta soportada en los sistemas operativos Windows 7 o posteriores y IOS, esta aplicación es generada por el servidor de correo electrónico postbox por lo tanto cuenta con una sincronización al 100% con este servidor, también soporta otros servidores de correo como gmail y Outlook; este cliente puede sincronizarse con dropbox, Onedive y mantenerte informado de tus redes sociales como Facebook, Twitter, entre otras. Puedes agregarle más funcionalidades con la instalación de APIs.

Una desventaja de esta aplicación es de código cerrado, pero gracias a que esta basado en código de mozilla puedes desarrollar APIs para agregarle tus propias funciones. [2]

* El cliente de correo electrónico zimbra es la aplicación más completa que analizamos, tiene compatibilidad con el servidor zimbra pero es capaz de soportar otros servidores de correo electrónico, se encuentra en los 3 sistemas para PC, Windows, IOS & Linux, te da la facilidad de agregarle funcionalidades por medio de la instalación de APIs y gracias a que su código es abierto se pueden programar tus propias funciones. Este cliente cuenta con la versión gratuita y la versión de paga. Una gran ventaja que tiene es que te puedes certificar en el desarrollo APIs para este cliente de correo electrónico. [3]

La única desventaja que encontramos en este cliente de correo es que la plataforma es demasiado grande y el tiempo que se necesita invertir al estudio del código es demasiado y el tiempo de desarrollo es este proyecto es muy corto.

* Opera mail es un cliente de correo electrónico que salió al mercado recientemente y se puede instalar en los sistemas operativos Windows, IOS & Linux, es capaz de comunicarse con diversos servidores de correo electrónico y su código es de libre acceso.

Su principal es que las funcionalidades que se quieran agregar no pueden ser adquiridas por medio de la instalación de complementos o APIs. [4]

* Por ultimo tenemos a Thunderbird que es un cliente de correo electrónico desarrollado por mozilla, este cliente es de código abierto y la instalación de APIs para agregar funcionalidad es demasiado sencilla; es un cliente de correo que puede ser instalado en los sistemas operativos Windows, IOS y Linux. [5]

Por lo tanto el cliente de correo electrónico que utilizaremos será Thunderbird al ser un cliente de correo electrónico casi tan completo como zimbra pero con la facilidad de desarrollar APIs más rápido.

1. Lenguajes de programación.

Uno de los objetivos que se tienen en este proyecto es generar un complemento para un cliente de correo electrónico por lo tanto al escoger a thunderbird como cliente tenemos que programar con el lenguaje que fue desarrollado para tener la mayor compatibilidad.

Para el desarrollo de este proyecto utilizaremos[5]:

* Lenguaje Python
* HTML 5
* CSS3

A pesar de ser una aplicación de escritorio este cliente de correo electrónico esta construido con lenguajes web.

1. Tipos de CAPTCHAS

En nuestro esquema de cifrado es necesario esconder la clave en un CAPTCHA para ser enviado al otro usuario y descifre el mensaje, pero existen varios tipos de CAPTCHAS que podemos utilizar[6].

Los CAPTCHAS se pueden clasificar de la siguiente forma:

* CAPTCHAS de texto: Este tipo de CAPTCHAS genera una pregunta sencilla donde la respuesta a la pregunta es la respuesta al CAPTCHA.
* CAPTCHAS de imágenes: Este tipo de CAPTCHAS nos muestran en una imagen una cadena de caracteres distorsionados, esta cadena de caracteres es la repuesta del CAPCHA transformada en una imagen.
* CAPTCHAS de audio: Este tipo de CAPTCHAS se caracterizan por ser un audio con ruidos de fondo, donde nos dirán la respuesta oculta.
* CAPTCHAS de video: Este tipo de CAPTCHAS nos muestran un video de unos cuantos segundos donde una palabra aparece alrededor de la pantalla, esta palabra es la respuesta al CAPTCHA.
* CAPTCHAS de acertijos: Este tipo de CPATCHAS es un diverso, ya que se trata de pequeños acertijos que resolver donde la respuesta no es una palabra si no una acción o serie de acciones. Los CAPTCHAS de acertijos pueden ser armar un rompecabezas pequeño, seleccionar la imagen que no corresponde, unir figurar geométricas, etc.

Para poder decidir qué tipo de CAPTCHAS utilizaremos se consideró los siguientes puntos:

* Facilidad de creación.
* Peso en byte del CAPTCHA.
* Forma del despliegue.
* Tipo de respuesta.

Por lo tanto nosotros necesitamos un tipo de CAPTCHA con poco peso, que su respuesta sea una cadena de caracteres y su forma de despliegue sea fácil de implementar.

Consideradas las necesidades anteriores las opciones son CAPTCHAS de imágenes y CAPTCHAS de texto, pero en este proyecto utilizaremos los CAPTCHAS de imágenes porque se almacenaras en los CAPTCHAS las claves de cifrado de los mensajes de correo electrónico y estas son cadenas de caracteres que no se les puede generar una pregunta coherente.

1. Bases de datos para almacenar los CAPTCHAS.

Para escoger un gestor de base de datos que controle la información delos usuarios registrados en nuestra aplicación, la información de los mensajes que envían entre usuarios y los CAPTCHAS asociados a los mensajes para ser descifrados consideramos 3 características principales en un gestor base de datos relacional:

* Rapidez en transferencias de información.
* Número de usuarios conectados que soporta.
* Facilidad de comunicación entre los lenguajes Python, HTML con los gestores de base de datos.

Los 3 gestores que se analizaron fueron SQLite, MySQL y PostGreSQL.

SQLite es un gestor de base de datos sumamente ligero que soporta hasta 2 terabytes de información, esta base de datos es compatible con python y lenguajes de programación web. Este gestor por su misma ligereza es posible ser implementada en dispositivos móviles, pero no es recomendable cuando múltiples usuarios están haciendo peticiones al mismo tiempo ya que su rendimiento baja[7].

MySQL es un gestor de base de datos que se caracteriza por su transferencia al hacer consultas de datos almacenados; es uno de los gestores libres más utilizados en aplicaciones web y cuenta con diferentes APIs para hacer la comunicación con los lenguajes Python, PHP, C++, etc. Y soporta peticiones de múltiples usuarios gracias a la implementación de peticiones por hilos.

Por ultimo tenemos a PostGreSQL que es un gestor de base de datos que puede hacer operaciones complejas como subconsultas, transacciones y rollbacks’s. Soporta las peticiones de múltiples usuarios pero en cuanto a la velocidad de transferencia de información, comparado con MySQL, es muy lenta pero lo compensa manteniendo una velocidad casi invariable a pesar de que la base se mantenga con un número grande de registros. [8]

Elegimos el gestor de base de datos MySQL porque el proyecto necesita mayor rapidez en la transferencia de información más que generar filtros muy complejos para la búsqueda de información.

Referencias

[1] <http://www.emclient.com/>

[2] https://www.postbox-inc.com/

[3] https://www.zimbra.com/

[4] http://www.opera.com/es-419/computer/mail

[5] https://www.mozilla.org/es-ES/thunderbird/

[6] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.444.8759&rep=rep1&type=pdf>

[7] https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems

[8] http://danielpecos.com/documents/postgresql-vs-mysql/