



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

CONSULTAS PERSONALIZADAS SOBRE EL DESEMPEÑO DE ESTUDIANTES EN UNA ASIGNATURA MEDIANTE BOTS

Alumno: Francisco Javier Martínez Bueno

Tutor: Prof. D. Víctor Manuel Rivas Santos
Dpto: Lenguajes y Sistemas Informáticos

Junio, 2021



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don **VICTOR MANUEL RIVAS SANTOS**, tutor del Trabajo Fin de Grado titulado: **Consultas personalizadas sobre el desempeño de estudiantes en una asignatura mediante bots**, que presenta **FRANCISCO JAVIER MARTÍNEZ BUENO**, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, junio de 2021

El alumno:

El tutor:

FRANCISCO JAVIER MARTÍNEZ BUENO

VICTOR MANUEL RIVAS SANTOS

AGRADECIMIENTOS

Mis agradecimientos a todas las personas que han colaborado y me han apoyado durante la realización de este Trabajo de Fin de Grado. Agradecer de forma expresa a mi tutor la labor y profesionalidad mostrada durante todo el proceso de redacción e implementación de este proyecto.

Agradecer también a mis amigos, haciendo que toda la carrera haya sido más fácil y llevadera, compartiendo horas de estudio, clases de teoría, prácticas y otros muy buenos momentos.

Gracias a Beatriz, mi pareja, por toda la ayuda ofrecida, por ser mi principal apoyo desde el comienzo de mi carrera universitaria, y, sobre todo, por creer en mi incluso cuando ni yo mismo lo hacía.

En especial, agradecer a mis padres, por ser los principales responsables de que hoy esté aquí, por vuestra confianza y vuestro apoyo, motivándome siempre para dar lo mejor de mí.

ÍNDICE

1. Introducción al proyecto	8
1.1. Definición del proyecto	8
1.2. Motivación	8
2. Telegram	9
2.1. Qué es Telegram	9
2.1.1. Chats privados.....	10
2.1.2. Chats secretos.....	10
2.1.3. Grupos y canales.....	10
2.2. Funcionamiento de Telegram.....	11
2.2.1. Cifrado servidor-cliente.....	11
2.2.2. Cifrado cliente-cliente	13
2.2.3. Cifrado simétrico AES 256-bit	14
2.2.4. Cifrado RSA 2048.....	15
2.2.5. Intercambio de claves seguras Diffie-Hellman	15
2.3. Plataformas de mensajería instantánea similares a Telegram.....	15
2.3.1. WhatsApp	15
2.3.2. Signal.....	17
2.3.3. WeChat.....	18
2.3.4. LINE.....	20
2.3.5. Facebook Messenger	21
2.3.6. Viber	22
2.3.7. Comparativa de Telegram con otras plataformas de mensajería instantánea	23
2.4. Otros servicios de Telegram	24
2.4.1. Widgets.....	24
2.4.2. Stickers animados	25
2.4.3. Temas personalizados en la nube.....	25
3. Bots y API de Telegram	26
3.1. Qué es un bot.....	26
3.2. Utilidades de un bot	26
3.3. Funcionamiento de un bot.....	27
3.4. Principales diferencias entre usuarios normales y bots	27
3.5. Ejemplos de bots importantes	28
3.5.1. Trello.....	28
3.5.2. GitLab	29
3.5.3. GitHub	30
3.6. Qué es una API	31
3.7. Telegram Bot API.....	31
3.7.1. BotFather.....	32
3.7.2. API de pagos	33
3.8. TDLib y Telegram API.....	34

4. Metodología de trabajo	35
4.1. Metodología para el desarrollo del software	35
4.1.1. Metodología tradicional	35
4.1.2. Metodología ágil	36
4.1.3. ¿Por qué utilizar una metodología ágil?	39
5. Planificación del proyecto	40
5.1. Análisis de riesgos	40
5.2. Planificación temporal	41
5.3. Estimación del coste del proyecto	41
6. Análisis del proyecto	43
6.1. Componentes lógicos	43
6.2. Conjunto inicial de historias de usuario	44
6.3. Escenarios basados en las historias de usuario iniciales	46
7. Creación y configuración de nuestro bot de Telegram	52
7.1. Elección del nombre de usuario de nuestro bot	52
7.2. Configuración inicial del bot utilizando BotFather	52
7.3. Creación y configuración de la hoja de cálculo y Google Apps Script	53
8. Incrementos	56
8.1. Versión 1.0	56
8.2. Versión 2.0	66
9. Conclusión	82
9.1. Reflexiones sobre el desarrollo del proyecto	82
9.2. Posibles mejoras futuras	83
10. Bibliografía	84
11. Anexos	86
11.1. Anexo 1. Ejemplo hoja de cálculo para una asignatura	86

ÍNDICE DE ILUSTRACIONES

<i>Ilustración 1: esquema de encriptación para un cifrado servidor-cliente</i>	12
<i>Ilustración 2: esquema de encriptación para un cifrado cliente-cliente</i>	13
<i>Ilustración 3: interacción con @trello_bot para la vinculación y creación de una nueva tarjeta</i>	28
<i>Ilustración 4: interacción con @gitlab_bot para la vinculación con un repositorio de GitLab</i>	29
<i>Ilustración 5: interacción con @GitHubBot para la vinculación con un repositorio de GitHub</i>	30
<i>Ilustración 6: ciclo de vida del desarrollo de software en una metodología ágil</i>	37
<i>Ilustración 7: tablero de Jira Software creado a partir de una plantilla Kanban</i>	39
<i>Ilustración 8: lluvia de ideas sobre los posibles nombres de usuario</i>	52
<i>Ilustración 9: creación y configuración inicial del bot</i>	53
<i>Ilustración 10: configuración inicial de la hoja de cálculo y conexión con el bot de Telegram a través de un webhook de Google</i>	54
<i>Ilustración 11: ventana externa para permitir la interacción entre el bot de Telegram y nuestra aplicación</i>	55
<i>Ilustración 12: diagrama de casos de uso de la versión 1.0</i>	59
<i>Ilustración 13: diagrama de secuencia del sistema para el caso de uso “Un estudiante se registra con su correo electrónico institucional”</i>	60
<i>Ilustración 14: diagrama de secuencia del sistema para el caso de uso “El profesor oculta información sensible”</i>	61
<i>Ilustración 15: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta información recogida en la hoja de cálculo”</i>	61
<i>Ilustración 16: diagrama de secuencia del sistema para el caso de uso “Un profesor bloquea temporalmente el acceso a los datos y vuelve a darlo”</i>	62
<i>Ilustración 17: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta qué órdenes le puede dar al bot”</i>	62
<i>Ilustración 18: diagrama de secuencia del sistema para el caso de uso “Un profesor elige con qué nombre se refiere a una columna/dato”</i>	63
<i>Ilustración 19: captura de pantalla con información relativa a la primera interacción de un estudiante con el bot</i>	64
<i>Ilustración 20: e-mail enviado por el bot de Telegram para verificar nuestra dirección de correo electrónico</i>	64
<i>Ilustración 21: verificación de la cuenta de correo electrónico y consulta sobre el desempeño de un estudiante en una asignatura</i>	65
<i>Ilustración 22: diagrama de casos de uso de la versión 2.0</i>	70
<i>Ilustración 23: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta el enlace web en el que se desarrollará una videoconferencia”</i>	71
<i>Ilustración 24: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta información relativa al sistema de evaluación de una asignatura”</i>	72
<i>Ilustración 25: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta información recogida en la hoja de cálculo sobre su desempeño en distintas asignaturas”</i>	72
<i>Ilustración 26: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta información sobre el propio bot”</i>	73
<i>Ilustración 27: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta información para contactar con el desarrollador del propio bot”</i>	73
<i>Ilustración 28: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta información para contactar con el profesor encargado de la asignatura”</i>	73
<i>Ilustración 29: ejecución del comando /informacionestudiante para obtener información relativa al desempeño de un estudiante en una asignatura</i>	74
<i>Ilustración 30: ejecución del comando /inforaciondocencia para obtener información relativa a los enlaces webs donde se desarrollan las videoconferencias de una asignatura</i>	74

<i>Ilustración 31: ejecución del comando /sistemaevaluacion con información relativa al sistema de evaluación de una asignatura</i>	<i>75</i>
<i>Ilustración 32: ejecución del comando /contactoprofesor para obtener información relativa a los diferentes métodos de contacto con un profesor.....</i>	<i>75</i>
<i>Ilustración 33: ejecución del comando /informacionbot para obtener información relativa al bot</i>	<i>75</i>
<i>Ilustración 34: ejecución del comando /contacto para obtener información relativa al desarrollador del bot</i>	<i>76</i>
<i>Ilustración 35: ejecución del comando /ayuda para obtener todos los comandos disponibles en el bot</i>	<i>76</i>
<i>Ilustración 36: encuesta de satisfacción. ¿Tuvo algún problema a la hora de registrarse en el bot?...77</i>	
<i>Ilustración 37: encuesta de satisfacción. En caso de que en la pregunta anterior haya contestado “Sí”, indique el/los motivo/s.....</i>	<i>77</i>
<i>Ilustración 38: encuesta de satisfacción. ¿Tuvo algún problema a la hora de verificar su correo electrónico?.....</i>	<i>78</i>
<i>Ilustración 39: encuesta de satisfacción. ¿Le resultó fácil consultar los comandos ofrecidos por el bot?</i>	<i>78</i>
<i>Ilustración 40: encuesta de satisfacción. ¿La interacción con el bot fue fácil e intuitiva?.....</i>	<i>78</i>
<i>Ilustración 41: encuesta de satisfacción. ¿Con qué probabilidad utilizaría este bot si todas las asignaturas lo implantasen?.....</i>	<i>79</i>
<i>Ilustración 42: encuesta de satisfacción. ¿Cuál es la calidad de la información recibida por el bot acerca de su desempeño en la asignatura?</i>	<i>79</i>
<i>Ilustración 43: encuesta de satisfacción. ¿Cuál es la probabilidad de que recomiende el bot a sus compañeros si se implantase en todas las asignaturas?.....</i>	<i>79</i>
<i>Ilustración 44: encuesta de satisfacción. En general, ¿cómo de satisfecho está con el bot?</i>	<i>80</i>
<i>Ilustración 45: encuesta de satisfacción. ¿Qué otra información le gustaría poder consultarle al bot?.....</i>	<i>80</i>
<i>Ilustración 46: encuesta de satisfacción. ¿Qué aspecto destacaría y qué aspecto mejoraría del bot?.....</i>	<i>81</i>
<i>Ilustración 47: hoja de cálculo de ejemplo perteneciente a la versión 1.0 del desarrollo del software</i>	<i>86</i>

ÍNDICE DE TABLAS

<i>Tabla 1: comparativa entre Telegram y diferentes plataformas de mensajería instantánea</i>	<i>23</i>
<i>Tabla 2: exposición de los riesgos del proyecto.....</i>	<i>40</i>
<i>Tabla 3: estimación del coste total del proyecto</i>	<i>42</i>
<i>Tabla 4: especificación del requisito FRQ-0001.....</i>	<i>56</i>
<i>Tabla 5: especificación del requisito FRQ-0002.....</i>	<i>56</i>
<i>Tabla 6: especificación del requisito FRQ-0003.....</i>	<i>57</i>
<i>Tabla 7: especificación del requisito FRQ-0004.....</i>	<i>57</i>
<i>Tabla 8: especificación del requisito FRQ-0005.....</i>	<i>58</i>
<i>Tabla 9: especificación del requisito FRQ-0006.....</i>	<i>58</i>
<i>Tabla 10: especificación del requisito FRQ-0007.....</i>	<i>66</i>
<i>Tabla 11: especificación del requisito FRQ-0008.....</i>	<i>67</i>
<i>Tabla 12: especificación del requisito FRQ-0009.....</i>	<i>67</i>
<i>Tabla 13: especificación del requisito FRQ-0010.....</i>	<i>68</i>
<i>Tabla 14: especificación del requisito FRQ-0011.....</i>	<i>68</i>
<i>Tabla 15: especificación del requisito FRQ-0012.....</i>	<i>69</i>

1. Introducción al proyecto

1.1. Definición del proyecto

En el transcurso de esta memoria se describirá el proceso de análisis, diseño y desarrollo de un bot para la plataforma de mensajería Telegram. Hoy en día es factible utilizar bots que permitan responder automáticamente a determinada información requerida por los usuarios.

El bot a desarrollar almacenará y accederá a una colección de datos guardada en la nube, tales como, hojas de cálculo y documentos de texto de Google Drive. Dicho bot permitirá a cada alumno solicitar información personalizada sobre su desarrollo en una asignatura, como, por ejemplo, sus puntuaciones en las diferentes pruebas prácticas o teóricas, fechas y horas de tutorías, enlaces a las videoconferencias para las clases prácticas y teóricas, etc.

El desarrollo del bot se llevará a cabo utilizando Google Apps Script, así mismo, seguiremos una metodología ágil para una correcta realización del proyecto.

1.2. Motivación

El desarrollo continuo de las nuevas tecnologías tiene como efecto la actualización constante de plataformas, dando lugar a aplicaciones de mensajería instantánea con propiedades innovadoras.

Una de las características diferenciadoras de Telegram es la capacidad de poder crear y programar un bot capaz de responder a cualquier instrucción dada por un usuario. El hecho de poder desarrollar una aplicación totalmente automatizada, sin nadie detrás de ella, preparada para ofrecer determinadas respuestas en función del comando que escribas, es fascinante, y sin duda, fue uno de los aspectos que hizo que finalmente me decantase por el tema expuesto en este Trabajo Fin de Grado.

2. Telegram

2.1. Qué es Telegram

Telegram (Telegram, 2013) es un servicio de mensajería multiplataforma enfocado en la velocidad y la seguridad, fundado por el empresario ruso Pavel Durov y su hermano Nikolai Durov.

Fue lanzado por primera vez en iOS el 14 de agosto de 2013 y en Android el 20 de octubre de 2013, ahora mismo la aplicación cuenta con más de 500 millones de usuarios activos mensuales. La base de datos de usuarios de esta aplicación tiende a aumentar cada vez que sale a la luz un problema de privacidad en algunas de las aplicaciones de mensajería de sus competidores.

Nikolai Durov se encargó de desarrollar un protocolo de datos personalizado y único, óptimo y seguro para trabajar con una gran cantidad de centros de datos, haciendo a cualquier red segura, fiable y veloz. El equipo encargado del desarrollo se vio obligado a abandonar Rusia debido a las políticas locales que conferían la regulación de las tecnologías de la información. Actualmente, el equipo se encuentra situado en la ciudad de Dubai, a pesar de que la gran cantidad de los desarrolladores encargados de este servicio de mensajería proceden de San Petersburgo.

Telegram es único gracias a su enfoque en la privacidad, el cifrado y una API de código abierto. Hay innumerables clientes no oficiales que acompañan a las aplicaciones y la interfaz web oficial de dicha aplicación.

Un hecho destacable de Telegram con respecto a la plataforma más popular del mundo, WhatsApp, es la posibilidad de tener una sesión iniciada de forma simultánea en diferentes dispositivos, mientras que en WhatsApp esto aún no es posible.

Con Telegram se pueden enviar mensajes, vídeos, fotos y archivos de cualquier tipo.

2.1.1. Chats privados

En los chats privados podemos buscar contactos sin necesidad de tener su número de teléfono, únicamente necesitamos su nombre de usuario. Aún así, si otorgamos permisos para acceder a nuestros contactos podremos entablar una conversación con cualquier usuario de nuestra agenda que utilizan Telegram.

Cada usuario puede compartir su nombre de usuario generando un enlace del tipo `t.me/NombreDeUsuario`, donde “NombreDeUsuario” sería el nombre de usuario establecido, de este modo, cualquier persona puede ponerse en contacto con nosotros sin conocer nuestro número de teléfono.

Telegram ofrece llamadas de voz y videollamadas con cifrado de extremo a extremo.

2.1.2. Chats secretos

Los chats secretos usan un cifrado end-to-end, es decir, únicamente el emisor y el receptor del mensaje tienen la posibilidad de acceder al contenido del mismo, por lo que, solo ellos pueden descifrarlo.

Los mensajes no pueden ser reenviados y, cuando eliminamos un mensaje de nuestra conversación, automáticamente ese mensaje desaparece de la aplicación del receptor.

Una vez que los archivos, videos, fotos y mensajes hayan sido leídos por el destinatario, será posible establecer un periodo de tiempo determinado en el que estos se autodestruirán.

El acceso a estos chats será posible únicamente desde el dispositivo de origen.

2.1.3. Grupos y canales

Telegram ofrece la posibilidad de crear grupos de hasta 200.000 personas, donde todos los usuarios pueden interaccionar respondiendo a mensajes, mencionando a un determinado usuario o grupo de usuarios o incluso haciendo uso de hastags para estructurar de una forma diferente los mensajes grupales de un determinado chat.

En Telegram se pueden crear canales donde únicamente el administrador o los administradores del mismo pueden enviar mensajes, cabría destacar que WhatsApp ha incluido hace poco esta funcionalidad en su aplicación.

Podemos silenciar un grupo para recibir únicamente las notificaciones de las menciones o cuando responden a uno de nuestros mensajes.

2.2. Funcionamiento de Telegram

La función principal de Telegram, al igual que la de la mayoría de las aplicaciones de mensajería, consiste en poder enviar mensajes a otros usuarios, buscando su nombre de usuario en el buscador de la aplicación, crear conversaciones grupales, enviar archivos de cualquier tipo y en cualquier formato, enviar “stickers”, “GIFs” y llamar a contactos; sin embargo, hay algunas características específicas de esta aplicación que hacen que funcione de manera diferente a otros servicios de mensajería similares.

La característica más relevante de Telegram es la privacidad y, para garantizar esto, emplea dos capas de cifrado seguro. El cifrado servidor-cliente se usa en los chats privados y grupales, mientras que para los chats secretos se utiliza una capa adicional de cifrado cliente-cliente.

Todos los datos, ya sea multimedia, textos o archivos son cifrados de la misma manera, basándose en un cifrado AES 256-bit symmetric, el cifrado RSA 2048 y el intercambio de claves seguras Diffie-Hellman.

2.2.1. Cifrado servidor-cliente

El cifrado servidor-cliente está basado en el protocolo MTProto 2.0, es decir, sincroniza todos los datos entre todos nuestros dispositivos y los carga a los servidores de Telegram.

Antes de que un mensaje se transmita a través de una red utilizando el protocolo de transporte, se cifra y se agrega un encabezado externo en la parte superior del mensaje, compuesto de un identificador de clave de 64 bits y una clave del mensaje de 128 bits.

Cada mensaje es encriptado haciendo uso del cifrado AES-256, utilizando una clave de autorización junto con la clave del mensaje, definiendo así una clave de 256 bits. La parte inicial del mensaje a cifrar contiene datos variables como la ID de la sesión, ID del mensaje, número de secuencia y sal del servidor, dando lugar a un esquema de cifrado por bloques del tipo AES (Advanced Encryption Standard).

En el cifrado MTProto 2.0, la clave del mensaje se define como los 128 bits intermedios del código hash SHA-256 del cuerpo del mensaje, incluyendo también los datos variables mencionados anteriormente, precedidos por 32 bits de la clave de autorización. La ilustración 1 muestra el esquema de cifrado utilizado entre el cliente y el servidor.

MTProto 2.0, part I

Cloud chats (server-client encryption)

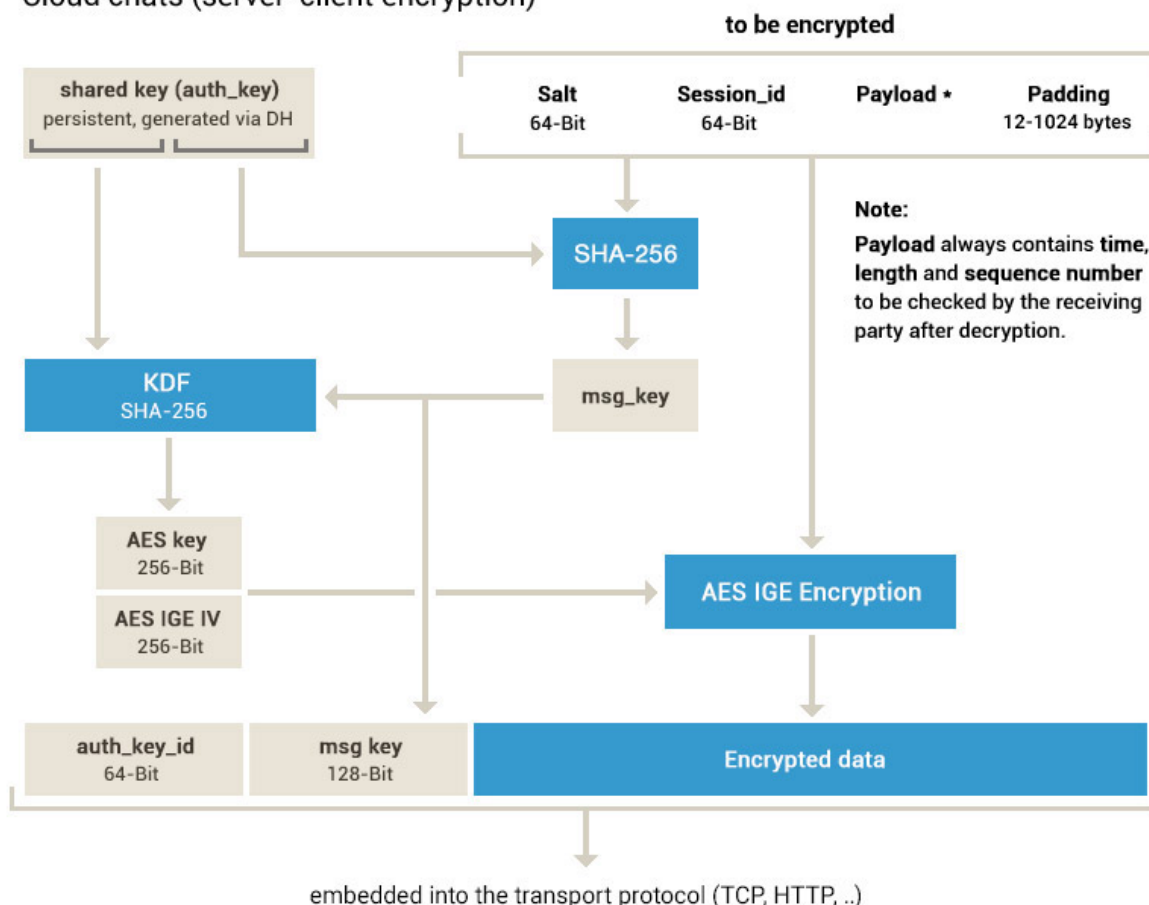


Ilustración 1: esquema de encriptación para un cifrado servidor-cliente

Telegram. Ilustración de MTProto Mobile Protocol. [Ilustración]. Recuperado de <https://core.telegram.org/mtproto> (Acceso por última vez 18-03-2021)

2.2.2. Cifrado cliente-cliente

El cifrado cliente-cliente se utiliza en los chats secretos; son chats one-on-one en los que los mensajes se cifran con una clave que únicamente tienen los participantes del propio chat.

El esquema de encriptación utilizado para estos chats secretos cifrados de un extremo a otro es diferente al utilizado para los chats comunes. Todas las claves son generadas utilizando el protocolo criptográfico Diffie-Hellman. La ilustración 2 muestra el esquema de cifrado utilizado en los chats secretos.

MTProto 2.0, part II

Secret chats (end-to-end encryption)

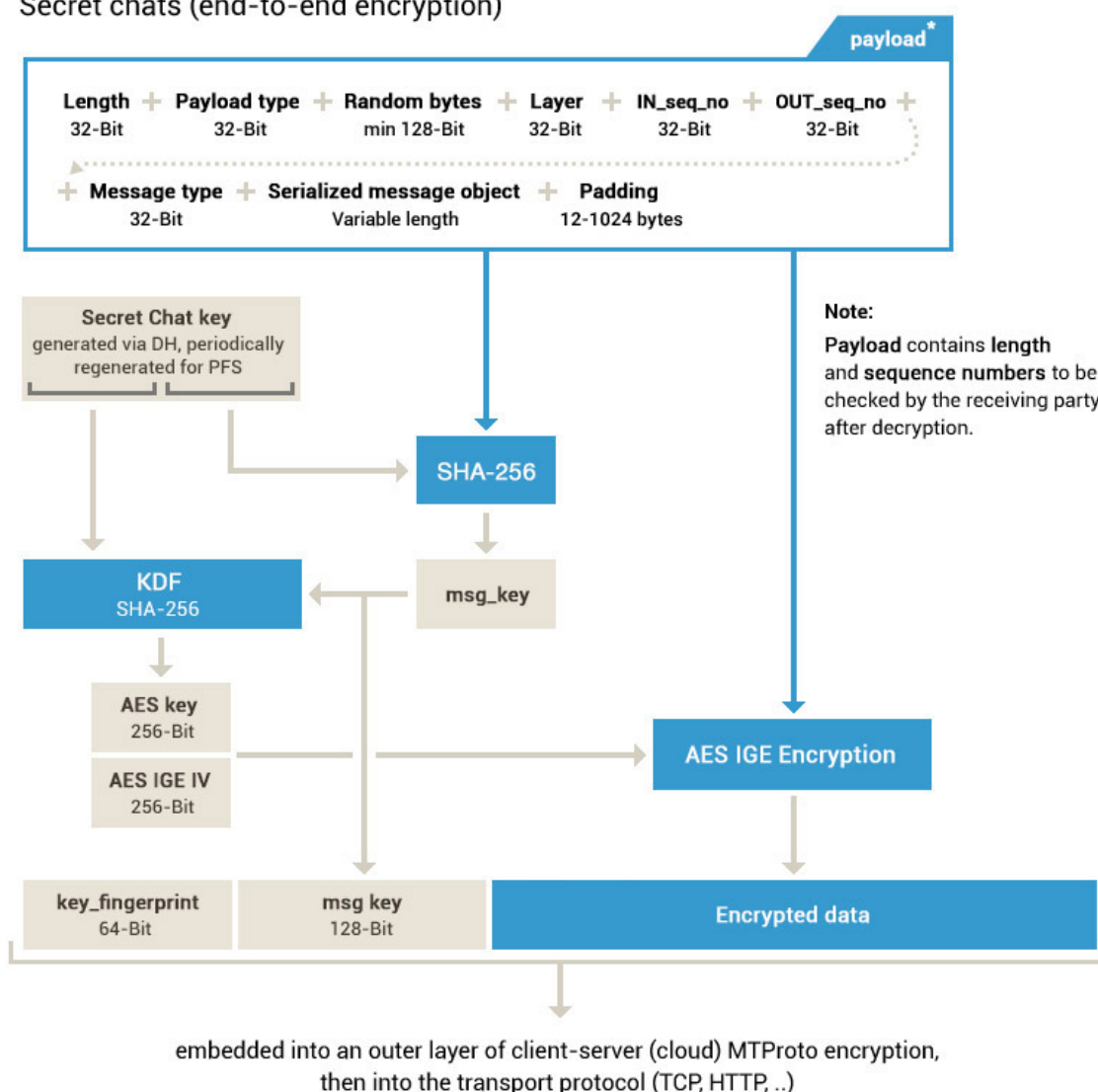


Ilustración 2: esquema de encriptación para un cifrado cliente-cliente

Telegram. Ilustración de End-to-End Encryption, Secret Chats. [Ilustración]. Recuperado de <https://core.telegram.org/api/end-to-end> (Acceso por última vez 18-03-2021)

2.2.3. Cifrado simétrico AES 256-bit

El cifrado simétrico AES 256-bit fue desarrollado por dos criptógrafos belgas, Vincent Rijmen y Joan Daemen, estudiantes de “Katholieke Universiteit Leuven” en Bélgica. Es el primer y único cifrado de acceso público aprobado por la Agencia de Seguridad Nacional de EEUU para proteger información ultrasecreta y, desde entonces, AES se ha convertido en el estándar de la industria para el cifrado.

Es un algoritmo de cifrado simétrico, es decir, se utiliza la misma clave tanto para el cifrado como para el descifrado. También se caracteriza por ser un cifrado en bloque ya que toda la información a cifrar se divide en secciones llamadas bloques. AES utiliza un tamaño de bloque de 128 bits, en el que los datos se dividen en una matriz de cuatro por cuatro que contiene 16 bytes.

AES proporciona seguridad adicional porque utiliza un proceso de expansión de claves en el que la clave inicial se utiliza para generar una serie de claves nuevas llamadas claves redondas, que se generan a través de múltiples rondas de modificación haciendo que sea más difícil romper el cifrado.

A continuación, veremos de manera muy resumida los pasos a seguir para encriptar un texto plano utilizando el algoritmo de cifrado simétrico AES 256-bit:

- La clave inicial se agrega al bloque usando un cifrado XOR.
- Cada byte de datos se sustituye por otro, siguiendo una tabla predeterminada.
- Las filas de la matriz 4x4 se desplazan
 - Los bytes de la segunda fila se mueven un espacio a la izquierda
 - Los bytes de la tercera fila se mueven dos espacios a la izquierda
 - Los bytes de la cuarta fila se mueven tres espacios a la izquierda
- Las columnas se mezclan
 - Una operación matemática combina los cuatro bytes de cada columna
- La clave redonda se agrega al bloque
- Se repite el proceso trece veces

2.2.4. Cifrado RSA 2048

Con el cifrado RSA 2048, los mensajes se cifran con una clave pública, que se puede compartir abiertamente; sin embargo, una vez que un mensaje se ha cifrado con la clave pública, solo se puede descifrar con una clave privada.

Este tipo de cifrado se utiliza en combinación con otros esquemas de cifrado, en nuestro caso, un archivo se cifrará utilizando un cifrado simétrico AES 256-bit, y luego la clave simétrica se cifra con un cifrado RSA 2048.

2.2.5. Intercambio de claves seguras Diffie-Hellman

El intercambio de claves seguras Diffie-Hellman permite establecer de forma segura claves públicas y privadas que se utilizan para proteger las comunicaciones.

Para que tanto el cliente como el servidor puedan generar conjuntamente una clave compartida, ambos deben elegir dos números públicos y, cada uno de ellos, un número secreto. A continuación, usando una fórmula matemática, cada uno debe hacer una serie de operaciones con los dos números públicos y el número secreto. Seguidamente, el servidor y el cliente se intercambian los resultados de forma pública. Por último, ambas partes utilizan por separado una fórmula matemática que combina los dos números secretos transformados con su número secreto, obteniendo así la clave compartida.

2.3. Plataformas de mensajería instantánea similares a Telegram

2.3.1. WhatsApp

WhatsApp¹ es una aplicación de mensajería instantánea multiplataforma (Gil, 2021) que nos permite enviar mensajes de texto, hacer llamadas de voz y vídeos, enviar archivos, con solo una conexión WiFi o datos móviles. Fue una de las primeras aplicaciones móviles en ofrecer mensajería gratuita basada en internet.

Esta aplicación cuenta con más de 2 mil millones de usuarios activos gracias a su accesibilidad, funcionalidad multiplataforma y algunas características que hacen que su uso sea sencillo y directo. Se envían más de 100 mil millones de mensajes cada día y se realizan más de mil millones de llamadas de forma diaria.

¹ URL de la plataforma WhatsApp: <https://www.whatsapp.com/?lang=es>

El fundador de esta aplicación es Jan Koum, un emprendedor ucraniano que con 16 años emigró a Estados Unidos. Empezó su formación en la Universidad Estatal de San José; sin embargo, no terminó sus estudios. Jan Koum fue contratado por Yahoo! en 1997, allí conoció a Brian Acton, su futuro socio y co-fundador de WhatsApp. En el año 2007 abandonan sus puestos de trabajo y en febrero de 2009 fundan la empresa WhatsApp, siendo el nombre resultado de un juego de palabras entre “What’s up?” y “App”, abreviatura de la palabra “application” en inglés.

La idea surgió cuando Jan Koum acudía a un gimnasio el cual prohibía el uso de teléfonos móviles, perdiendo así una gran cantidad de llamadas telefónicas durante la duración de su entrenamiento. Fue entonces cuando se le ocurrió que sería útil que apareciese junto a nuestro nombre, en la lista de contactos, el estado actual de cada persona, permitiendo al resto de usuarios saber si esa persona estaba disponible o no.

En 2014 Mark Zuckerberg, creador de Facebook, se interesa por la aplicación y la adquiere por 21.800 millones de dólares.

En 2018 se destapa el escándalo de “Cambridge Analytica”, empresa que se encargó en 2016 de la campaña política de Donald Trump, y que estaba manejando información proporcionada por las compañías de Mark Zuckerberg. Se estima que se utilizó información sin consentimiento de más de 50 millones de personas. Tras este revuelo, Jan Koum y Brian Acton renuncian a su puesto en la compañía que ellos mismos fundaron.

Algunas de las principales características de WhatsApp son las siguientes:

- **Mensajes seguros:** utiliza un cifrado de extremo a extremo, donde únicamente el emisor y el receptor pueden leer o escuchar lo que se envía.
- **Llamadas de voz y vídeo:** tanto en chats privados como en grupos, pudiéndose conectar como máximo ocho participantes de forma simultánea.
- **Mensajes de voz**

- **Versión de escritorio:** podemos acceder a WhatsApp a través de nuestro ordenador, aunque será posible únicamente si nuestro móvil está conectado a una red WiFi o móvil.
- **Uso compartido de fotos, vídeos y documentos:** el tamaño máximo permitido para los archivos multimedia es de 16 MB, mientras que para documentos es de 100 MB.
- **Compartir ubicación:** WhatsApp permite compartir nuestra ubicación actual y nuestra ubicación en tiempo real, siendo esto de gran utilidad si necesitamos que alguien sepa donde nos encontramos en un momento determinado.

2.3.2. Signal

Signal² es una de las plataformas de mensajería instantánea (Lumb, 2018) más seguras que existen actualmente, pero también es lo suficientemente simple como para que cualquier usuario sea capaz de configurarla y usarla.

Moxie Marlinspike y Stuart Anderson fundaron en 2010 TextSecure y RedPhone para enviar mensajes de texto y hacer llamadas de voz proporcionando una comunicación cifrada de extremo a extremo. Al cabo de un año, Twitter compró la empresa y relanzaron ambos servicios como aplicaciones de mensajería de código abierto. En 2013 ambos socios crearon el proyecto de código abierto “Open Whisper Systems”, donde continuarían desarrollando TextSecure y RedPhone para finalmente combinarlos y lanzar Signal en noviembre de 2015.

De forma paralela, Moxie Marlinspike y Stuart Anderson comenzarían a desarrollar el protocolo de cifrado de Signal para posteriormente ofrecer un paquete con el método criptográfico de extremo a extremo y que otras plataformas de mensajería lo implementaran. En julio de 2016, Facebook Messenger empezó a utilizar dicho protocolo en sus conversaciones secretas, Google Hallo lo haría en septiembre de 2016 en su modo incógnito y seguidamente lo implantaría Microsoft en enero de 2018 en las conversaciones privada de Skype.

² URL de la plataforma Signal: <https://signal.org/es/>

En febrero de 2018, los cofundadores de Signal y WhatsApp, Moxie Marlinspike y Brian Acton respectivamente, lanzaron “Signal Foundation” con el único objetivo de apoyar, acelerar y ampliar la misión de Signal para hacer que la comunicación privada sea accesible y ubicua.

Algunas de las principales características de Signal son las siguientes:

- La aplicación de Signal en Android puede configurarse como la aplicación predeterminada de SMS/MMS.
- Llamas de voz y vídeo, de hasta ocho integrantes, totalmente privadas gracias a la encriptación de extremo a extremo.
- Chats grupales: un usuario puede unirse a un grupo a través de un enlace de invitación o código QR. Se puede configurar un temporizador de mensajes para que sean eliminados automáticamente. Hay un máximo de 1000 personas por grupo.
- “Note to Self”: es un chat privado cifrado para enviarnos mensajes a nosotros mismo, utilizando así el chat a modo de aplicación de notas o incluso para compartir mensajes y archivos con nuestros dispositivos vinculados.

2.3.3. WeChat

WeChat³ es la aplicación de mensajería instantánea (Zheng, 2020) más conocida de China, con más de mil millones de usuarios activos de forma mensual. La versión original fue creada en octubre de 2010 por Allen Zhang en el centro de investigación de Tencent Guangzhou, una de las empresas más grandes de Asia por su capitalización de mercado; sin embargo, fue lanzada en 2011. Actualmente servicios como Facebook o WhatsApp están prohibidos en China, por tanto, WeChat es el estándar para los negocios y la comunicación.

En 2012 la aplicación contaba con 100 millones de usuarios, en 2016 pasó a tener más de 889 millones de usuarios activos mensuales y fue en 2019 cuando la cifra aumentó por encima de los mil millones de usuarios.

³ URL de la plataforma WeChat: <https://www.wechat.com/es/>

La seguridad de esta aplicación es prácticamente nula ya que la propia empresa desarrolladora comparte datos de usuarios con el gobierno Chino. WeChat no cuenta con ningún tipo de cifrado, por tanto, nuestros mensajes pueden ser interceptados y leídos por una tercera persona.

A día de hoy es fácil descargarse la aplicación; sin embargo, el registrarse en la misma no es tan sencillo como crear un nombre de usuario y una contraseña. Para poder acceder a este servicio de mensajería, alguien que ya tenga una cuenta deberá escanear nuestro código QR para verificar la creación de nuestro perfil.

Algunas de las principales características de WeChat son las siguientes:

- **Aplicación de mensajería instantánea:** WeChat permite enviar mensajes, nuestra localización actual, fotos y vídeos a cualquier persona registrada en esta plataforma. También podemos realizar llamadas de voz y vídeo, crear chats grupales con hasta 500 personas y realizar llamadas en grupo con hasta nueve personas. Esta aplicación cuenta con una opción para conocer gente nueva haciendo click sobre “Personas cercas” o agitando el teléfono móvil.
- **“WeChat Mini Programs”:** WeChat permite que empresas de terceros desarrollen mini aplicaciones capaces de brindar funciones avanzadas a los usuarios, ejecutándose dentro de la propia aplicación de mensajería, permitiendo así a WeChat agrupar funciones y capacidades en una única aplicación móvil nativa.
- **“Moments”:** es una especie de red social donde se pueden publicar actualizaciones textuales, subir música y compartir imágenes y vídeos, similar a la línea de tiempo de Facebook o Twitter. La principal diferencia con respecto a las dos redes sociales citadas anteriormente es el nivel de intimidad, siendo este mayor en “Moments”, ya que nuestra lista de contactos no es visible para el resto de usuarios y nuestras actualizaciones únicamente las pueden ver los contactos que nosotros hayamos verificado.

- **Cuentas oficiales:** cualquier marca, celebridad o empresa puede registrar una cuenta oficial verificada en WeChat para enviar noticias y mensajes a sus seguidores.
- **Pagos y compras con “WeChat Pay”:** desde la propia aplicación tenemos la opción de escanear códigos QR para pedir comida desde el interior de un local y pagar la misma. Podemos enviar y recibir dinero de nuestros contactos de forma instantánea haciendo uso de esta.

2.3.4. LINE

LINE⁴ es una plataforma de mensajería instantánea japonesa (Iqbal, 2021) desarrollada en 2011 por NHN como una solución de emergencia para los empleados de la empresa tras el terremoto de la costa del Pacífico en la región de Tōhoku. Este terremoto provocó una caída de todos los servicios de telefonía, dando lugar al desarrollo de LINE para su personal como medio de comunicación basado en internet.

La aplicación cuenta con más de 215 millones de usuarios activos mensuales.

LINE utiliza su propio protocolo de cifrado de extremo a extremo (Technical Whitepaper, 2019), Letter Sealing, para encriptar cualquier tipo de mensaje o llamada de voz o vídeo. Los mensajes se cifran localmente en el dispositivo cliente antes de enviarse a los servidores de la plataforma, pudiendo descifrarse únicamente por el destinatario previsto.

Algunas de las principales características de LINE son las siguientes:

- **“LINE Games”:** fue creado en 2011 y únicamente los usuarios con una cuenta en la aplicación de LINE tienen acceso a estos juegos, pudiendo conectarnos con amigos, ganar puntos e incluso enviar y recibir recompensas.
- **“LINE Pay”:** fue introducido dentro de la aplicación el 2014. Este servicio permite a cualquier usuario de la plataforma enviar y solicitar dinero a los usuarios de su agenda de contactos. Así mismo, este

⁴ URL de la plataforma LINE: <https://line.me/en/>

servicio permite hacer transferencias bancarias e incluso retirar e ingresar dinero haciendo uso de un cajero automático.

- **“LINE Taxi”**: lanzado en 2015, es un servicio de taxis local de Japón para pedir y pagar taxis, siendo usable únicamente a través de la aplicación LINE.
- **“LINE Wow”**: servicio de envío y entrega de productos alimenticios.
- **“LINE Today”**: sección de noticias dentro de la aplicación.
- **“Line Shopping”**: los usuarios de la aplicación obtienen descuentos adicionales comprando a través de este portal.

2.3.5. Facebook Messenger

Facebook Messenger⁵ (Kacey Kroh, 2016) es un servicio de mensajería instantánea, cuenta con más de 1300 millones de usuario activos mensuales y es propiedad de Facebook. Esta aplicación empezó como un chat interno propio de la red social, llamado Facebook Chat. En marzo de 2011, esta empresa adquirió Beluga, una plataforma de mensajería grupal para lanzar así, en agosto de 2011, Facebook Messenger, una plataforma independiente a Facebook, con el fin de reemplazar Facebook Chat.

Los chats privados y secretos de Facebook Messenger cuentan con un cifrado de extremo a extremo; sin embargo, la comunicación en chats grupales no está cifrada y, por tanto, no es segura.

Algunas de las principales características de Facebook Messenger son las siguientes:

- **Enviar fotos, vídeo, stickers o GIFs**: al igual que en otras plataformas, desde Facebook Messenger podemos enviar cualquier tipo de archivo multimedia que tengamos almacenado en nuestro dispositivo.
- **Enviar dinero y solicitar fondos**: los pagos son gratuitos, sin embargo, para hacer uso de esta funcionalidad hay que residir en EE.UU.
- **Llamas de voz y vídeo.**

⁵ URL de la plataforma Facebook Messenger: <https://www.messenger.com/>

- **Viajes en Uber o Lyft:** desde la propia aplicación podemos hacer uso de ambos servicios de transporte a particulares.
- **Bots:** Facebook Messenger cuenta con esta funcionalidad gracias a la introducción de la API en 2016 para que cualquier desarrollador pueda construir los mismos. Por ejemplo, a día de hoy podemos enviar un mensaje al bot de Burger King para hacer un pedido.

2.3.6. Viber

Viber⁶ es una aplicación de mensajería multiplataforma (Johnson, 2021) desde la que podemos enviar mensajes de texto, imágenes, vídeos e incluso hacer llamadas telefónicas.

Viber permite crear chats grupales de hasta 250 personas e incluso realizar llamadas grupales con hasta 20 usuarios de forma simultánea, siendo todo esto cifrado de extremo a extremo, evitando así que nuestras conversaciones sean interceptadas por terceros.

Viber cuenta con más de un millón de usuarios registrados gracias a sus opciones de seguridad y la privacidad brindada por dicha plataforma de mensajería instantánea.

Algunas de las principales características de Viber son las siguientes:

- **Registro por número de teléfono:** para poder iniciar sesión en Viber debemos introducir nuestro número de teléfono, evitando así que cualquier usuario oculte su identidad tras la pantalla haciendo uso de un nombre de usuario y una contraseña.
- **Mensajes de texto, llamadas de voz y video:** Viber nos permite enviar cualquier tipo de mensaje a cualquier usuario de nuestra agenda de contactos.
- **“Viber Out”:** Viber cuenta con un servicio por suscripción para poder realizar llamadas a teléfonos móviles.
- **Suministro de noticias personalizado:** Viber cuenta con una función para mostrarnos noticias en función de las fuentes que nos interesen.

⁶ URL de la plataforma Viber: <https://www.viber.com/es/>

2.3.7. Comparativa de Telegram con otras plataformas de mensajería instantánea

La tabla 1 muestra algunas características diferenciadoras de Telegram con respecto al resto de plataformas de mensajería instantánea.

	Telegram	WhatsApp	Signal	WeChat	LINE	Facebook Messenger	Viber
Gratuito	✓	✓	✓	✓	✓	✓	✓
Llamadas de audio y vídeo	✓	✓	✓	✓	✓	✓	✓
Servicio de noticias	x	x	x	x	✓	x	✓
Pagos a amigos desde la aplicación	x	x	x	✓	✓	✓	✓
Eliminar mensajes	✓	✓	✓	✓	✓	✓	✓
Editar mensajes	✓	x	x	✓	x	x	✓
Programar envío de mensajes	✓	x	x	x	x	x	x
Versión de escritorio	✓	✓	✓	✓	✓	✓	✓
Copia de seguridad de mensajes en la nube	✓	✓	x	x	✓	✓	✓
Creación de bots	✓	x	x	x	x	✓	x
Software código abierto	✓	x	✓	x	x	x	x
Cifrado de extremo a extremo por defecto	x	✓	✓	x	✓	x	✓

Tabla 1: comparativa entre Telegram y diferentes plataformas de mensajería instantánea

2.4. Otros servicios de Telegram

2.4.1. Widgets

Telegram ofrece la posibilidad de crear widgets (Telegram, 2018). Un ejemplo de estos es, la inclusión de un botón para compartir nuestra página web o post, añadir la previsualización de un mensaje, iniciar sesión en una página web utilizando como método de autenticación nuestra cuenta de Telegram o inclusive, insertar un chat de discusión de cualquier canal público.

Las principales utilidades de los widgets de Telegram son las siguientes:

- **Botón para compartir:** el botón para compartir nos permite reenviar contenido desde un sitio web a cualquier conversación de nuestra cuenta de Telegram. Al presionar sobre el botón de compartir, el usuario deberá seleccionar el chat privado, grupo o canal donde quiere que sea enviado el mensaje, pudiendo editar el mismo previamente al envío.
- **Previsualización de mensajes:** los widgets de Telegram permiten incrustar mensajes de grupos y canales públicos en cualquier página web, pudiendo abrir dicho mensaje en la aplicación de mensajería en cualquier momento y desde cualquier lugar.
- **Inicio de sesión utilizando las credenciales de Telegram:** podemos iniciar sesión en una página web haciendo uso del widget de inicio de sesión de Telegram, siempre y cuando el administrador del sitio haya habilitado dicha opción. Para activar esta funcionalidad hace falta únicamente crear un bot de Telegram y enviar el comando “/setdomain” para vincular el dominio de nuestro sitio web con BotFather, por último, incluiremos el código fuente en nuestro sitio web.
- **Inclusión de chat de discusión:** unas de las opciones más interesantes que nos proporcionan los widgets de Telegram es la capacidad de poder incluir un chat de discusión en nuestra propia página web.

2.4.2. Stickers animados

Los stickers animados fueron incluidos en la versión 5.9 de Telegram (Telegram, 2019). Cualquier usuario puede crear y cargar nuevos stickers, siendo necesario lo siguiente:

- Editor de gráficos vectoriales que nos permita exportar cualquier imagen en formato vectorial a Adobe After Effects para convertirlo en animación.
- Adobe After Effects.
- Extensión de Adobe After Effects llamada “Bodymovin” para poder exportar las animaciones en formato .tgs compatible con la plataforma “Telegram Animated Stickers”.

Para poder subir un sticker animado, el mismo, debe cumplir algunos requerimientos técnicos. El tamaño máximo del sticker de 512 x 512 píxeles, no pudiendo superar un peso de 64 KB, así mismo, la duración máxima de la animación de 3 segundos y todas las animaciones deben ejecutarse a 60 fotogramas por segundo.

2.4.3. Temas personalizados en la nube

Los temas personalizados en la nube fueron incluidos en la versión 5.11 de Telegram. Cualquier usuario puede crear y compartir sus temas personalizados en la nube.

Para la creación de un nuevo tema o modificación de un tema existente debemos acceder al [Editor de temas en línea](#)⁷ (Acceso por última vez 24-03-2021). Una vez que guardemos el tema, Telegram nos enviará un mensaje con un enlace para compartirlo.

Cada vez que editemos, guardemos y apliquemos cualquier cambio sobre nuestro tema, este se actualizará de forma automática para todos los usuarios que estén haciendo uso del mismo.

⁷ URL del servicio “Telegram Theme Editor”: <https://themes.contest.com/>

3. Bots y API de Telegram

3.1. Qué es un bot

Un bot (Telegram, 2015) es una aplicación de terceros programada para llevar a cabo determinadas tareas. Los bots están automatizados, por tanto, se ejecutan en función de sus instrucciones y sin necesidad de órdenes específicas de los humanos. Cualquier usuario de Telegram puede interactuar con un bot enviándole mensajes o comandos o incluso mencionando al mismo en un grupo dentro de la aplicación de mensajería instantánea.

3.2. Utilidades de un bot

A continuación, se exponen algunas utilidades prácticas de un bot de Telegram:

- **Recibir noticias personalizadas:** existen bots programados para actuar como un periódico inteligente, enviando así contenido relevante que sea de interés para un usuario.
- **Integración con otros servicios:** existe la posibilidad de acceder a servicios externos a través de bots de Telegram creados por terceros.
- **Pagos de usuarios de Telegram:** un bot puede ofrecer servicios de pago o funcionar como una tienda virtual.
- **Herramientas personalizadas:** un bot puede proporcionarnos cualquier tipo de información o herramienta, por ejemplo, pronósticos meteorológicos, traducciones a cualquier idioma o votaciones y encuestas.
- **Juegos:** un bot ofrece la posibilidad de jugar desde simples juegos arcade hasta disparos en 3D o incluso juegos de estrategia en tiempo real, haciendo uso del lenguaje de marcado de hipertexto HTML5.
- **Creación de servicios sociales:** Telegram permite crear cualquier tipo de bot y las posibilidades son infinitas. Un aspecto muy interesante es la capacidad de poder acceder a nuestra ubicación y nuestros intereses para así crear un bot capaz de conectar a gente que busque entablar una conversación con personas desconocidas en función de la proximidad y los intereses comunes.

3.3. Funcionamiento de un bot

El funcionamiento de los bots de Telegram es sencillo, siendo estos, cuentas especiales que no requieren de un número de teléfono adicional para su configuración o utilización.

Cualquier usuario puede interactuar con un bot de dos maneras:

- Abriendo un chat privado con el bot y enviándole mensajes o comandos al mismo.
- Añadiendo al bot a un grupo existente y mencionándolo escribiendo “@nombredelbot” seguido de una consulta, permitiendo así que cada integrante del grupo pueda interactuar con el mismo.

El servidor de Telegram que hace de intermediario maneja todo el cifrado y la comunicación entre el bot y la API de Telegram.

3.4. Principales diferencias entre usuarios normales y bots

Aunque los bots pueden añadirse y ser mencionados en cualquier grupo, al igual que un usuario normal, tienen algunas peculiaridades.

- Los bots no tienen estado “en línea” ni marcas de tiempo “por última vez”, en su lugar, la interfaz muestra la etiqueta “bot”.
- Los bots tienen almacenamiento en la nube limitado, por tanto, el servidor elimina los mensajes más antiguos tras ser procesados.
- Un bot no puede iniciar una conversación con un usuario, siendo el usuario el que debe enviarle un mensaje o agregarlo a un grupo primero.
- Todos los nombres de los bots terminan en “bot”.
- Cuando un bot es agregado a un grupo, solamente recibe los mensajes que empiecen con una barra inclinada “/”, respuestas a los propios mensajes del bot y mensajes de canales de los que el bot es miembro. El hecho de que el bot no reciba todos los mensajes hace que estemos seguros de que no nos robará información y permite al desarrollador del bot ahorrar muchos recursos, ya que no necesitará procesar miles de mensajes irrelevantes de forma diaria.

3.5. Ejemplos de bots importantes

3.5.1. Trello

La plataforma Ingram ha desarrollado un bot para Telegram, integrando así, el software de administración de proyectos Trello, capaz de gestionar y crear nuevas tarjetas correspondientes a las Historias de Usuario y tareas asociadas a la creación de un proyecto software.

El nombre del bot es [@trello_bot](https://t.me/trello_bot)⁸ (Acceso por última vez 15-04-2021) y fue desarrollado por la plataforma de código abierto, Ingram. A día de hoy, el bot es mantenido por el alemán Roman Khafizianov.

Las posibilidades del bot de Trello son infinitas, desde la creación de una nueva tarjeta en nuestro tablero de Trello hasta la reorganización y redistribución de la misma. En la Ilustración 3 podemos observar la solicitud de autorización por parte del bot y la selección del tablero con el que queremos interaccionar.

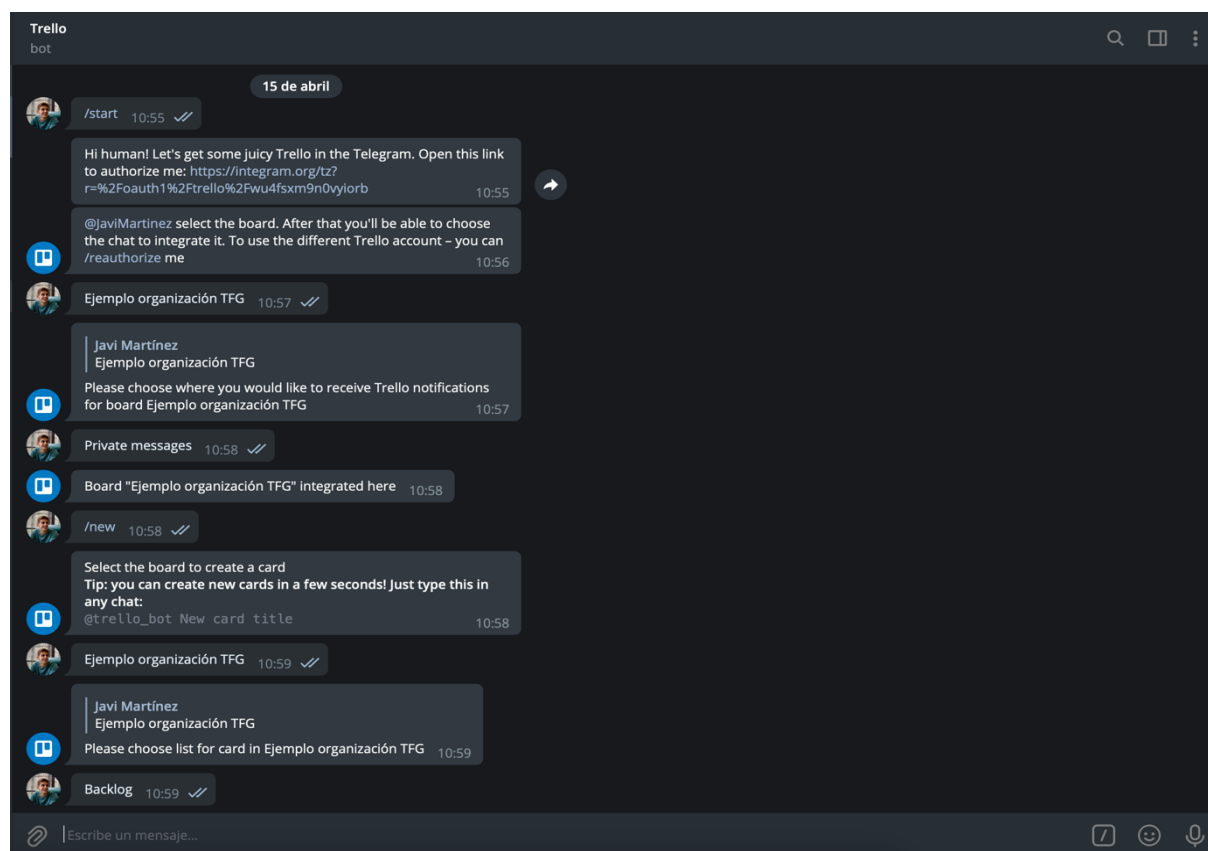


Ilustración 3: interacción con @trello_bot para la vinculación y creación de una nueva tarjeta

⁸ URL del bot de Trello: https://telegram.me/trello_bot

3.5.2. GitLab

Integram también permite la interlocución de la plataforma GitLab con Telegram gracias al bot [@gitlab_bot](https://telegram.me/gitlab_bot)⁹ (Acceso por última vez 15-04-2021).

La configuración con respecto a Trello es un poco más compleja y tediosa, pues hay que añadir una dirección URL dada por el bot a la sección “webhooks” del proyecto del que queramos obtener notificaciones de cada interacción con el repositorio de GitLab.

En la Ilustración 4 podemos observar cómo el bot nos facilita la URL a incluir en el apartado de “webhooks” de nuestro repositorio de GitLab. Así mismo, si una vez que añadimos el “webhook” hacemos click sobre el botón “Test”, obtendremos un ejemplo sobre las posibles notificaciones que el bot nos puede enviar al haber interacción en nuestro repositorio.

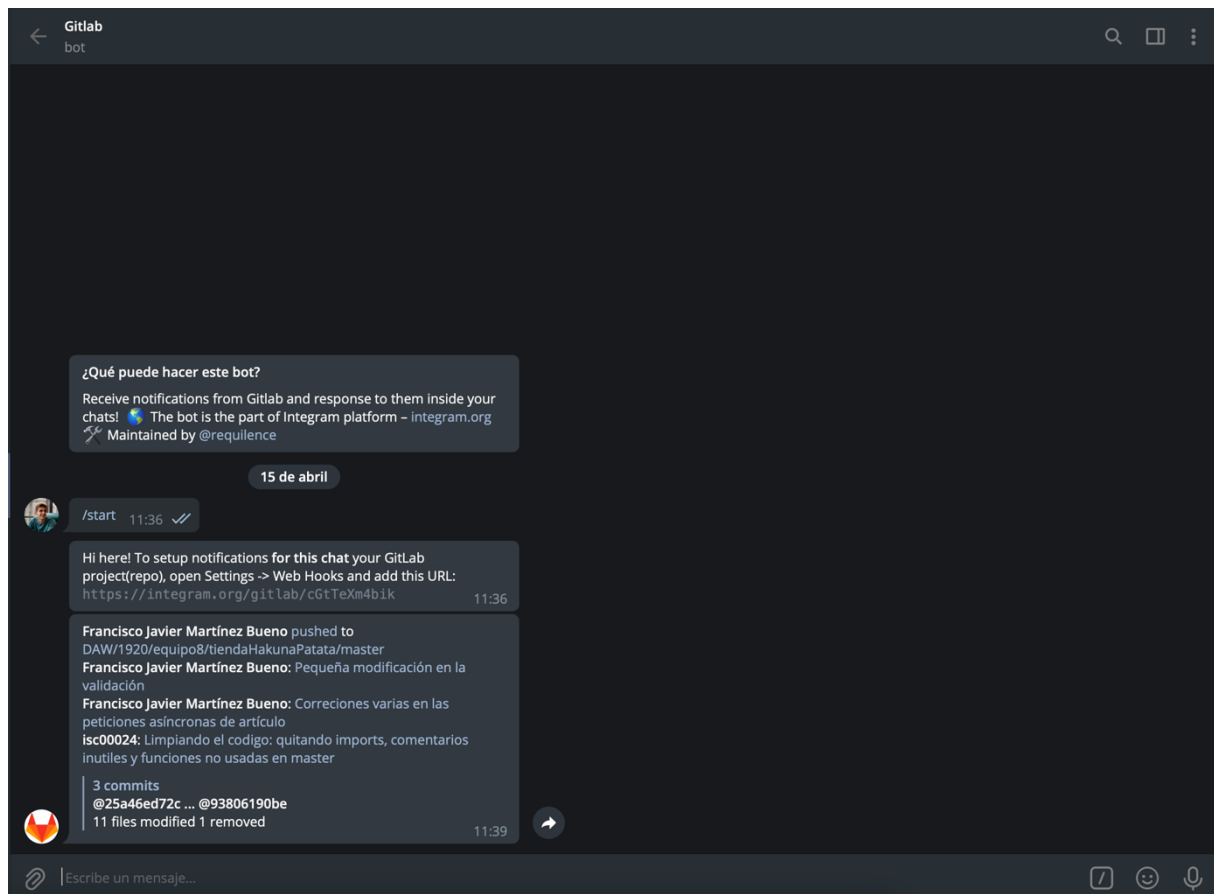


Ilustración 4: interacción con [@gitlab_bot](https://telegram.me/gitlab_bot) para la vinculación con un repositorio de GitLab

⁹ URL del bot de GitLab: https://telegram.me/gitlab_bot

3.5.3. GitHub

El bot de Telegram de Github nos facilita la interacción con los repositorios que tenemos en dicha plataforma de desarrollo colaborativo, pudiendo ser notificados cuando alguien haga alguna modificación sobre alguno de nuestros repositorios públicos.

El nombre del bot es [@GitHubBot](https://t.me/githubbot)¹⁰ (Acceso por última vez 24-06-2021) y fue desarrollado por Igor Zhukov, quien también creó Webogram en 2014, un proyecto sobre la que está basada la actual aplicación web de Telegram.

La utilización del bot es sencilla e intuitiva, siendo únicamente necesario acceder a un enlace para permitir al bot el acceso a nuestra cuenta de GitHub.

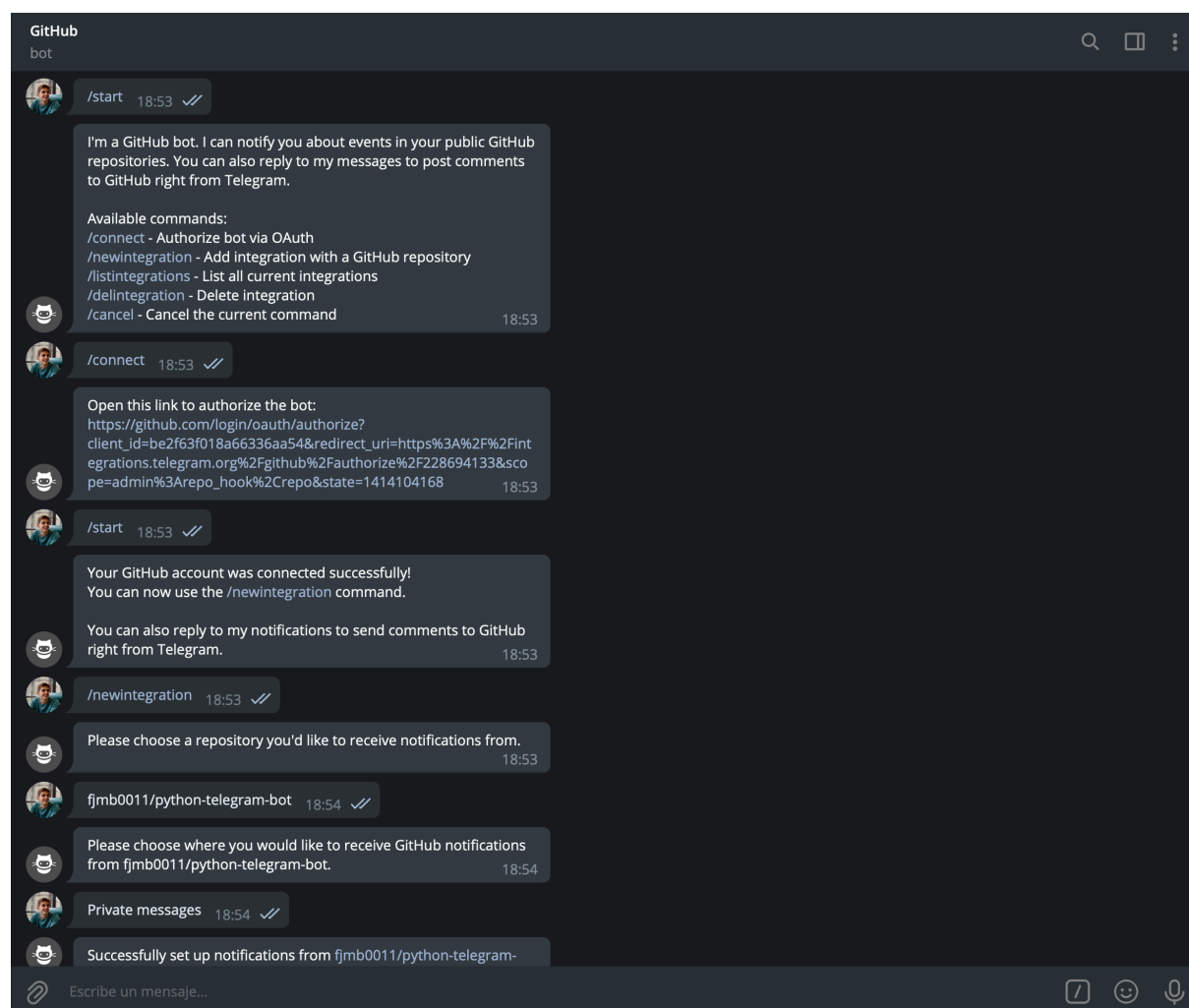


Ilustración 5: interacción con @GitHubBot para la vinculación con un repositorio de GitHub

¹⁰ URL del bot de GitHub: <https://telegram.me/githubbot>

3.6. Qué es una API

API son las siglas de “Application Programming Interface” (Siegel, 2020), es decir, interfaz de programación de aplicaciones. Una API es un conjunto de funciones y procedimientos que permiten la creación de aplicaciones que acceden a datos y características de otras aplicaciones o servicios.

Telegram proporciona dos tipos de APIs para desarrolladores de forma gratuita, ambas explicadas a continuación.

3.7. Telegram Bot API

Telegram Bot API (Telegram, 2015) nos permite conectar un bot al sistema de Telegram, sirviendo así como una interfaz entre el usuario y el código que se ejecuta en algún lugar del servidor.

Para hacer uso de esta API no es necesario saber nada sobre el protocolo de cifrado MTProto de Telegram ya que el propio servidor intermediario del cliente de mensajería manejará toda la encriptación y comunicación con la API. Todas las consultas a la Bot API de Telegram deben ser enviadas a través del protocolo HTTPS.

Al crear un bot obtenemos un token de autenticación único compuesto por letras y números, del tipo *0123456789:ABF0J5LKKjhY-ojtY4sPOHHTqDYk4JIMHvZ*, que utilizaremos para hacer peticiones a la API de Telegram, tal y como se observa en la Ilustración 9, dentro del apartado “7.2 Configuración inicial del bot utilizando BotFather”.

Obteniendo actualizaciones

Hay dos formas mutuamente excluyentes de recibir actualizaciones para nuestro bot: mediante el método `getUpdates` o a través de un webhook. Estos son algunos de los métodos más utilizados para obtener actualizaciones sobre nuestro bot:

- **getUpdates:** este método se utiliza para recibir actualizaciones entrantes mediante un sondeo largo, es decir, el servidor enviará una respuesta al cliente cuando tenga información disponible para el mismo.
- **setWebhook:** un webhook nos permite especificar una URL y recibir actualizaciones a través de la misma cuando se produce un evento, haciendo una llamada a esa dirección web mediante una solicitud HTTPS POST.
- **deleteWebhook:** este método se utiliza para eliminar la integración del webhook en caso de que queramos obtener actualizaciones a través del método getUpdates.

Algunos de los métodos disponibles

Los métodos de la API no distinguen entre mayúsculas y minúsculas. Una llamada exitosa a la API devuelve un objeto JSON con el resultado. Estos son algunos de los métodos disponibles más utilizados:

- **getMe:** es un método simple para probar el token de autenticación de nuestro bot que devuelve información sobre el mismo.
- **sendPhoto/sendAudio/sendDocument:** estos métodos nos permiten enviar fotos, audios y documentos. Si el envío se ha sido realizado con éxito, el mensaje enviado es devuelto.
- **getFile:** este método nos permite obtener información básica sobre un archivo y descargar el mismo. El tamaño máximo del archivo que puede descargar un bot es de 20 MB. Si el envío se ha realizado con éxito, el bot devuelve un objeto File.

3.7.1. BotFather

BotFather es un bot desarrollado por Telegram capaz de crear nuevos bots y cambiar la configuración de los existentes.

Para la creación de un nuevo bot, BotFather nos pedirá un nombre y un nombre de usuario para posteriormente generar un token de autorización para nuestro bot. El nombre de usuario de nuestro bot debe terminar con la palabra “bot”.

El token de autorización es necesario para autorizar al bot y enviar solicitudes a la API del mismo.

BotFather nos facilita el cambio de nombre de nuestro bot, el cambio de la descripción del mismo, las imágenes de perfil del bot, etc. a través de comandos.

Unas de las principales características de Telegram es su velocidad, por tanto, el bot también debe ser veloz. Telegram enviará alertas de estado a los desarrolladores de los bots si encuentra alguna anomalía en la cantidad de respuestas y la tasa de conversión de solicitudes/respuestas.

3.7.2. API de pagos

Los desarrolladores de bots tienen acceso a la API de pagos de Telegram para aceptar pagos de usuarios de Telegram en todo el mundo.

Los bots que implementan pagos son capaces de enviar mensajes de facturas, incluyendo estos una foto y una descripción del producto junto con un botón Pagar y la cantidad a desembolsar. Una vez que hacemos click sobre ese botón, se abre una interfaz de pago especial de Telegram, donde se solicitan detalles adicionales como dirección de envío, número de teléfono o dirección de correo electrónico.

Una vez ingresada la información necesaria, se reenvía al bot, para seguidamente ofrecer varias opciones de envío en función de la dirección de envío introducida. Una vez seleccionado el envío, el usuario deberá elegir una de las tarjetas de crédito guardadas o introducir la información de una nueva tarjeta. También podemos elegir como método de pago Apple Pay o Google Pay. Telegram no procesa los pagos de los usuarios, siendo estos gestionados por proveedores de pagos en todo el mundo.

Telegram no cobra ninguna comisión por utilizar la API de pagos; sin embargo, los proveedores de pagos sí que lo hacen, cobrando estos un porcentaje del importe de la transacción y una cantidad fija por cada compra del producto.

Algunos de los métodos disponibles

- **sendInvoice:** este método nos permite enviar facturas. En caso de éxito, el mensaje enviado es devuelto.
- **answerShippingQuery:** este método permite especificar un precio para el envío del producto en función de la dirección de envío del mismo.

3.8. TDLib y Telegram API

TDLib

TDLib (Telegram Database Library) es un cliente de Telegram (Telegram, 2018) multiplataforma para ayudar a los desarrolladores a crear sus propias aplicaciones personalizadas de Telegram.

TDLib ofrece una herramienta segura, encargándose por si misma de todos los detalles de implementación de red, cifrado y almacenamiento de datos locales, para que los desarrolladores puedan centrarse en el diseño y la interfaz, dejando así de lado el cómo hacerlos funcionar correctamente. TDLib se mantiene estable en conexiones a internet lentas y poco fiables.

Esta herramienta es compatible con todas las funciones de Telegram, haciendo que el desarrollo de aplicaciones sea muy sencillo para cualquier plataforma. La biblioteca es compatible con cualquier lenguaje de programación que pueda ejecutar funciones C.

Telegram API

La API de Telegram permite crear nuestros propios clientes totalmente personalizados, siendo nosotros los encargados de autorizar el acceso a un nuevo usuario a través de la verificación de su número de teléfono, acceder a datos a través de los métodos de la API y gestionar los errores de retorno de la misma.

4. Metodología de trabajo

El principal objetivo de este TFG es desarrollar un bot para la plataforma de mensajería Telegram. Para ello, es necesario conocer la metodología que vamos a utilizar para el desarrollo del software y las etapas en las que se suele dividir el desarrollo del mismo.

4.1. Metodología para el desarrollo del software

Actualmente podemos diferenciar (Mavuru, 2018) dos grandes grupos de metodologías para el desarrollo de software: las metodologías tradicionales y las metodologías ágiles. A continuación, haremos un breve análisis de las características de cada una de ellas.

4.1.1. Metodología tradicional

Las metodologías tradicionales de desarrollo de software están basadas en fases preorganizadas del ciclo de vida del desarrollo de software. El flujo de desarrollo es unidireccional, desde la definición inicial de los requisitos hasta el mantenimiento del producto final.

Los enfoques tradicionales son adecuados en sectores donde se comprenden bien los requisitos; sin embargo, en industrias tan cambiantes como las tecnologías de la información, los procedimientos de desarrollo tradicionales pueden no lograr los objetivos del proyecto.

Las principales desventajas de las metodologías tradicionales son la declaración del problema con mucha antelación, debiendo determinar la solución de antemano sin poder cambiarla ni modificarla, y la definición completa de todos los requisitos desde la fase inicial sin ninguna posibilidad de cambio o modificación después de comenzar con el desarrollo del proyecto.

En una metodología tradicional, el cliente no puede realizar evaluaciones intermediarias para asegurar el correcto desarrollo del producto, por tanto, de forma ocasional, los proyectos se descartan, siendo esto una muestra de ineficiencia y una pérdida de esfuerzo y dinero.

4.1.2. Metodología ágil

Una metodología ágil proporciona un enfoque preciso y fácil de usar para el desarrollo de software. Los clientes cuentan con la posibilidad de realizar modificaciones a lo largo de las fases del desarrollo del proyecto, aunque la declaración del problema y la posible solución del mismo se acuerden con anterioridad.

Las metodologías ágiles promueven un proceso de gestión de proyectos que fomenta la inspección y adaptación, con una filosofía de liderazgo que impulsa el trabajo en equipo, la autoorganización y la responsabilidad sobre el producto.

En las metodologías ágiles, el conjunto inicial de requisitos se divide en historias de usuario. El proyecto se divide en iteraciones, cada una de las cuales da lugar a un nuevo incremento o “release”. Una vez realizada la planificación de la iteración, las historias de usuario se dividen en tareas. Al final de cada iteración el cliente puede revisar el entregable y proponer algún cambio si realmente fuese necesario.

Fases del ciclo de vida del desarrollo de software en una metodología ágil

En cada incremento deberemos pasar por cada una de las siguientes etapas:

- **Requisitos:** para poder implementar una aplicación funcional con un código limpio es necesario que el equipo de desarrollo y la parte interesada se reúnan para recopilar los requisitos necesarios de cada “release”.
- **Diseño:** durante la fase de diseño, se crea la arquitectura de software detallada que cumple con los requisitos específicos del incremento. Esta etapa es de gran importancia, pues un buen diseño añadirá valor a nuestro producto y nos ayudará a previsualizar como será el resultado y funcionamiento de nuestros entregables.
- **Implementación:** la fase de implementación consiste en escribir todo el código perteneciente a cada historia de usuario de la “release” y convertir el diseño en parte del software real. Generalmente esta etapa del ciclo de vida del desarrollo de software es la más larga, ya que el

equipo de ingeniería del software debe asegurarse de que su código cumpla con todos los requisitos de las partes interesadas en el desarrollo del producto.

- **Pruebas:** una vez que las Historias de Usuario del incremento están totalmente implementadas, comienza la fase de pruebas del sistema. Los ingenieros de control de calidad son los encargados de probar el software utilizando una amplia gama de frameworks para la realización de pruebas unitarias, garantizando así la limpieza y claridad del código para cumplir con los objetivos comerciales de la solución.
- **Despliegue:** es el mecanismo mediante el cual las aplicaciones, módulos, actualizaciones y parches se entregan a los usuarios, de forma que puedan hacer uso del software para posteriormente facilitarnos una opinión acerca de su experiencia de uso.
- **Revisión:** la etapa final del ciclo de vida del desarrollo de software incluye actualizaciones periódicas y mantenimiento. Durante esta etapa, el producto se actualiza, mejora y afina de acuerdo con los comentarios de los interesados sobre el rendimiento del mismo, incluyendo modificaciones en función de las necesidades reales del usuario final.

En la Ilustración 6 podemos ver el ciclo de vida del desarrollo de software de una metodología ágil.



Ilustración 6: ciclo de vida del desarrollo de software en una metodología ágil

Anurina, O. (2021). Agile SDLC: Skyrocketing Your Project with Agile Principles. [Ilustración]. Traducción propia. Recuperado de <https://mlsdev.com/blog/agile-sdlc> (Acceso por última vez 20-04-2021)

Kanban

Kanban es un método de administración de flujo de trabajo para definir y administrar el desarrollo de software de manera ágil. Esta metodología consiste en visualizar el trabajo, limitar el trabajo en curso y maximizar la eficiencia, reduciendo así el tiempo que se tarda en llevar un proyecto del principio al fin del proceso.

Un tablero Kanban suele contar con las columnas básicas “Backlog”, “Seleccionado para desarrollo”, “En curso” y “Listo” para comenzar a visualizar nuestro trabajo. Las tareas o trabajos que deben realizarse se representan mediante tarjetas. Al mirar a nuestro tablero Kanban podremos ver fácilmente cómo progresa nuestro proyecto.

En Kanban se debe limitar el trabajo en proceso, es decir, debemos limitar el WIP (Work In Progress). Los límites de trabajo en proceso restringen la cantidad máxima de tarjetas en las diferentes etapas del flujo de trabajo. Los límites de WIP ayudan al equipo a enfocarse únicamente en tareas actuales, permitiendo así terminar más rápido con elementos de trabajo individuales.

Utilizaremos Jira Software, desarrollado por Atlassian, para la administración de tareas de nuestro proyecto, siguiendo una metodología ágil Kanban.

En la Ilustración 7 podemos observar el tablero de Jira Software; cuenta con 4 columnas, “Backlog”, “Seleccionado para desarrollo”, “En curso” y “Listo”, creadas automáticamente a partir de la plantilla “Kanban” tras seleccionar “Proyecto clásico” como tipo de proyecto.

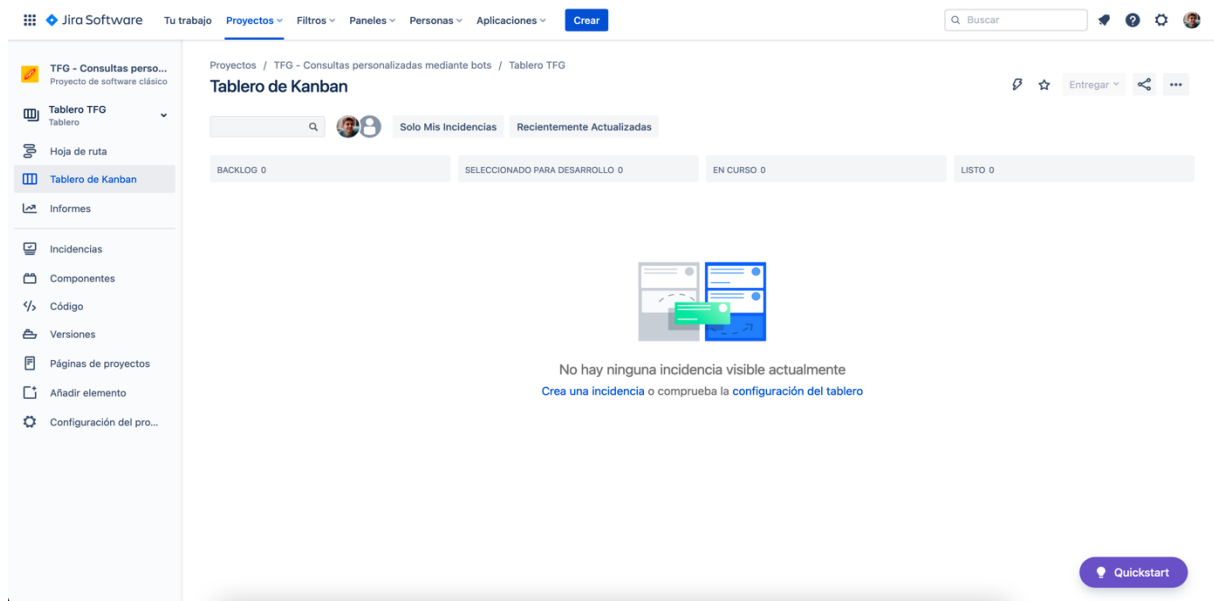


Ilustración 7: tablero de Jira Software creado a partir de una plantilla Kanban

4.1.3. ¿Por qué utilizar una metodología ágil?

Una metodología tradicional sigue fases estrictas, respetando los requisitos originales y el plan de diseño creado al comienzo del proyecto, mientras que en una metodología ágil pueden definirse e incluirse nuevos requisitos en cualquier etapa. Actualmente no tenemos muy claros los requisitos de nuestro software, por tanto, hacer uso de una metodología ágil es lo más lógico y coherente.

Así mismo, el tiempo con el que contamos para desarrollar el software es limitado, y entregar un software que funcione haciendo uso de una metodología tradicional lleva mucho tiempo; sin embargo, en una metodología ágil, tras cada iteración, el cliente obtiene un entregable funcional.

Por otra parte, el líder del proyecto facilita el trabajo al equipo de desarrollo, eliminando así los cuellos de botella y ayudando al equipo a mantenerse enfocado para entregar iteraciones de software de manera regular, proporcionando un mejor resultado del proyecto desarrollado.

En una metodología tradicional, el proceso de prueba comienza una vez finalizado el desarrollo, por tanto, hay muchas posibilidades de encontrar errores tras el desarrollo del mismo, siendo muy costoso corregirlos. En una metodología ágil, el software se va entregando conforme se van desarrollando los requisitos.

5. Planificación del proyecto

5.1. Análisis de riesgos

El análisis de riesgos es la estimación del grado de exposición a que una amenaza se materialice sobre uno o más activos causando daños o perjuicios a la organización.

El análisis y gestión de riesgos es parte fundamental de sistema de seguridad y deberá mantenerse actualizado. La inmensa mayoría de las veces es obligatorio aplicar el Esquema Nacional de Seguridad (Esquema Nacional de Seguridad - ENS, 2019), que tiene por objeto establecer la política de seguridad en la utilización de medios electrónicos y está constituido por principios básicos y requisitos mínimos que permitan una protección adecuada de la información tratada.

Una de las formas más utilizadas para analizar riesgos es determinar la exposición o impacto en cada riesgo identificado. La exposición de un riesgo puede estimarse a partir de la probabilidad de pérdida no esperada, en formato porcentaje, multiplicada por la magnitud de pérdida, en semanas.

En la tabla 2 podemos ver una aproximación de los riesgos encontrados a la hora de desarrollar nuestro software.

Riesgo	Probabilidad de pérdida	Magnitud de la pérdida (semanas)	Exposición a riesgo (semanas)
Diseño inadecuado del software	0,2 (20%)	6	1,2
Diseño del software en un lenguaje no explorado previamente	0,3 (30%)	4	1,2
Metodología de desarrollo del software inapropiada	0,1 (10%)	2	0,2
Difícil utilización de la API de Telegram	0,2 (20%)	4	0,8
Poca comunicación por parte del cliente	0,4 (40%)	4	1,6
Mala estimación de tiempos	0,1 (10%)	2	0,2

Tabla 2: exposición de los riesgos del proyecto

Tras calcular la exposición de cada riesgo podemos averiguar la media de retraso total sin gestión de riesgos, siendo esta de 0,9 semanas aproximadamente.

¿Qué hacemos con el riesgo?

- Asumirlo: cuando el riesgo es muy bajo, asumimos el riesgo puesto que no afectará demasiado.
- Eliminarlo: eliminar la posibilidad de que ese riesgo se de.
- Mitigarlo: aplicamos salvaguardas o controles para reducir ese riesgo a unos niveles aceptables.
- Compartirlo: se puede contratar un seguro para la empresa que a cambio de una prima se hace cargo de un posible problema futuro.

5.2. Planificación temporal

Teniendo como base el período docente establecido por la Universidad de Jaén durante el segundo cuatrimestre del curso 2020/2021, y como último día para la entrega del proyecto la fecha de finalización del periodo de entrega de Trabajos de Fin de Grado, se ha planteado la posible planificación de horas respecto a los días lectivos entre el día 1 de marzo de 2021 y 30 de junio de 2021, con un total de 370 horas, contando que se trabajan unas 5 horas diarias, durante 74 días lectivos.

- Día de comienzo del proyecto: 1 de marzo de 2021.
- Día de finalización del proyecto: 30 de junio de 2021.
- Duración total del proyecto: 74 días.
- Horas diarias trabajadas: 5 horas.
- Horas totales trabajadas para la realización del proyecto: 370 horas.

5.3. Estimación del coste del proyecto

El desarrollo del software y la especificación del mismo se llevará a cabo por un único ingeniero de software, que trabajará desde principios del mes de marzo hasta finales del mes de junio, a razón de 5 días a la semana, de lunes a viernes, con un desempeño diario de 5 horas. El salario medio de un desarrollador software junior en

España es de aproximadamente 1.100,00€¹¹ netos al mes, por lo que su salario mensual será de 687,50€, trabajando unas 100 horas mensuales.

Así mismo, hay que tener en cuenta el desgaste del equipo informático con el que se desarrollará nuestro software. Un equipo informático correcto para el desarrollo de nuestro software puede costar unos 700,00€, con una vida media de 4 años, es decir, 48 meses, por tanto, el gasto por el equipo será de unos 15,00€ mensuales. Además, deberemos tener en cuenta el gasto de luz y material de oficina que usaremos durante los meses de desarrollo del software, que será aproximadamente unos 10,00€ al mes. La suma total de todos los gastos supondrá un desembolso mensual de 712,50€.

Teniendo en cuenta que el mantener el bot de Telegram no supondrá inicialmente un coste adicional y que toda la información se almacenará en los servidores de Google, ya que utilizaremos Google Sheets, podemos concluir el coste total del proyecto en $712,50€ \times 4 \text{ meses} = \mathbf{2.850,00€}$, tal y como podemos observar en el desglose de la tabla número 3 mostrada a continuación.

Descripción	Meses	Coste/mes	Subtotal	Total	% del total
1. Gestión del proyecto				2.750,00€	96,50%
1.1 Equipo de desarrollo	4	687,50€	2.750,00€		
2. Hardware				60,00€	2,10%
2.1 Dispositivo	4	15,00€	60,00€		
3. Suministros				40,00€	1,40%
3.1 Luz	4	5,00€	20,00€		
3.2 Material de oficina	4	5,00€	20,00€		
Coste total estimado del proyecto				2.850,00€	

Tabla 3: estimación del coste total del proyecto

El monto final obtenido hace alusión a la inversión inicial necesaria aproximada para poder hacer frente económicamente al desarrollo del software planteado.

¹¹ Fuente: <https://es.indeed.com/career/desarrollador-junior/salaries>

6. Análisis del proyecto

El análisis del proyecto ayuda al desarrollador a entender con mayor facilidad el futuro funcionamiento del software, redactándose en él un posible conjunto inicial de historias de usuario y los escenarios por los que pasará tanto un profesor como un estudiante a la hora de interactuar con el software creado.

6.1. Componentes lógicos

Para la correcta implementación de nuestro bot de Telegram haremos uso de los siguientes componentes lógicos:

- **Hoja de cálculo de Google Sheets:** utilizaremos este programa de hojas de cálculo incluido en el paquete gratuito de editores de documentos de Google basado en la web para almacenar toda la información relativa al desempeño de un estudiante en una asignatura. Así mismo, la herramienta para editar secuencias de comandos de Google Sheets nos ayudará a automatizar la consulta de información en nuestra hoja de cálculo de forma dinámica.

Los profesores serán los únicos usuarios capaces de acceder a la hoja de cálculo con toda la información sobre el desempeño de los estudiantes.

- **Bot de Telegram:** crearemos un nuevo bot utilizando BotFather, para que posteriormente, el mismo pueda hacer peticiones a nuestra hoja de cálculo de Google a través de un webhook.

Los estudiantes matriculados en una asignatura podrán registrarse para tener acceso al bot y así hacer consultas sobre su desempeño en las distintas asignaturas.

Los profesores también podrán registrarse en el bot para hacer pruebas y consultas sobre la hoja de cálculo que ellos hayan cumplimentado.

6.2. Conjunto inicial de historias de usuario

El conjunto inicial de Historias de Usuario, clasificadas en función de la prioridad de cada una de ellas, es el siguiente:

Alta prioridad

- Un profesor debe poder decidir qué columnas de la hoja de cálculo van a estar accesibles para un estudiante y cuales no.
 - **COMO** profesor **QUIERO** poder decidir cuales de las columnas de la hoja de cálculo van a estar accesibles para el estudiante y cuales no **PARA** poder tener cierto control sobre la información mostrada
- Un estudiante debe poder preguntar por algunos de los aspectos recogidos en la hoja de cálculo.
 - **COMO** estudiante **QUIERO** poder preguntar por algunos de los aspectos recogidos en la hoja de cálculo **PARA** que esté informado de ese aspecto.
- Un profesor debe poder bloquear temporalmente el acceso y volver a darlo.
 - **COMO** profesor **QUIERO** poder bloquear temporalmente el acceso y volver a darlo, de forma que, si el estudiante solicita un dato, este reciba un mensaje diciendo que la base de datos está en mantenimiento **PARA** que cuando esté haciendo cálculos no puedan acceder y obtener datos erróneos.
- Un estudiante debe poder acceder únicamente a sus notas.
 - **COMO** estudiante **QUIERO** poder acceder únicamente a mis notas y no a las de otros compañeros **PARA** mantener así la privacidad de las mismas.
- Un estudiante debe poder consultar qué órdenes le puede dar al bot.
 - **COMO** estudiante **QUIERO** poder consultar qué órdenes le puedo dar al bot **PARA** tener acceso a todos los comandos para interactuar con el mismo.

Baja prioridad

- Un profesor debe poder elegir con qué nombre se refiere a una columna/dato.
 - **COMO** profesor **QUIERO** poder decidir con qué nombre se refiere a una columna/dato, pudiendo ser ese nombre distinto al nombre de la columna de la hoja de cálculo **PARA** poder personalizar cada dato en función de las necesidades.
- Un estudiante debe poder acceder a cada una de sus notas en distintas asignaturas.
 - **COMO** estudiante **QUIERO** poder acceder a cada una de mis notas en distintas asignaturas **PARA** que el sistema esté disponible para cuantas más asignaturas, mejor.
- Un profesor debe poder tener acceso a estadísticas.
 - **COMO** profesor **QUIERO** tener acceso a estadísticas, como por ejemplo, saber qué personas han accedido, qué dato han preguntado y en qué momento lo han preguntado **PARA** llevar un control sobre el uso del bot.
- Un profesor debe poder configurar grupos de columnas bajo un solo identificador.
 - **COMO** profesor **QUIERO** poder configurar grupos de columnas bajo un solo identificador, de forma que cuando el alumno pregunte por ese identificador, se devuelva la información de varias columnas **PARA** obtener una visión más general acerca del desempeño del alumno en una asignatura
- Un estudiante debe poder preguntar de forma fácil por un grupo de columnas preconfiguradas o creadas por un profesor.
 - **COMO** estudiante **QUIERO** poder preguntar de forma fácil por un grupo de columnas (Por ejemplo: ¿Cuáles son mis notas en prácticas?, Ver informe) **PARA** obtener una visión más general acerca de mi desempeño en una asignatura

- Un estudiante debe ser capaz de poder contactar con el profesor en caso de que tengan alguna duda.
 - **COMO** estudiante **QUIERO** poder mandar un mensaje al profesor encargado de una asignatura por Telegram o por correo electrónico **PARA** solucionar las dudas lo más rápido posible.
- Un estudiante debe poder gestionar una cita para tutoría.
 - **COMO** estudiante **QUIERO** poder gestionar una cita para tutoría **PARA** poder tener una buena planificación temporal.
- Un profesor debe poder gestionar una cita para tutoría a un estudiante.
 - **COMO** profesor **QUIERO** poder gestionar una cita para tutoría **PARA** poder tener una buena planificación temporal.
- Un estudiante debe poder consultar el enlace web en el que se desarrollará una videoconferencia.
 - **COMO** estudiante **QUIERO** poder consultar el enlace web en el que se desarrollará una videoconferencia **PARA** poder acceder a la misma de forma rápida.
- Un estudiante debe tener acceso a un apartado de "Sistema evaluación".
 - **COMO** estudiante **QUIERO** que haya un apartado de "Sistema evaluación" **PARA** solucionar las dudas más frecuentes acerca de mi evaluación de una asignatura.

6.3. Escenarios basados en las historias de usuario iniciales

Los escenarios previos al desarrollo del software son útiles para entender los aspectos que directa o indirectamente intervienen en el proceso interactivo entre el cliente y el producto desarrollado.

La redacción de los posibles escenarios supone un punto de partida idóneo para la fase de desarrollo de nuestro software.

A continuación, se exponen los posibles escenarios de nuestro bot de Telegram partiendo de las historias de usuario iniciales:

- **Un profesor debe poder decidir qué columnas de la hoja de cálculo van a estar accesibles para un estudiante y cuales no:**
 1. El profesor accede a la hoja de cálculo de Google Sheets.
 2. El profesor se posiciona sobre la columna que quiere que sea accesible o inaccesible.
 3. El profesor escribe “Accesible” si quiere que sea accesible o cualquier otro valor si quiere que la columna sea inaccesible, en el campo habilitado para ello.
- **Un estudiante debe poder preguntar por algunos de los aspectos recogidos en la hoja de cálculo:**
 1. El estudiante accede al bot de Telegram.
 2. El estudiante envía un comando.
 3. El bot de Telegram contestará al estudiante con información sobre su desempeño en la asignatura.
- **Un profesor debe poder bloquear temporalmente el acceso y volver a darlo:**
 1. El profesor accede a la hoja de cálculo de Google Sheets.
 2. El profesor se posiciona sobre la celda que determina la accesibilidad de una hoja de cálculo.
 3. El profesor despliega el desplegable.
 - a. El profesor selecciona “Sí” si quiere que los datos de la hoja de cálculo sean accesibles.
 - b. El profesor selecciona “No” si quiere que los datos de la hoja de cálculo no sean accesibles.
 - c. El profesor deberá indicar el motivo de la inaccesibilidad.
- **Un estudiante debe poder acceder únicamente a sus notas:**
 1. El estudiante accede al bot de Telegram por primera vez.
 2. El estudiante se registra para poder guardar su ID y el correo electrónico asociado a la misma
 3. El estudiante verifica su correo electrónico enviándole al bot el código de verificación que ha recibido en el correo electrónico facilitado.
 4. El estudiante pide información sobre su desempeño en una asignatura haciendo uso del comando “/informacionestudiante”.

5. El estudiante recibe únicamente información relativa a su desempeño en la asignatura.

- **Un estudiante debe poder consultar qué órdenes le puede dar al bot:**
 1. El estudiante accede al bot de Telegram.
 2. El estudiante escribe “/”.
 - a. El bot de Telegram le muestra una lista con todos los comandos disponibles.
 3. El estudiante escribe el comando “/ayuda”.
 - a. El bot de Telegram le devuelve un mensaje con todos los comandos disponibles.
- **Un profesor debe poder elegir con qué nombre se refiere a una columna/dato:**
 1. El profesor accede a la hoja de cálculo de Google Sheets.
 2. El profesor se posiciona sobre una columna con información sobre el desempeño de un estudiante en su asignatura.
 3. El profesor cambia el valor del título de la columna por el que crea conveniente.
- **Un estudiante debe poder acceder a cada una de sus notas en distintas asignaturas:**
 1. El estudiante accede al bot de Telegram.
 2. El estudiante escribe el comando “/informacionestudiante”.
 3. El bot de Telegram le devuelve una lista con el código y el nombre de las asignaturas en las que está matriculado.
 4. El estudiante vuelve a escribir el comando “/informacionestudiante” seguido de un espacio y el código de la asignatura.
 5. El bot de Telegram le muestra una lista con información sobre su desempeño en una asignatura.
- **Un profesor debe poder tener acceso a estadísticas:**
 1. El profesor accede a la hoja de cálculo de Google Sheets.
 2. El profesor se posiciona sobre la hoja de cálculo llamada “Estadísticas”.
 3. El profesor consulta información relativa a la interacción de los estudiantes con el bot.

- a. El profesor consulta qué personas han accedido al bot para consultar información sobre el desempeño del estudiante en su asignatura.
 - b. El profesor consulta qué dato le han preguntado al bot sobre el desempeño de un estudiante en su asignatura.
 - c. El profesor consulta en qué momento le han preguntado un dato al bot.
- **Un profesor debe poder configurar grupos de columnas bajo un solo identificador:**
 1. El profesor accede a la hoja de cálculo de Google Sheets.
 2. El profesor se posiciona sobre el grupo de columnas que quiere configurar bajo un solo identificador.
 3. El profesor le asigna a todas esas columnas un identificador único.
 - **Un estudiante debe poder preguntar de forma fácil por un grupo de columnas preconfiguradas o creadas por un profesor:**
 1. El estudiante accede al bot de Telegram.
 2. El estudiante escribe un comando para consultar los grupos de columnas configurados por el profesor bajo un único identificador.
 3. El bot de Telegram le envía un mensaje al estudiante con información relativa a la información que hay en el grupo de columnas.
 - **Un estudiante debe ser capaz de poder contactar con el profesor en caso de que tengan alguna duda:**
 1. El estudiante accede al bot de Telegram.
 2. El estudiante escribe el comando “/contactoprofesor”.
 3. El bot de Telegram le devuelve una lista con el código y el nombre de las asignaturas en las que está matriculado.
 4. El estudiante vuelve a escribir el comando “/contactoprofesor” seguido de un espacio y el código de la asignatura.
 5. El bot de Telegram le muestra una lista con información para contactar con el profesor encargado de la asignatura.
 - **Un estudiante debe poder gestionar una cita para tutoría:**
 1. El estudiante accede al bot de Telegram.
 2. El estudiante escribe el comando “/reservartutoria”.

3. El bot de Telegram le devuelve una lista con el código y el nombre de las asignaturas en las que está matriculado.
 4. El estudiante vuelve a escribir el comando “/reservartutoria” seguido de un espacio y el código de la asignatura.
 5. El bot de Telegram le muestra una lista con el horario disponible de los profesores de esa asignatura.
- **Un profesor debe poder gestionar una cita para tutoría a un estudiante:**
 1. El profesor accede a la hoja de cálculo de Google Sheets.
 2. El profesor accede a la hoja de cálculo con información relativa a las tutorías que él tiene disponibles.
 3. El profesor puede modificar el horario de la tutoría que un estudiante ha reservado y posteriormente el estudiante es notificado.
 4. El profesor puede eliminar una tutoría que el estudiante ha reservado y posteriormente el estudiante es notificado.
 5. El profesor puede asignar una cita para tutoría a un estudiante dado su correo electrónico y posteriormente el estudiante es notificado.
 - **Un estudiante debe poder consultar el enlace web en el que se desarrollará una videoconferencia:**
 1. El estudiante accede al bot de Telegram.
 2. El estudiante escribe el comando “/informaciondocencia”.
 3. El bot de Telegram le devuelve una lista con el código y el nombre de las asignaturas en las que está matriculado.
 4. El estudiante vuelve a escribir el comando “/informaciondocencia” seguido de un espacio y el código de la asignatura.
 5. El bot de Telegram le muestra una lista con los enlaces webs donde se desarrollan las videoconferencias de la asignatura.
 - **Un estudiante debe tener acceso a un apartado de "Sistema evaluacion":**
 1. El estudiante accede al bot de Telegram.
 2. El estudiante escribe el comando “/sistemaevaluacion”.

3. El bot de Telegram le devuelve una lista con el código y el nombre de las asignaturas en las que está matriculado.
4. El estudiante vuelve a escribir el comando “/sistemaevaluacion” seguido de un espacio y el código de la asignatura.
5. El bot de Telegram le muestra una lista con información relativa al sistema de evaluación de la asignatura.

7. Creación y configuración de nuestro bot de Telegram

7.1. Elección del nombre de usuario de nuestro bot

La elección del nombre de usuario se llevó a cabo haciendo una lluvia de ideas, tal y como podemos observar en la Ilustración 8.

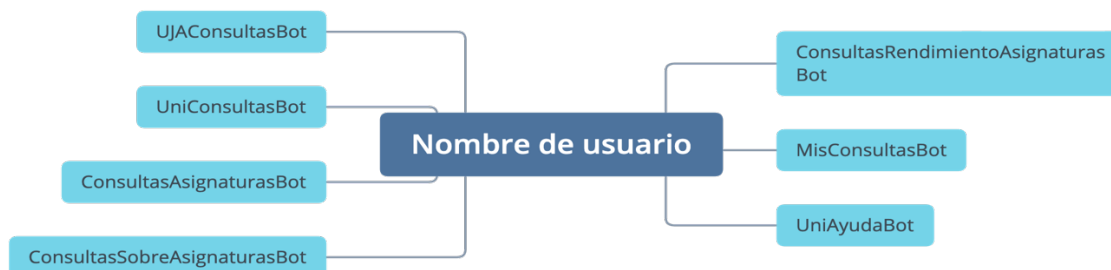


Ilustración 8: Lluvia de ideas sobre los posibles nombres de usuario

Después de plantear varios posibles nombres de usuario y tras la revisión de cada uno de ellos, concluí que UniConsultasBot, resultado de la combinación de las abreviaturas de “Universidad” y “Consultas”, era el más adecuado, ya que resume de manera clara y concisa el propósito por el cuál este Bot fue creado.

7.2. Configuración inicial del bot utilizando BotFather

Telegram ofrece la posibilidad de crear un bot con ayuda de BotFather, el propio bot de Telegram para llevar un control de todos nuestros bots. Para ello, abriremos una conversación con @BotFather, pulsaremos el botón “Iniciar” y teclearemos el comando “/newbot”. Seguidamente, BotFather nos pedirá un nombre y un nombre de usuario, en nuestro caso, UniConsultas y UniConsultasBot respectivamente. Al terminar la configuración inicial, el bot nos enviará nuestro token de autenticación único para hacer peticiones a la API de Telegram.

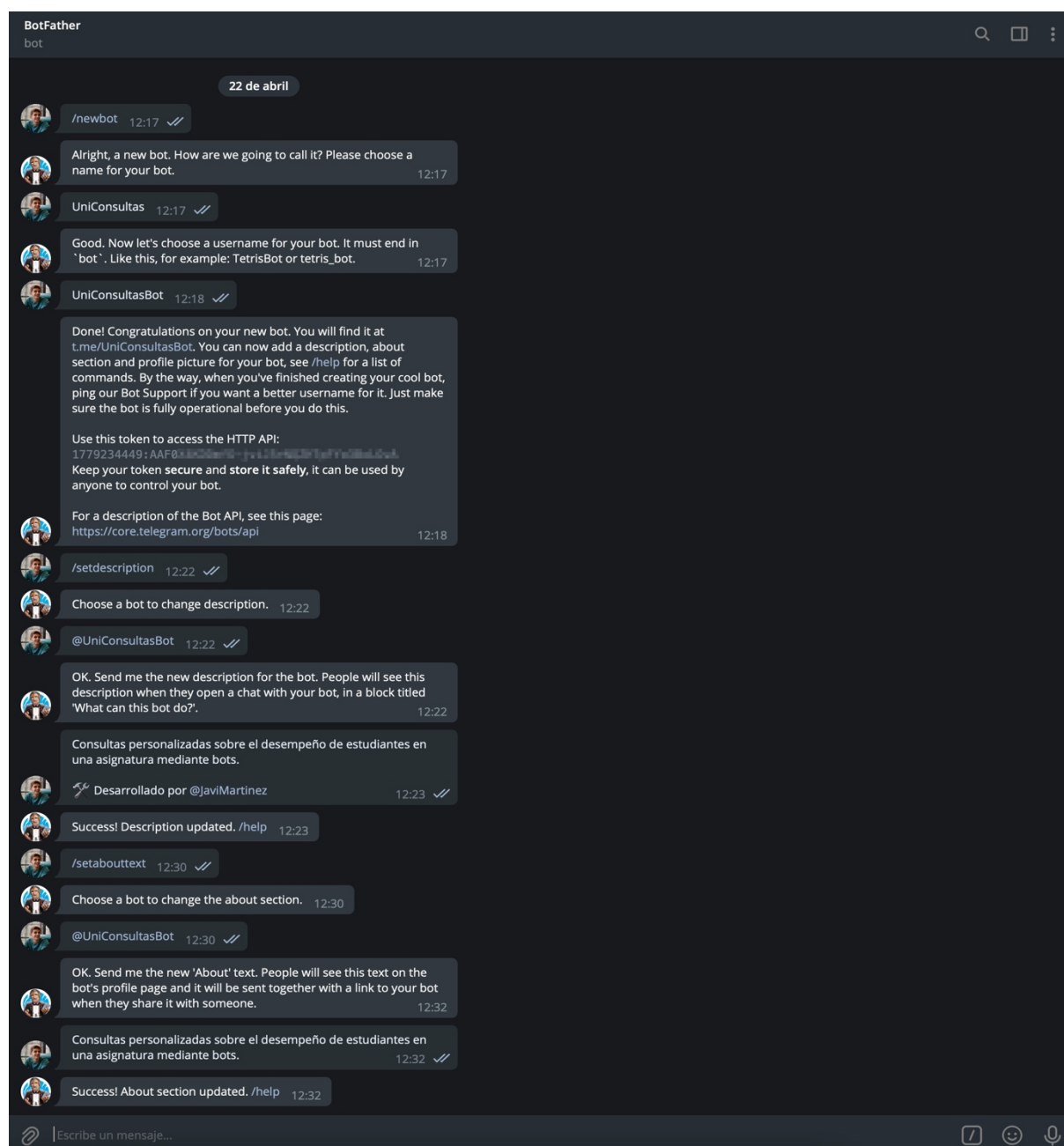


Ilustración 9: creación y configuración inicial del bot

7.3. Creación y configuración de la hoja de cálculo y Google Apps Script

El bot de Telegram accederá a una hoja de cálculo de Google Sheets para consultar información relativa al desempeño de un estudiante en una asignatura. Así mismo, el código encargado de consultar la información de la hoja de cálculo se almacenará dentro del apartado “Editor de secuencia de comandos” de nuestra hoja de cálculo.

Lo primero que haremos será crear una hoja de cálculo en blanco y acceder al “Editor de secuencia de comandos”, donde desarrollaremos nuestro código inicial de configuración para establecer la conexión con nuestro bot de Telegram a través de un webhook de Google. En la ilustración 10 podemos ver varias funciones creadas para la configuración inicial:

- **function getMe():** función que nos muestra información relativa a la configuración de nuestro bot de Telegram, en formato JSON.
- **function getUpdates():** función que nos muestra información sobre cualquier interacción que tiene un usuario con nuestro bot, en formato JSON. Este método es únicamente funcional cuando no tenemos nuestro webhook activo.
- **function setWebhook():** función para establecer nuestra hoja de cálculo como la aplicación a la que debe hacer peticiones nuestro bot de Telegram.

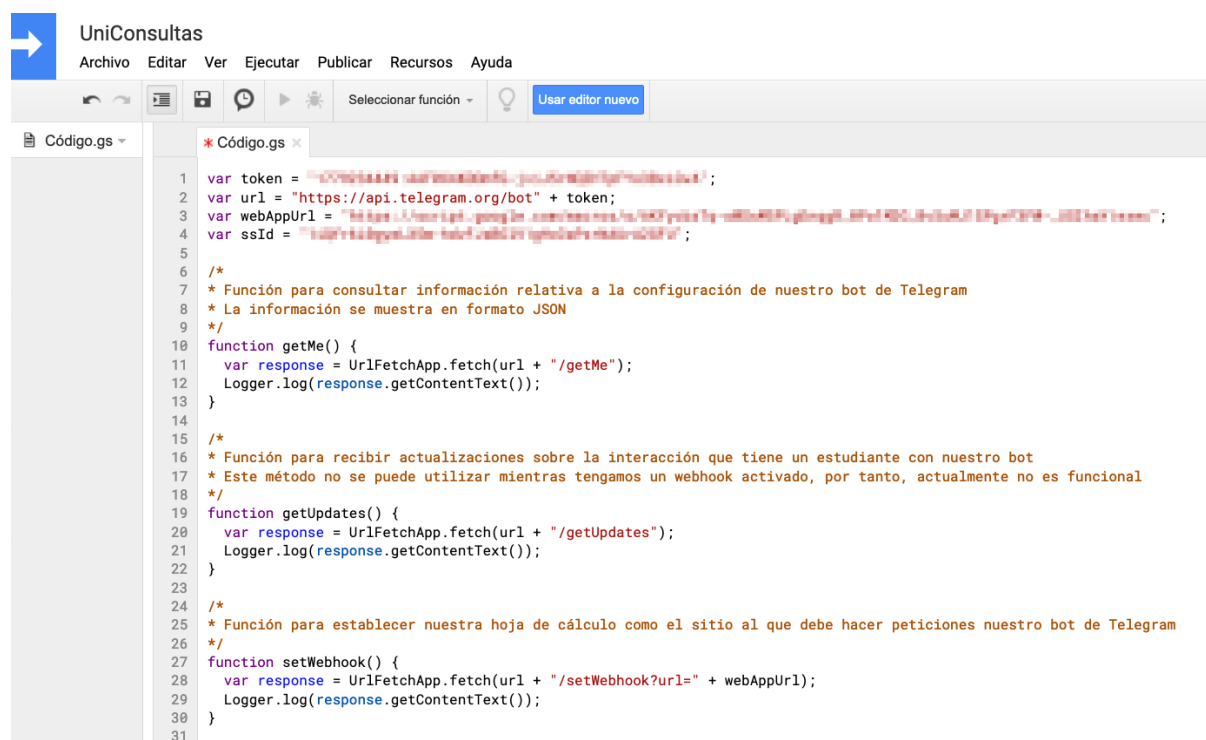


Ilustración 10: configuración inicial de la hoja de cálculo y conexión con el bot de Telegram a través de un webhook de Google

Al intentar conectarnos a la API de Telegram y hacer una petición “/getMe” para consultar información relativa a nuestro bot, Google nos pedirá permitir dicha acción a través de una ventana externa.

Así mismo, en un futuro, Google volverá a pedirnos permitir el acceso a GMail para que nuestro Bot de Telegram pueda enviar correos electrónicos a los estudiantes para verificar el correo electrónico institucional.

También deberemos concederle permisos para acceder a la hoja de cálculo y consultar cualquier información necesaria de la misma, tal y como podemos observar en la ilustración 11.

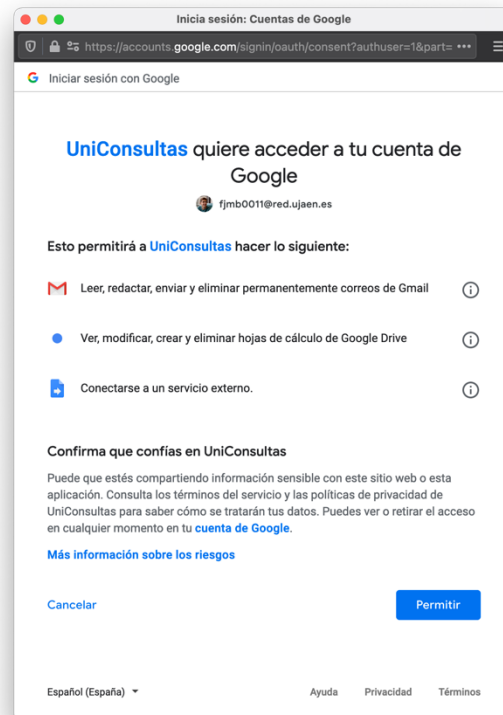


Ilustración 11: ventana externa para permitir la interacción entre el bot de Telegram y nuestra aplicación

8. Incrementos

8.1. Versión 1.0

Requisitos

FRQ-0001	El sistema debe solicitar el correo electrónico institucional al estudiante para poder registrarse
Versión	Versión 1.0 (23/04/2021)
Autor/es	Francisco Javier Martínez Bueno
Fuentes	Estudiante no registrado
Propósito	Asociar el ID del usuario de Telegram al correo electrónico institucional para poder hacer consultas sobre su desempeño en una asignatura.
Descripción	Mientras que el correo electrónico del estudiante no esté almacenado en la hoja de cálculo, este no podrá realizar ninguna acción.
Prioridad	Alta
Comentarios	Ninguno

Tabla 4: especificación del requisito FRQ-0001

FRQ-0002	El sistema debe permitir a un profesor ocultar información sensible
Versión	Versión 1.0 (23/04/2021)
Autor/es	Francisco Javier Martínez Bueno
Fuentes	Profesor
Propósito	Ocultar información sensible en función de las necesidades del profesor.
Descripción	Un profesor debe poder decidir qué columnas de la hoja de cálculo van a estar accesibles para un estudiante y cuales no.
Prioridad	Alta
Comentarios	Ninguno

Tabla 5: especificación del requisito FRQ-0002

FRQ-0003	El sistema debe permitir al estudiante consultar información recogida en la hoja de cálculo sobre su desempeño en una asignatura
Versión	Versión 1.0 (23/04/2021)
Autor/es	Francisco Javier Martínez Bueno
Fuentes	Estudiante registrado
Propósito	Acceder a la información recogida en la hoja de cálculo acerca de su desempeño en una asignatura.
Descripción	Un estudiante debe poder preguntar por algunos de los aspectos recogidos en la hoja de cálculo.
Prioridad	Alta
Comentarios	Ninguno

Tabla 6: especificación del requisito FRQ-0003

FRQ-0004	El sistema debe permitir a un profesor bloquear temporalmente el acceso a la hoja de cálculo y volver a darlo
Versión	Versión 1.0 (23/04/2021)
Autor/es	Francisco Javier Martínez Bueno
Fuentes	Profesor
Propósito	Evitar que un estudiante acceda a información errónea mientras se hacen modificaciones y cálculos en la hoja de cálculo.
Descripción	Un profesor debe poder bloquear temporalmente el acceso y volver a darlo, de forma que, si el estudiante solicita un dato, este reciba un mensaje diciendo que la base de datos está en mantenimiento.
Prioridad	Alta
Comentarios	Ninguno

Tabla 7: especificación del requisito FRQ-0004

FRQ-0005	El sistema debe permitir a un estudiante consultar qué órdenes le puede dar al bot
Versión	Versión 1.0 (23/04/2021)
Autor/es	Francisco Javier Martínez Bueno
Fuentes	Estudiante registrado
Propósito	Saber en todo momento qué comandos puede enviar un estudiante al bot para interactuar con el mismo.
Descripción	Un estudiante debe poder consultar en todo momento las órdenes que puede darle al bot, para ello, el bot debe contar con un comando “/comandos” capaz de listar todas las órdenes que se le pueden dar al bot ese momento dado.
Prioridad	Alta
Comentarios	Ninguno

Tabla 8: especificación del requisito FRQ-0005

FRQ-0006	El sistema debe permitir a un profesor elegir con qué nombre se refiere a una columna/dato
Versión	Versión 1.0 (23/04/2021)
Autor/es	Francisco Javier Martínez Bueno
Fuentes	Profesor
Propósito	Cambiar en un momento dado el nombre con el que un profesor quiere referirse a una columna o dato, haciendo que la inserción de valores en la hoja de cálculo sea dinámica en función de la asignatura.
Descripción	Un profesor debe poder cambiar en cualquier momento el nombre con el que el mismo se refiere a una columna o dato, en función de las necesidades de cada uno.
Prioridad	Alta
Comentarios	Ninguno

Tabla 9: especificación del requisito FRQ-0006

Diseño

- **Diagrama de casos de uso**

El diagrama de casos de uso representa al conjunto de responsabilidades que debe desempeñar nuestra aplicación durante un incremento.

En este caso, tal y como podemos ver en la Ilustración 12, un estudiante no registrado debe ser capaz de registrarse con su correo electrónico institucional, un estudiante registrado debe ser capaz de consultar información recogida en la hoja de cálculo y consultar qué órdenes le puede dar al bot, mientras que un profesor debe ser capaz de bloquear temporalmente el acceso a los datos y volver a darlo, ocultar información sensible que no quiera que pueda ser consultada por un estudiante y elegir con qué nombre se quiere referir a una columna o dato.

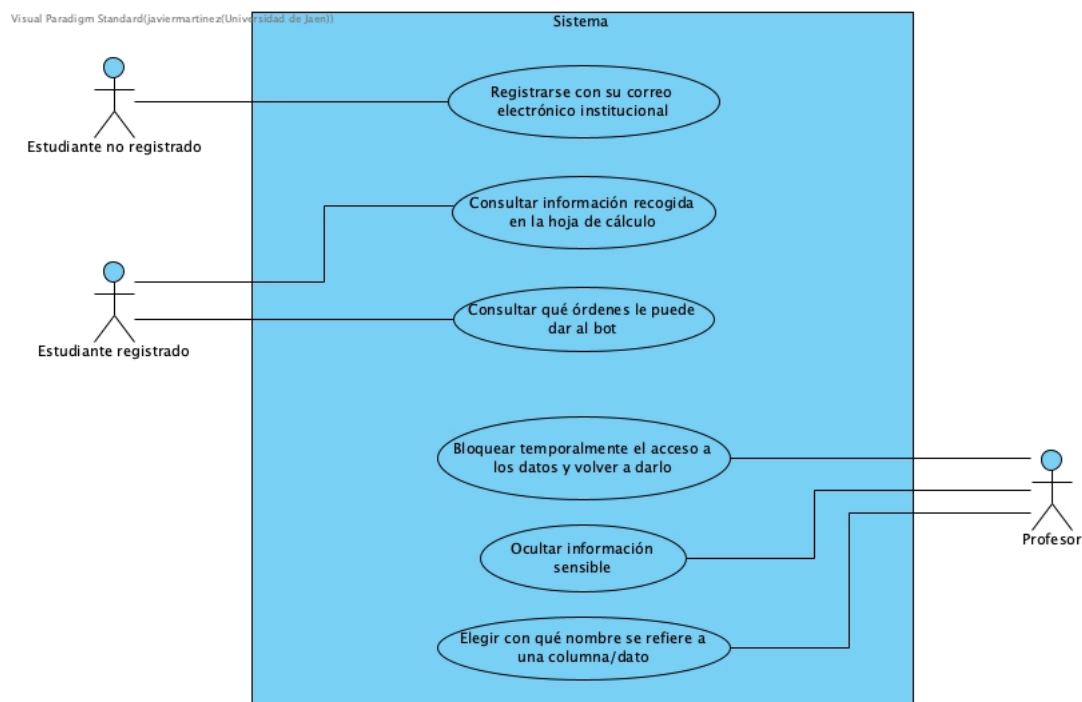


Ilustración 12: diagrama de casos de uso de la versión 1.0

- **Diagramas de secuencia del sistema**

A continuación, se exponen los escenarios principales de los casos de uso de la versión 1.0, representados mediante diagramas de secuencia del sistema.

- Un estudiante se registra con su correo electrónico institucional

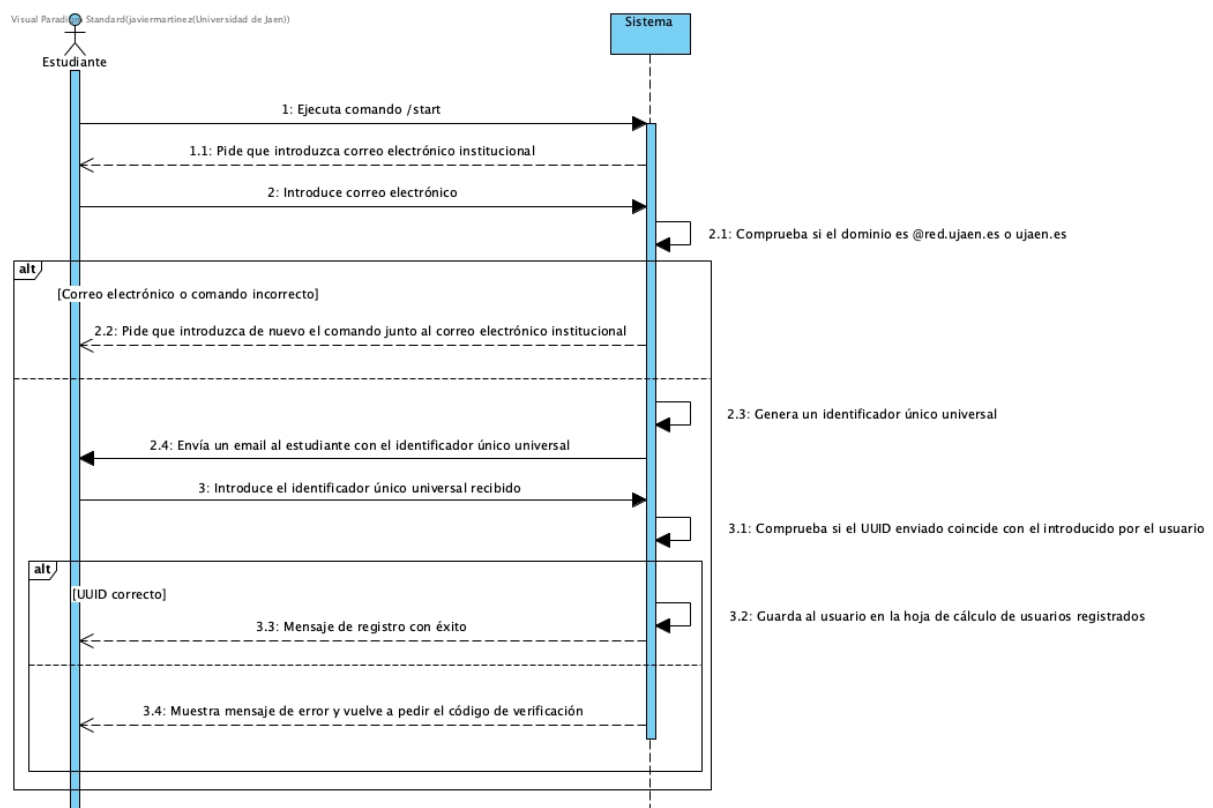


Ilustración 13: *diagrama de secuencia del sistema para el caso de uso “Un estudiante se registra con su correo electrónico institucional”*

- El profesor oculta información sensible

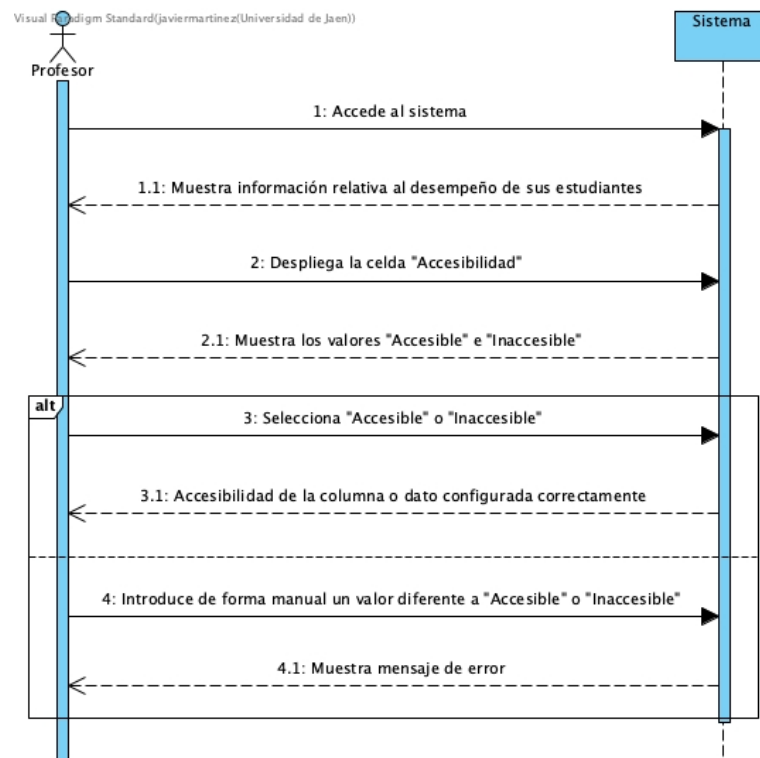


Ilustración 14: diagrama de secuencia del sistema para el caso de uso “El profesor oculta información sensible”

- Un estudiante consulta información recogida en la hoja de cálculo

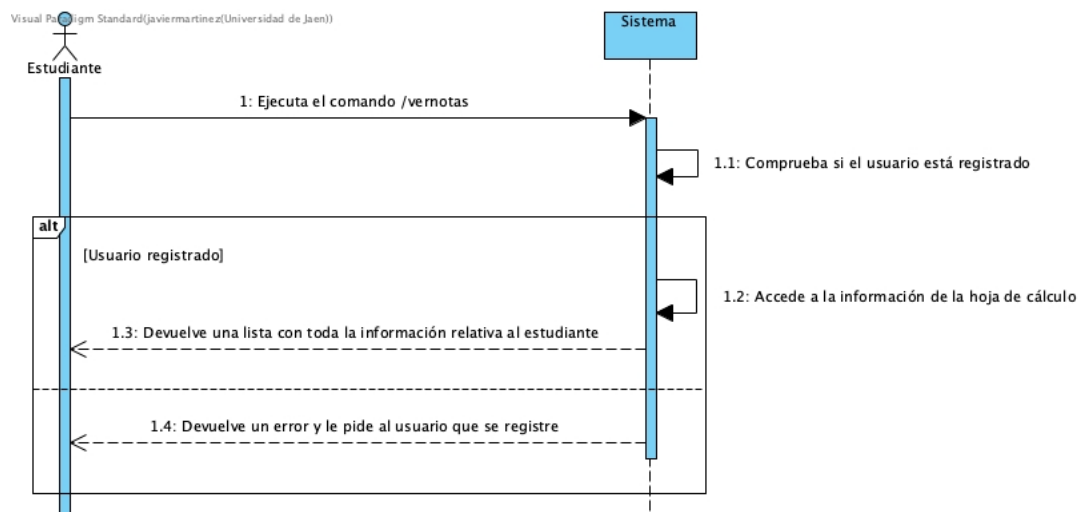


Ilustración 15: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta información recogida en la hoja de cálculo”

- Un profesor bloquea temporalmente el acceso a los datos y vuelve a darlo

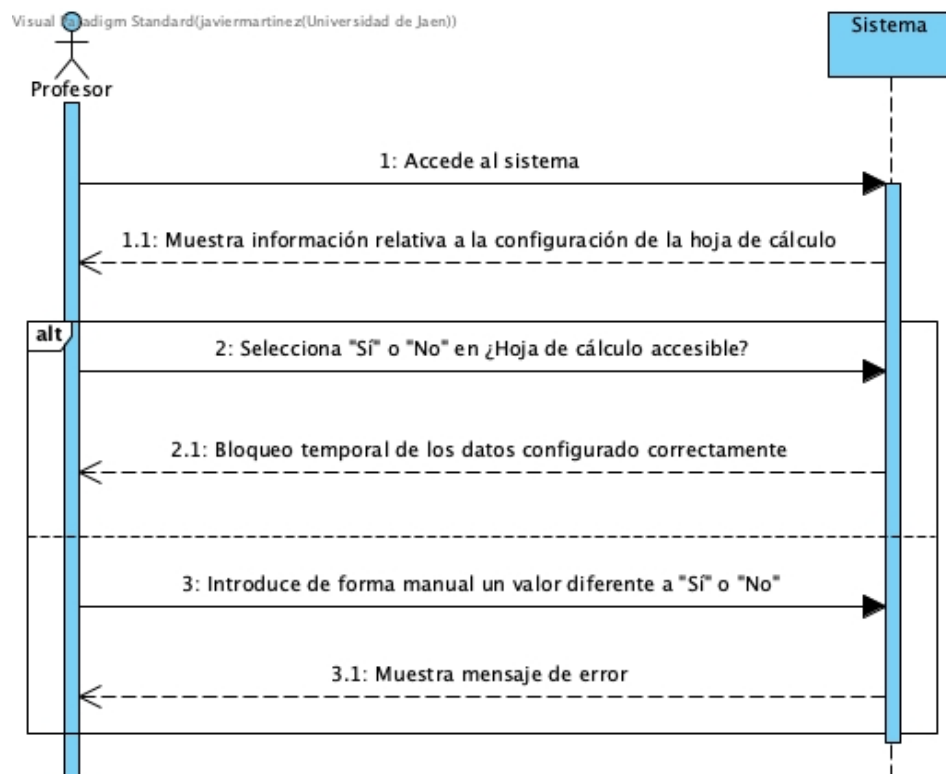


Ilustración 16: diagrama de secuencia del sistema para el caso de uso "Un profesor bloquea temporalmente el acceso a los datos y vuelve a darlo"

- Un estudiante consulta qué órdenes le puede dar al bot

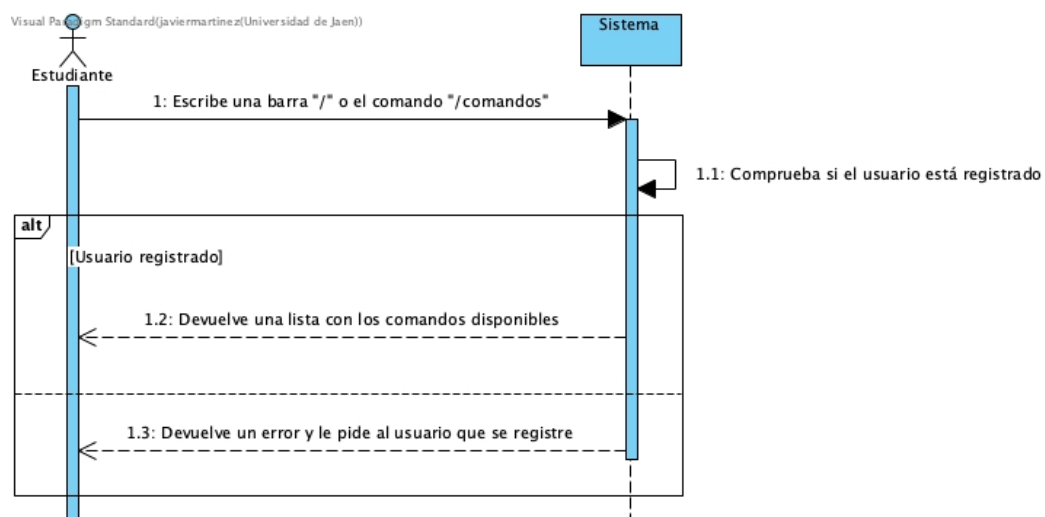


Ilustración 17: diagrama de secuencia del sistema para el caso de uso "Un estudiante consulta qué órdenes le puede dar al bot"

- Un profesor elige con qué nombre se refiere a una columna/dato

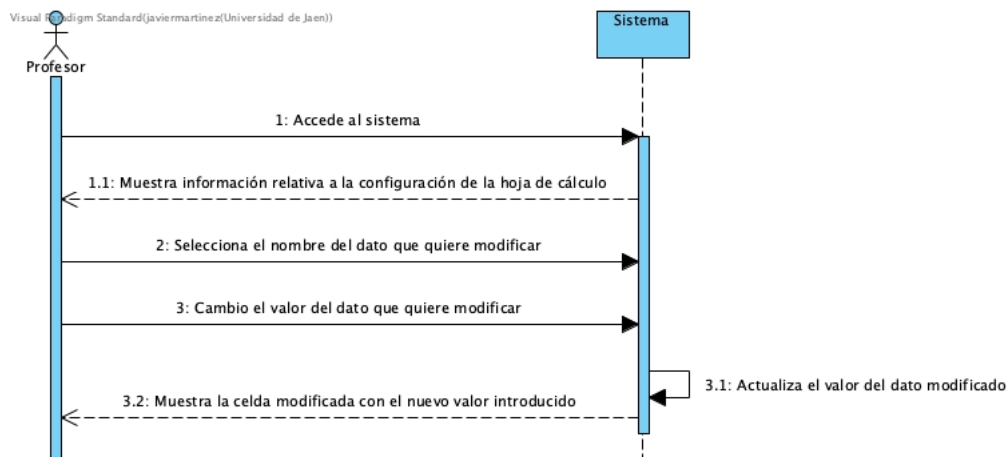


Ilustración 18: diagrama de secuencia del sistema para el caso de uso “Un profesor elige con qué nombre se refiere a una columna/dato”

Implementación

Debido a la extensión de la implementación de todas las funcionalidades, el código fuente de la aplicación está disponible en un archivo anexo.

Pruebas

La versión inicial de la hoja de cálculo del profesor se encuentra en los anexos al final de este trabajo.

Al iniciar una conversación con el bot de Telegram este nos pedirá registrarnos para poder almacenar nuestro correo electrónico asociado a nuestro ID de Telegram. Tal y como podemos ver en la ilustración 19, cuando el usuario ejecuta el comando “/registrar fjmb0011@red.ujaen.es”, el bot nos devuelve un mensaje informándonos sobre el correo electrónico que nos acaba de enviar con el código de verificación.

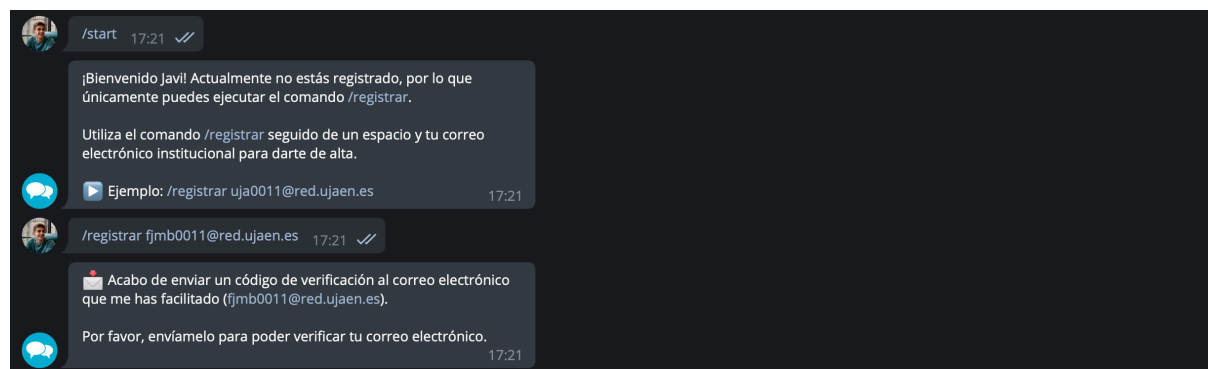


Ilustración 19: captura de pantalla con información relativa a la primera interacción de un estudiante con el bot

El correo electrónico enviado por el bot de Telegram cuenta con un asunto personalizado en función del nombre del estudiante en Telegram. Así mismo, en el cuerpo del mensaje, se encuentra el identificador único universal (UUID) que se utiliza como código de verificación para verificar el correo electrónico institucional del estudiante.

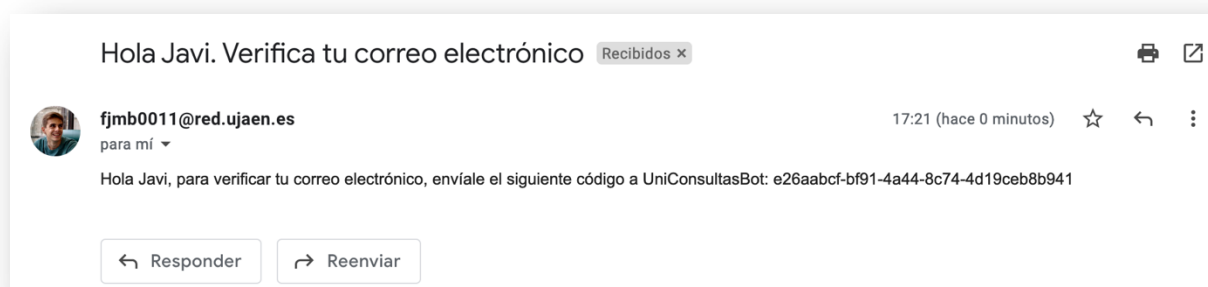


Ilustración 20: e-mail enviado por el bot de Telegram para verificar nuestra dirección de correo electrónico

Tras verificar el correo electrónico, el estudiante puede interactuar con el bot sin ningún tipo de restricción, tal y como observamos en la ilustración 21.

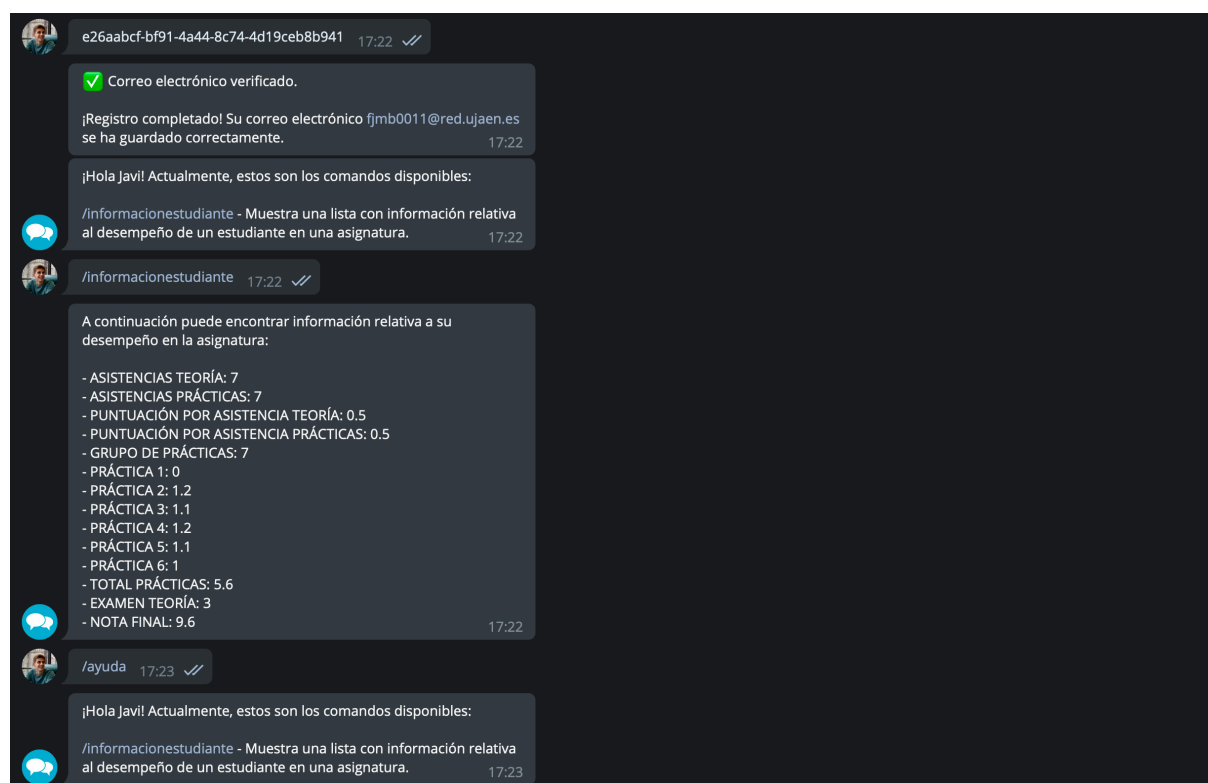


Ilustración 21: verificación de la cuenta de correo electrónico y consulta sobre el desempeño de un estudiante en una asignatura

Despliegue

Victor Manuel Rivas Santos, profesor de la Universidad de Jaén, introduce información relativa al desempeño de sus estudiantes en la hoja de cálculo de Desarrollo ágil, siendo esta una asignatura que él imparte.

Todos los estudiantes de la asignatura acceden al bot de Telegram y se registran en él para consultar información acerca de su desempeño en la misma.

Revisión

Tras el despliegue inicial del bot de Telegram en la versión 1.0, se encontraron una serie de problemas o posibles mejoras, que serán implementadas y corregidas en la versión 2.0, dando lugar a una interacción estudiante-bot óptima:

- Un estudiante no es capaz de registrarse porque al intentar guardar su ID de Telegram cifrado en la hoja de cálculo, este empieza por un símbolo “+”, siendo entonces detectado por Google Sheets como una fórmula.
 - Solución: guardar el dato como una cadena, haciendo que la misma empiece por “_” o algún otro símbolo.
- Un profesor no es capaz de registrarse porque únicamente se permite el registro a correos electrónicos cuyo dominio es “@red.ujaen.es”.
 - Solución: añadir condición OR y el dominio “@ujaen.es” a la sentencia condicional que permite que un usuario se registre si su correo electrónico termina en “red.ujaen.es”.
- Añadir el comando “/informacionbot” para poder consultar información relativa a este. Por ejemplo, se podrá consultar quién es su creador, el motivo de la creación del bot de Telegram, qué utilidad tiene el bot de Telegram, etc.
- Añadir el comando “/contacto” para poder contactar con el desarrollador del bot de Telegram en caso de tener alguna duda o problema sobre la utilización del mismo.

8.2. Versión 2.0

Requisitos

FRQ-0007	El sistema debe permitir al estudiante consultar el enlace web en el que se desarrollará una videoconferencia
Versión	Versión 2.0 (20/05/2021)
Autor/es	• Francisco Javier Martínez Bueno
Fuentes	• Estudiante registrado
Propósito	Acceder de forma rápida al enlace web en el que se desarrollará una videoconferencia perteneciente a una clase teórica o práctica.
Descripción	El sistema debe permitir a un estudiante consultar el enlace web en el que se desarrollará una videoconferencia.
Prioridad	Media
Comentarios	Ninguno

Tabla 10: especificación del requisito FRQ-0007

FRQ-0008	El sistema debe permitir al estudiante consultar información relativa al sistema de evaluación de una asignatura
Versión	Versión 2.0 (20/05/2021)
Autor/es	Francisco Javier Martínez Bueno
Fuentes	Estudiante registrado
Propósito	Acceder de forma rápida a la nota mínima necesaria en prácticas y teoría para poder aprobar la asignatura, así como la ponderación sobre la nota final de ambas.
Descripción	El sistema debe permitir a un estudiante acceder a la información relativa al sistema de evaluación de la asignatura.
Prioridad	Media
Comentarios	Ninguno

Tabla 11: especificación del requisito FRQ-0008

FRQ-0009	El sistema debe permitir a un estudiante consultar información recogida en la hoja de cálculo sobre su desempeño en distintas asignaturas
Versión	Versión 2.0 (20/05/2021)
Autor/es	Francisco Javier Martínez Bueno
Fuentes	Estudiante registrado
Propósito	Acceder a la información recogida en la hoja de cálculo acerca de su desempeño en más de una asignatura.
Descripción	Un estudiante debe poder preguntar por algunos de los aspectos recogidos en la hoja de cálculo en más de una asignatura.
Prioridad	Media
Comentarios	Ninguno

Tabla 12: especificación del requisito FRQ-0009

FRQ-0010	El sistema debe permitir a un estudiante consultar información sobre el propio bot
Versión	Versión 2.0 (20/05/2021)
Autor/es	Francisco Javier Martínez Bueno
Fuentes	Estudiante no registrado / estudiante registrado
Propósito	Consultar información sobre el propio bot, como, ¿quién ha creado el bot?, ¿por qué se ha creado el bot?, ¿para qué sirve el bot?, etc
Descripción	Un estudiante debe poder tener acceso a información sobre el propio bot.
Prioridad	Media
Comentarios	Ninguno

Tabla 13: especificación del requisito FRQ-0010

FRQ-0011	El sistema debe permitir a un estudiante consultar información para contactar con el desarrollador del bot
Versión	Versión 2.0 (20/05/2021)
Autor/es	Francisco Javier Martínez Bueno
Fuentes	Estudiante no registrado / estudiante registrado
Propósito	Contactar con el desarrollador del bot para poder enviarle impresiones, dudas y posibles formas de mejorar el mismo.
Descripción	Un estudiante debe ser capaz de ejecutar una orden y obtener diferentes métodos de contacto para hablar con el desarrollador.
Prioridad	Media
Comentarios	Ninguno

Tabla 14: especificación del requisito FRQ-0011

FRQ-0012	El sistema debe permitir a un estudiante consultar información para contactar con el profesor encargado de la asignatura
Versión	Versión 2.0 (20/05/2021)
Autor/es	Francisco Javier Martínez Bueno
Fuentes	Estudiante registrado
Propósito	Contactar con el profesor encargado de la asignatura para poder preguntarle información acerca de cualquier inquietud sobre el desempeño del estudiante en la misma.
Descripción	Un estudiante debe ser capaz de ejecutar una orden y obtener diferentes métodos de contacto para hablar con el profesor encargado de la asignatura.
Prioridad	Media
Comentarios	Ninguno

Tabla 15: especificación del requisito FRQ-0012

Diseño

- **Diagrama de casos de uso**

En esta iteración, tal y como podemos ver en la Ilustración 22, un estudiante registrado debe ser capaz de consultar un enlace web en el que se desarrollará una videoconferencia, consultar información sobre el sistema de evaluación de una asignatura, consultar información recogida en la hoja de cálculo sobre su desempeño en distintas asignaturas y contactar con el profesor encargado de la asignatura. Así mismo, tanto un estudiante registrado como un estudiante no registrado deben ser capaces de consultar información sobre el propio bot y consultar información para contactar con el desarrollador del bot.

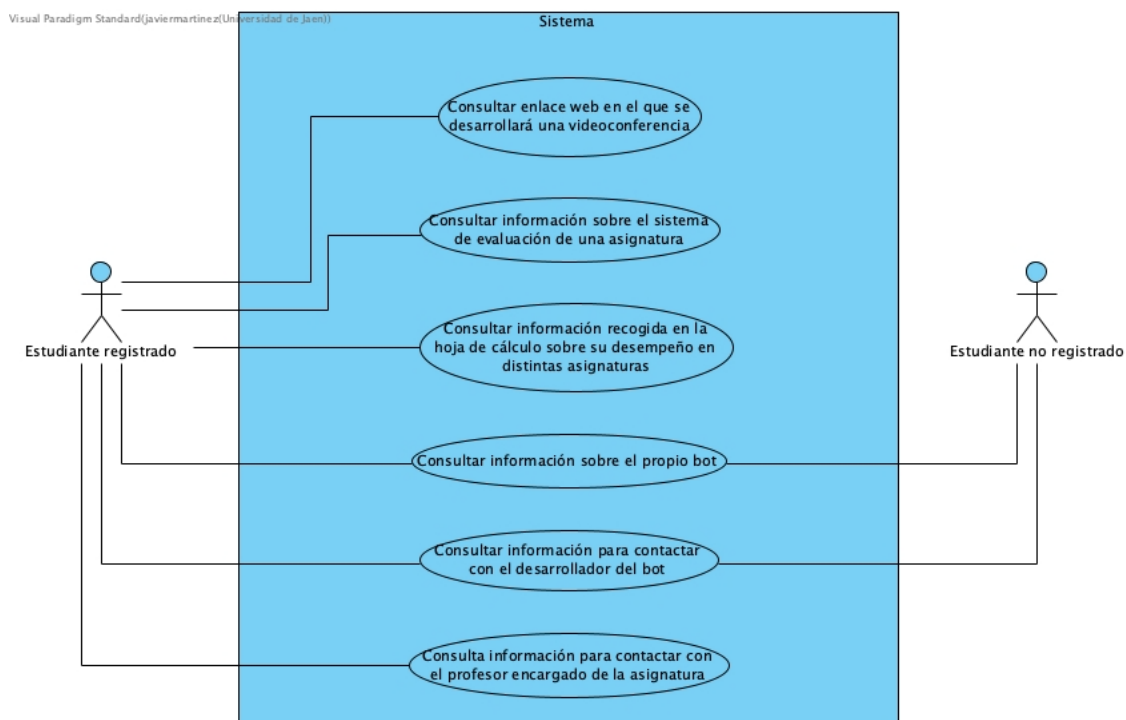


Ilustración 22: diagrama de casos de uso de la versión 2.0

- **Diagramas de secuencia del sistema**

A continuación, se exponen los escenarios principales de los casos de uso de la versión 2.0, representados mediante diagramas de secuencia del sistema.

- Un estudiante consulta el enlace web en el que se desarrollará una videoconferencia

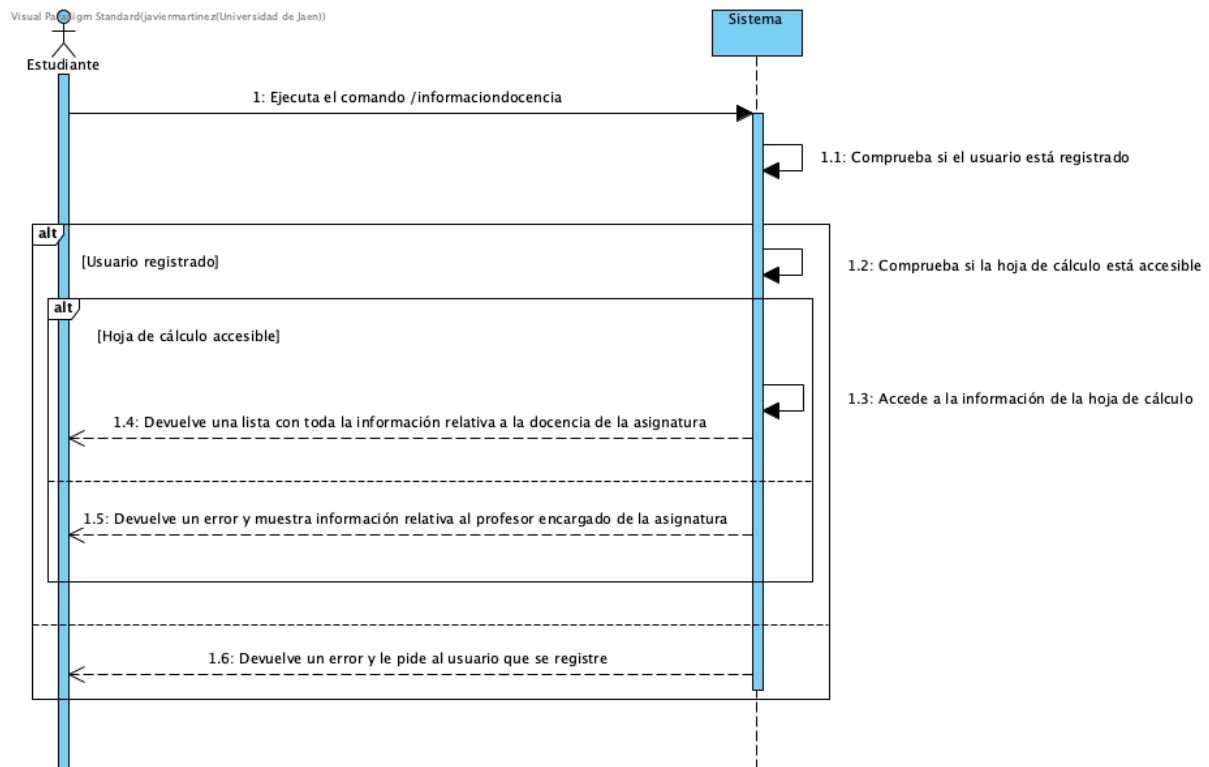


Ilustración 23: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta el enlace web en el que se desarrollará una videoconferencia”

- Un estudiante consulta información relativa al sistema de evaluación de una asignatura

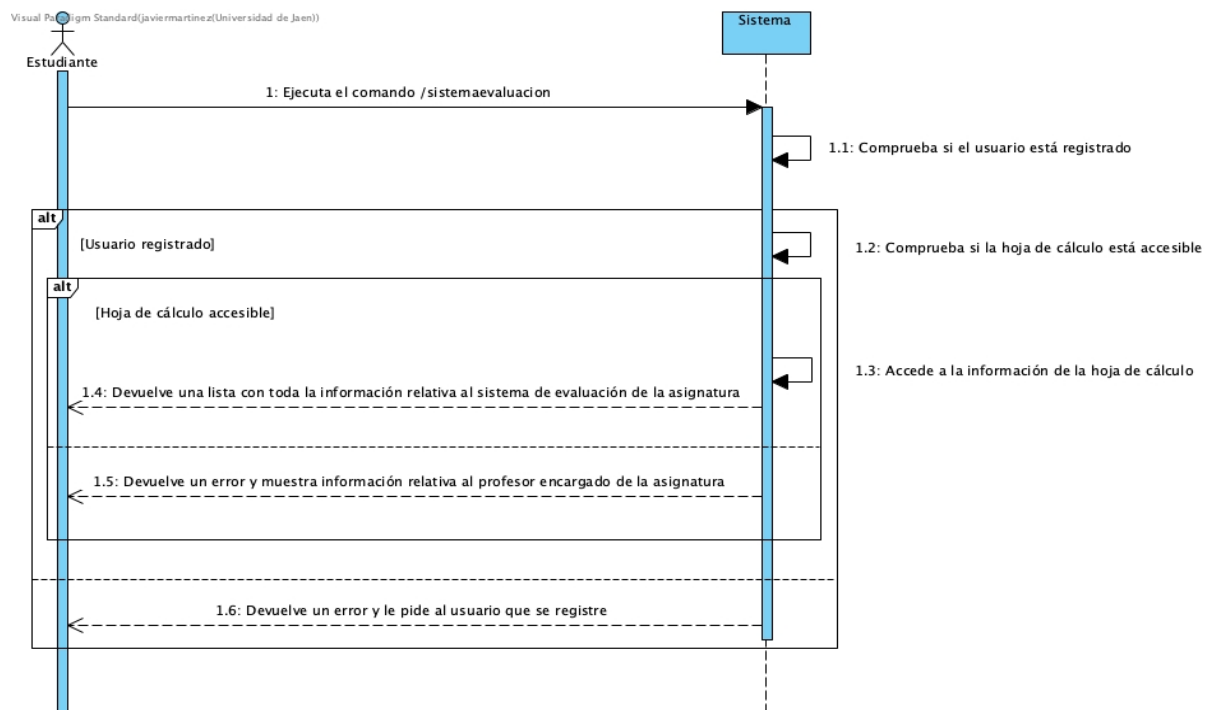


Ilustración 24: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta información relativa al sistema de evaluación de una asignatura”

- Un estudiante consulta información recogida en la hoja de cálculo sobre su desempeño en distintas asignaturas

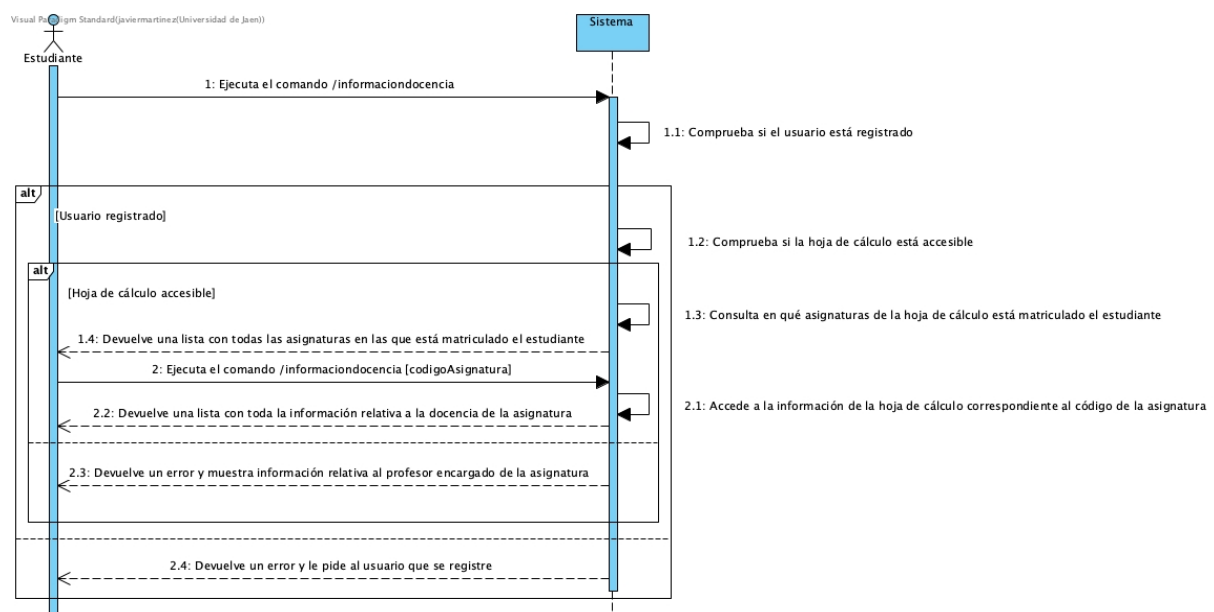


Ilustración 25: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta información recogida en la hoja de cálculo sobre su desempeño en distintas asignaturas”

- Un estudiante consulta información sobre el propio bot

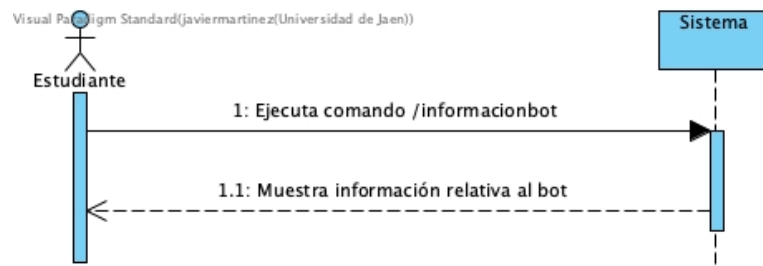


Ilustración 26: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta información sobre el propio bot”

- Un estudiante consulta información para contactar con el desarrollador del propio bot

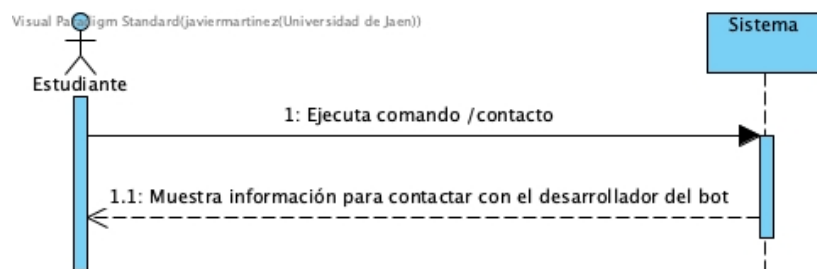


Ilustración 27: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta información para contactar con el desarrollador del propio bot”

- Un estudiante consulta información para contactar con el profesor encargado de la asignatura

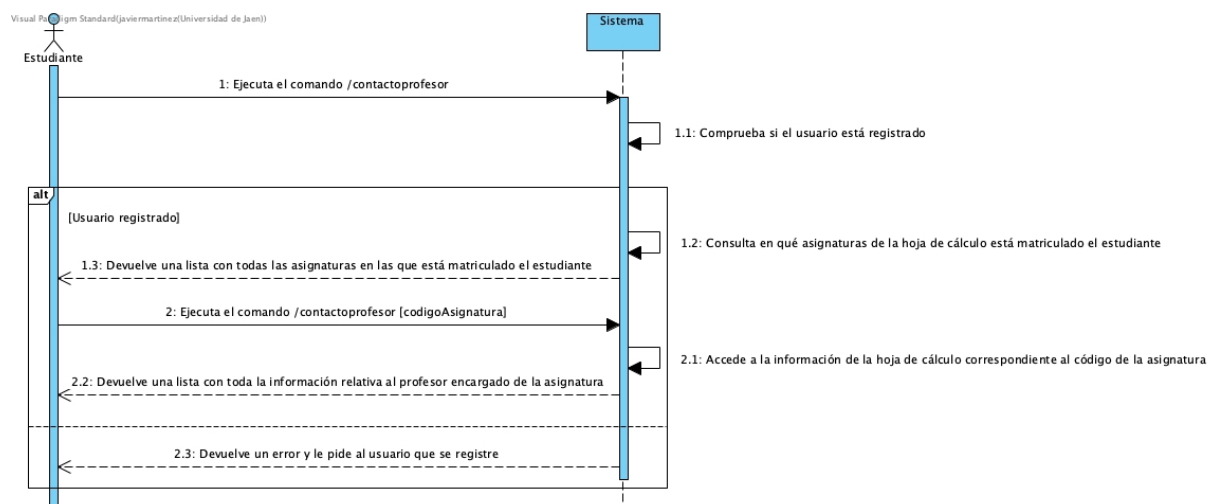


Ilustración 28: diagrama de secuencia del sistema para el caso de uso “Un estudiante consulta información para contactar con el profesor encargado de la asignatura”

Implementación

Debido a la extensión de la implementación de todas las funcionalidades, el código fuente de la aplicación está disponible en un archivo anexo.

Pruebas

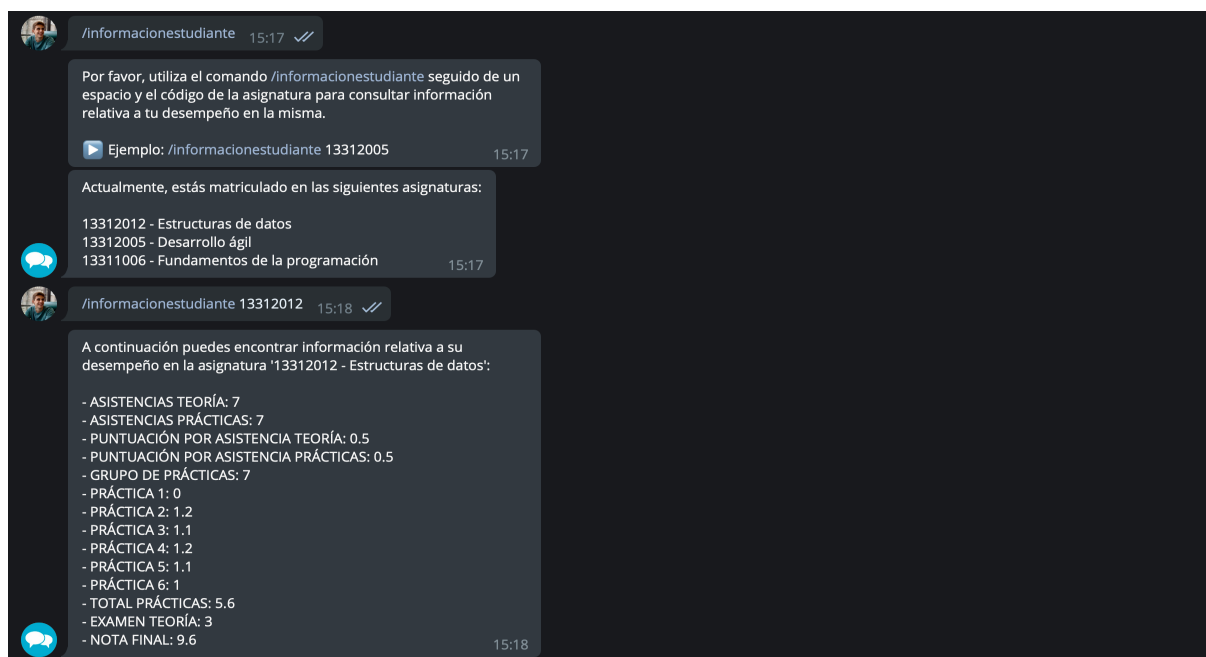


Ilustración 29: ejecución del comando `/informacionestudiante` para obtener información relativa al desempeño de un estudiante en una asignatura

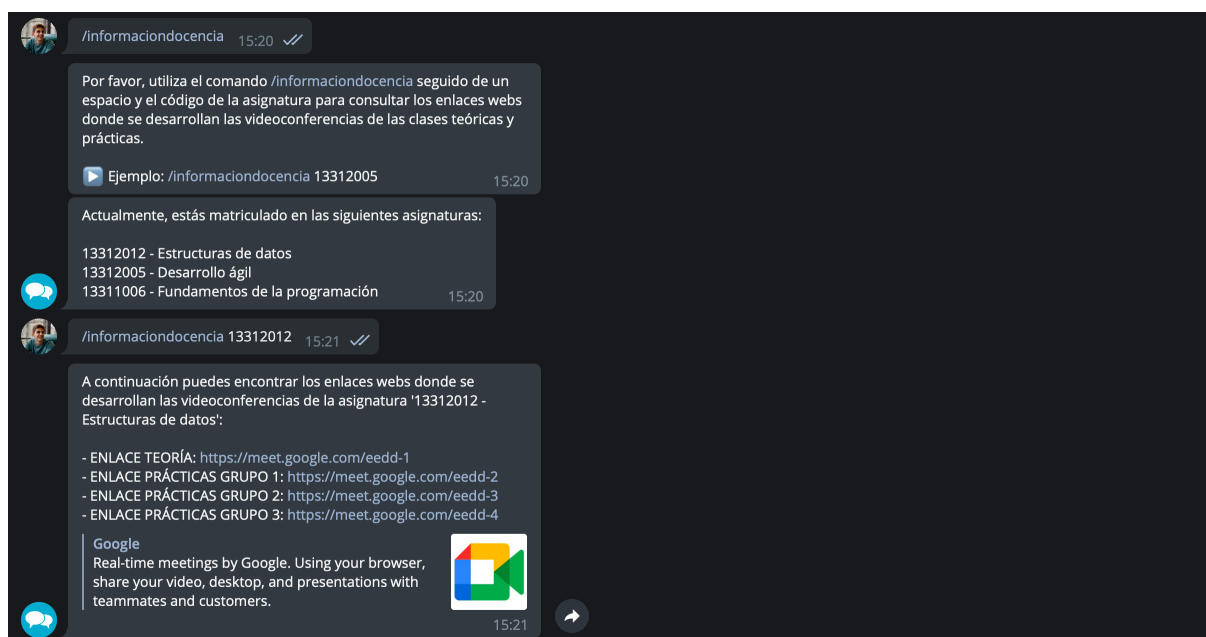


Ilustración 30: ejecución del comando `/informaciondocencia` para obtener información relativa a los enlaces webs donde se desarrollan las videoconferencias de una asignatura

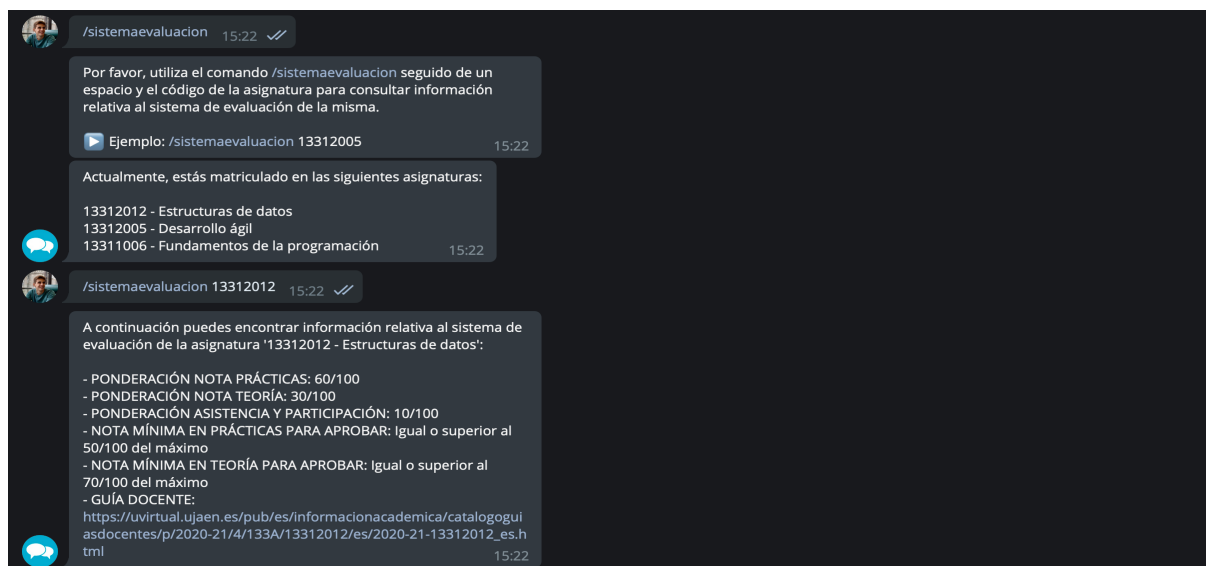


Ilustración 31: ejecución del comando /sistemaevaluacion con información relativa al sistema de evaluación de una asignatura

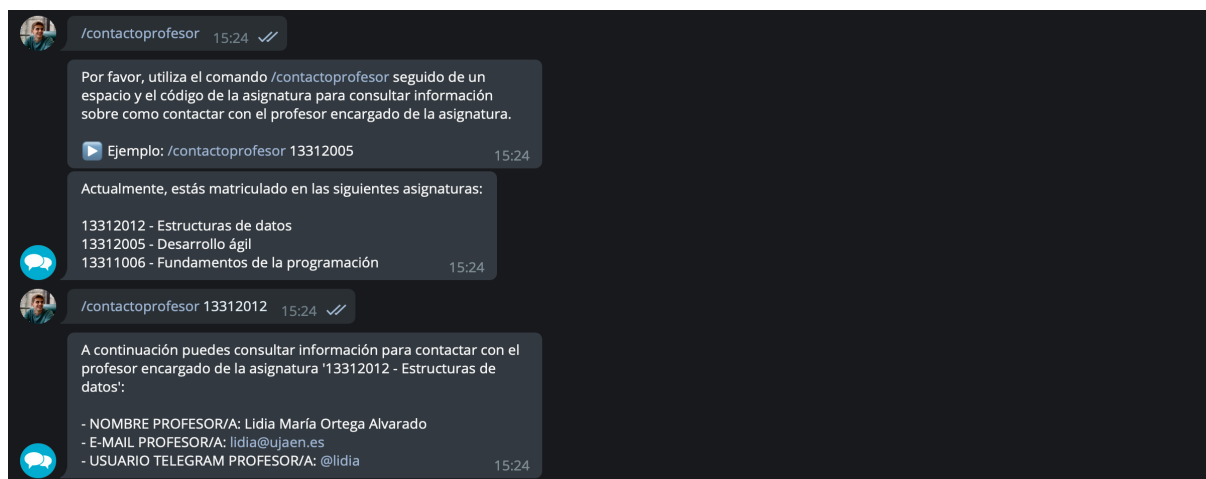


Ilustración 32: ejecución del comando /contactoprofesor para obtener información relativa a los diferentes métodos de contacto con un profesor

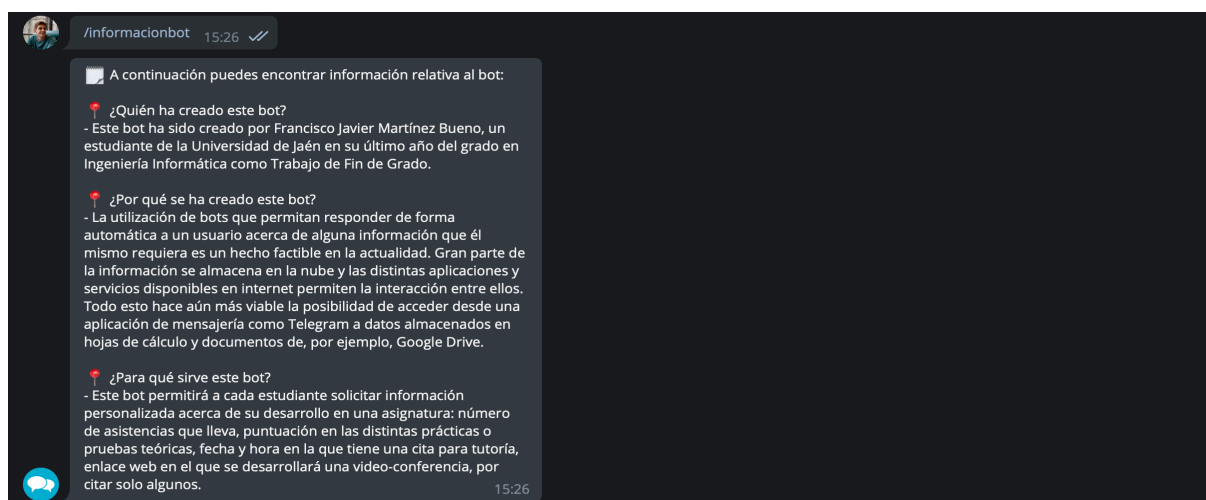


Ilustración 33: ejecución del comando /informacionbot para obtener información relativa al bot

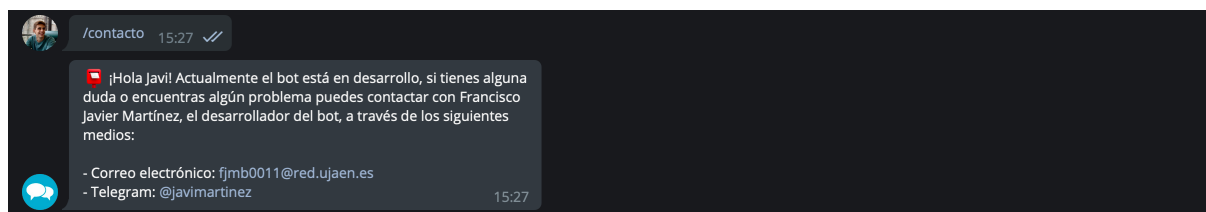


Ilustración 34: ejecución del comando /contacto para obtener información relativa al desarrollador del bot

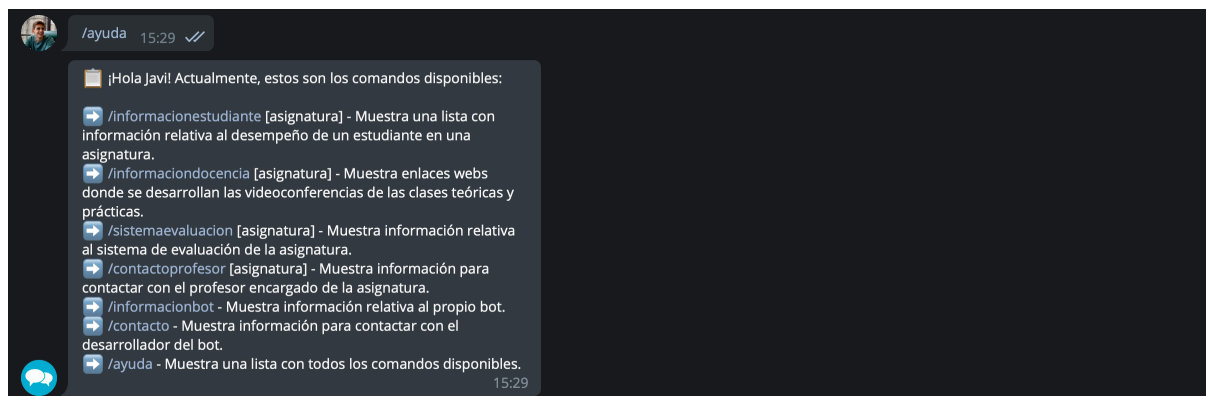


Ilustración 35: ejecución del comando /ayuda para obtener todos los comandos disponibles en el bot

Despliegue

La interacción con el bot por parte de los estudiantes de la asignatura de desarrollo ágil ha sido continua, pudiendo evaluar en todo momento cualquier anomalía producida al añadir una nueva funcionalidad.

Revisión

Una vez implementado el requisito **FRQ-0009** (el sistema debe permitir a un estudiante consultar información recogida en la hoja de cálculo sobre su desempeño en distintas asignaturas), se observa que los estudiantes no introducen el nombre de la asignatura tal y como se muestra en el bot, y debido a que este es sensible a las mayúsculas y minúsculas, no es capaz de relacionar el nombre introducido con el nombre correspondiente a la hoja de cálculo.

La solución propuesta y finalmente implementada consiste en mostrar el código de la asignatura junto al nombre de la misma a la hora de listar las asignaturas en las que un estudiante está matriculado, teniendo que introducir ahora el código de la asignatura en lugar del nombre.

Aparte de los cambios y modificaciones que el desarrollador del bot hizo conforme iban surgiendo los distintos problemas, entre ellos los comentados anteriormente, se envió una encuesta a cada usuario que se había registrado en el servicio, siendo estos estudiantes de la Universidad de Jaén matriculados en la asignatura de desarrollo ágil del tercer curso del grado en Ingeniería Informática. Así mismo, la encuesta también fue enviada al profesor que probó e introdujo datos en la hoja de cálculo de Google.

La encuesta fue enviada en dos ocasiones, recibiendo un total de 8 respuestas frente a los 19 usuarios encuestados que estaban registrados en el bot.



Ilustración 36: encuesta de satisfacción. ¿Tuvo algún problema a la hora de registrarse en el bot?

En caso de que en la pregunta anterior haya contestado "Sí", indique el/los motivo/s.

2 respuestas

Al principio, el bot solo aceptaba correos de estudiantes (@red.ujaen.es), siendo necesario que también aceptara los que empiezan por @ujaen.es

No podía entrar aunque estuviese verificado debido a mi ID de Telegram

Ilustración 37: encuesta de satisfacción. En caso de que en la pregunta anterior haya contestado "Sí", indique el/los motivo/s

¿Tuvo algún problema a la hora de verificar su correo electrónico?
8 respuestas

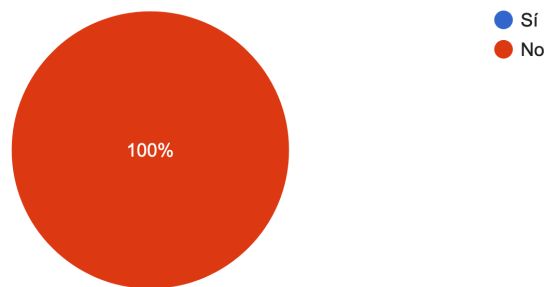


Ilustración 38: encuesta de satisfacción. ¿Tuvo algún problema a la hora de verificar su correo electrónico?

¿Le resultó fácil consultar los comandos ofrecidos por el bot?
8 respuestas

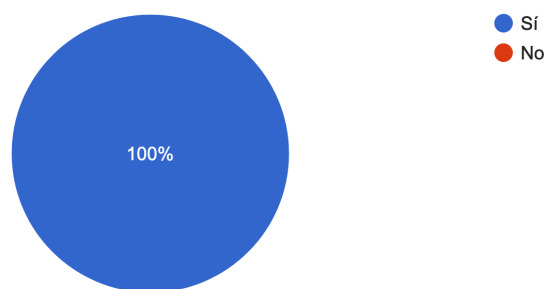


Ilustración 39: encuesta de satisfacción. ¿Le resultó fácil consultar los comandos ofrecidos por el bot?

¿La interacción con el bot fue fácil e intuitiva?
8 respuestas

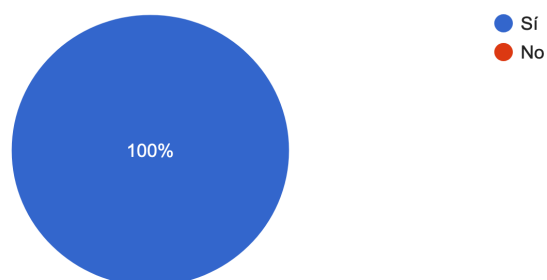


Ilustración 40: encuesta de satisfacción. ¿La interacción con el bot fue fácil e intuitiva?

¿Con qué probabilidad utilizaría este bot si todas las asignaturas lo implantasen?

8 respuestas

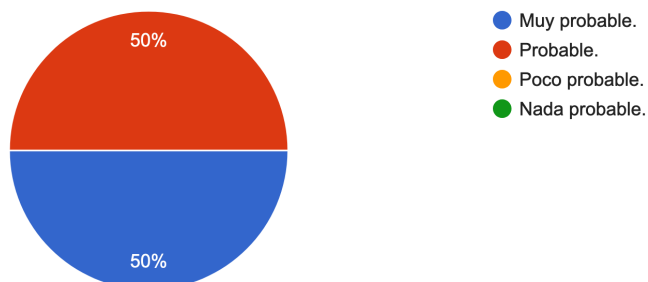


Ilustración 41: encuesta de satisfacción. ¿Con qué probabilidad utilizaría este bot si todas las asignaturas lo implantasen?

¿Cuál es la calidad de la información recibida por el bot acerca de su desempeño en la asignatura?

8 respuestas



Ilustración 42: encuesta de satisfacción. ¿Cuál es la calidad de la información recibida por el bot acerca de su desempeño en la asignatura?

¿Cuál es la probabilidad de que recomiende el bot a sus compañeros si se implantase en todas las asignaturas?

8 respuestas

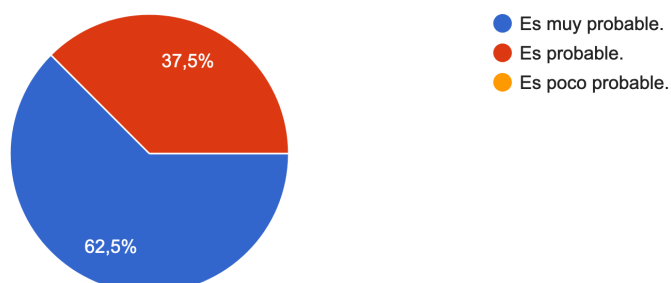


Ilustración 43: encuesta de satisfacción. ¿Cuál es la probabilidad de que recomiende el bot a sus compañeros si se implantase en todas las asignaturas?

En general, ¿cómo de satisfecho está con el bot?

8 respuestas

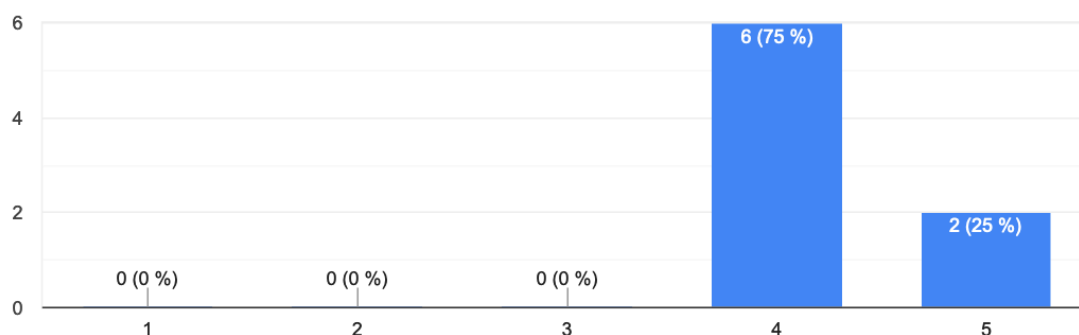


Ilustración 44: encuesta de satisfacción. En general, ¿cómo de satisfecho está con el bot?

¿Qué otra información le gustaría poder consultarle al bot?

8 respuestas

Actividades relacionadas con la asignaturas
Cuando estarán disponibles las notas aprox
Nombre de todos los profesores que imparten la asignatura, grupos de prácticas que hay y horarios y aulas para los mismos.
Los horarios de tutorías de los profesores que haya en la asignatura, por ejemplo.
Mi expediente académico
Creo que la información que se muestra es bastante y tampoco se debería de sobrecargar la información a mostrar en el bot.
Información de cualquier asignatura
Horario

Ilustración 45: encuesta de satisfacción. ¿Qué otra información le gustaría poder consultarle al bot?

¿Qué aspecto destacaría y qué aspecto mejoraría del bot?

8 respuestas

Mejoraría que al darle a elegir la opción del comando deseado, directamente se mandaba y no podía completarlo con otras palabras añadidas

Es bastante útil el comando informacionestudiante y no tener que entrar en docencia virtual o universidad virtual, por otro lado podría ser interesante un comando como help en el que se indique todos los comando disponibles en el bot.

Aunque es fácil de usar, los comandos son muy largos; debería existir alias para ellos, para no tener que escribir tanto.

Destacaría que es muy intuitivo

El aspecto que destacaría sería que es muy fácil y rápido de utilizar y en un solo comando puedes obtener información importante y no hace falta que te metas en docencia o en otro sitio a buscarla.

El aspecto que mejoraría del bot sería que a la hora de poner la asignatura de la que se quiere información hay que ponerlo en mayúsculas, por ejemplo, en mi caso "DESARROLLO ÁGIL". No es algo muy grave pero otros compañeros y yo escribimos "desarrollo ágil" o "Desarrollo Ágil" al principio cuando interactuamos por primera vez y nos salía la información.

Destaco su sencillez, y mejoraría si se implanta en más asignaturas.

Que pudiera detectar la asignatura sin escribir necesariamente el código

Me parece todo bien

Ilustración 46: encuesta de satisfacción. ¿Qué aspecto destacaría y qué aspecto mejoraría del bot?

Tras la revisión de las respuestas de la encuesta podemos concluir que la interacción por parte de los estudiantes ha sido fácil e intuitiva, pudiendo consultar cualquier comando que el estudiante precisase.

La mayoría de los estudiantes muy probablemente utilizarían y recomendarían el bot si todas las asignaturas de la Universidad de Jaén lo implantasen, siendo buena la calidad de la información enviada por parte del mismo.

Para concluir, los estudiantes encuestados dicen estar muy satisfechos con el bot.

9. Conclusión

9.1. Reflexiones sobre el desarrollo del proyecto

En primer lugar, me gustaría destacar que la utilización de bots para responder consultas personalizadas está en pleno auge, teniendo una relación muy estrecha con la inteligencia artificial, y viéndose reflejado en la cantidad de información que podemos encontrar en la web y la demanda de desarrolladores altamente cualificados por parte de grandes empresas.

El hecho de poder crear un bot capaz de acceder a una hoja de cálculo que está en la nube, consultar información y procesarla, sin ningún coste, me fascina. Los servicios basados en la web son cada vez más utilizados y demandados por pequeñas y grandes empresas ya que facilitan la interacción y sincronización entre los usuarios y la información desde cualquier punto del mundo.

Una hoja de cálculo es accesible por cualquier persona, ya que actualmente no supone mucha complejidad interactuar con las mismas. Esto hace que sea factible el hecho de que un profesor introduzca información relativa a un estudiante en una de ellas para que un bot de Telegram procese dicha información y se la muestre al usuario final, haciendo posible la futura utilización del bot en toda la Universidad de Jaén.

Para mi suponía un reto el tener que utilizar Google Script, aunque estuviese basado en JavaScript, ya que nunca antes había desarrollado ninguna aplicación utilizando esa tecnología en la carrera universitaria; sin embargo, he sido capaz de desarrollar e implementar cualquier funcionalidad que me he propuesto gracias a los conocimientos y la mentalidad que he desarrollado a lo largo de estos años para abordar cualquier problema.

Animo a cualquier estudiante a intentar hacer un pequeño bot de Telegram utilizando como lenguaje de programación Google Script para guardar y consultar información en una hoja de cálculo de Google de forma sencilla y totalmente gratuita.

9.2. Posibles mejoras futuras

Aunque actualmente el bot de Telegram es totalmente funcional para todas las asignaturas de un grado en un año académico concreto, podría seguir desarrollándose para que fuese utilizable por cualquier grado de la Universidad de Jaén en cualquier año académico.

Algunas de las posibles mejoras futuras, basándome en las historias de usuario que han quedado por implementar y en las respuestas de la encuesta que realicé a todos los estudiantes que habían utilizado el bot, serían:

- Inclusión de los nombres de todos los profesores que imparten una asignatura, así como una manera de contactar con los profesores de prácticas en función del grupo de prácticas al que pertenezca un estudiante.
- Un estudiante debería poder consultar el horario de tutorías de un profesor, reservar o cancelar una cita para tutoría, consultar qué día tiene una cita para una tutoría y a qué hora, etc.
- Un profesor debería poder gestionar una tutoría a un estudiante, es decir, asignar citas, cancelar citas, consultar información sobre una cita, etc.
- Estadísticas accesibles por un profesor para consultar qué personas han accedido al bot para consultar información sobre su asignatura, qué dato le han preguntado al bot acerca de su asignatura, en qué momento han preguntado un dato concreto, etc.
- Un profesor debería ser capaz de configurar grupos de columnas bajo un solo identificador para poder mostrar únicamente información sobre las notas obtenidas en prácticas, por ejemplo.
- Un estudiante debería ser capaz de consultar todo su expediente académico.
- Acortar de alguna forma los comandos y hacer una utilización mas sencilla de los mismos, por ejemplo, incluyendo la utilización de botones y teclados propios de Telegram.

10. Bibliografía

- Esquema Nacional de Seguridad - ENS*. (26 de Julio de 2019). Obtenido de Portal de Administración Electrónica: https://administracionelectronica.gob.es/pae_Home/pae_Estrategias/pae_Seguridad_Inicio/pae_Esquema_Nacional_de_Seguridad.html
- Gil, C. (25 de Febrero de 2021). *Historia de Whatsapp: cómo se creó la app que ha revolucionado la comunicación social*. Obtenido de Marketing4ecommerce: <https://marketing4ecommerce.net/historia-de-whatsapp/>
- Iqbal, M. (8 de Marzo de 2021). *Line Revenue and Usage Statistics (2020)*. Obtenido de BusinessOfApps: <https://www.businessofapps.com/data/line-statistics/>
- Johnson, D. (2 de Febrero de 2021). *What is Viber? Everything you need to know about the encrypted messaging platform*. Obtenido de BusinessInsider: <https://www.businessinsider.com/what-is-viber-messaging-app>
- Kacey Kroh, A. H. (1 de Agosto de 2016). *What is Facebook Messenger?* Obtenido de Dummies: <https://www.dummies.com/social-media/facebook/what-is-facebook-messenger/>
- Lumb, D. J. (7 de Octubre de 2018). *The story of Signal*. Obtenido de Increment: <https://increment.com/security/story-of-signal/>
- Marlinspike, M. (21 de Febrero de 2018). *Signal Foundation*. Obtenido de Signal: <https://signal.org/blog/signal-foundation/>
- Mavuru, I. (22 de Junio de 2018). *Traditional vs. Agile Software Development Methodologies*. Obtenido de KPIPartners: <https://www.kpipartners.com/blog/traditional-vs-agile-software-development-methodologies>
- Siegel, C. (28 de Octubre de 2020). *What is an API?* Obtenido de APIfriends: <https://apifriends.com/api-management/what-is-an-api/>
- Technical Whitepaper*. (28 de Octubre de 2019). Obtenido de LINE Encryption Overview: <https://d.line-scdn.net/stf/linecorp/en/csr/line-encryption-whitepaper-ver2.0.pdf>
- Telegram. (14 de Agosto de 2013). *Preguntas frecuentes*. Obtenido de Telegram: <https://telegram.org/faq/es/>
- Telegram. (24 de Junio de 2015). *Bots: An introduction for developers*. Obtenido de Telegram: <https://core.telegram.org/bots>
- Telegram. (24 de Junio de 2015). *Telegram Bot API*. Obtenido de Telegram: <https://core.telegram.org/bots/api>
- Telegram. (18 de Mayo de 2017). *Bots Payments API*. Obtenido de Telegram: <https://core.telegram.org/bots/payments>
- Telegram. (31 de Enero de 2018). *TDLib – Build Your Own Telegram*. Obtenido de Telegram: <https://telegram.org/blog/tdlib>

Telegram. (7 de Febrero de 2018). *Telegram Widgets*. Obtenido de <https://core.telegram.org/widgets>

Telegram. (6 de Julio de 2019). *Creating Animated Stickers*. Obtenido de https://core.telegram.org/animated_stickers

Zheng, W. (09 de Agosto de 2020). *What is WeChat and what can it do?* . Obtenido de CGTN: <https://newseu.cgtn.com/news/2020-08-09/What-is-WeChat-and-what-can-it-do--SNepY1rgNG/index.html>

11. Anexos

11.1. Anexo 1. Ejemplo hoja de cálculo para una asignatura

INFORMACIÓN HOJA DE CÁLCULO	INFORMACIÓN ASIGNATURA	INFORMACIÓN ASIGNATURA	INFORMACIÓN ASIGNATURA	INFORMACIÓN PROFESORES/AS	INFORMACIÓN PROFESORES/AS	INFORMACIÓN DOCENCIA	INFORMACIÓN DOCENCIA	INFORMACIÓN DOCENCIA	INFORMACIÓN DOCENCIA	SISTEMA EVALUACIÓN	SISTEMA EVALUACIÓN	SISTEMA EVALUACIÓN	SISTEMA EVALUACIÓN	SISTEMA EVALUACIÓN	SISTEMA EVALUACIÓN				
¿HOJA DE CÁLCULO ACCESIBLE?	ASIGNATURA	GRADO	CURSO	NOMBRE PROFESOR/IA	E-MAIL PROFESOR/IA	ENLACE TEORÍA	ENLACE PRÁCTICAS GRUPO 1	ENLACE PRÁCTICAS GRUPO 2	ENLACE PRÁCTICAS GRUPO 3	PONDERACIÓN NOTA PRÁCTICAS	PONDERACIÓN NOTA TEORÍA	PONDERACIÓN ASISTENCIA Y BALANCEO FINAL	NOTA MÍNIMA EN PRÁCTICAS PARA ASISTENCIA	NOTA MÍNIMA EN TEORÍA PARA ASISTENCIA	GUÍA DOCENTE				
SI	Inaccesible	Inaccesible	Inaccesible	Siempre accesible	Siempre accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible				
Motivo	Desarrollo Ágil	Ingeniería Informática	2019-2020	Victor Manuel Rivas Santos	vivas@ujaen.es	https://meet.google.com/desarrolloagil-1	https://meet.google.com/desarrolloagil-2	https://meet.google.com/desarrolloagil-3	https://meet.google.com/desarrolloagil-4	60/100	30/100	10/100	Igual o superior al 50/100 del máximo	Igual o superior al 70/100 del máximo	https://virtual.ujaen.es/que-es/informacion-accesible-y-no-accesible				
CORREO ELECTRÓNICO INSTITUCIONAL	DNI	PRIMER APELLIDO	SEGUNDO APELLIDO	NOMBRE	NOMBRE COMPLETO	ASISTENCIAS TEORÍA	ASISTENCIAS PRÁCTICAS	PUNTUACIÓN POR ASISTENCIA TEORÍA	PUNTUACIÓN POR ASISTENCIA PRÁCTICAS	GRUPO DE PRÁCTICAS	PRÁCTICA 1	PRÁCTICA 2	PRÁCTICA 3	PRÁCTICA 4	PRÁCTICA 5	PRÁCTICA 6	TOTAL PRÁCTICAS	EXAMEN TEORÍA	NOTA FINAL
Inaccesible	Inaccesible	Inaccesible	Inaccesible	Inaccesible	Inaccesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible	Accesible
fmb0011@red.ujaen.es	77435663E	MARTÍNEZ	BUENO	FRANCISCO JAVIER	MARTÍNEZ BUENO, FRANCISCO JAVIER	7	7	0,50	0,50	7	0,00	1,20	1,10	1,20	1,10	1,00	5,60	3,00	9,60
uja0011@red.ujaen.es	11111111A	APELLIDO1	APELLIDO2	NOMBRE1	APELLIDO1 APELLIDO2, NOMBRE1	4	3	0,40	0,38	8	0,00	1,12	0,95	1,19	0,39	0,82	4,47	2,84	8,09
uja0012@red.ujaen.es	11111111B	APELLIDO3	APELLIDO4	NOMBRE2	APELLIDO3 APELLIDO4, NOMBRE2	2	3	0,20	0,38	10	0,00	0,86	0,96	0,77	0,69	0,84	4,12	1,47	6,17
uja0013@red.ujaen.es	11111111C	APELLIDO5	APELLIDO6	NOMBRE3	APELLIDO5 APELLIDO6, NOMBRE3	5	1	0,50	0,13	9	0,00	0,65	0,35	0,47	1,02	0,97	3,46	2,91	7,00
uja0014@red.ujaen.es	11111111D	APELLIDO7	APELLIDO8	NOMBRE4	APELLIDO7 APELLIDO8, NOMBRE4	3	3	0,30	0,38	1	0,00	1,19	1,11	0,2	0,12	0,29	2,91	2,68	6,27
uja0015@red.ujaen.es	11111111E	APELLIDO9	APELLIDO10	NOMBRE5	APELLIDO9 APELLIDO10, NOMBRE5	2	3	0,20	0,38	6	0,00	0,6	0,58	0,34	0,76	0,84	3,12	1,25	4,95
uja0016@red.ujaen.es	11111111F	APELLIDO11	APELLIDO12	NOMBRE6	APELLIDO11 APELLIDO12, NOMBRE6	1	4	0,10	0,50	7	0,00	0,85	0,42	1,04	0,18	0,42	2,91	2,91	6,42
uja0017@red.ujaen.es	11111111G	APELLIDO13	APELLIDO14	NOMBRE7	APELLIDO13 APELLIDO14, NOMBRE7	2	1	0,20	0,13	5	0,00	0,33	0,73	0,78	0,38	0,53	2,75	1,25	4,33
uja0018@red.ujaen.es	11111111H	APELLIDO15	APELLIDO16	NOMBRE8	APELLIDO15 APELLIDO16, NOMBRE8	5	3	0,50	0,38	2	0,00	0,8	0,97	0,1	0,33	0,53	2,73	1,36	4,97
uja0019@red.ujaen.es	11111111I	APELLIDO17	APELLIDO18	NOMBRE9	APELLIDO17 APELLIDO18, NOMBRE9	4	1	0,40	0,13	9	0,00	0,46	0,33	1,02	1,2	1,06	4,07	2,4	7,00
uja0020@red.ujaen.es	11111111J	APELLIDO19	APELLIDO20	NOMBRE10	APELLIDO19 APELLIDO20, NOMBRE10	1	2	0,10	0,25	3	0,00	1,05	0,27	0,87	0,77	0,33	3,29	2,14	5,78
uja0021@red.ujaen.es	11111111K	APELLIDO21	APELLIDO22	NOMBRE11	APELLIDO21 APELLIDO22, NOMBRE11	5	1	0,50	0,13	1	0,00	0,85	0,9	0,57	0,86	1,09	4,27	2,14	7,04
uja0022@red.ujaen.es	11111111L	APELLIDO23	APELLIDO24	NOMBRE12	APELLIDO23 APELLIDO24, NOMBRE12	5	2	0,50	0,25	8	0,00	0,27	0,35	0,79	0,57	1	2,98	1,14	4,87
uja0023@red.ujaen.es	11111111M	APELLIDO25	APELLIDO26	NOMBRE13	APELLIDO25 APELLIDO26, NOMBRE13	1	5	0,10	0,50	1	0,00	0,56	1,19	0,73	0,69	0,63	3,80	1,04	5,44
uja0024@red.ujaen.es	11111111N	APELLIDO27	APELLIDO28	NOMBRE14	APELLIDO27 APELLIDO28, NOMBRE14	3	3	0,30	0,38	8	0,00	0,06	1,16	0,83	0,95	1,13	4,13	1,86	6,67
uja0025@red.ujaen.es	11111111O	APELLIDO29	APELLIDO30	NOMBRE15	APELLIDO29 APELLIDO30, NOMBRE15	4	2	0,40	0,25	4	0,00	0,48	1,11	0,05	0,88	0,21	2,73	2,57	5,95
uja0026@red.ujaen.es	11111111P	APELLIDO31	APELLIDO32	NOMBRE16	APELLIDO31 APELLIDO32, NOMBRE16	1	5	0,10	0,50	7	0,00	0,37	0,71	0,6	0,46	0,29	2,43	2,62	5,65
uja0027@red.ujaen.es	11111111Q	APELLIDO33	APELLIDO34	NOMBRE17	APELLIDO33 APELLIDO34, NOMBRE17	5	1	0,50	0,13	10	0,00	0,44	0,76	0,49	0,55	0,78	3,02	2,96	6,61
uja0028@red.ujaen.es	11111111R	APELLIDO35	APELLIDO36	NOMBRE18	APELLIDO35 APELLIDO36, NOMBRE18	2	4	0,20	0,50	6	0,00	0,29	0,82	0,11	0,5	0,75	2,47	2,65	5,82
uja0029@red.ujaen.es	11111111S	APELLIDO37	APELLIDO38	NOMBRE19	APELLIDO37 APELLIDO38, NOMBRE19	2	4	0,20	0,50	6	0,00	0,79	0,34	0,39	0,85	0,21	2,58	1,04	4,32
uja0030@red.ujaen.es	11111111T	APELLIDO39	APELLIDO40	NOMBRE20	APELLIDO39 APELLIDO40, NOMBRE20	1	2	0,10	0,25	7	0,00	0,01	1,18	0,59	0,25	0,44	2,47	1,71	4,53
uja0031@red.ujaen.es	11111111U	APELLIDO41	APELLIDO42	NOMBRE21	APELLIDO41 APELLIDO42, NOMBRE21	3	1	0,30	0,13	3	0,00	0,92	0,18	0,16	1,07	1,05	3,38	1,36	5,17
uja0032@red.ujaen.es	11111111V	APELLIDO43	APELLIDO44	NOMBRE22	APELLIDO43 APELLIDO44, NOMBRE22	2	4	0,20	0,50	1	0,00	0,05	1,13	1	0,24	0,64	3,06	1,03	4,79
uja0033@red.ujaen.es	11111111W	APELLIDO45	APELLIDO46	NOMBRE23	APELLIDO45 APELLIDO46, NOMBRE23	1	2	0,10	0,25	5	0,00	0,38	0,04	0,87	1,03	1,02	3,34	2,32	6,01
uja0034@red.ujaen.es	11111111X	APELLIDO47	APELLIDO48	NOMBRE24	APELLIDO47 APELLIDO48, NOMBRE24	2	4	0,20	0,50	8	0,00	0,3	0,73	0,47	0,14	0,76	2,40	1,37	4,47
uja0035@red.ujaen.es	11111111Y	APELLIDO49	APELLIDO50	NOMBRE25	APELLIDO49 APELLIDO50, NOMBRE25	4	2	0,40	0,25	5	0,00	0,97	0,97	0,78	0,37	0,06	3,15	1,69	5,49
uja0036@red.ujaen.es	11111111Z	APELLIDO51	APELLIDO52	NOMBRE26	APELLIDO51 APELLIDO52, NOMBRE26	5	5	0,50	0,50	2	0,00	0,97	1,06	1,15	0,38	0,6	4,16	1,89	7,05
uja0037@red.ujaen.es	11111111A	APELLIDO53	APELLIDO54	NOMBRE27	APELLIDO53 APELLIDO54, NOMBRE27	2	5	0,20	0,50	5	0,00	0,71	0,66	0,47	0,71	0,68	3,23	1,12	5,05

Ilustración 47: hoja de cálculo de ejemplo perteneciente a la versión 1.0 del desarrollo del software