



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

SISTEMA DE PUBLICACIÓN DE DATOS DE POLEN A TRAVÉS DE REDES SOCIALES

Alumno: Juan Manuel Hermoso Ureña

Tutor: Arturo Montejo Ráez
Dpto: Informática

Septiembre, 2021

(Página intencionalmente en blanco)



Universidad de Jaén

Departamento de Informática

Don Arturo Montejo Ráez, tutor del Trabajo Fin de Grado titulado: '**Sistema de publicación de datos de polen a través de redes sociales**', que presenta Don Juan Manuel Hermoso Ureña, otorga el visto bueno para su entrega y defensa en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2021

El alumno:

El tutor:

Juan Manuel Hermoso Ureña

Arturo Montejo Ráez

(Página intencionalmente en blanco)

Agradecimientos

A mi familia, por apoyarme incondicionalmente ante mi deseo de ser Ingeniero Informático. En especial a mis padres, por creer en mí e inculcarme los valores necesarios para llegar a ser lo que hoy soy y por aconsejarme y guiarme durante toda mi vida académica a pesar de mis peculiaridades. Todo lo que soy y seré se lo debo a ellos.

También querría agradecer a mi pareja, ya que sin su motivación constante seguramente no estaría escribiendo esto.

A mi tutor, Arturo Montejo, por comprender mis obligaciones profesionales y prestar la ayuda justa en cada momento, así como a todo el personal docente que ha ayudado a mi formación.

Gracias.

Índice

1. INTRODUCCIÓN	11
1.1. Motivación.....	11
1.2. Objetivos	13
1.3. Metodología de desarrollo	14
2. ANÁLISIS.....	15
2.1. Antecedentes y herramientas actuales.....	16
2.1.1. Aplicaciones en Play Store.....	16
2.1.2. Web SEAIC	17
2.1.3. Web www.eltiempo.es.....	18
2.1.4. Web Hospital Universitario de Jaén.....	18
2.2. Pruebas de la utilidad real de la información difundida.....	19
2.3. Especificaciones del sistema.....	21
2.3.1. Flujo de comunicación del sistema con el administrador	21
2.3.2. Requisitos funcionales	23
2.3.3. Requisitos no funcionales	24
2.3.4. Casos de uso	25
2.4. Estudios previos del origen de los datos	32
2.4.1. Estudio del procedimiento de la medición de los niveles por parte del Hospital de Jaén	32
2.4.2. Estudio de la periodicidad de la publicación de los datos por parte del Hospital de Jaén	34
2.5. Estimación de costes	37
2.5.1. Hardware	37
2.5.2. Software.....	38
2.5.3. Amazon Web Services.....	39
2.5.4. Recursos humanos	41
2.5.5. Costes indirectos.....	42
3. PLANIFICACIÓN TEMPORAL	43
3.1. Diagrama de Gantt.....	44
4. DISEÑO INICIAL.....	45
4.1. Definición de estrategias a seguir.....	46
4.1.1. Web scraping para monitorizar cambios en la web del Hospital de Jaén.....	46
4.1.2. Captura de los datos nuevos detectados.....	48

4.1.3.	Edición del frame capturado del vídeo	49
4.2.	Workflow	53
4.3.	Elecciones previas al desarrollo	54
4.3.1.	Servicio de computación en la nube.....	54
4.3.2.	Lenguaje de programación.....	56
4.3.3.	Software.....	57
4.3.4.	Redes sociales para la difusión de los niveles de polen	59
4.4.	Arquitectura del sistema.....	61
5.	DESARROLLO DEL SISTEMA	63
5.1.	Configuraciones previas.....	64
5.1.1.	Amazon Web Services.....	64
5.1.2.	Programador de tareas de Windows	67
5.1.3.	API de Telegram y configuración de canales	70
5.1.4.	API de Facebook.....	72
5.1.5.	API de Instagram	73
5.1.6.	API de Twitter	74
5.2.	Implementación del código.....	76
5.2.1.	Librerías usadas.....	76
5.2.2.	Explicación del código.....	82
5.3.	Repositorio del proyecto.....	88
5.4.	Manual de despliegue	90
5.5.	Prueba real	92
6.	LÍNEAS DE FUTURO.....	98
6.1.	Posible monetización del sistema.....	99
6.2.	Posible adaptación a otras ciudades	102
6.3.	Más redes sociales futuras.....	105
6.4.	Cambios en la web del Hospital de Jaén.....	106
7.	CONCLUSIONES.....	106
7.1.	Posibles errores que se pueden producir	107
7.2.	Dificultades encontradas	107
7.3.	Conclusiones finales	109
	Bibliografía	111

Índice de Ilustraciones

Ilustración 1.1 Situación de este TFG en el proceso de publicación de los niveles de polen	12
Ilustración 1.2 Solución propuesta con los objetivos de este TFG a la ilustración 1.1	14
Ilustración 2.1 Apps de polen en Play Store	16
Ilustración 2.2 Reseñas de Apps de polen en Play Store	17
Ilustración 2.3 Web SEAIC	17
Ilustración 2.4 Web www.eltiempo.es	18
Ilustración 2.5 Web Hospital de Jaén 2011-2020 y Web Hospital de Jaén 2021	18
Ilustración 2.6 Seguidores en Telegram, Twitter, Facebook e Instagram a fecha 10/06/2021.....	19
Ilustración 2.7 Opiniones reales en la página de Facebook “Polen Jaén”	20
Ilustración 2.8 Noticia de jaen24h.com sobre los canales de difusión de Polen Jaén	21
Ilustración 2.9 Diagrama del flujo de comunicación Sistema-Administrador	22
Ilustración 2.10 Caso de uso: Inicio del sistema	26
Ilustración 2.11 Caso de uso: Detección de cambios	27
Ilustración 2.12 Caso de uso: Captura de los niveles y edición de la plantilla.....	28
Ilustración 2.13 Caso de uso: Solicitud de consentimiento para publicación en redes sociales	29
Ilustración 2.14 Caso de uso: Publicación de los niveles en redes sociales	30
Ilustración 2.15 Caso de uso: Informe de resolución	31
Ilustración 2.16 Captador de polen tipo Hirst	32
Ilustración 2.17 Microscopio de laboratorio.....	33
Ilustración 2.18 Mes de junio de 2021 señalando el día 11 festivo	36
Ilustración 2.19 Calculadora de precios de AWS	39
Ilustración 2.20 Estimación de la calculadora de precios AWS	40
Ilustración 2.21 Servicio Amazon EC2 gratuito los 12 primeros meses	40
Ilustración 4.1 Vídeo con los niveles de polen en el código fuente de la web del Hospital de Jaén	47
Ilustración 4.2 Secuencia del vídeo de la web del Hospital de Jaén	48
Ilustración 4.3 Ejemplo de frame capturado del vídeo de la web del Hospital de Jaén.....	49
Ilustración 4.4 Imagen terminada lista para publicar en las redes sociales	50
Ilustración 4.5 Frame original con fecha y logo borrados	51
Ilustración 4.6 Recorte del frame original con fecha y logo borrados	51
Ilustración 4.7 Recorte editado para promocionar los canales de difusión del sistema	52
Ilustración 4.8 Recorte editado para promocionar los canales de difusión del sistema y con fecha....	52
Ilustración 4.9 Resumen del proceso de edición del frame	52
Ilustración 4.10 Diagrama del flujo de trabajo del sistema.....	54
Ilustración 4.11 Logo Amazon Web Services.....	55
Ilustración 4.12 Logo de Python.....	56
Ilustración 4.13 Logo de Ninja IDE.....	57
Ilustración 4.14 Captura de pantalla del entorno Ninja IDE	58
Ilustración 4.15 Logo de Adobe Photoshop cc 2019.....	58
Ilustración 4.16 Logo de Microsoft Windows 10.....	59
Ilustración 4.17 Logo de Microsoft Office 2016	59
Ilustración 4.18 Iconos redes sociales para difusión de niveles de polen	60
Ilustración 4.19 Esquema general de la arquitectura y módulos del sistema	63
Ilustración 5.1 Arquitectura instancia EC2 + EBS en AWS.....	65
Ilustración 5.2 Elección de una imagen AMI para la instancia en AWS	66
Ilustración 5.3 Elección del tipo de instancia EC2 en AWS	66
Ilustración 5.4 Escritorio remoto de la instancia ec2 creada en AWS	67
Ilustración 5.5 Programador de tareas de Windows	68
Ilustración 5.6 Configuración del desencadenador del programador de tareas de Windows.....	69
Ilustración 5.7 Configuración de la acción a realizar por el programador de tareas de Windows	70
Ilustración 5.8 Esquema de la arquitectura del sistema con respecto a la API bot de Telegram.....	72

Ilustración 5.9 Panel del explorador de la API Graph de Facebook	73
Ilustración 5.10 Permisos para publicar en Instagram añadidos a la app creada en Facebook	74
Ilustración 5.11 Panel de desarrollador de Twitter de la aplicación creada	75
Ilustración 5.12 Panel de desarrollador de Twitter con Keys y Tokens de la aplicación creada	76
Ilustración 5.13 Librería BeautifulSoup	76
Ilustración 5.14 Librería requests	77
Ilustración 5.15 Librería time	77
Ilustración 5.16 Módulo datetime	78
Ilustración 5.17 Librería cv2	78
Ilustración 5.18 Librería PIL.....	79
Ilustración 5.19 Librería facebook	79
Ilustración 5.20 Librería tweepy	79
Ilustración 5.21 Librería telegram.....	80
Ilustración 5.22 Módulos InlineKeyboardButton, InlineKeyboardMarkup, Updater y CallbackQueryHandler	80
Ilustración 5.23 Librería threading	80
Ilustración 5.24 Librería json	81
Ilustración 5.25 Librería os	81
Ilustración 5.26 Librería dotenv	81
Ilustración 5.27 Función leerHtmlFichero	82
Ilustración 5.28 Función obtenerHtmlWeb	82
Ilustración 5.29 Función filtrarHtml.....	83
Ilustración 5.30 Procedimiento actualizacionHtmlFichero.....	83
Ilustración 5.31 Procedimiento descargarVideo	83
Ilustración 5.32 Procedimiento capturarFrame	84
Ilustración 5.33 Procedimiento avisoBotTelegram	85
Ilustración 5.34 Procedimiento publicarFacebook	85
Ilustración 5.35 Función publicarTwitter	86
Ilustración 5.36 Procedimiento publicarInstagram	87
Ilustración 5.37 Procedimiento publicarTelegram	87
Ilustración 5.38 Repositorio del proyecto	88
Ilustración 5.39 Contenido de la carpeta 'Images' del repositorio del proyecto	89
Ilustración 5.40 Comunicaciones al administrador de comienzo del sistema.....	93
Ilustración 5.41 Comunicaciones al administrador de cambios detectados y petición de continuación.....	94
Ilustración 5.42 Comunicaciones al administrador de detección del sistema por orden recibida.....	95
Ilustración 5.43 Comunicaciones al administrador de publicación en RRSS tras orden recibida	96
Ilustración 5.44 Publicación de los niveles de polen en Facebook, Instagram, Twitter y Telegram.....	97
Ilustración 5.45 Comunicaciones al administrador de finalización del sistema por falta de cambios...	98
Ilustración 6.1 Medicamento Rino-Ebastel.....	100
Ilustración 6.2 Empresa Persianas Efalú	100
Ilustración 6.3 Grupo Aldes	101
Ilustración 6.4 Clínica Fisioterapia, Haloterapia y Osteopatía	101
Ilustración 6.5 Mapa de captadores de SEAIC	102
Ilustración 6.6 Información del captador de Granada	103
Ilustración 6.7 Información de los captadores de Castilla y León	103
Ilustración 6.8 Información de los captadores de Extremadura	104
Ilustración 6.9 Información de los captadores de Castilla-La Mancha	104
Ilustración 6.10 Información diaria de los captadores de la Comunidad de Madrid	105
Ilustración 7.1 Comparativa de las webs de publicación de niveles de polen del Hospital de Jaén antigua y nueva	108
Ilustración 7.2 Salida por consola del error producido con la librería instabot en la instancia de AWS	109

Índice de Tablas

Tabla 2.1 Descripción de las comunicaciones Sistema-Administrador	23
Tabla 2.2 Caso de uso: Inicio del sistema.....	26
Tabla 2.3 Caso de uso: Detección de cambios.....	27
Tabla 2.4 Caso de uso: Captura de los niveles y edición de la plantilla	28
Tabla 2.5 Caso de uso: Solicitud de consentimiento para publicación en redes sociales.....	29
Tabla 2.6 Caso de uso: Publicación de los niveles en redes sociales.....	30
Tabla 2.7 Caso de uso: Informe de resolución	31
Tabla 2.8 Horarios de publicación de los niveles de polen por parte del Hospital de Jaén en 2021 ...	36
Tabla 2.9 Costes de Hardware.....	38
Tabla 2.10 Costes de Software	39
Tabla 2.11 Costes AWS	41
Tabla 2.12 Costes de recursos humanos.....	41
Tabla 2.13 Costes indirectos	42
Tabla 2.14 Costes totales del proyecto	42
Tabla 3.1 Diagrama de Gantt (1).....	45
Tabla 3.2 Diagrama de Gantt (2).....	45

1. INTRODUCCIÓN

Como introducción a este Trabajo Fin de Grado (TFG) se va a comenzar explicando qué ha motivado su realización, exponiendo el cómo y por qué surgió esta idea y qué necesidad se detectó para intentar cubrir con este proyecto.

También se pasará a definir el propósito y los objetivos que se pretenden alcanzar al finalizar la elaboración de este proyecto.

Por último, se analizarán y razonará la elección del seguimiento de una metodología de desarrollo software a la hora de elaborar este proyecto.

1.1. Motivación

En el año 2011 el Complejo Hospitalario de Jaén se hizo con un captador de polen tipo Hirst, el cual fue instalado en la terraza de su centro Materno-Infantil. Desde entonces, se vienen recabando diariamente los niveles de los distintos pólenes presentes en la atmósfera de la ciudad entre los meses de abril y junio, el periodo de mayor floración en la zona.

Estos datos son recogidos y enviados a la red de la Sociedad Española de Alergología e Inmunología Clínica, cuyo principal objetivo es potenciar la investigación en el área de la aerobiología (estudio de los pólenes y hongos que se encuentran en el aire) y polinosis (enfermedades alérgicas producidas por los pólenes). Los niveles recogidos por los captadores de la red repartidos por la geografía española son añadidos días más tarde a su web (www.polenes.com) donde se pueden consultar libremente.

Esos datos son útiles a nivel de investigación, sin embargo, carecen de interés para las personas alérgicas acceder a los datos de días atrás. La ventaja de conocerlos con rapidez es que los pacientes puedan saber cuándo y con qué intensidad iniciar medidas físicas, como es evitar su contacto con el polen, y farmacológicas, como son la toma de antihistamínicos o inhaladores, que les reporte un mayor control de su patología alérgica y, por tanto, una mejora de su calidad de vida durante estas fechas.

Es por eso que desde el Hospital de Jaén empezaron a publicar los datos en su web inmediatamente después de que los correspondientes expertos lo analizaran en el laboratorio, provocando así alcanzar esa utilidad para las personas alérgicas de Jaén de conocer los niveles de polen en la atmósfera que les rodea en un espacio temporal prácticamente inmediato.

Es aquí donde radica la principal motivación de este proyecto, ya que aprovechando esa situación favorable que existe en la ciudad Jaén en cuanto a la inmediatez de la publicación de los datos de polen, se ha elaborado este trabajo con la finalidad de que esos datos sean realmente aprovechados por las personas alérgicas y demás interesados difundiendo y haciéndoles llegar esa información a través de la vía más directa, sencilla y asumible para el ciudadano de a pie: las redes sociales.

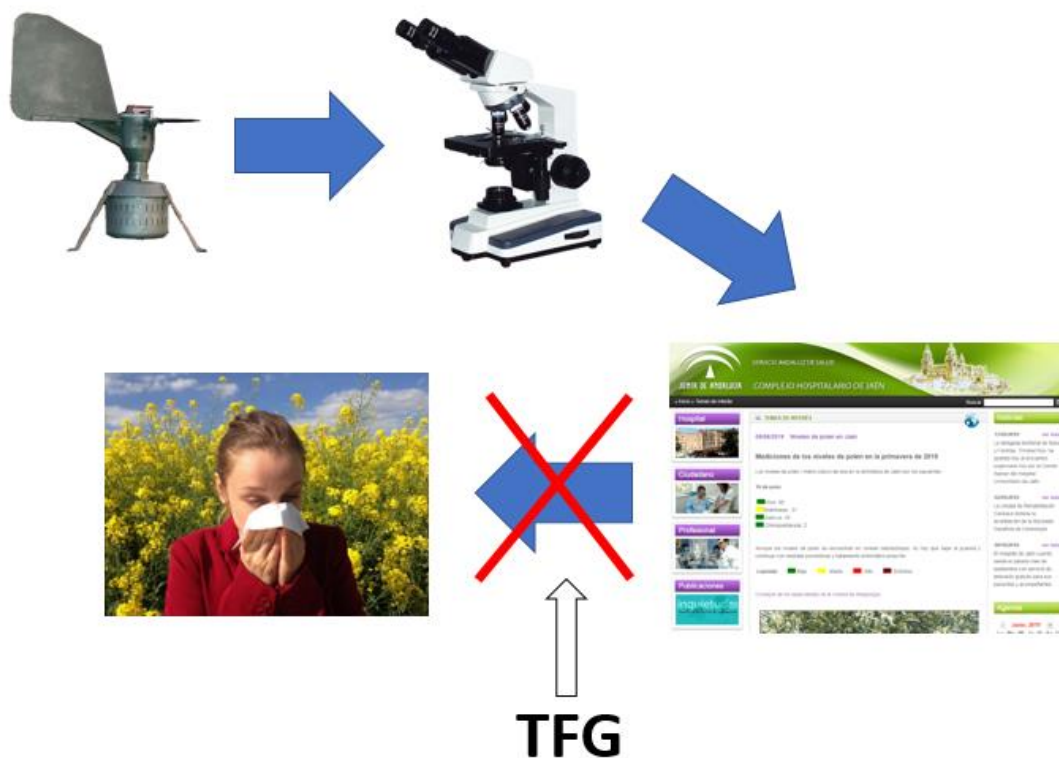


Ilustración 1.1 Situación de este TFG en el proceso de publicación de los niveles de polen

1.2. Objetivos

Se procede ahora a detallar los principales objetivos de este trabajo y que se esperan alcanzar tras finalizar su desarrollo.

- Crear y configurar un sistema encargado de automatizar el proceso de publicación en redes sociales de los niveles de polen para la ciudad de Jaén que se actualizan en la web del Hospital.
- Monitorizar la web del Hospital de Jaén para detectar justo el momento en el que se actualizan los niveles de polen.
- Que el proceso automatizado se ejecute de manera independiente de cualquier equipo local, aprovechando así todas las ventajas que ello conlleva.
- Avisar mediante una notificación al móvil personal del administrador de cuándo se ha producido el cambio.
- Personalizar convenientemente la plantilla con los niveles de polen que posteriormente se difundirán en las distintas redes sociales.
- Solicitar aprobación del administrador justo antes de publicar los niveles de polen para evitar que se publique información errónea o en formato defectuoso.
- Publicar la plantilla personalizada con los niveles de polen en las redes sociales que servirán de canales de comunicación con las personas interesadas.
- Comunicar al administrador, para tranquilidad de este, cada uno de los pasos importantes por los que va transcurriendo el flujo de ejecución del proceso de automatización.

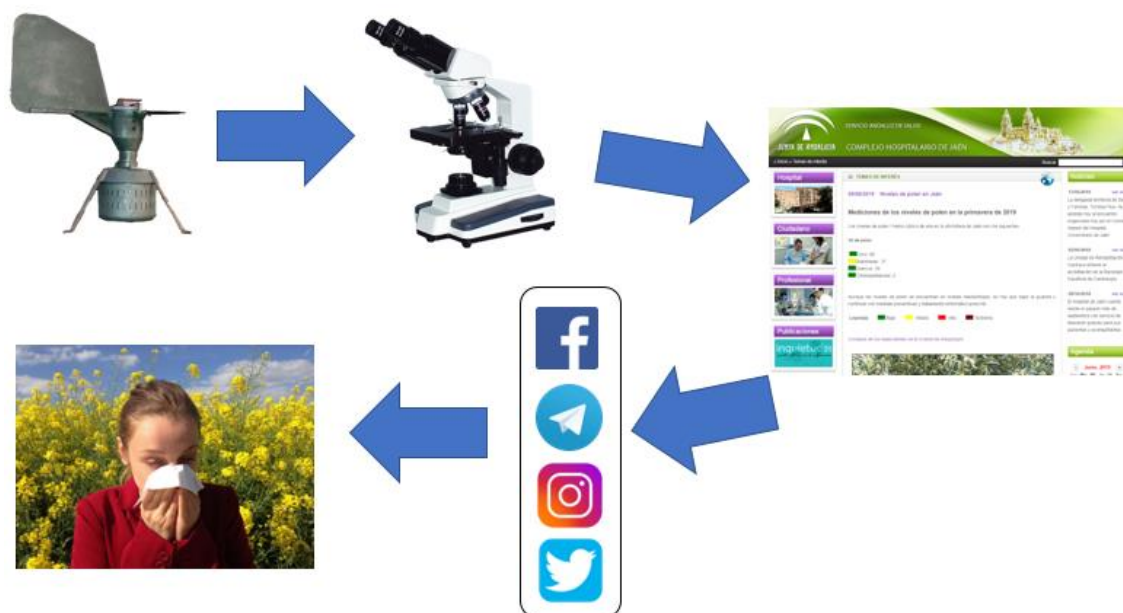


Ilustración 1.2 Solución propuesta con los objetivos de este TFG a la ilustración 1.1

1.3. Metodología de desarrollo

En un principio, el presente TFG pretende ser desarrollado utilizando alguna de las metodologías aprendidas en ciertas asignaturas impartidas durante de la carrera, en concreto, alguna metodología ágil [1]. Estos métodos de ingeniería del software están basados en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.

Primeramente, se pensó en seguir una metodología Scrum [2], que fue la más presente en los años de enseñanza durante la carrera. Scrum, basada en pequeños ciclos temporales (sprints) en los que al final de cada uno de ellos, hemos de ser capaces de entregar una versión del producto estable.

Finalmente, esta metodología ha sido descartada, ya que mis obligaciones profesionales hacen que el tiempo que voy a dedicar a este proyecto sea irremediabilmente irregular y poco continuado. Habrá semanas que mi dedicación al desarrollo sea prácticamente nula y otras en las que me sea posible avanzar notablemente. Además, se prevé que los cambios sean continuados durante el progreso de este TFG, circunstancia a evitar en dicho marco de trabajo.

La metodología elegida y que se considera más adaptada a este proyecto es Kanban [3], surgida a finales de los años 40, cuando Toyota empezó a optimizar sus procesos de ingeniería a partir del modelo que utilizaban los supermercados para llenar las estanterías. Los supermercados almacenan únicamente los productos suficientes para suplir la demanda del cliente, una práctica que optimiza el flujo entre el supermercado y el cliente. Como las existencias coinciden con las pautas de consumo, el supermercado aumenta considerablemente su eficacia en gestión de existencias reduciendo la cantidad de excedentes que debe mantener en cualquier momento. Mientras tanto, el supermercado puede seguir garantizando que ese producto que el cliente necesita siempre esté disponible.

Flujo único y continuo de trabajo, flexibilidad en los entregables y su gran adaptación a los cambios constituyen las razones principales para esta elección.

2. ANÁLISIS

En este capítulo clave del proyecto se van a examinar las diferentes herramientas actuales que hay disponibles para la consulta de los niveles de polen en la provincia de Jaén, destacando las deficiencias de cada una de ellas, las cuales han tenido como consecuencia la elaboración de este sistema.

Posteriormente, se mostrarán pruebas reales de que el trabajo que llevo realizando durante algunos años en primavera publicando los niveles de polen manualmente en las distintas redes sociales son útiles según experiencias y opiniones de algunos de sus seguidores.

Tras esto, se van a examinar las principales especificaciones requeridas que se deben cumplir junto con los casos de uso, y además, se analizarán todos los estudios previos a realizar sobre el origen de los datos para un mayor conocimiento del contexto en el que son extraídos.

Por último, se hará una estimación orientativa de los gastos en los que se pueden incurrir durante el desarrollo de este proyecto.

2.1. Antecedentes y herramientas actuales

Se va proceder ahora a explicar la situación en cuanto a las escasas posibilidades que tienen actualmente los alérgicos al polen en Jaén para estar al tanto sobre dichos niveles, siempre dejando al margen las publicaciones en redes sociales que se realizan en este trabajo.

2.1.1. Aplicaciones en Play Store

Existen algunas aplicaciones en la tienda de Android para tal fin a nivel nacional. Puedes elegir la provincia en la que estás interesado y los tipos de pólenes de los que quieres recibir los niveles. Se ha procedido a descargar y probar tres de las aplicaciones con más relevancia de este tipo.



Ilustración 2.1 Apps de polen en Play Store

Sin embargo, la eficacia de estas herramientas es prácticamente nula. Se han observado datos notablemente desactualizados, niveles genéricos e inexactos (alto, medio, bajo, en lugar de datos cuantitativos) además de otras deficiencias a nivel usabilidad de las aplicaciones. Como ejemplo, se exponen algunas reseñas reales dejadas en dicha tienda de aplicaciones:



Ilustración 2.2 Reseñas de Apps de polen en Play Store

2.1.2. Web SEAIC

La web SEAIC (www.polenes.com), de la Sociedad Española de Alergología e Inmunología Clínica, tiene resumidos para su consulta los datos de todos los captadores pertenecientes a su red. Sin embargo, una vez más nos encontramos con que los niveles son actualizados varios días después de ser medidos.



Ilustración 2.3 Web SEAIC

2.1.3. Web www.eltiempo.es

En la conocida web meteorológica www.eltiempo.es están disponibles los niveles genéricos a modo de previsión para la semana en curso. Por tanto, se trata únicamente de niveles orientativos, dentro de una previsión que puede cumplirse o no y referentes a un espacio de tiempo de varios días.



Tipos de polen	Valor
Olivo	A
Gramíneas	M
Ortigas	B
Robles y encinas	B

Ilustración 2.4 Web www.eltiempo.es

2.1.4. Web Hospital Universitario de Jaén

Como se explicó anteriormente en el apartado 'Motivación', en la web de Hospital de Jaén se llevan publicando los niveles durante la primavera con inmediatez y con exactitud desde el año 2011.



Ilustración 2.5 Web Hospital de Jaén 2011-2020 y Web Hospital de Jaén 2021

Por tanto, en esta ocasión sí conseguimos esa utilidad para las personas alérgicas, sin embargo, con este proyecto se quiere conseguir que estos datos lleguen a esas personas con la misma exactitud e inmediatez, pero, además, por la vía directa de las redes sociales.

2.2. Pruebas de la utilidad real de la información difundida

En la primavera del 2016 creé una página en Facebook llamada “Polen Jaén” donde empecé a difundir manualmente los niveles de polen que se publicaban en la web del Hospital de Jaén. Poco después, creé un canal en Telegram llamado también “Polen Jaén” para realizar el mismo trabajo y que llegara esa información a los interesados cada día a través de una notificación en su móvil. Durante las siguientes primaveras también hice lo propio creando y publicando los niveles en un perfil de Twitter y otro de Instagram.

Con el paso del tiempo la cantidad de usuarios interesados en la información que se difundía en cada una de las redes sociales fue creciendo y creciendo hasta alcanzar las estadísticas de hoy en día. A continuación, se muestran los seguidores totales en cada red social:

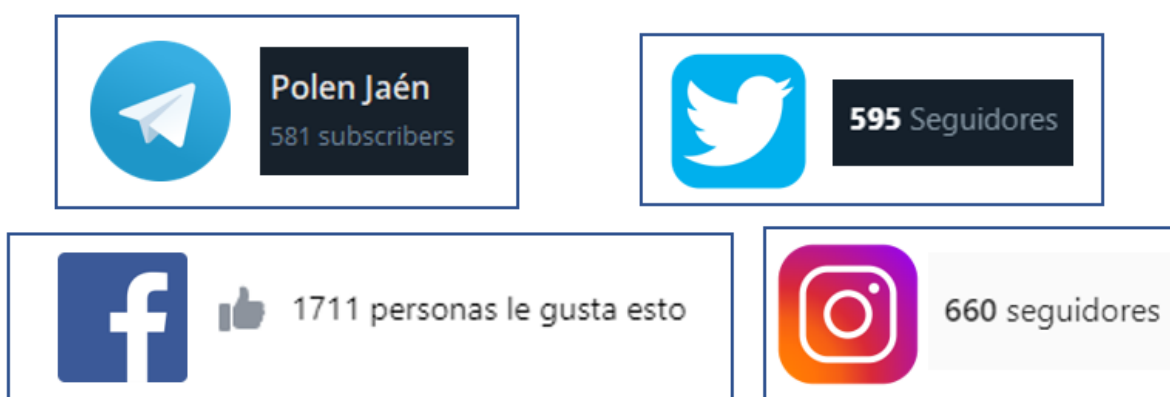
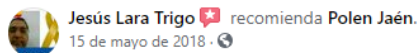


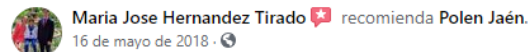
Ilustración 2.6 Seguidores en Telegram, Twitter, Facebook e Instagram a fecha 10/06/2021

Estos datos suman un total de 3547 seguidores que por una vía u otra están al tanto de los niveles de polen en Jaén gracias a la información que se difunde.

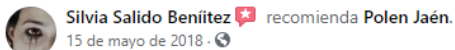
En la siguiente ilustración se dejan también algunas reseñas vertidas en la página de Facebook con opiniones reales de usuarios que se informan de los niveles de polen a través de estas publicaciones.



Es una página muy útil para saber los niveles de polen de toda la provincia



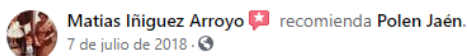
Nos ayuda a todos los alérgicos a estar informados.



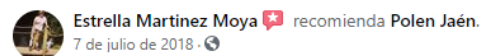
Por toda la información q nos ofrecen. Gracias por vuestro trabajo



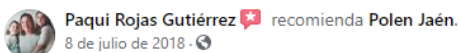
Se agradece el trabajo y la dedicación. Sin duda, el mejor sitio para estar informado. El extra de tener un canal en Telegram es lo mejor!



Es una maravilla. Gracias por vuestra ayuda



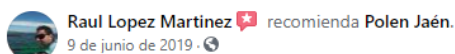
La mejor información para los alérgicos



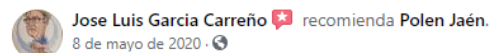
Agradecida de mantenerme informada para enfrentarme a la terrible alergia...



Muy útil para mantenernos informados sobre los niveles del polen



por su gran labor diaria, muchas gracias



Estay haciendo una buena labor con los Alérgicos. Muchas gracias

Ilustración 2.7 Opiniones reales en la página de Facebook “Polen Jaén”

Además, tras consultar numerosas listas elaboradas por expertos alergólogos con recomendaciones para paliar los síntomas de la alergia primaveral, en prácticamente todas se aconseja estar al tanto de los niveles de polen de tu ciudad [4].

Por último, también destacar que esta forma de publicar los niveles de polen ha sido incluso noticia en diferentes medios digitales en forma de artículos informativos. Aquí se muestra a modo de ejemplo uno publicado por jaen24h.com en el año 2018 (<http://jaen24h.com/actualidad/altos-niveles-de-polen-en-jaen-nuevo-canal-de-telegram-para-estar-informado-de-todo>)

Altos niveles de polen en Jaén... ¡Nuevo canal de Telegram para estar informado de todo!

Publicado por [Sergio López](#) el Jue, 24/05/2018 - 08:42



La primavera es una época del año temida para los alérgicos al polen, y como sabemos, muy especialmente aquí en Jaén para los alérgicos al polen del olivo. Uno de los muchos consejos que dan los especialistas es estar al tanto de los niveles polínicos en la atmósfera para así tomar las medidas preventivas necesarias. Y es aquí justamente donde surge la idea. ¿Por qué no habilitar un medio donde se reciba a diario esa información con una notificación en el móvil? Pues así se ha hecho apoyándose en la aplicación Telegram.

Para los que no la conozcan, Telegram es una app de mensajería instantánea del estilo de WhatsApp. Entre otras muchas de las funcionalidades que tiene se encuentran los canales, que son plataformas en las que el administrador puede enviar mensajes los cuales son recibidos por todos los miembros suscritos al canal.

Relacionado con:

Salud

Así pues, para todos los interesados en recibir esa información a diario, solo deben tener instalado Telegram si no lo tienen aún (disponible en Google Play y App Store), acceder al canal buscando @polenjaen o directamente a través del siguiente enlace <https://t.me/polenjaen> y unirse.

Cabe destacar que el origen de todos estos datos son los recabados por el captador de polen tipos Hirst que se encuentra en el Complejo Hospitalario de Jaén.

Y por último, informar que también pueden seguir a través de las redes sociales a Polen Jaén, donde publicamos gráficas semanales y noticias de interés sobre la alergia al polen en la provincia.

- Facebook: facebook.com/polenjaen
- Instagram: instagram.com/polenjaen
- Twitter: twitter.com/polen_jaen

Ilustración 2.8 Noticia de jaen24h.com sobre los canales de difusión de Polen Jaén

Todas estas razones entre otras muchas más, conforman pruebas reales de que conocer los niveles de polen de tu zona en un corto espacio de tiempo es indudablemente útil y aconsejable para todas las personas alérgicas.

2.3. Especificaciones del sistema

Se procede ahora a definir las características generales que se espera que se cumplan en el sistema a desarrollar. Para ello, se comienza describiendo el flujo de comunicación deseado entre el administrador del sistema y el propio sistema, para después enumerar los correspondientes requisitos funcionales y no funcionales junto con los casos de uso que se implementarán.

2.3.1. Flujo de comunicación del sistema con el administrador

En esta sección se va a proceder a definir y detallar los distintos puntos de comunicación entre el sistema y el administrador. Este flujo de comunicación se realizará apoyándose en un canal privado de Telegram al que únicamente tendrá

acceso el administrador y desde el que podrá estar informado del transcurrir del sistema e incluso llegar a interactuar con él en el momento de confirmar la publicación en redes sociales.

Esto último debe considerarse requisito indispensable para el correcto funcionamiento del sistema, para después de eso poder elegir si se continua con la ejecución del programa o, por el contrario, se decide cancelar por la detección de algún error y evitar así publicar algo indeseado.

Dotando al sistema de esta funcionalidad se consiguen dos cosas que considero muy importantes: conocer el seguimiento de la ejecución del programa y dar mi consentimiento o revocarlo antes de publicar los datos en las redes sociales que son seguidas por miles de usuarios.

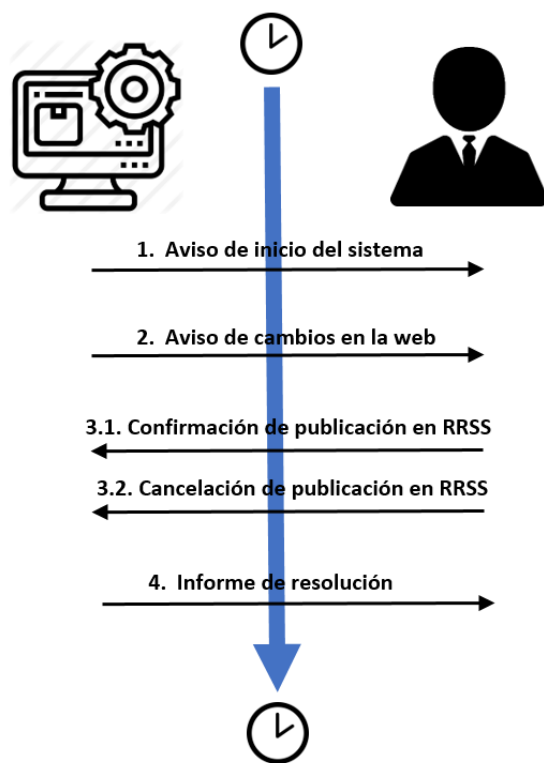


Ilustración 2.9 Diagrama del flujo de comunicación Sistema-Administrador

COMUNICACIÓN	EMISOR	RECEPTOR	DESCRIPCIÓN
1. Aviso de inicio del sistema	Sistema	Administrador	Se produce al iniciarse el sistema en el día y la hora a la que se haya programado.
2. Aviso de cambios en la web	Sistema	Administrador	Se envía la plantilla final al administrador justo antes de publicarla en RRSS, notificando así a la vez que se ha producido cambios en la web del Hospital de Jaén.
3.1. Confirmación de publicación en RRSS	Administrador	Sistema	Si le parece todo correcto, el administrador enviará una confirmación al sistema para publicar en RRSS.
3.2. Cancelación de publicación en RRSS	Administrador	Sistema	Si algo ha ido mal, el administrador cancelará la publicación en RRSS. Se necesitará revisión del sistema por su parte para identificar el error.
4. Informe de resolución	Sistema	Administrador	El sistema anunciará la finalización del proceso en el día actual informando de cómo ha sido la resolución.

Tabla 2.1 Descripción de las comunicaciones Sistema-Administrador

En el supuesto caso en el que un día transcurra el tiempo de monitorización máximo estimado y no lleguen a producirse cambios, las comunicaciones 2, 3.1 y 3.2 no se efectuarán, pasando directamente al informe de la resolución estipulada cuando se produce este acontecimiento.

Otro tipo de comunicación entre el sistema y el administrador, en este caso totalmente asíncrona, se realizará cuando este último revise cada día y después de la finalización del proceso la salida por consola en el que se informa con detalle de cómo ha funcionado todo. Esto será de especial importancia para la identificación de los posibles errores y su posterior resolución.

2.3.2. Requisitos funcionales

Tras la redacción del apartado anterior, se está en disposición de enumerar los requisitos funcionales [5] del sistema, que son aquellas funcionalidades que un sistema software implementa, definidas como un conjunto de entradas, comportamientos y salidas.

RF1. Inicio del sistema. El administrador configura el sistema para que se ejecute los días determinados a una hora específica. Al comenzar, se debe iniciar la

monitorización de la web del Hospital de Jaén a la espera de que se produzcan cambios en los niveles de polen.

RF2. Detección de cambios. El sistema detecta cuándo se producen dichos cambios para empezar a realizar los pasos siguientes.

RF3. Captura de los niveles y edición de la plantilla. El sistema deberá capturar los nuevos datos de los niveles y realizará las ediciones oportunas para poder publicarlos posteriormente.

RF4. Solicitud de consentimiento para publicación en redes sociales. El sistema informará al administrador de que se han producido los cambios y enviará la plantilla ya editada para que dé su consentimiento para publicar en las diferentes redes sociales o lo revoque.

RF5. Publicación de los niveles en redes sociales. El sistema publicará con el consentimiento del administrador la plantilla con los niveles de polen en Telegram, Facebook, Instagram y Twitter.

RF6. Informe de resolución. El sistema informará al administrador de si se ha publicado en redes sociales o no y de la finalización del sistema.

2.3.3. Requisitos no funcionales

Se trata de requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento y propiedades del sistema: rendimiento, seguridad, disponibilidad... En otras palabras, no hablan de “qué” hace el sistema, sino de “cómo” lo hace.

Por tanto, se puede concluir que existen numerosos requisitos no funcionales [6] que cumplir en el desarrollo del cualquier proyecto. Sin embargo, al tratarse de un trabajo de ámbito académico y no depender de ningún usuario que nos imponga restricciones, se optará en un principio por flexibilizar estos requisitos.

En este proyecto se aspira a desarrollar un sistema capaz de automatizar la publicación en redes sociales de los niveles de polen en la ciudad de Jaén. Se va a proceder pues, a especificar los requisitos no funcionales requeridos para conseguir alcanzar los objetivos pretendidos en este trabajo.

RNF1. Hacer uso de técnicas de web scraping, tanto para monitorizar la web del Hospital de Jaén para detectar cuándo se producen cambios, como para obtener los niveles actualizados de la manera más conveniente para ser tratados.

RNF2. El proceso se ejecutará en un sistema de computación en la nube como es Amazon Web Services (AWS) para la correcta independencia del proceso automatizado.

RNF3. El sistema será capaz de comunicarse con el smartphone del administrador a través de un canal privado de Telegram para mantenerlo informado del flujo de trabajo en el que se encuentra el proceso, además de para solicitar confirmación en el momento crítico justo antes de realizar la publicación en redes sociales.

RNF4. Será de gran importancia, por motivos que se aclararán más adelante, que se personalice convenientemente la plantilla final antes de su publicación en las redes sociales.

RNF5. Por último, y siempre con el visto bueno del administrador, el sistema será capaz de publicar la plantilla final personalizada de manera correcta en Facebook, Telegram, Instagram y Twitter, donde los usuarios podrán consultarlos libremente.

2.3.4. Casos de uso

Una vez definidos todos los requisitos del sistema, se está en disposición de definir los casos de uso [7]. Un caso de uso es la descripción de una acción o actividad del sistema, y su definición está apoyada en los requisitos funcionales descritos anteriormente.

Caso de uso – Inicio del sistema.

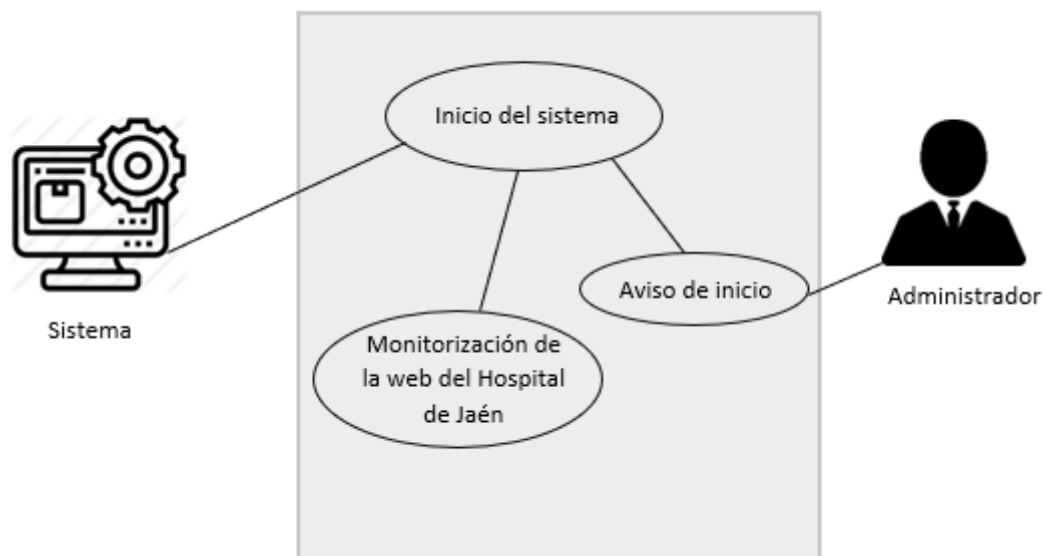


Ilustración 2.10 Caso de uso: Inicio del sistema

Inicio del sistema	
Actor primario	Sistema
Actores participantes	Sistema, administrador
Precondiciones	1. Se produce automáticamente en la fecha y hora establecida por el administrador.
Flujo	1. El script comienza su ejecución. 2. Inicio de la monitorización de la web del Hospital de Jaén. 3. Se notifica al administrador de dicho acontecimiento.
Poscondiciones	1. El sistema se queda monitorización la web del Hospital de Jaén a la espera de nuevos cambios.

Tabla 2.2 Caso de uso: Inicio del sistema

Caso de uso – Detección de cambios.

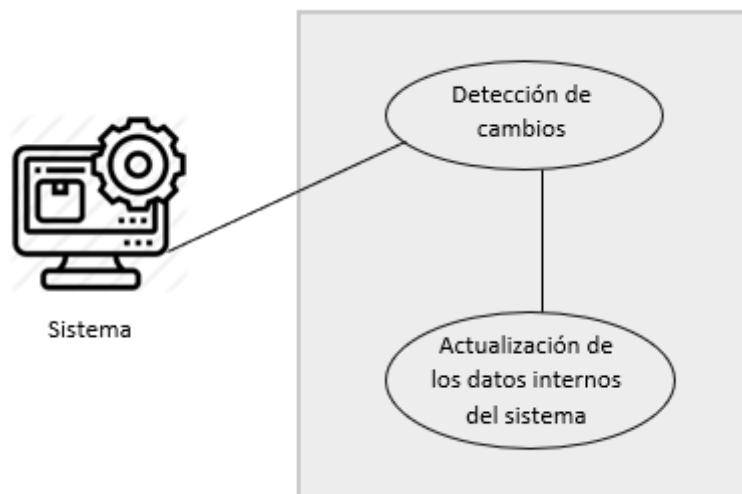


Ilustración 2.11 Caso de uso: Detección de cambios

Detección de cambios	
Actor principal	Sistema
Actores participantes	Sistema
Precondiciones	1. La web del Hospital de Jaén actualiza los datos de los niveles de polen.
Flujo	1. El sistema detecta dichos cambios. 2. Se paraliza la monitorización. 3. Se guardan los nuevos datos en el sistema para la monitorización del próximo inicio.
Poscondiciones	1. El sistema ha detectado que existen nuevos datos de niveles de polen correspondientes al día actual.

Tabla 2.3 Caso de uso: Detección de cambios

Caso de uso – Captura de los niveles y edición de la plantilla.

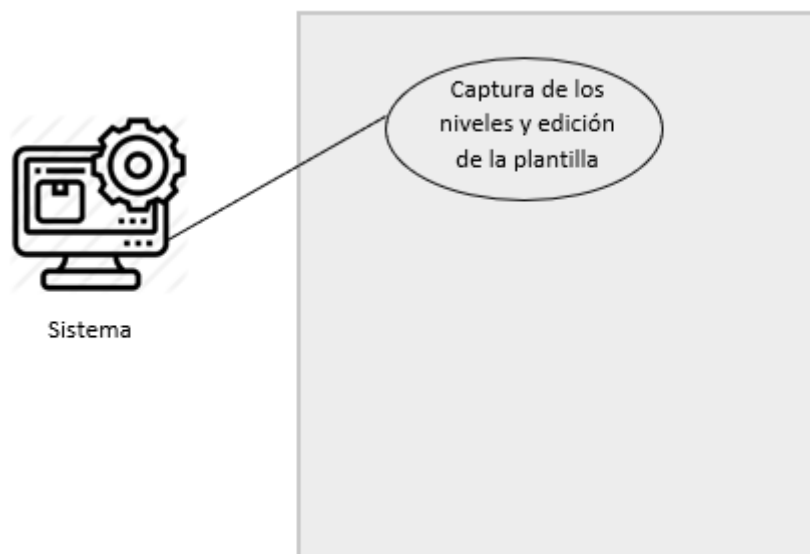


Ilustración 2.12 Caso de uso: Captura de los niveles y edición de la plantilla

Captura de los niveles y edición de la plantilla	
Actor primario	Sistema
Actores participantes	Sistema
Precondiciones	1. El sistema ha detectado actualización de los niveles de polen en la web del Hospital de Jaén.
Flujo	1. Se capturan los nuevos datos detectados. 2. Se edita convenientemente la plantilla con dichos niveles.
Poscondiciones	1. El sistema ya tiene preparada la plantilla final con los nuevos niveles de polen.

Tabla 2.4 Caso de uso: Captura de los niveles y edición de la plantilla

Caso de uso – Solicitud de consentimiento para publicación en redes sociales.

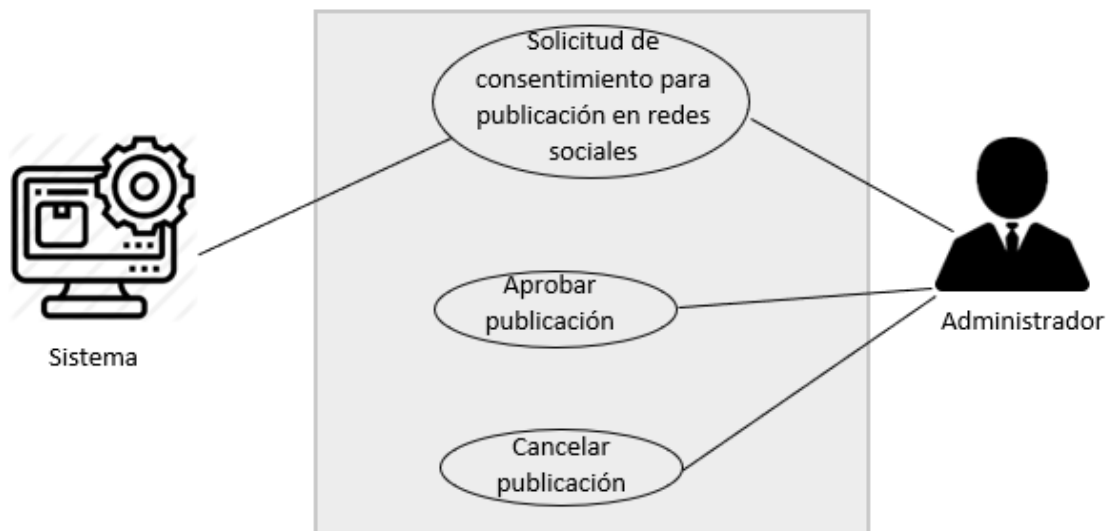


Ilustración 2.13 Caso de uso: Solicitud de consentimiento para publicación en redes sociales

Solicitud de consentimiento para publicación en redes sociales	
Actor primario	Administrador
Actores participantes	Administrador, sistema
Precondiciones	1. El sistema envía la plantilla con los nuevos niveles de polen al administrador para que lo pueda visualizar.
Flujo	1 El administrador observa la plantilla tras las ediciones realizadas. 2 El administrador decide si proceder a la publicación en redes sociales o cancelar.
Poscondiciones	1. El sistema posee la información de lo que el administrador desea hacer para actuar en consecuencia.

Tabla 2.5 Caso de uso: Solicitud de consentimiento para publicación en redes sociales

Caso de uso – Publicación de los niveles en redes sociales.**Ilustración 2.14 Caso de uso: Publicación de los niveles en redes sociales**

Publicación de los niveles en redes sociales	
Actor primario	Sistema
Actores participantes	Sistema, redes sociales
Precondiciones	1. El administrador ha autorizado la publicación de la plantilla en las redes sociales.
Flujo	1. Se publica la plantilla en la página de Facebook "Polen Jaén" 2. Se publica la plantilla en el perfil de Twitter "Polen Jaén". 3. Se publica la plantilla en el perfil de Instagram "Polen Jaén". 4. Se publica la plantilla en el canal de Telegram "Polen Jaén".
Poscondiciones	1. La plantilla con los datos del día actual queda publicada en las cuatro vías de difusión.

Tabla 2.6 Caso de uso: Publicación de los niveles en redes sociales

Caso de uso – Informe de resolución.

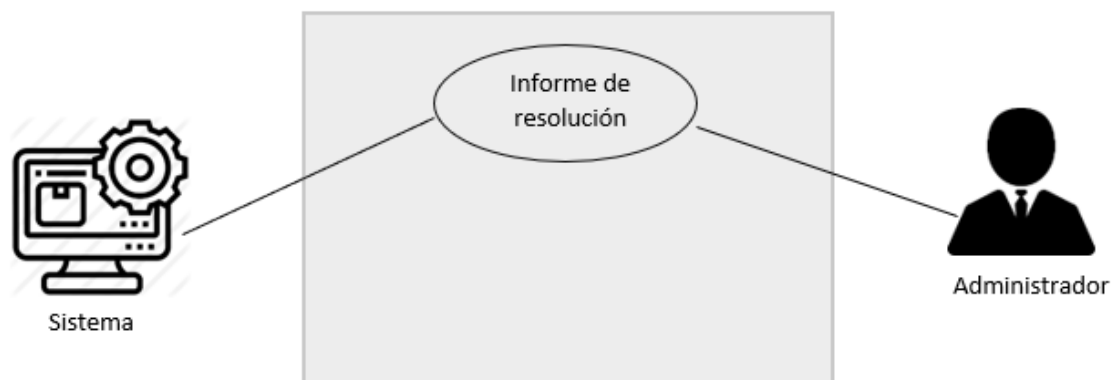


Ilustración 2.15 Caso de uso: Informe de resolución

Informe de resolución	
Actor primario	Sistema
Actores participantes	Sistema, administrador
Precondiciones	1. El administrador ha aceptado o rechazado la publicación de los niveles de polen en redes sociales.
Flujo	1. El sistema envía el informe de finalización de la ejecución del programa según las instrucciones recibidas por el administrador.
Poscondiciones	1. El sistema finaliza su ejecución hasta que vuelva a ejecutarse el siguiente día programado.

Tabla 2.7 Caso de uso: Informe de resolución

2.4. Estudios previos del origen de los datos

Para comenzar este capítulo se va realizar un estudio detallado del origen de los datos de los que se va a surtir nuestro sistema. Para poder utilizar estos datos en tiempo y forma correcta, se debe analizar la fuente en profundidad y así poder configurar y optimizar el entorno del proyecto en consecuencia.

En el caso que acontece y como hemos comentado anteriormente, los datos capturados y manipulados por el sistema serán los niveles de polen publicados en la web del Hospital Universitario de Jaén. Así pues, se va a proceder a analizar la forma en la que realizan esa medición por los especialistas y los periodos en los que suelen ser publicados en su web para así comprender mejor el origen de los datos.

2.4.1. Estudio del procedimiento de la medición de los niveles por parte del Hospital de Jaén

Como hemos podido saber estos últimos años de primera mano por los encargados de dicha tarea en el Hospital Universitario de Jaén a través de artículos periodísticos o entrevistas en telediarios y medios de comunicación digitales, el procedimiento para que los expertos consigan conocer los niveles de polen es el siguiente:

1. Captador.



Ilustración 2.16 Captador de polen tipo Hirst

Se dispone de un captador de pólenes en la novena planta del Hospital Materno Infantil como el que se muestra en la anterior, que mediante el método

'Hirst' capta partículas volumétricas por succión según el principio de impacto. Este equipamiento es el empleado por todas las estaciones de muestreo adscritas a la Red Española de Aerobiología (REA), así como por los grupos de trabajo de los diferentes países que forman parte de la European Aeroallergen Network (EAN).

Este captador tipo Hirst consta de tres unidades, la unidad de impacto, una veleta y bomba de vacío.

La unidad de impacto tiene un orificio de entrada y un soporte circular (tambor) al que se adhieren las partículas. El tambor está conectado a un reloj y gira a razón de 2 mm/hora (48 mm/día) de forma que se pueden obtener datos tanto diarios como semanales. Sobre el tambor se fija una cinta impregnada de una sustancia adhesiva para que las partículas succionadas desde el exterior se adhieran.

La veleta adosada al exterior protege la unidad de impacto, y su función es mantener el orificio de entrada en la dirección de los vientos dominantes, con lo que la eficacia de captación de partículas aerotransportadas es mayor.

La bomba de vacío permite la succión de un volumen de aire determinado (10 litros/minuto).

2. Intervención humana: Laboratorio.



Ilustración 2.17 Microscopio de laboratorio

Como temíamos, este procedimiento no es automático, sino que requiere de la intervención de un especialista para recoger la cinta del captador a la que se adhieren los distintos pólenes y tratarlos en un laboratorio por un técnico. De ahí a que en la mayoría de ciudades tarden tanto tiempo en publicar los datos.

Por tanto, diaria o semanalmente se recoge la cinta sobre la que se han adherido los pólenes. Se coloca sobre el portador y se tiñe la muestra con fucsina para su visión al microscopio. Posteriormente se realiza la lectura en el microscopio, y mediante una fórmula matemática se determina el número medio de granos de polen por metro cúbico de aire cada 24 horas.

3. Publicación.

Por último, se procede a la publicación de los resultados del laboratorio en la web del Complejo Hospitalario de Jaén. Desde este año 2021, han cambiado la web y la publicación se realiza con un vídeo de 25 segundos. En el periodo de 2011-2020 la publicación se realizaba en su web anterior en formato texto. Ambas se pueden observar en las ilustraciones 2.5 y 2.6.

Web anterior:

<http://www.juntadeandalucia.es/servicioandaluzdesalud/chjtemas.php?id=1303>

Web actual:

<https://www.hospitaljaen.es/mediciones-de-los-niveles-de-polen-en-la-primavera-de-2021/>

2.4.2. Estudio de la periodicidad de la publicación de los datos por parte del Hospital de Jaén

Tras varios años observando y estudiando durante la primavera la publicación de los niveles de polen por el Hospital Universitario de Jaén, se han podido concluir los patrones en cuanto a los periodos de tiempo en los que se suelen publicar los datos.

Los niveles son publicados diariamente entre el 15 de abril y el 15 de junio exceptuando los sábados, domingos y festivos. En cuanto al horario, desde el hospital nos aseguran que las mediciones se realizan hasta las 8 de la mañana, que es la hora a la que se extrae la muestra del captador de polen y se lleva a su correspondiente estudio en el laboratorio. Bajo mi experiencia, puedo concluir que los datos son publicados como muy pronto a las 10:30 y como muy tarde a las 13:30.

Como podemos comprobar la publicación de los datos no tienen un horario fijo y varía cada día, pero siempre se publican en un intervalo de horas concreto. Todo esto se debe de tener en cuenta para configurar nuestro sistema correctamente y estudiar en qué momentos es conveniente que se esté comprobando si hay cambios, ya que, evidentemente es innecesario estar comprobándolo fuera de ese horario (como por ejemplo los fines de semana o durante la noche).

Por tanto, para una optimización de nuestro sistema, sería conveniente limitar la comprobación de nuevos cambios en la web del Hospital de Jaén (es decir, el momento en el que se publican los nuevos datos del día) solo si es lunes, martes, miércoles, jueves o viernes. Comprobar si ha habido cambios cada dos minutos a partir de las 10:30, y en el momento en el que se detectan cambios se detiene la búsqueda hasta el próximo día conveniente.

Para salvar el hecho de que algún día de los configurados pueda coincidir con un día particular, pero a la vez un día festivo en el que no se publican datos, se ha configurado el sistema para que se detenga si pasadas 8 horas aún sigue buscando cambios. Con esto evitamos, por ejemplo, que en el peor de los casos, siendo un viernes festivo, el sistema esté monitorizando cambios innecesariamente cada dos minutos hasta el siguiente lunes.

Esto ha sucedido este año 2021 el viernes día 11 de junio, festividad de la Virgen de la Capilla en Jaén y día no laboral en la ciudad. Como se ha explicado, es conveniente que cuando en el día 11 transcurran varias horas sin que se produzcan cambios en la web del Hospital de Jaén, se detenga la monitorización automáticamente para no mantener el proceso innecesariamente ese mismo día ni los sucesivos 12 y 13.

A calendar for the month of June 2021. The title 'JUNIO' is in a red box at the top. The days of the week are abbreviated as L, M, X, J, V, S, D. The dates are arranged in a grid. The 11th of June is highlighted with a blue background, indicating it is a holiday. The 12th, 13th, 19th, 20th, 26th, and 27th are also highlighted in red.

JUNIO						
L	M	X	J	V	S	D
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

Ilustración 2.18 Mes de junio de 2021 señalando el día 11 festivo

Para terminar, se muestra una tabla con los horarios exactos de publicación del Hospital de Jaén durante esta primavera del 2021, donde se puede apreciar la consonancia con todo lo comentado en este apartado.

15-abr	12:10	01-may	x	17-may	10:36	01-jun	12:28
16-abr	11:00	02-may	x	18-may	10:53	02-jun	11:00
17-abr	x	03-may	10:52	19-may	10:31	03-jun	11:15
18-abr	x	04-may	12:25	20-may	13:06	04-jun	10:49
19-abr	11:36	05-may	10:48	21-may	11:46	05-jun	x
20-abr	11:15	06-may	11:44	22-may	x	06-jun	x
21-abr	11:17	07-may	11:12	23-may	x	07-jun	11:50
22-abr	11:18	08-may	x	24-may	10:50	08-jun	10:32
23-abr	10:43	09-may	x	25-may	11:39	09-jun	11:54
24-abr	x	10-may	11:11	26-may	12:56	10-jun	11:15
25-abr	x	11-may	11:00	27-may	10:54	11-jun	x
26-abr	10:59	12-may	11:12	28-may	11:30	12-jun	x
27-abr	10:38	13-may	11:32	29-may	x	13-jun	x
28-abr	10:41	14-may	10:58	30-may	x	14-jun	11:54
29-abr	11:15	15-may	x	31-may	11:27	15-jun	11:30
30-abr	11:36	16-may	x				

Tabla 2.8 Horarios de publicación de los niveles de polen por parte del Hospital de Jaén en 2021

2.5. Estimación de costes

Tras realizar un análisis de las especificaciones del sistema y una estimación temporal para su puesta en marcha, se está en disposición de proceder a desglosar detalladamente un presupuesto aproximado del análisis, desarrollo y mantenimiento del sistema.

Se tendrán en cuenta el equipamiento hardware, el software necesario, los gastos en equipo humano para analizar y desarrollar, además de algunos costes indirectos provenientes de recursos como la electricidad y el internet.

Por último, también se realizará una estimación del coste por año del sistema de computación en la nube elegido: AWS.

2.5.1. Hardware

Para el desarrollo del proyecto se requerirá de dos ordenadores no necesariamente muy potentes, un sobremesa ubicado en casa y un portátil, este último obligatoriamente necesario pues voy a compaginar TFG con mis obligaciones laborales, por lo que serán numerosas las veces que me veré forzado a trabajar en ubicaciones distintas con dicho ordenador portátil.

El equipo de sobremesa fue adquirido en 2017 por el precio de 350€ y periféricos reutilizados de un anterior ordenador. Se trata de un AMD A10-7700K Radeon R7, con 8GB de RAM, un SSD de 120GB combinado con un HDD de 1TB como características principales. Para una estimación realista hay que tener en cuenta la amortización de cada componente hardware, que para un equipo de estas características puede tener una duración media de unos 7 años, es decir, 84 meses. Por tanto, el precio de este equipo cada mes sería de $350\text{€}/84\text{meses}=4'17\text{€/mes}$, normalizado a los 7 meses de la duración del proyecto, se tendría un valor asociado de 29'19€ por este ordenador de sobremesa.

En cuanto al ordenador portátil se trata de un Sony VAIO SVE1512U1ESI cuyos componentes principales son un procesador Intel Core i5-3210, memoria RAM de 6GB, y un disco duro SSD de 240GB. Este equipo fue adquirido en el año 2013

por 650€, pero teniendo en cuenta que la vida útil para este tipo de equipos suele ser de unos 4 años y ya hace tiempo que pasó ese plazo, se puede considerar ya totalmente amortizado y podría contar como gasto nulo para esta estimación.

También será necesario un smartphone para la comunicación Sistema-Administrador. En mi caso se trata del modelo Xiaomi Redmi Note 8 Pro de 6GB de RAM y 64GB de almacenamiento, adquirido en mayo de 2020 por 195€. Considerando una vida útil para este tipo de smartphones de gama media de unos 2 años y la duración estimada para el proyecto de 7 meses, se podría atribuir al proyecto un coste de 52'81€ por este componente.

Componente	Coste
Ordenador sobremesa	29'19€
Ordenador portátil	0'00€
Smartphone	56'88€
Total	86'07€

Tabla 2.9 Costes de Hardware

2.5.2. Software

En cuanto al coste software debemos tener en cuenta el paquete de Microsoft Office 2016 para el desarrollo de la memoria y algunos diseños básicos (con licencia gratuita para estudiantes de la Universidad de Jaén) y la licencia de Adobe Photoshop CC 2019 para las ediciones realizadas sobre la imagen capturada de la web del Hospital de Jaén cuya licencia ronda los 24€/mes y que ascendería el coste por los 7 meses del proyecto a 168€ por este concepto.

También se necesitarán sin coste añadido a este presupuesto por su gratuidad, las licencias de Microsoft Windows que ya vienen incorporadas en ambos ordenadores, el entorno de desarrollo en python Ninja IDE y, además, el paquete Python 3.8.2 y todas sus librerías necesarias.

Software	Coste
Microsoft Windows 10	0€
Microsoft Office 2016	0€
Adobe Photoshop CC 2019	168€
Ninja IDE	0€
Python 3.8.2 + librerías	0€
Total	168€

Tabla 2.10 Costes de Software

2.5.3. Amazon Web Services

Como hemos comentado anteriormente, el servicio de computación en la nube elegido para este proyecto es Amazon Web Services. Utilizando la calculadora de precios que pone a disposición esta plataforma podemos hacer un cálculo de lo que nos cuesta una reserva de una instancia EC2 con recursos adaptados a nuestro proyecto.

Ilustración 2.19 Calculadora de precios de AWS

Para estimar los costes, elegimos en la calculadora un modelo de precios bajo demanda, una instancia t2.nano con un uso del 100% al mes y un almacenamiento SSD de uso general (gp2) de 30GB. Esto nos calcula una estimación de 9'98 USD al mes, que convertidos a euros equivalen a 8'36€.

Amazon EC2 estimación

Instancias bajo demanda de Amazon EC2 (monthly)	6,50 USD
Precios de Amazon Elastic Block Storage (EBS) (monthly)	3,48 USD
Costo total mensual:	9,98 USD

Ilustración 2.20 Estimación de la calculadora de precios AWS

Sin embargo, por suerte, Amazon Web Services ofrece los 12 primeros meses gratuitos para este tipo de reservas, por lo que estos costes se pueden obviar en lo que respecta al desarrollo de este proyecto.



COMPUTACIÓN

Capa gratuita **12 MESES GRATIS**

Amazon EC2

750 horas

al mes

Capacidad de cómputo de tamaño variable en la nube.

750 horas por mes de uso de instancias t2.micro con Linux, RHEL o SLES

750 horas por mes de uso de instancias t2.micro con Windows

Ilustración 2.21 Servicio Amazon EC2 gratuito los 12 primeros meses

Aun así, hay que considerar que sí tendrán que tenerse en cuenta para el mantenimiento los costes obtenidos en la calculadora de AWS, ya que cada año durante los dos meses de primavera que se publican los niveles se estiman unos 16'72€ de gastos.

Servicio	Coste
AWS (12 primeros meses)	0€
Total	0€

Tabla 2.11 Costes AWS

2.5.4. Recursos humanos

Aunque todo el desarrollo de este trabajo ha sido desarrollado por mi persona, el proyecto tiene que tener asociados unos costes de personal y se va a tratar de asociarlos al presupuesto de la manera más realista.

Los roles integrados en este proyecto y su división más acertada serían los de analista, encargado de todo lo que respecta a la fase de “Análisis y Diseño”, y el de desarrollador, al que se le asocian las fases de “Estudio de las APIs de las redes sociales y configuración”, “Configuración de un sistema de computación en la nube”, “Desarrollo del código” y “Pruebas”.

Para el caso del analista se podría establecer un coste aproximado de 26€ brutos por hora. La fase a la que está asociado tiene una duración asignada de 5 semanas con una media de 12 horas de trabajo en cada una de ellas. Esto hace un total de 60 horas, por lo que el coste asociado a este puesto es de $60 \times 26 = 1560$ € brutos.

Por otro lado, las fases para el puesto de desarrollador abarcan un total de 18 semanas, también con una media de 12 horas de trabajo en cada una de ellas, es decir, 216 horas totales. Atribuyéndole un coste bruto de 17€ por hora, el cómputo general para este rol sería de $216 \times 17 = 3672$ € brutos.

Puesto	Coste
Analista	1560€
Desarrollador	3672€
Total	5232€

Tabla 2.12 Costes de recursos humanos

2.5.5. Costes indirectos

Aquí se van a estudiar los dos costes indirectos más relevantes que son los de electricidad e internet.

Para el consumo eléctrico se tiene en cuenta que el gasto en mi residencia habitual es de media de unos 70€ al mes y se va a asociar a este proyecto un 10% de ese gasto, que serían 7€ al mes. Teniendo en cuenta la duración del proyecto de 7 meses, quedarían atribuidos en total 49€ al presupuesto en el concepto de electricidad.

Para el cálculo del gasto de internet se tiene en cuenta el precio total de la tarifa contratada en mi residencia habitual, que sería de 29€ al mes con la compañía Jazztel. A razón de los 7 meses de duración del proyecto, quedan asociados 203€ de este concepto.

Servicio	Coste
Electricidad	49€
Internet	203€
Total	252€

Tabla 2.13 Costes indirectos

Por tanto, tras analizar cada punto por separado, ya se está en disposición de elaborar el presupuesto total previsto para la elaboración del proyecto y que se desglosa en la siguiente tabla.

Concepto	Coste
Hardware	86'07€
Software	168'00€
Amazon Web Services	0'00€
Recursos Humanos	5232'00€
Costes indirectos	252'00€
Total	5738'07€

Tabla 2.14 Costes totales del proyecto

3. PLANIFICACIÓN TEMPORAL

La planificación temporal para realizar por completo este proyecto es muy genérica y poco precisa, pues al principio esta idea surgió y comenzó a desarrollarse sin saber que acabaría siendo un proyecto formal como defensa de un TFG. Por otro lado, hay partes del este proyecto que pueden realizarse exclusivamente en las fechas en las que el Hospital de Jaén publica los niveles en su web. Además, su progreso estará constantemente ligado a mis obligaciones laborales a jornada completa, por lo que habrá semanas que pueda dedicarle más tiempo y otras menos.

Para ser más realistas, se va a exponer una planificación con fechas no específicas, así pues, se detallarán las tareas y se asociarán a intervalos de tiempo relativos. También se establecerá una media de 12 horas dedicadas al proyecto por cada semana adjudicada.

1. **Análisis y Diseño** (5 semanas). Durante esta primera fase se realizarán los primeros análisis de la situación y se diseñarán las primeras pinceladas de lo que será nuestro sistema. También se definirá una planificación aproximada tras definir los objetivos y requisitos y se especificará todo lo posible antes de su configuración y desarrollo.
2. **Estudio de las APIs de las redes sociales y configuración** (3 semanas). Como se comentó en las especificaciones, serán 4 las vías de difusión utilizadas: Facebook, Telegram, Instagram y Twitter. Será necesario el correcto estudio y configuración a través de sus respectivas APIs para la posterior comunicación con dichas redes sociales.
3. **Configuración de un sistema de computación en la nube** (2 semanas). En esta fase del proyecto se estudiarán las diferentes posibilidades de alojamiento en un sistema de computación en la nube, para posteriormente, crear ese espacio y configurarlo convenientemente.
4. **Desarrollo del código** (10 semanas). Esta fase constará de todo el proceso de implementación del código fuente del proyecto.
5. **Pruebas, corrección de errores y mejoras** (3 semanas). A pesar de que las pruebas y testeos serán realizados constantemente por la

naturaleza del proyecto, se identifica como una fase más por su importancia y relevancia en el resultado final.

6. **Redacción de la memoria** (5 semanas). Aunque durante las fases anteriores se estará constantemente tomando anotaciones, apuntando referencias, realizando bocetos y demás detalles útiles para la posterior redacción de la memoria, al final llega el momento de darle forma y concretar el documento.

Como se ha mencionado, previamente a todas estas tareas se estuvo madurando poco a poco esta idea de proyecto durante algunas primaveras en años anteriores. Por tanto, se podría añadir una tarea previa adicional no planificada cuya duración puede constar de unos 8 meses (suma total de los dos meses de primavera en los que se publican niveles de los años 2016, 2017, 2018 y 2019) y que consistió en el surgimiento de la idea, su evolución y su resultado final como base de este TFG.

Como vemos, la suma total de la duración del proyecto asciende a 28 semanas, es decir, 7 meses. En las conclusiones finales se explicará si se han cumplido las previsiones aquí redactadas.

3.1. Diagrama de Gantt

Según su definición, el diagrama de Gantt es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

Se procede ahora a determinar el diagrama de Gantt consecuente con las fases y tareas arriba definidas.

		Semana														
Tarea	Duración	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	5 semanas															
2	3 semanas															
3	2 semanas															
4	10 semanas															
5	3 semanas															
6	5 semanas															

Tabla 3.1 Diagrama de Gantt (1)

		Semana													
Tarea	Duración	16	17	18	19	20	21	22	23	24	25	26	27	28	
1	5 semanas														
2	3 semanas														
3	2 semanas														
4	10 semanas														
5	3 semanas														
6	5 semanas														

Tabla 3.2 Diagrama de Gantt (2)

4. DISEÑO INICIAL

En esta sección se realizará un diseño preliminar del sistema que se desea elaborar, explicando algunas estrategias claves a seguir para alcanzar los objetivos de este proyecto, sin olvidar igualmente, las elecciones más importantes que hay que tomar previas al desarrollo.

El flujo de trabajo será otro de los temas que se abordarán en este capítulo, donde también se planteará la arquitectura general del sistema. Todo en su conjunto nos dará una idea muy aproximada del sistema final que se desea conseguir en este TFG.

4.1. Definición de estrategias a seguir

En esta sección se van a definir una serie de estrategias que necesariamente se deben plantear para lograr los objetivos de este proyecto y cumplir las especificaciones planteadas anteriormente.

Entre ellas destacan el estudio en profundidad de la web del Hospital de Jaén para plantear cómo se van a realizar las técnicas de web scraping en ella con el fin de detectar el momento en el que se han actualizado los niveles de polen.

Por otro lado, se necesita plantear cuál será la manera más óptima de capturar dichos datos de la web del Hospital de Jaén.

Para terminar con esta sección, se definirá la estrategia de la edición de los datos capturados con la finalidad de personalizarlos para más tarde publicarlos en las redes sociales.

Recalcar que es en esta parte del proyecto donde se produce uno de los cambios más radicales durante su desarrollo, pues después de un periodo de 10 años (2011-2020 ambos inclusive) en las que el hospital de Jaén ha ido publicando los niveles en su web, en este año 2021 han optado por crear una nueva más innovadora y moderna. Esto hace que todos los estudios y estrategias planteadas durante el año 2020 y anteriores hayan quedado obsoletas y, por tanto, se hayan tenido que volver a realizar desde el principio. Todo esto quedará explicado más concretamente en las conclusiones de esta memoria.

4.1.1. Web scraping para monitorizar cambios en la web del Hospital de Jaén

Se procede ahora a explicar la estrategia a seguir para monitorizar los niveles de polen en la web del Hospital de Jaén y detectar cuándo se actualizan.

Lo primero es realizar un primer vistazo general a la web objetivo (ilustración 2.6). Como vemos, la forma que tiene de publicar los niveles es con un vídeo de una duración fija de 25 segundos. Por tanto, como estrategia de web scraping [8] se

procederá a obtener el código fuente de dicha web y buscar en él el correspondiente vídeo con los niveles de polen.

```
ze:cover;background-size:cover;padding: 0px 0px 0px 0px;"></div><style type="text/css">  
op : 0px;margin-bottom : 30px;}.fusion-builder-column-1 >.fusion-column-wrapper {padding-right : 9.6%;padding-bottom : 0px !important;padding-left : 0px !important;-body .fusion-builder-column-1{width:100% !important;order : 0;}.fusion-builder-column-1{width:92%;}}@media only screen and (max-width:640px) {.fusion-body .fusion-builder-column-1.fusion-column-wrapper {margin-right : 1.92%;margin-left : 1.92%;}}</style></div><div class="fusion-builder-column-2 fusion_builder_column_3_5_3_5 fusion-flex-column"><div class="fusion-column-wrapper ut-column" style="background-position:left top;background-repeat:no-repeat;-webkit-background-size:cover;background-size:cover;padding: 0px 0px 0px 0px;"><div class="fusion-video deo-wrapper"><video playsinline="true" width="100%" style="object-fit: cover;" autoplay=""><source src="https://www.hospitaljaen.es/wp-content/uploads/2021/04/Polen-15.mp4"></source></div></div></div></div><style type="text/css">}.fusion-body .fusion-builder-column-builder-column-2 >.fusion-column-wrapper {padding-top : 0px !important;padding-right : 0px !important;padding-left : 0px !important;margin-left : 3.2%;}@media only screen and (max-width:640px) {.fusion-body .fusion-builder-column-2 >.fusion-column-wrapper {margin-left : 1.92%;}}@media only screen and (max-width:640px) {.fusion-body .fusion-builder-column-2{width:100% !important;order : 0;}.fusion-builder-column-2 >.fusion-column-wrapper {margin-left : 1.92%;}}</style></div><div class="fusion-layout-column fusion_builder_flex-column"><div class="fusion-column-wrapper fusion-flex-justify-content-flex-start background-repeat:no-repeat;-webkit-background-size:cover;-moz-background-size:cover;-ms-background-size:cover;min-height:100px;min-width:100px;"></div><style type="text/css">}.fusion-body .fusion-builder-column-builder-column-3 >.fusion-column-wrapper {padding-top : 0px !important;padding-right : 0px !important;padding-left : 0px !important;margin-left : 9.6%;}@media only screen and (max-width:640px) {.fusion-body .fusion-builder-column-3 >.fusion-column-wrapper {margin-left : 1.92%;}}@media only screen and (max-width:640px) {.fusion-body .fusion-builder-column-3{width:100% !important;order : 0;}.fusion-builder-column-3 >.fusion-column-wrapper {margin-left : 1.92%;}}</style></div>
```

Ilustración 4.1 Vídeo con los niveles de polen en el código fuente de la web del Hospital de Jaén

Analizando el código fuente detectamos el enlace al vídeo objetivo en formato .mp4. El formato es el siguiente:

```
href="https://www.hospitaljaen.es/wp-content/uploads/[año]/
[mes]/Polen-[día].mp4"
```

donde cada elemento entre corchetes equivale al valor correspondiente a la fecha de los niveles que se muestran en el vídeo. Por ejemplo, para el día 5 de junio de 2021 el enlace sería:

href="https://www.hospitaljaen.es/wp-content/uploads/2021/06/Polen-5.mp4"

Por tanto, la estrategia a seguir para saber si se han producido cambios en el día actual será obtener el código fuente completo de la web, localizar y extraer el enlace del vídeo objetivo y compararlo con el enlace del vídeo objetivo obtenido del día anterior. Si se ha producido algún cambio, extraer los datos según la estrategia que se explicará en el siguiente apartado, si no se ha producido, volver a realizar todo el proceso después de dejar transcurrir un intervalo de tiempo estimado.

Esta estrategia aquí planteada no estará constantemente realizándose, sino que se optimizará el sistema para que se realice dentro de los tiempos estudiados ya en la sección 2.4.2.

4.1.2. Captura de los datos nuevos detectados

Una vez detectado que se han actualizado los niveles de polen en la web del Hospital de Jaén hay que elegir cómo se van a capturar y obtener dichos datos de la manera más conveniente para posteriormente tratarlos y difundirlos.

Como se ha podido observar, son publicados cada día con un vídeo de 25 segundos de duración y siempre siguiendo el mismo patrón con la siguiente secuencia:

1. Introducción con el logo del Hospital Universitario de Jaén de unos 7 segundos de duración.
2. Niveles de polen del día correspondiente de unos 15 segundos de duración.
3. Parte final con el logo de la Junta de Andalucía de unos 3 segundos de duración.

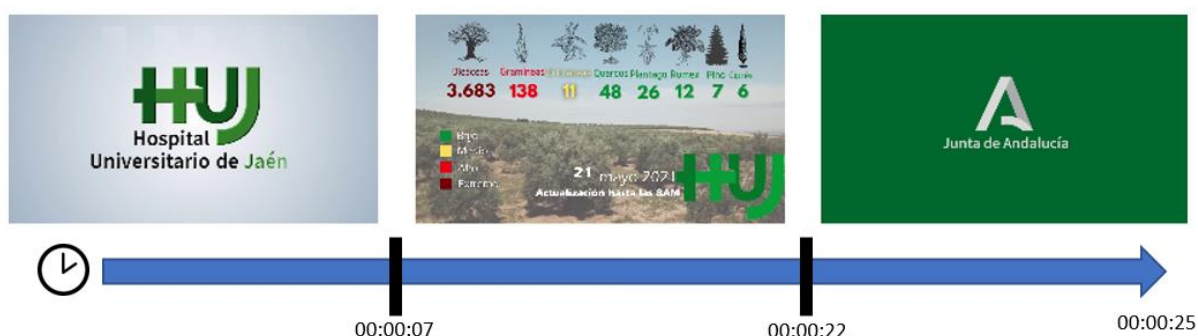


Ilustración 4.2 Secuencia del vídeo de la web del Hospital de Jaén

Guardar los datos en el sistema y difundirlos en las redes sociales conforme están publicados en la web del Hospital creemos que no es lo más conveniente para las personas interesadas, pues tener que ver un vídeo de 25 segundos cada día para estar al tanto de los niveles de polen es tedioso y poco práctico. Además, si la

intención de este proyecto es personalizar la plantilla final, hacerlo sobre un vídeo complica mucho más la tarea.

Por tanto, como estrategia más adecuada a la intención de este proyecto se ha escogido la de capturar un frame de dicho vídeo en el intervalo de tiempo en el que se muestran los niveles de polen, para mayor seguridad se escogería uno del ecuador del vídeo (sobre el segundo 00:12) para tener la certeza de guardar una imagen correcta.

Así se consigue que manipular posteriormente la imagen para personalizar la plantilla final sea mucho más sencillo, a la vez que se tiene la certeza de que difundir los niveles de polen en las redes sociales con una imagen es mucho más acertado y práctico para los seguidores que hacer a través de un vídeo.

4.1.3. Edición del frame capturado del vídeo

Una vez capturada la imagen del vídeo y guardada en el repositorio se va a explicar cuál sería la estrategia a seguir para su correcta edición y poder así obtener el resultado deseado.

Se parte del frame capturado del vídeo tal y como se muestra en la siguiente ilustración.

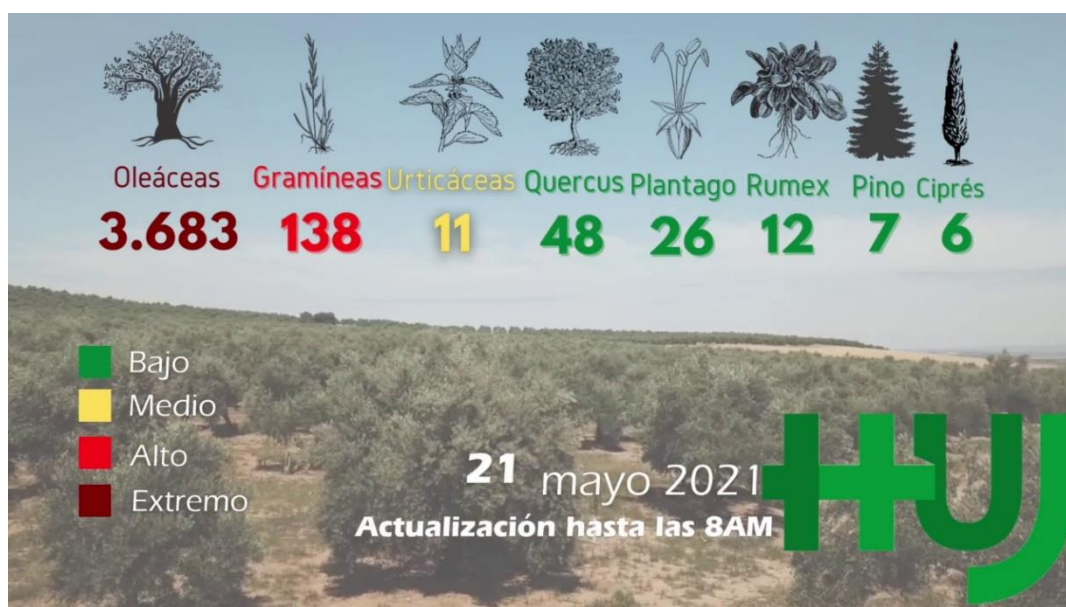


Ilustración 4.3 Ejemplo de frame capturado del vídeo de la web del Hospital de Jaén

Se ha comprobado que el frame guarda la misma estructura cada día, cosa que simplifica las ediciones. El objetivo de esta estrategia es tener como resultado una imagen equivalente a la de la siguiente ilustración, en la que se muestra la fecha de manera mucho más visible y en la que se personaliza para dar difusión a nuestras diferentes redes de difusión.



Ilustración 4.4 Imagen terminada lista para publicar en las redes sociales

Para llegar a eso, primero se necesitará realizar con algún software de edición de imágenes el borrado de las letras con la fecha y el logo de cualquiera de los frames originales. Como ejemplo se muestra la siguiente ilustración.

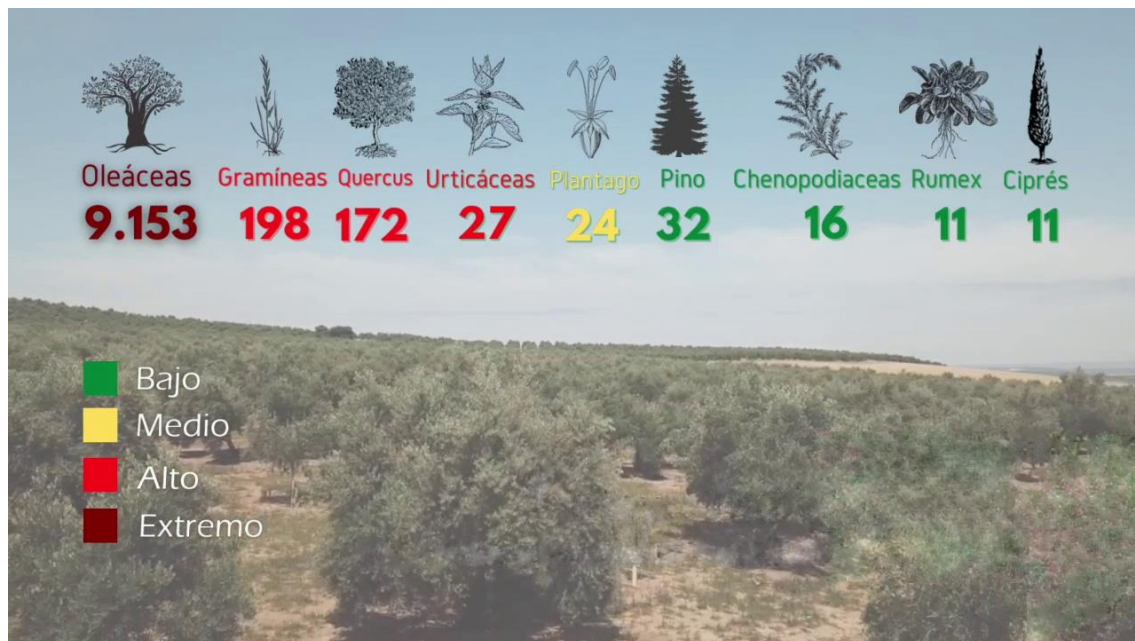


Ilustración 4.5 Frame original con fecha y logo borrados

Posteriormente se recortará la sección clave inferior derecha donde siempre se realizarán las ediciones para trabajar en los siguientes pasos con ella.



Ilustración 4.6 Recorte del frame original con fecha y logo borrados

Llegados a este punto se añadirá a este recorte la personalización que sea requerida. Para el caso normal se promocionarán nuestras propias redes sociales y que lleguen así a más gente, sin embargo, como se explicará próximamente, hay pensada una estrategia para una posible monetización del sistema donde se podrá promocionar cualquier producto o empresa que se crea conveniente.

Siempre se tendrá en cuenta que hay que reservar el espacio en la parte superior para dibujar posteriormente la fecha del día actual.



Ilustración 4.7 Recorte editado para promocionar los canales de difusión del sistema

Tras esta última modificación, se procederá mediante programación a imprimir sobre este recorte la fecha del día que corresponde. El recorte final quedaría como el siguiente ejemplo.

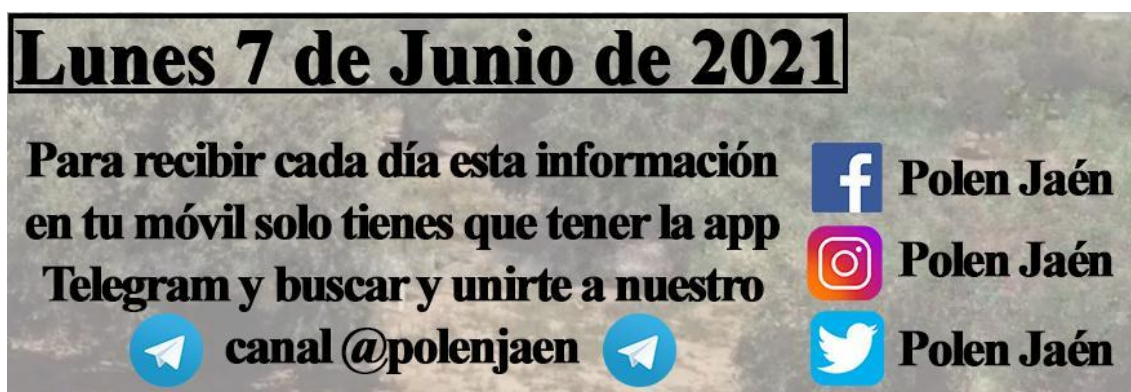


Ilustración 4.8 Recorte editado para promocionar los canales de difusión del sistema y con fecha

Para terminar, se procederá a la adecuada fusión del frame original de la ilustración 4.3 con el recorte terminado de la ilustración 4.8 y dará como resultado la imagen final para difundir por redes sociales como la mostrada en la ilustración 4.4. En la siguiente imagen se resume todo el proceso de una forma clara.



Ilustración 4.9 Resumen del proceso de edición del frame

Todo este proceso es la razón por la que sería muy conveniente, en un paso posterior, pedir autorización al administrador antes de realizar las respectivas publicaciones en las diferentes redes sociales. Así se consigue evitar que se difunda una imagen final incorrecta por producirse alguna variación en el formato publicado por parte de la web del Hospital de Jaén o por algún error producido en el proceso explicado en esta sección.

4.2. Workflow

En este capítulo se va a desarrollar un diseño genérico del flujo de trabajo del sistema con los pasos por los que debe ir transcurriendo según los resultados obtenidos. Tras este diseño se tendrá una idea mucho más certera de la idea que se quiere llevar a cabo con este trabajo.

Se procede ahora a enumerar los pasos clave del flujo de trabajo de este sistema:

1. Inicio del sistema a la hora definida.
2. Comprobación de cambios en la web del Hospital de Jaén.
3. Captura del frame del vídeo de la web del Hospital de Jaén.
4. Ediciones correspondientes sobre el frame capturado.
5. Solicitud de confirmación al administrador del sistema.
6. Publicación en las distintas redes sociales.
7. Finalización del sistema.

Cabe recalcar que, además del paso 5 en el que ya viene intrínseco la comunicación del sistema con el administrador para la solicitud de confirmación, en los pasos 1, 2 y 7 también será necesario el aviso por parte del sistema al administrador para que este tenga conocimiento de los pasos por los que vas transcurriendo el sistema. Todo esto va en concordancia con lo explicado anteriormente en el capítulo '*2.3.1 Flujo de comunicación del sistema con el administrador*'.

Ahora se va a definir el siguiente diagrama donde se puede apreciar de manera más visual el movimiento del sistema por los pasos enumerados anteriormente.

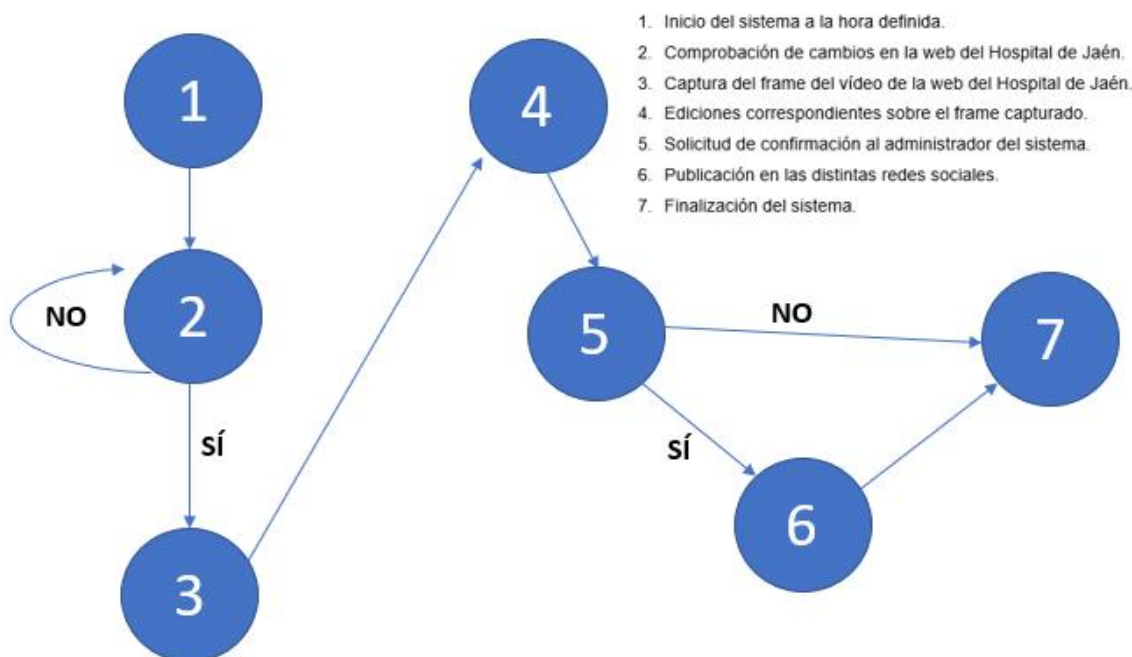


Ilustración 4.10 Diagrama del flujo de trabajo del sistema

4.3. Elecciones previas al desarrollo

En este capítulo se van a explicar algunas elecciones previas claves que hay que tomar antes de centrarse en desarrollar el sistema, acompañado siempre del correspondiente razonamiento de por qué ha sido esa la elección.

4.3.1. Servicio de computación en la nube

Por las características de nuestro proyecto, al formar un sistema que debe de estar comprobando cambios constantemente durante un periodo de tiempo concreto y realizando acciones, su ejecución debe de ser fiable e independiente de cualquier arquitectura local. Es por eso que se ha buscado una plataforma en la nube que cubra esas necesidades para así estar despreocupados de posibles caídas de internet o de la red eléctrica en casa entre otros imprevistos.

Para ello se ha elegido Amazon Web Services (AWS abreviado). Es una colección de servicios de computación en la nube pública (también llamados servicios web) que en conjunto forman una plataforma de computación en la nube, ofrecidas a través de Internet por Amazon.com. Es considerado como un pionero en este campo.



Ilustración 4.11 Logo Amazon Web Services

Entre todos los servicios que ofrecen, el que se ha elegido es el Amazon Elastic Compute Cloud (Amazon EC2) que es un servicio web que proporciona capacidad informática en la nube segura y de tamaño modificable. Está diseñado para simplificar el uso de la informática en la nube a escala web para los desarrolladores. La sencilla interfaz de servicios web de Amazon EC2 permite obtener y configurar capacidad con una fricción mínima. Proporciona un control completo sobre los recursos informáticos y puede ejecutarse en el entorno informático acreditado de Amazon.

Para ello se reservará una capa EC2 con capacidades y servicios básicos, que es gratuita durante doce meses. La instancia será desplegada incluyendo la instalación de Microsoft Windows Server 2019 desde donde se va a ejecutar el código principal. El entorno será preparado convenientemente para el correcto funcionamiento del sistema, instalando en él todos los requisitos necesarios para que todo funcione correctamente, como son el software de Python, las librerías y

dependencias correspondientes, configuración del programador de tareas de Windows, etc.

4.3.2. Lenguaje de programación

Entre la amplia gama de posibilidades en cuanto a lenguajes de programación existentes en la actualidad, se ha optado por realizar el código en Python.

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta parcialmente la orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.



Ilustración 4.12 Logo de Python

Durante la carrera únicamente adquirimos algunas nociones básicas sobre Python en las prácticas de la asignatura del segundo cuatrimestre del tercer curso “Procesamiento de Información Visual”, por lo que anteriormente al desarrollo de este proyecto mis conocimientos sobre Python eran prácticamente nulos.

Aun así, se ha escogido este lenguaje de programación ya que es un lenguaje que se adapta perfectamente para llegar a alcanzar los objetivos de este proyecto. Dispone de numerosas librerías para funcionalidades tan útiles para este trabajo como lo son web scraping, edición de imágenes y vídeos, comunicación con API de redes sociales...

Además de la versatilidad y potencia de este lenguaje de programación, el gran seguimiento y la gran comunidad que existe detrás hace que cualquier problema o duda surgida durante la programación pueda ser resuelta de una manera más sencilla que con otros lenguajes.

Así pues, la elaboración de este trabajo me ha servido no solo para afianzar las nociones básicas que ya aprendimos en la carrera sobre Python sino, sobre todo, para ampliar en gran medida esos conocimientos sobre este lenguaje de programación tan utilizado en la actualidad.

4.3.3. Software

El software principal que se usará para el desarrollo del código en python será el Ninja-IDE. Según la descripción en wikipedia se define así:

“Ninja-IDE es un entorno de desarrollo integrado multiplataforma diseñado para construir aplicaciones Python. Proporciona herramientas para simplificar el desarrollo del software Python y maneja muchos tipos de situaciones gracias a su amplia extensibilidad.”



Ilustración 4.13 Logo de Ninja IDE

Se ha optado por este entorno de desarrollo porque, aunque existiendo algunos más famosos y potentes, este fue el que utilizamos durante las prácticas de “Procesamiento de Información Visual” y por eso ya estaba familiarizado con él.

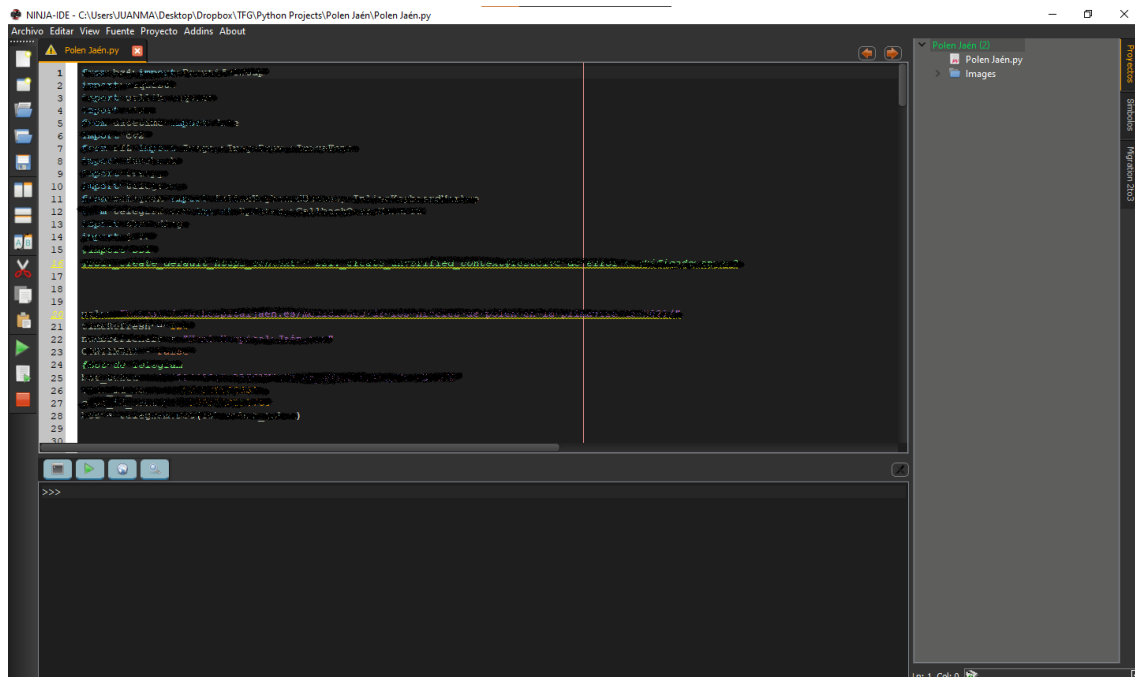


Ilustración 4.14 Captura de pantalla del entorno Ninja IDE

Aparte también se utilizará el software Adobe Photoshop CC 2019 con el fin de realizar las ediciones oportunas sobre la imagen capturada del vídeo de los niveles de polen que publica el Hospital de Jaén y poder obtener como resultado la imagen final adaptada a las necesidades de este proyecto.



Ilustración 4.15 Logo de Adobe Photoshop cc 2019

Todo esto se realizará trabajando sobre Windows 10 como Sistema Operativo base, pues es sobre el que yo trabajo tanto en mi ordenador de sobremesa como en el portátil.



Ilustración 4.16 Logo de Microsoft Windows 10

También destacar que para la elaboración de la memoria del proyecto se utilizará Microsoft Word 2016 que también coincide con el editor de texto principal que dispongo en mis dos equipos.



Ilustración 4.17 Logo de Microsoft Office 2016

4.3.4. Redes sociales para la difusión de los niveles de polen

Una vez que los datos nuevos del polen son detectados y capturados por el sistema, quizá la opción más clásica sería su difusión a través de una aplicación móvil para llegar a los usuarios. Sin embargo, como se ha explicado en el apartado “2.1.1 Aplicaciones en Play Store”, se ha indagado y comprobado que ese tipo de aplicaciones ya existen en dicha tienda de aplicaciones de Android, y también

hemos detectado su escasa ineficacia como solución a la necesidad que en este proyecto queremos cubrir.

Son aplicaciones que ofrecen datos muy desactualizados y que, en su mayoría, son muy poco específicos, ofreciéndolos en medidas muy inexactas como por ejemplo nivel bajo, medio o alto de un polen específico.

Además de esos problemas comentados, creo que, aun siendo una información útil, a la gran mayoría de usuarios el hecho de hacerle descargarse una aplicación en su móvil solo para saber los niveles de polen de su ciudad es un esfuerzo que pocos harían.

Por tanto, la idea de este sistema, es hacerle llegar esa información a las personas interesadas de la forma más sencilla y directa, eligiendo un medio en el que los usuarios no tengan que preparar nada y que todo el mundo ya utilice habitualmente en su día a día: las redes sociales.

Así pues, se ha elegido publicar los niveles de polen en la ciudad de Jaén a través de perfiles en las redes sociales que se consideran más apropiadas por sus características y seguimiento:



Ilustración 4.18 Iconos redes sociales para difusión de niveles de polen

- **Facebook:** he creado una “página” de Facebook para publicar los niveles además de otro tipo de publicaciones aparte que sean de utilidad. Por el tipo de usuarios que por tendencia usan Facebook mayoritariamente hoy en día, con esta red social copamos mayormente a personas alérgicas de edad algo avanzada o padres y madres de

niños alérgicos a los que les interesa conocer los niveles diarios de polen.

- **Instagram:** se ha creado un perfil estándar en esta red social que más tarde se ha convertido en una cuenta empresarial que nos facilita la obtención de estadísticas y demás herramientas para monitorizar el seguimiento de nuestras publicaciones. Además, he conectado este perfil a nuestra página de Facebook equivalente, requisito para poder trabajar con las API's de ambas redes sociales. Enfocado a usuarios más jóvenes que los de Facebook.
- **Twitter:** se ha creado una cuenta equivalente en esta red social, interesante por la idiosincrasia que tiene la plataforma. La mayoría de los usuarios de Twitter usan esta red social para estar al tanto de la actualidad en cuanto a noticias, acontecimientos o temas que consideran útiles donde podrían encajar en este caso los niveles de polen de su ciudad.
- **Telegram:** la aplicación de mensajería instantánea más potente en cuanto a funcionalidades. Entre ellas están los canales, una herramienta de envío de mensajes de 1 a muchos. Funcionan de la siguiente manera: el creador del canal envía un mensaje (es el único que puede hacerlo) que es recibido por todos los suscriptores al canal. Por tanto, nos aprovechamos de estos canales para difundir nuestra información de manera que las personas que se unan al canal la reciben con una notificación en su móvil.

4.4. Arquitectura del sistema

Tras finalizar todo el diseño inicial de este proyecto, se da paso a definir la arquitectura general del sistema enumerando los diferentes módulos que lo componen.

1. **Módulo AWS.** Es el módulo central de todo el sistema, encargado de gestionar todas las comunicaciones con el resto de componentes. Estará

alojado en los servidores de Amazon y en él se ejecutará el código en lenguaje Python que controla todo el sistema.

2. **Módulo Telegram.** El segundo módulo en cuanto a importancia, pues es el encargado de una doble función de comunicación: con el administrador y con los alérgicos, mediante un canal privado y otro público respectivamente. Todo esto se realizará a través de su correspondiente API.
3. **Módulo Facebook.** Se centra en la publicación en la correspondiente página de Facebook de los niveles de polen y hace llegar esa información a los seguidores. Se realiza a través de su correspondiente API.
4. **Módulo Instagram.** Como el módulo anterior pero a través del perfil correspondiente en Instagram. Se hace uso también de su correspondiente API.
5. **Módulo Twitter.** Como los dos módulos anteriores pero a través de su correspondiente perfil en Twitter. Se realiza a través de su respectiva API.

Para finalizar, se muestra, para una observación más visual, la ilustración con el esquema general de la arquitectura definida.

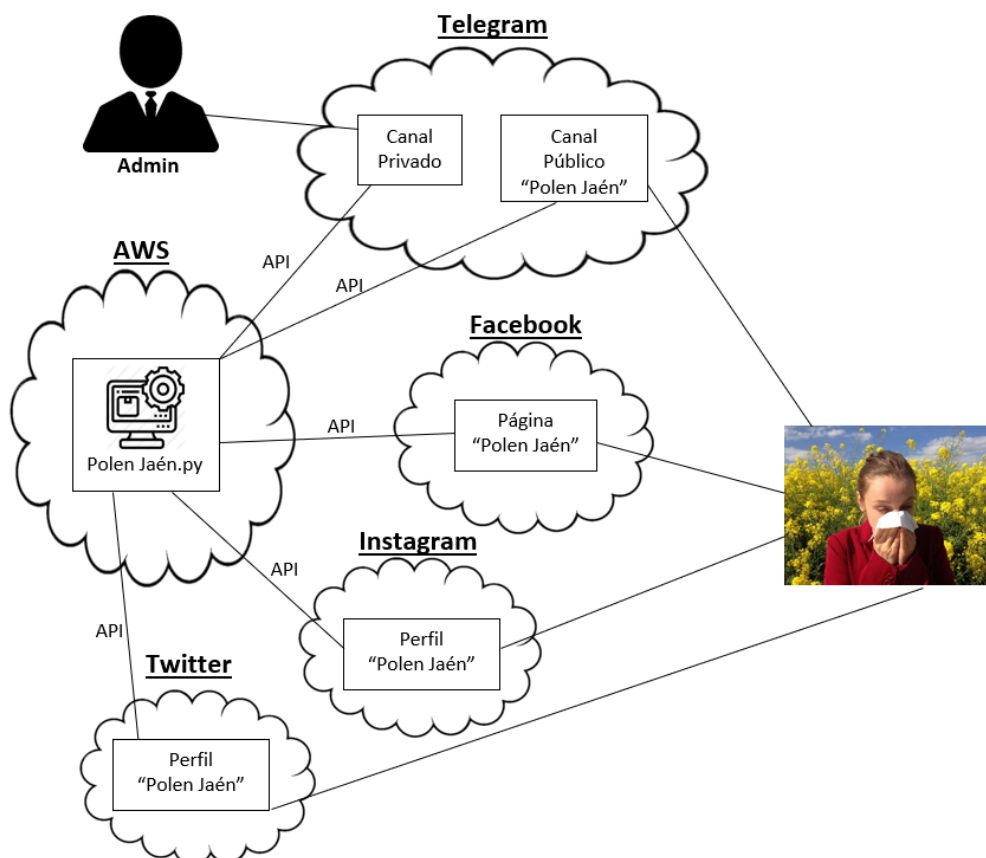


Ilustración 4.19 Esquema general de la arquitectura y módulos del sistema

5. DESARROLLO DEL SISTEMA

En este capítulo se va explicar paso por paso cómo se ha configurado y desarrollado el sistema para cumplir todos los objetivos de este proyecto.

Para empezar, se exponen las configuraciones previas de plataformas y protocolos que se han tenido que utilizar como apoyos necesarios al sistema, los cuales han tenido que ser estudiados a fondo para conocerlos al detalle y poder ser utilizados convenientemente.

Posteriormente, se realizará un desglose al detalle punto por punto del código en Python del sistema.

Por último, se detallarán las diferentes partes del repositorio Python, concretando qué es cada fichero y cada archivo contenido en él. También se hará lo

propio con los requisitos a cumplir para una correcta ejecución del sistema y una prueba real con capturas para una prueba visual del su funcionamiento.

5.1. Configuraciones previas

Una serie de configuraciones previas al desarrollo y elaboración del código en Python son necesarias para su correcto funcionamiento. Entre ellas se encuentran el estudio y configuración de Amazon Web Services como sistema de computación en la nube y el programador de tareas de Windows Server dentro de este, además de las APIs de Telegram, Facebook, Instagram y Twitter para la correcta comunicación con las diferentes plataformas.

5.1.1. Amazon Web Services

Como se ha comentado anteriormente, Amazon Web Services es el sistema de computación en la nube y EC2 la herramienta utilizada para dar soporte al sistema con independencia de cualquier equipo local.

Personalmente nunca he utilizado esta plataforma por lo que se ha necesitado un estudio en profundidad de la gran cantidad de herramientas y usos que ofrece. Para crear una instancia EC2 y configurarla correctamente se han seguido los propios tutoriales que ofrece Amazon en su web [9][10].

Los primeros pasos han consistido en inscribirme y crear una cuenta en Amazon Web Services en el portal disponible para ello (<https://portal.aws.amazon.com/billing/signup#/start>). Durante el proceso se ha requerido teclear un código de verificación en el teléfono tras recibir una llamada.

El siguiente paso ha sido la creación de un par de claves criptográficas para proteger la información de inicio de la correspondiente instancia. Se ha realizado siguiendo los pasos de las instrucciones de Amazon y guardando en mi equipo la clave privada asociado al par de claves creado. Esto permite iniciar sesión mediante

Remote Desktop Protocol (RDP) que permite la comunicación en la ejecución de una aplicación entre una terminal y un servidor.

Tras esto se ha proseguido creando un grupo de seguridad. Esto actúa como firewall para las instancias asociadas al controlar el tráfico entrante y saliente en el ámbito de la instancia

Después de estos pasos previos ya está todo preparado para lanzarse y conectarse a una instancia EC2 con Windows Server. Esta instancia es un servidor virtual en la nube de Amazon con respaldo en Amazon Elastic Block Store (EBS) que es el almacenamiento a nivel de bloque de Amazon.

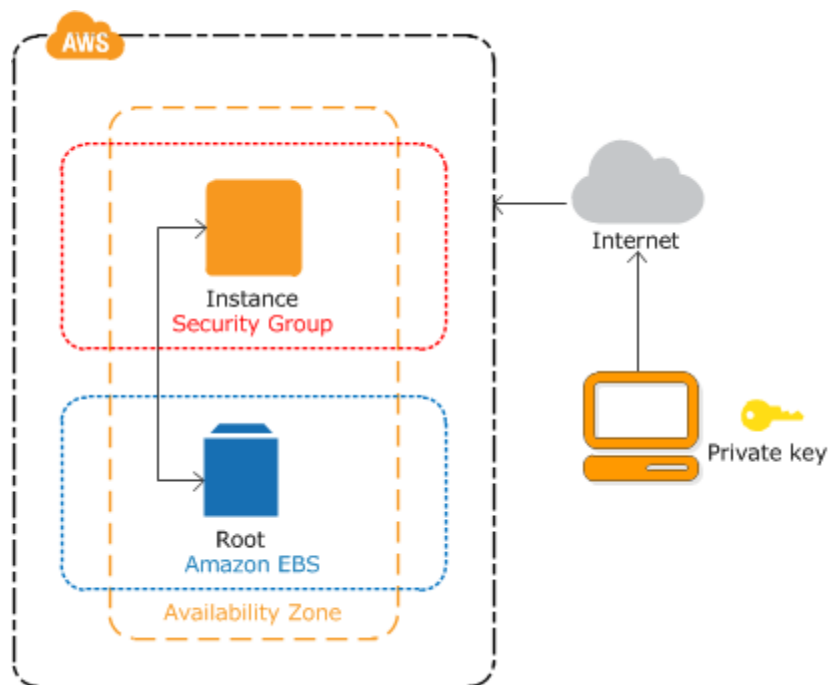


Ilustración 5.1 Arquitectura instancia EC2 + EBS en AWS

Para lanzar definitivamente una instancia se hace uso de la consola Amazon EC2 y se elige la opción correspondiente. Es necesario seleccionar una imagen de Amazon Machine (AMI) que es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. En nuestro caso procedemos a elegir la imagen con Microsoft Windows Server 2019.

Paso 1: Elegir una imagen de Amazon Machine (AMI)

[Cancelar y salir](#)

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Puede seleccionar una AMI proporcionada por AWS, nuestra comunidad de usuarios o AWS Marketplace, o puede seleccionar una de sus propias AMI.

Search results for 'windows' (1 a 19 de 19 AMI):

- Microsoft Windows Server 2019 Base** - ami-022d7dc38f1289513
Microsoft Windows 2019 Datacenter edition. [English]
Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí
Seleccionar
- Microsoft Windows Server 2019 Base with Containers** - ami-02f9a111a7be1fbd8
Microsoft Windows 2019 Datacenter edition with Containers. [English]
Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí
Seleccionar
- Microsoft Windows Server 2019 with SQL Server 2017 Standard** - ami-0264c15fdb6aa3612
Microsoft Windows 2019 Datacenter edition, Microsoft SQL Server 2017 Standard. [English]
Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí
Seleccionar
- Microsoft Windows Server 2019 with SQL Server 2017 Enterprise** - ami-0effbc2738c22f2a19
Microsoft Windows 2019 Datacenter edition, Microsoft SQL Server 2017 Enterprise. [English]
Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí
Seleccionar
- Microsoft Windows Server 2019 with SQL Server 2019 Standard** - ami-0a01ee3b0eca54e95
Microsoft Windows 2019 Datacenter edition, Microsoft SQL Server 2019 Standard. [English]
Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí
Seleccionar

Ilustración 5.2 Elección de una imagen AMI para la instancia en AWS

En el siguiente paso toca elegir el tipo de instancia. En nuestro caso cumple con las necesidades del sistema la instancia *t2.micro* que viene seleccionada por defecto y que está incluida en la capa gratuita de 12 meses de AWS.

Paso 2: Página Choose an Instance Type

Amazon EC2 proporciona una amplia selección de tipos de instancias optimizados para adaptarse a diferentes casos de uso. Las instancias son servidores virtuales que pueden ejecutar aplicaciones. Tienen distintas combinaciones de CPU, memoria, almacenamiento y capacidad de red, lo que proporciona una gran flexibilidad para elegir la combinación de recursos adecuada para las aplicaciones. [Más información](#) acerca de los tipos de instancias y cómo pueden satisfacer sus necesidades de computación.

Filtrar por: Todas las familias de instancias Generación actual Mostrar/ocultar columnas

Seleccionada actualmente: t2.micro (- ECU, 1 vCPU, 2.5 GHz, -, 1 GiB memoria, EBS solo)

	Familia	Tipo	vCPU	Memoria (GiB)	Almacenamiento de la instancia (GB)	Optimizado para EBS disponible	Desempeño de la red	Compatibilidad con IPv6
☐	t2	t2.nano	1	0.5	EBS solo	-	De bajo a moderado	Sí
☒	t2	t2.micro Apto para la capa gratuita	1	1	EBS solo	-	De bajo a moderado	Sí
☐	t2	t2.small	1	2	EBS solo	-	De bajo a moderado	Sí
☐	t2	t2.medium	2	4	EBS solo	-	De bajo a moderado	Sí
☐	t2	t2.large	2	8	EBS solo	-	De bajo a moderado	Sí

Ilustración 5.3 Elección del tipo de instancia EC2 en AWS

Por último, ya solo queda asociar a la instancia el grupo de seguridad y el par de claves creados con anterioridad y lanzarla, tarea que puede requerir de algunos minutos hasta que esté lista para conectarse.

Pasados esos minutos, ya podemos usar una conexión a nuestra instancia mediante escritorio remoto (RDP) usando la clave privada generada anteriormente. Para ello, desde el panel de configuración de AWS se puede descargar el archivo de escritorio remoto a nuestro ordenador local y verificando correctamente el certificado de seguridad se establece la conexión.

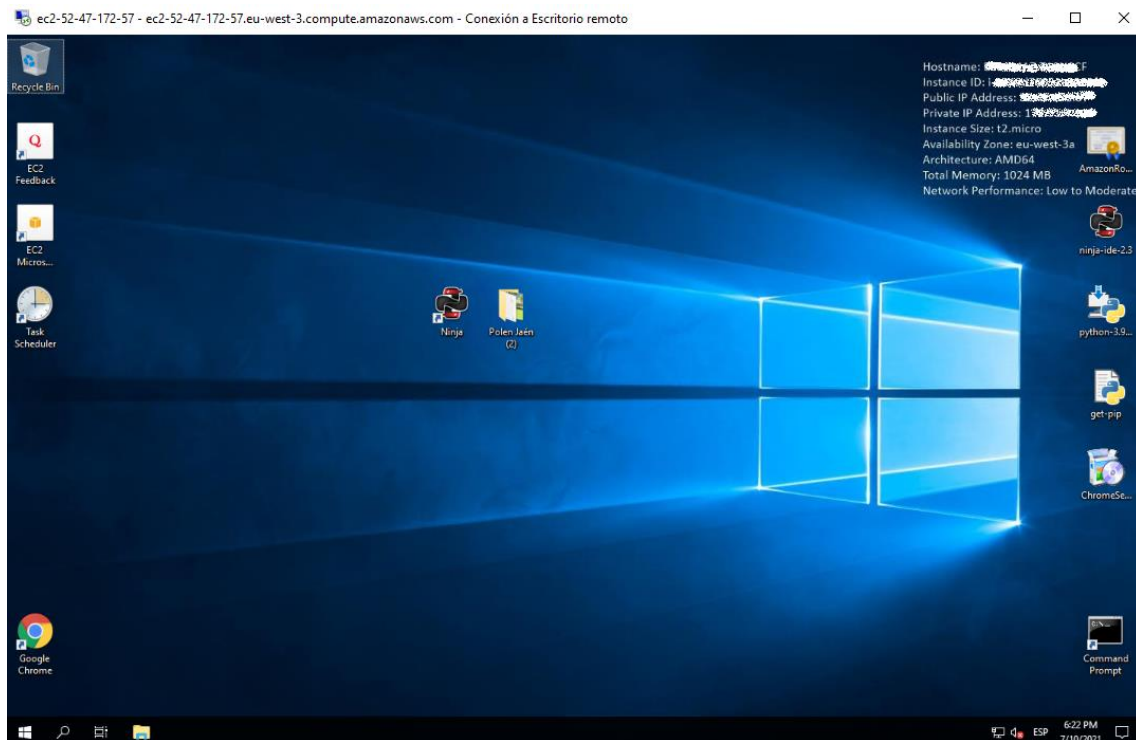


Ilustración 5.4 Escritorio remoto de la instancia ec2 creada en AWS

Así pues, ya se dispone de una instancia en la nube con Windows Server para desplegar aquí todas nuestras aplicaciones y poder ejecutarlas con independencia de cualquier equipo local.

5.1.2. Programador de tareas de Windows

Tras el estudio realizado en el apartado 2.4.2 se concluyó que existe la necesidad de acotar el tiempo en el que el sistema está ejecutándose a los periodos

útiles para así maximizar la optimización y evitar estar monitorizando en periodos que no son necesarios. Para ello se va a hacer uso del propio programador de tareas del que dispone Microsoft Windows [11] y que cumple con los requisitos que se necesitan.

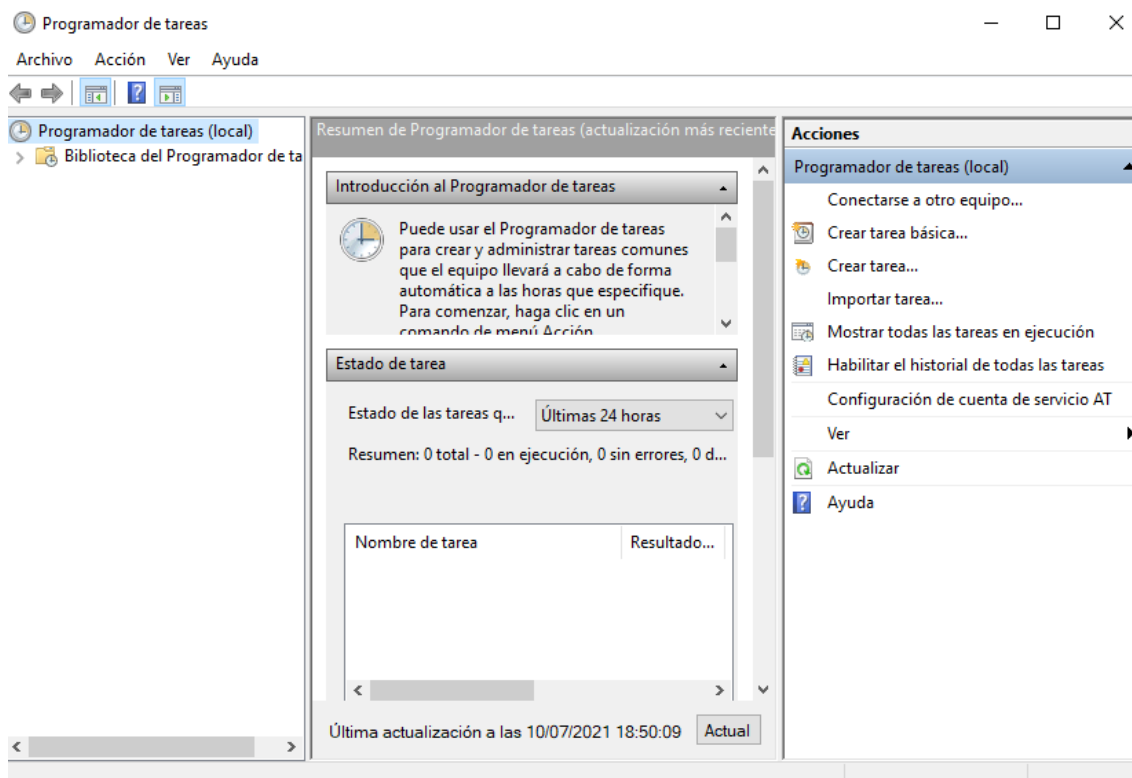


Ilustración 5.5 Programador de tareas de Windows

Lo primero que se necesita es crear una nueva tarea a la que vamos a nombrar “Polen Jaén” y en la cual vamos a especificar las condiciones que la activarán (desencadenador). Para ser coherentes con el estudio realizado en el análisis de este proyecto, se va a programar la tarea para su ejecución de lunes a viernes a las 10:30:00.

Editar desencadenador ✕

Iniciar la tarea: Según una programación ▼

Configuración

☐ Una vez
☐ Diariamente
☒ Semanalmente
☐ Mensualmente

Inicio: 25/04/2021 10:30:00 ☐ Sincronizar zonas

Repetir cada: 1 semanas en:

☐ Domingo ☒ Lunes ☒ Martes ☒ Miércoles
☒ Jueves ☒ Viernes ☐ Sábado

Configuración avanzada

☐ Retraso máx. (retraso aleatorio): 1 hora ▼

☐ Repetir cada: 1 hora ▼ durante: 1 día ▼

☐ Detener todas las tareas en ejecución al final de la duración de repetición

☐ Detener la tarea si se ejecuta durante más de: 3 días ▼

☐ Expiración: 10/07/2022 19:02:40 ☐ Sincronizar zonas horaria

☒ Habilitado

Aceptar Cancelar

Ilustración 5.6 Configuración del desencadenador del programador de tareas de Windows

Para finalizar, como es lógico, al crear una tarea se debe especificar la acción que se producirá cuando se inicie la tarea. Para este caso será la de iniciar un programa, que será el ejecutable Python que se va a proceder a desarrollar.

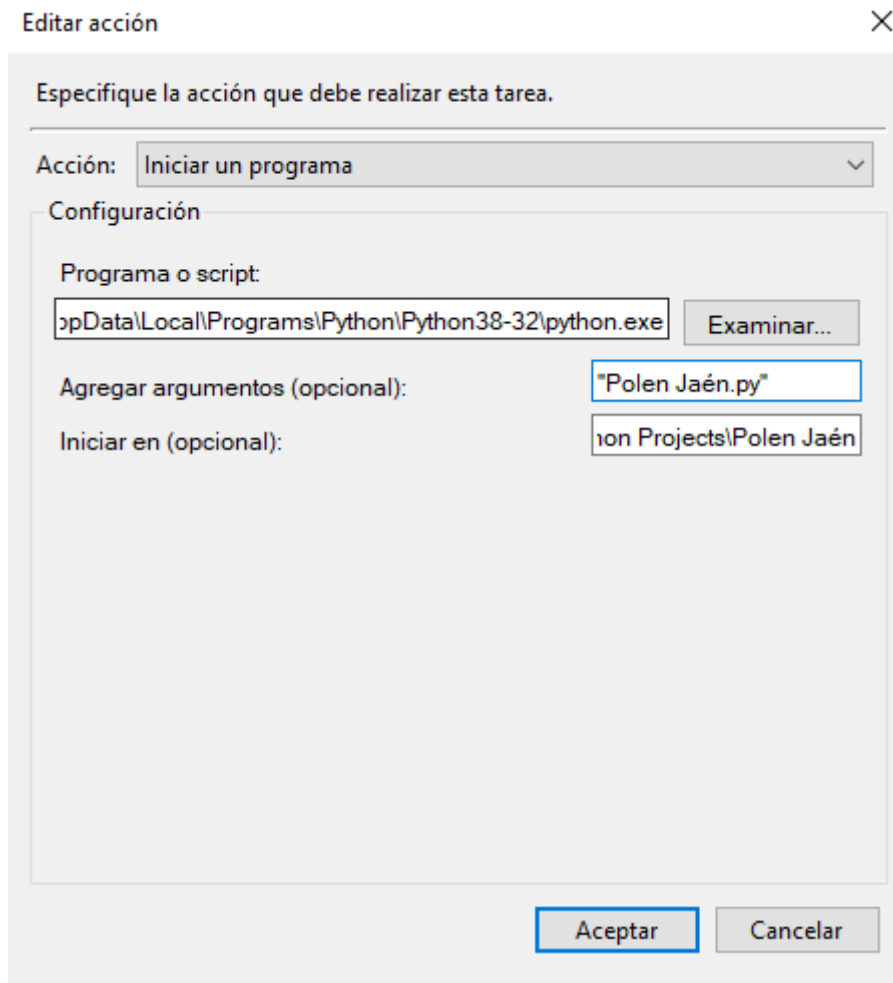


Ilustración 5.7 Configuración de la acción a realizar por el programador de tareas de Windows

Según la imagen mostrada, en el apartado “Programa o script” debemos seleccionar la ruta de instalación de Python en el equipo, en “Agregar argumentos” el nombre del ejecutable programado en Python y en “Iniciar en” la ruta donde se encuentra dicho ejecutable.

5.1.3. API de Telegram y configuración de canales

La aplicación Telegram va a ser doblemente utilizada para nuestro proyecto, ya que servirá para la difusión de los niveles de polen a través de un canal público donde las personas interesadas se suscribirán y los recibirán con una notificación en su móvil, y por otro lado servirá de herramienta de comunicación con el sistema mediante un canal privado al que solo tendrá acceso el administrador.

Al canal público se puede acceder y unirse libremente a través del siguiente enlace: <https://t.me/polenjaen>. Por el contrario, al canal privado no se puede acceder sin invitación.

Toda la comunicación con la API de Telegram se realiza mediante los conocidos bots, que son una serie de aplicaciones de terceros que se ejecutan dentro de la aplicación de mensajería y se integran en ésta de manera que se utilizan como si fueran una persona real con la que interactúas.

Por consiguiente, siguiendo las instrucciones de la documentación de Telegram, ayudados del asistente se ha creado un bot [12] llamado “Bot Polen Jaén” el cual puede ser abierto a través de este enlace: https://t.me/polenjaen_bot. Sin embargo, se puede observar que no tiene interacción ninguna pues su uso va a ser exclusivamente privado para las tareas que aquí se requieren.

La creación del bot se ha realizado con la ayuda del bot “BotFather” (<https://telegram.me/botfather>) y en el proceso se le asigna el nombre correspondiente y se recibe su token de autorización, que no es más que un código alfanumérico para poder comunicarme con la API bot de Telegram.

Una vez creado y configurado todo, será el código en Python ejecutado en la nube de AWS el encargado de controlar al bot que utilizará los canales privado y público según las necesidades del proyecto.

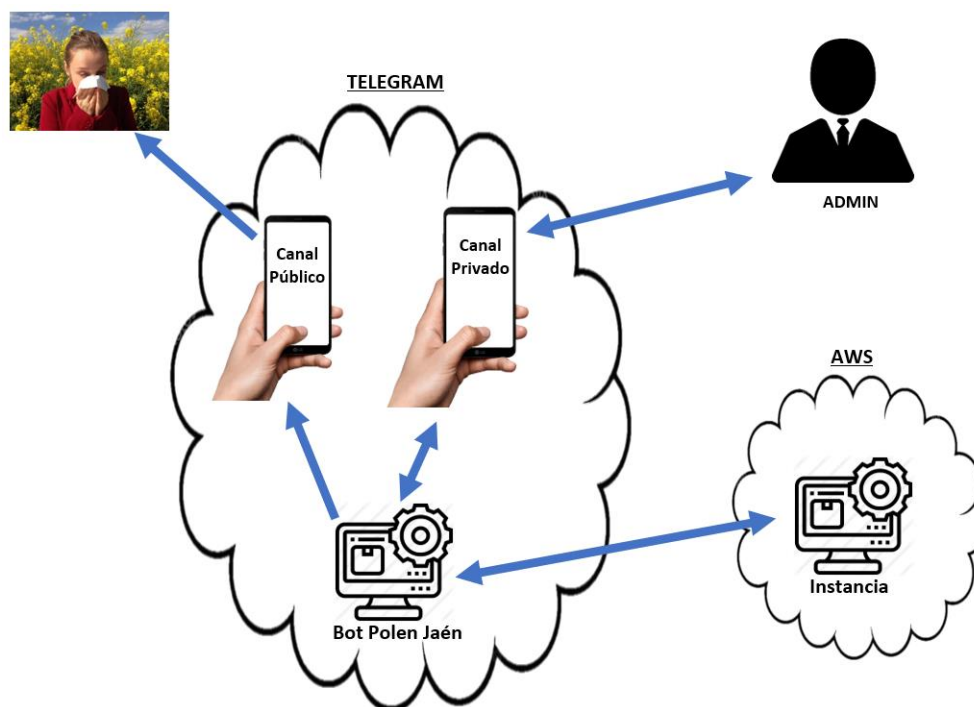


Ilustración 5.8 Esquema de la arquitectura del sistema con respecto a la API bot de Telegram

5.1.4. API de Facebook

En este capítulo se va a resumir el proceso seguido en Facebook para poder configurar su API Graph [13] con el objetivo de poder automatizar posteriormente en el programa en Python la publicación de los niveles de polen en la página de Facebook llamada “Polen Jaén”, la cual ya fue creada hace varios años (<https://www.facebook.com/polenjaen>).

Para empezar, hay que acceder al entorno de desarrolladores de Facebook (<https://developers.facebook.com>). Allí se debe crear una aplicación a la que se le asignará un nombre y que servirá de base para la comunicación con la API.

Tras crearla vamos a acceder dentro de ella al panel del explorador de la API Graph, donde se permite probar, crear y autenticar llamadas a la API y depurar los errores de las respuestas.

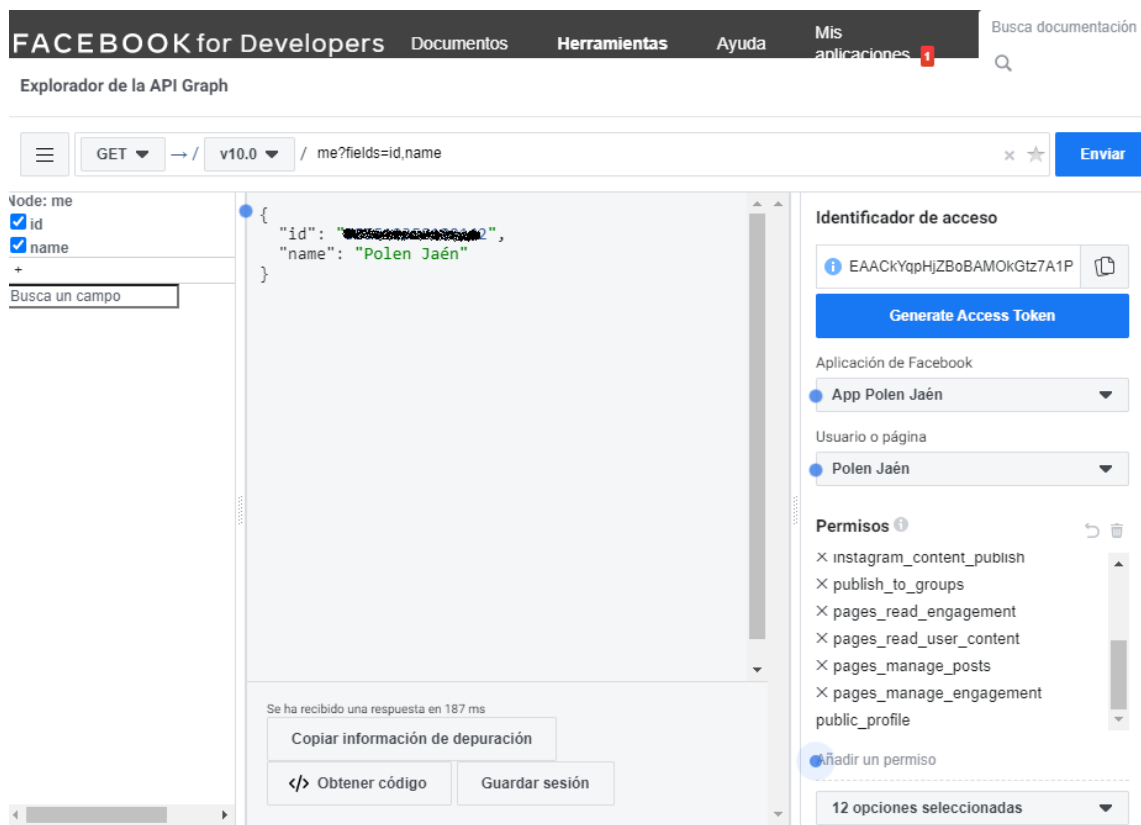


Ilustración 5.9 Panel del explorador de la API Graph de Facebook

Será desde aquí desde donde se permita habilitar los permisos necesarios para poder acceder al perfil de la página, leer información, publicar contenido, etc. También podremos obtener datos importantes como el id de la página y el token de acceso necesario para poder realizar todas esas acciones.

Destacar que el token dado por defecto por Facebook es de corta duración y es válido solo durante una hora. Así pues, hay que realizar una serie de pasos [14] para convertirlo en un token de larga duración, válido durante 60 días, y, posteriormente, con ese token, convertirlo a uno permanente válido para tiempo indefinido.

5.1.5. API de Instagram

Para poder hacer lo equivalente y conseguir publicar automáticamente los niveles de polen en el perfil “Polen Jaén” de Instagram (<https://www.instagram.com/polenjaen>) debemos utilizar y configurar el mismo

explorador de la API Graph de Facebook, ya que Instagram pertenece a Facebook y comparten esta arquitectura. [15]

Hay que recalcar que, para realizar este procedimiento, el perfil de Instagram debe de ser un perfil de tipo “Empresa” y estar conectada a la página de Facebook equivalente.

Una vez confirmado eso, se accede al explorador de la API Graph de Facebook de la aplicación creada en el apartado anterior y, seleccionando el token permanente y la configuración anterior, se añaden los permisos necesarios referentes a Instagram.

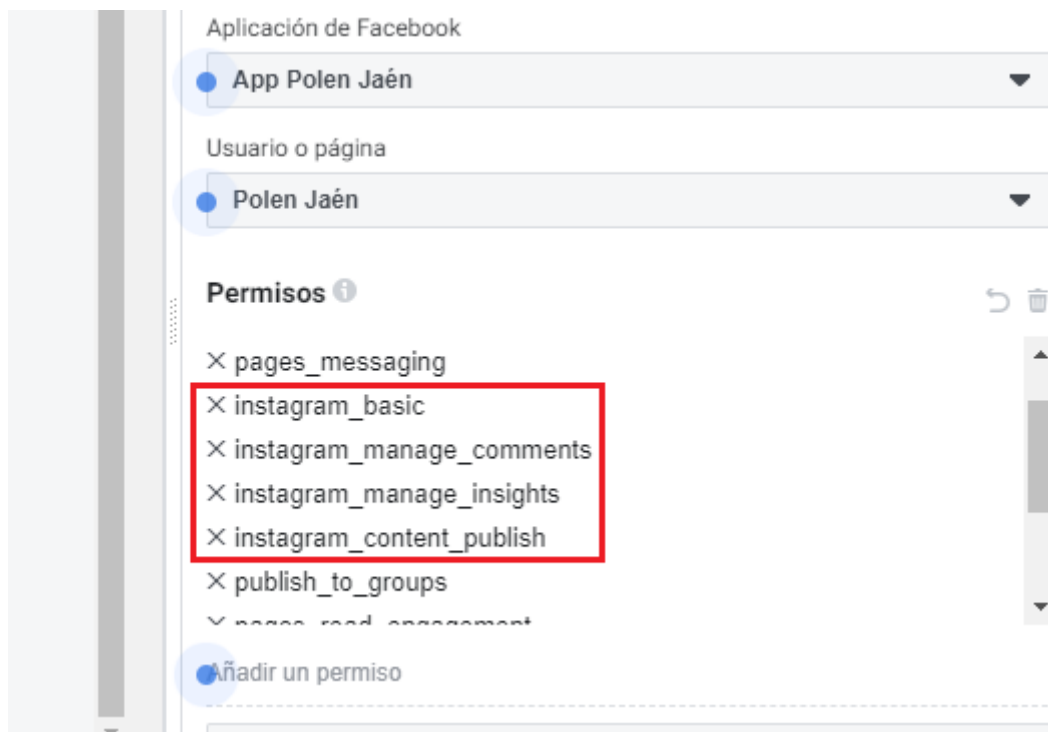


Ilustración 5.10 Permisos para publicar en Instagram añadidos a la app creada en Facebook

5.1.6. API de Twitter

Para que el código del script pueda comunicarse con la red social Twitter y poder así publicar los niveles de polen a través de la cuenta “Polen Jaén” creada para tal fin (https://twitter.com/polen_jaen) es necesario realizar configuraciones que se van a explicar a continuación.

En primer lugar, será requisito necesario que la cuenta en Twitter esté verificada con un número de teléfono para así poder crear una aplicación [16] asociada a la cuenta en la página de desarrolladores de Twitter. Esta aplicación es la que dotará de la funcionalidad deseada de comunicar con la API de la red social y poder publicar tuits a través de librerías de Python.

Para ello, hay que acceder al panel habilitado para tal fin (<https://developer.twitter.com/en/apps>) y crear la aplicación determinando el nombre, la descripción y los permisos entre otros parámetros y aceptando los términos y condiciones de Twitter.

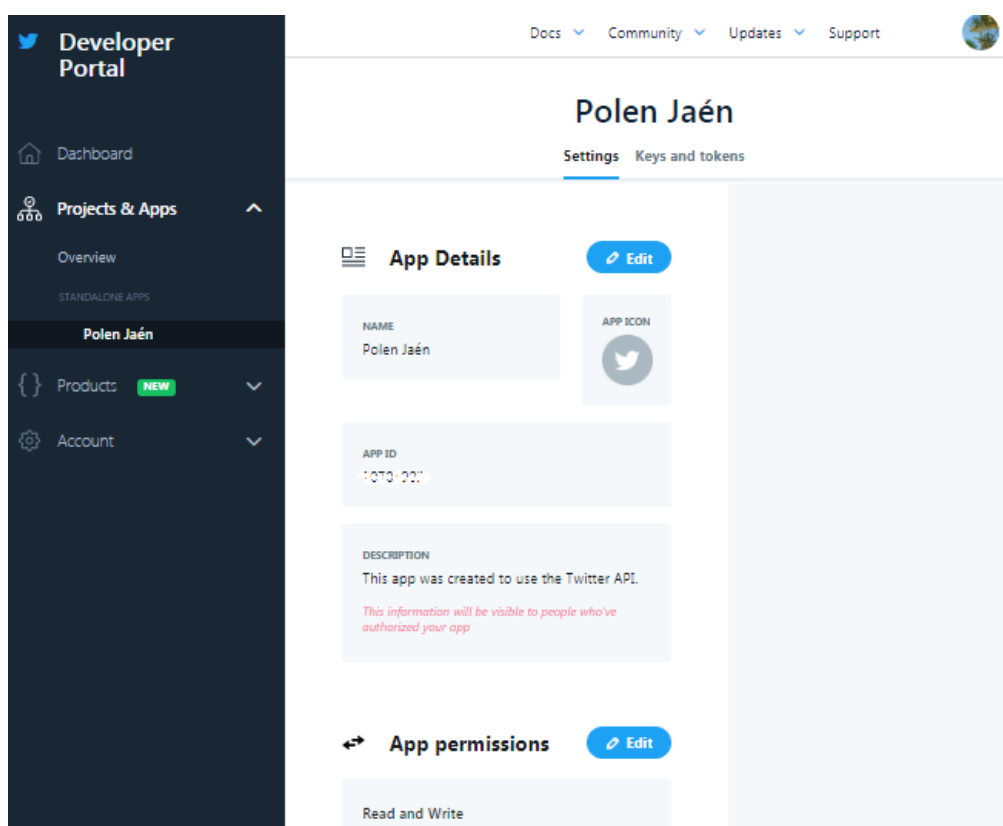


Ilustración 5.11 Panel de desarrollador de Twitter de la aplicación creada

Una vez creada la aplicación podemos acceder a los parámetros generados que posteriormente servirán de comunicación con nuestra cuenta. Estos parámetros son Access Token, Access Token Secret, API Key y API Secret Key.

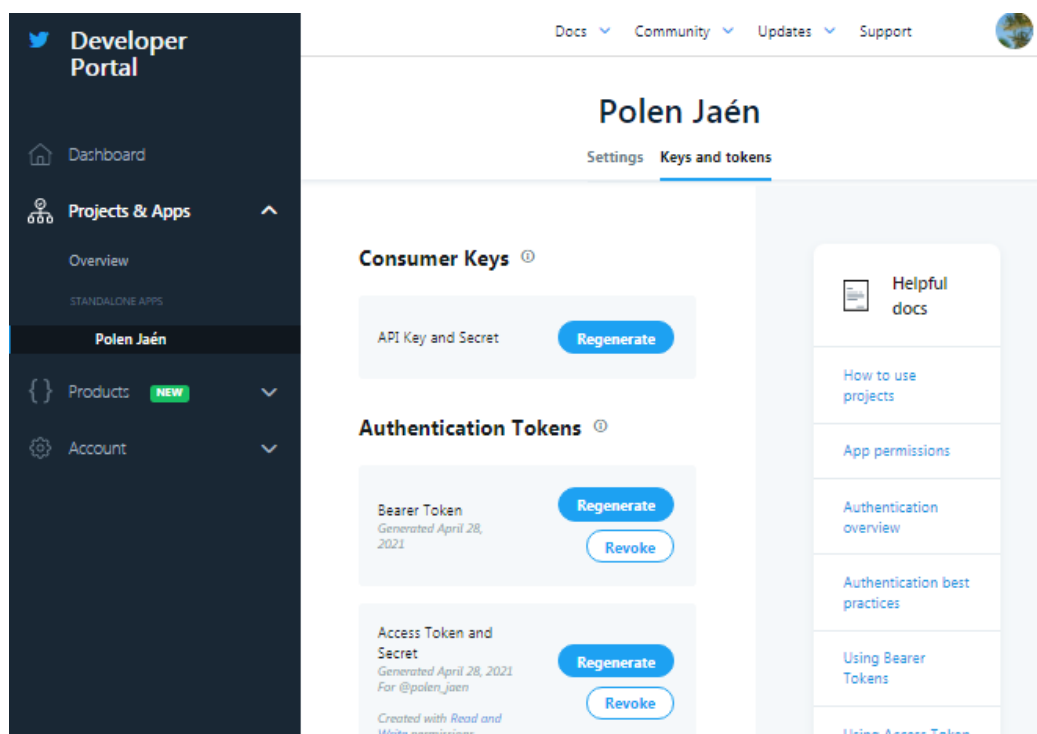


Ilustración 5.12 Panel de desarrollador de Twitter con Keys y Tokens de la aplicación creada

5.2. Implementación del código

En este capítulo del proyecto se va a explicar paso a paso todo el código en Python programado, código que se encargará de conducir y controlar al sistema a través de todo su flujo de ejecución. Dicho script puede considerarse el núcleo principal de la arquitectura del sistema.

5.2.1. Librerías usadas

Para empezar, se van a nombrar todas las librerías usadas en el código a la vez que se describirán sus funcionalidades para así razonar el motivo de su uso.

```
from bs4 import BeautifulSoup
```

Ilustración 5.13 Librería BeautifulSoup

Beautiful Soup [17][18] es una librería Python que permite extraer información de contenido en formato HTML o XML. Para usarla, es necesario especificar un parser, que es responsable de transformar un documento HTML o XML en un árbol complejo de objetos Python. Esto permite, por ejemplo, que podamos interactuar con los elementos de una página web como si estuviésemos utilizando las herramientas del desarrollador de un navegador.

Esta librería es usada en este proyecto con el fin de realizar web scraping en la web del hospital de Jaén. Me ayudo de ella para así moverme a través del código HTML y obtener el elemento clave, que en este caso es un vídeo en formato mp4 con los niveles de polen del día.

```
import requests
import urllib.request
```

Ilustración 5.14 Librería requests

Requests es una librería Python que facilita enormemente el trabajo con peticiones HTTP. En este caso se hace uso de esta librería para realizar una petición Get a la URL del Hospital de Jaén y poder así obtener su código fuente.

En cuanto al módulo urllib.request, proporciona una interfaz de programación para usar recursos de Internet identificados por URLs.

En el sistema se usa este módulo para abrir la URL (urlopen) del vídeo con extensión .mp4 contenido en la web del Hospital de Jaén para guardarlo posteriormente en la carpeta local del proyecto.

```
import time
```

Ilustración 5.15 Librería time

Este módulo proporciona varias funciones relacionadas con el tiempo. Es útil en este caso para tres cosas:

- Mostrar la hora del sistema mientras está comprobando cambios y llevar así un seguimiento más intuitivo.

- Controlar el intervalo de tiempo entre comprobaciones de cambios en la web del Hospital de Jaén. Concretamente, detenemos la ejecución del bucle de detección de cambios durante 120 segundos para que no se produzcan un exceso de peticiones en un espacio corto de tiempo.

- Detectar si ha llegado la hora límite para seguir comprobando cambios. Concretamente, si llegan las 19:00 y no se han actualizado los niveles en la web del Hospital de Jaén, se procede a cortar la ejecución del programa.

```
from datetime import date
```

Ilustración 5.16 Módulo datetime

El módulo datetime proporciona clases para manipular fechas y horas.

Con su utilización se consigue obtener la fecha y la hora con el formato deseado para posteriormente, durante la edición del frame capturado, incrustarlo en la plantilla final.

```
import cv2
```

Ilustración 5.17 Librería cv2

OpenCV es una conocida librería gratuita de código abierto utilizada en el procesamiento de imágenes, vídeos e incluso secuencias en directo.

Es usada en la función que captura el frame del vídeo. Con ella se carga el vídeo, se elige el frame interesado y se guarda como imagen en formato .jpg en la ruta del proyecto.

```
from PIL import Image, ImageDraw, ImageFont
```

Ilustración 5.18 Librería PIL

Python Imaging Library es una librería gratuita que permite la edición de imágenes directamente desde Python.

En este caso me facilita las herramientas necesarias para la edición deseada del frame capturado del vídeo. Con ellas se consigue plasmar en la imagen de los niveles la fecha del día correspondiente en un formato de visualización agradable, además del recuadro probablemente destinado a espacio publicitario como posible monetización del sistema, detalles que serán explicados más adelante.

```
import facebook
```

Ilustración 5.19 Librería facebook

El módulo Facebook facilita las herramientas necesarias para interactuar con la API Graph de Facebook y también con la de Instagram.

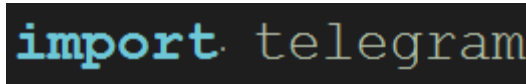
Así pues, es usada en el script para identificarse en Facebook e Instagram a través del respectivo token para después publicar la imagen con los niveles de polen en ambas redes sociales.

```
import tweepy
```

Ilustración 5.20 Librería tweepy

Es una librería a través de la cual se permite comunicar con la API de Twitter de una manera sencilla con Python.

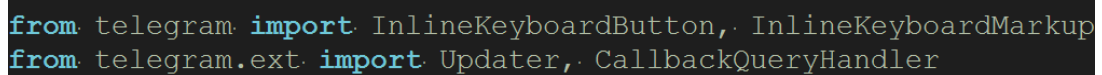
Es por eso que se usa para conectarse a Twitter con los correspondientes credenciales y publicar la imagen con los niveles de polen en esta red social. También se utiliza para guardar la url de la imagen del tuit que luego sirve para subir esa misma imagen a Instagram.

A screenshot of a code editor showing the text 'import telegram' in a monospaced font. The word 'import' is in blue and 'telegram' is in green.**Ilustración 5.21 Librería telegram**

De manera equivalente, se utiliza esta librería para esta vez comunicarse con la API de Telegram a través de sus conocidos bots.

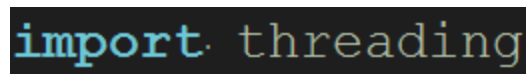
En concreto, se conecta al bot con su correspondiente Token y se publica en el canal público de Telegram los niveles de polen.

También se publica a través del canal privado de Telegram los mensajes más relevantes sobre la ejecución del programa para que el administrador pueda realizar el seguimiento del sistema.

A screenshot of a code editor showing two lines of Python code. The first line is 'from telegram import InlineKeyboardButton, InlineKeyboardMarkup' and the second line is 'from telegram.ext import Updater, CallbackQueryHandler'. The keywords 'from', 'import', and 'from' are in blue, while the module names are in green.**Ilustración 5.22 Módulos InlineKeyboardButton, InlineKeyboardMarkup, Updater y CallbackQueryHandler**

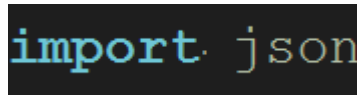
Estos módulos son utilizados de manera complementaria a la librería anterior.

Son usados para el momento de enviar la plantilla terminada al canal privado de Telegram para que el administrador pueda decidir si todo está correcto y se procede a su publicación, o si, por el contrario, algo no está bien y se debe detener.

A screenshot of a code editor showing the text 'import threading' in a monospaced font. The word 'import' is in blue and 'threading' is in green.**Ilustración 5.23 Librería threading**

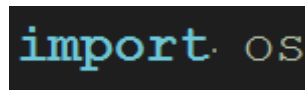
Es un módulo que facilita la utilización de hilos en Python.

En este caso se usa simplemente para cerrar la escucha del bot de Telegram una vez que el administrador ha decidido si continuar la ejecución del programa o ha decidido abortarla.

A screenshot of a code editor with a dark background. The text 'import json' is written in a light blue monospaced font. The word 'import' is in blue, and 'json' is in a lighter blue.**Ilustración 5.24 Librería json**

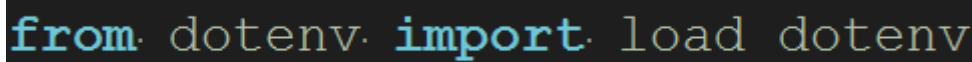
JSON (JavaScript Object Notation) se puede considerar como un método simple de crear y almacenar estructuras de datos. Con este módulo, Python hace que sea sencillo trabajar con archivos JSON.

En este script solo se va a utilizar esta librería para tratar algunos datos que devuelve la API Graph de Facebook al publicar la imagen de los niveles de polen en Instagram.

A screenshot of a code editor with a dark background. The text 'import os' is written in a light blue monospaced font. The word 'import' is in blue, and 'os' is in a lighter blue.**Ilustración 5.25 Librería os**

El módulo os de Python permite realizar operaciones dependientes del Sistema Operativo como crear una carpeta, listar contenidos de una carpeta, conocer acerca de un proceso, finalizar un proceso, etc.

Para este caso se utiliza para facilitar la carga de parámetros desde el fichero correspondiente en la ruta del proyecto.

A screenshot of a code editor with a dark background. The text 'from dotenv import load_dotenv' is written in a light blue monospaced font. The word 'from' is in blue, 'dotenv' is in a lighter blue, 'import' is in blue, and 'load_dotenv' is in a lighter blue.**Ilustración 5.26 Librería dotenv**

La librería dotenv es la que permite manipular archivos `.env`, que es un archivo de texto donde, en cada línea, se define una variable de entorno y se le asigna su valor mediante el operador `=`.

En este caso tenemos un archivo `.env` con variables y sus correspondientes valores, las cuales deben ser privadas y no estar accesibles a otros usuarios, es por eso que no están definidas en el propio código.

5.2.2. Explicación del código

Ahora llega al momento de mostrar las principales y más relevantes partes del código fuente desarrollado en Python y explicar qué hacen.

```
def leerHtmlFichero(nombreFichero):  
    ... fichero = open(nombreFichero, 'r')  
    ... #Lectura completa del fichero  
    ... htmlFichero = fichero.read()  
    ... #Se cierra el fichero  
    ... fichero.close()  
    ... return htmlFichero
```

Ilustración 5.27 Función leerHtmlFichero

Esta función es la encargada de leer el contenido del fichero que se la pasa por parámetro y devolverlo en tipo string. En el código se usa para leer el contenido del fichero URL_MP4.txt.

```
def obtenerHtmlWeb(url):  
    ... #Petición get a la url  
    ... req = requests.get(url)  
  
    ... #Obtenemos el status code y el código html  
    ... status_code = req.status_code  
    ... cod_html = req.text  
  
    ... #Si la respuesta request es correcta  
    ... if(status_code == 200):  
    ...     ... #Se pasa el contenido HTML de la web a un objeto BeautifulSoup  
    ...     ... html = BeautifulSoup(cod_html, "html.parser")  
    ...     ... return html  
    ... else:  
    ...     ... print('Status Code: ' % status_code)  
    ...     ... return 0
```

Ilustración 5.28 Función obtenerHtmlWeb

Con esta función se realiza una petición Http a la url del Hospital de Jaén pasada por parámetro y se obtiene su código Html completo que se devuelve con un objeto BeautifulSoup para tratarlo y filtrarlo posteriormente.

```
def filtrarHtml(html):  
    ...etiquetaClave = html.find('source').get('src')  
    ...return str(etiquetaClave)
```

Ilustración 5.29 Función filtrarHtml

Con esta sencilla función se filtra el código Html como objeto BeautifulSoup conseguido en la función anterior. Con el método *find()* se consigue la url del vídeo con los niveles de polen contenido en la web.

Esa url es devuelta ahora como tipo string para posteriormente compararla con la url que se tenía guardada en nuestro fichero y detectar si se han producido cambios o no.

```
def actualizacionHtmlFichero(nombreFichero, htmlNuevo):  
    ...fichero = open(nombreFichero, 'r+')  
    ...#Borrado del fichero completo  
    ...fichero.truncate(0)  
    ...fichero.writelines(htmlNuevo)  
    ...#Cierre del fichero  
    ...fichero.close()
```

Ilustración 5.30 Procedimiento actualizacionHtmlFichero

Este procedimiento es el encargado de borrar el contenido del fichero URL_MP4.txt y sustituirlo por la nueva url del vídeo cuando ya se han producido cambios en la web del Hospital de Jaén. Así se consigue tener la información actualizada y guardada para comparar y detectar los cambios en la web del Hospital de Jaén al día siguiente.

```
def descargarVideo(url):  
    ...r = urllib.request.urlopen(url)  
    ...video = "Video.mp4"  
    ...f = open(video, 'wb')  
    ...f.write(r.read())  
    ...f.close()
```

Ilustración 5.31 Procedimiento descargarVideo

Con este procedimiento se descarga el vídeo con los niveles de polen actualizados de la web del Hospital de Jaén y es guardado en el repositorio del proyecto con el nombre “Vídeo.mp4”.

```
def capturarFrame() :  
    ... vid = cv2.VideoCapture("Vídeo.mp4")  
    ... contadorFrames = 0  
  
    ... #El frame 386 está a la mitad del vídeo y valdría  
    ... while contadorFrames < 386:  
    ...     ret, frame = vid.read()  
    ...     if ret:  
    ...         contadorFrames += 1  
    ...     else:  
    ...         break  
  
    ... cv2.imwrite('Images\FrameUtil.jpg', frame)  
  
    ... vid.release()  
    ... cv2.destroyAllWindows()
```

Ilustración 5.32 Procedimiento capturarFrame

Este procedimiento es el utilizado para capturar el frame del vídeo descargado en nuestro repositorio para así disponer de la imagen con los niveles de polen. Se elige el frame 386 porque es el que está en mitad de los 772 frames que tiene el vídeo completo y los niveles siempre se encuentran en la parte central de éste. La imagen es guardada en la carpeta “Images” del repositorio con el nombre “FrameUtil.jpg”. Todo este proceso se realiza con la ayuda de la librería de OpenCV.

```
def avisoBotTelegram():
    """#Apagado de la escucha del bot"""
    def shutdown():
        updater.stop()
        updater.is_idle = False

    def continuar(bot, update):
        global CONTINUAR
        CONTINUAR = True
        threading.Thread(target=shutdown).start()

    def cancelar(bot, update):
        threading.Thread(target=shutdown).start()

    """#Creación del teclado"""
    keyboard = [[InlineKeyboardButton('CONTINUAR PROGRAMA \U00002705', callback_data='continuar')],
                [InlineKeyboardButton("Cancelar ejecución del programa \U0000274c", callback_data='cancelar')]]
    reply_markup = InlineKeyboardMarkup(keyboard)
    """#Envío del frame con teclado para confirmar o cancelar"""
    with open('Images\FrameUtil_terminado.jpg', 'rb') as photo_file:
        bot.sendPhoto(chat_id=CHAT_ID_CANAL_PRIVADO,
                      photo=photo_file,
                      caption='',
                      reply_markup=reply_markup)
    """#Creación del updater y el dispatcher"""
    updater = Updater(token=BOT_TOKEN, use_context=True)
    dp = updater.dispatcher
    """#Configuración del dispatcher para recibir la respuesta elegida"""
    dp.add_handler(CallbackQueryHandler(pattern='continuar', callback=continuar))
    dp.add_handler(CallbackQueryHandler(pattern='cancelar', callback=cancelar))
    """#El bot se queda a la escucha"""
    updater.start_polling()
    updater.idle()
```

Ilustración 5.33 Procedimiento avisoBotTelegram

Este procedimiento es el encargado de activar el modo escucha del bot de Telegram al enviar la imagen a través del canal privado generando un teclado con dos opciones en las que el administrador debe elegir una. La ejecución se detiene hasta que se detecta una respuesta, la cual es identificada para poder actuar en consecuencia a través de la función *continuar(bot, updater)* o *cancelar(bot, updater)*

```
def publicarFacebook():
    """#The Graph API allows you read and write data to and from the facebook social graph"""
    fb = facebook.GraphAPI(ACCESS_TOKEN_FACEBOOK)
    """#Post a photo with captions"""
    text = "Telegram: t.me/polenjaen\nInstagram: instagram.com/polenjaen\nTwitter: twitter.com/polen_jaen"
    fb.put_photo(open('Images\FrameUtil_terminado.jpg', 'rb'), message=text)
```

Ilustración 5.34 Procedimiento publicarFacebook

Con este procedimiento, mediante su API, se consigue publicar en la página de Facebook “Polen Jaén” la imagen seleccionada de la plantilla terminada que se encuentra ubicada en el repositorio del proyecto y un texto especificado.

```
def publicarTwitter():  
    ''' # Configuración de acceso con las credenciales  
    auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)  
    auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)  
  
    ''' # Conexión con el API  
    api = tweepy.API(auth)  
  
    ''' # Publico tuit con la imagen y texto deseados  
    filename = 'Images\FrameUtil_terminado.jpg'  
    status = "Telegram: t.me/polenjaen"  
    api.update_with_media(filename, status)  
  
    ''' # Navego a través de las clases para obtener el parámetro 'media_url_https'  
    ''' # que me servirá para publicar en instagram  
    ''' # devuelve class dict  
    url_media = api.me().status.entities  
    ''' # devuelve class list  
    url_media = url_media['media']  
    ''' # devuelve class dict  
    url_media = url_media[0]  
    ''' # devuelve str con la url de la imagen  
    url_media = url_media['media_url_https']  
    return url_media
```

Ilustración 5.35 Función publicarTwitter

Con esta función se consigue lo equivalente para el perfil de Twitter “Polen Jaén”. Conectado con la API de la red social a través de las credenciales de acceso se consigue publicar la imagen alojada en el repositorio local con la plantilla terminada y el texto deseado. En esta ocasión, obtengo la url de la imagen que acabo de publicar y la devuelvo, pues será necesario para la posterior publicación en Instagram.

```
def publicarInstagram(url_media):
    '''image_location_l' = url_media
    post_url = 'https://graph.facebook.com/v10.0/17841405395759208/media'

    payload = {
        'image_url': image_location_l,
        'caption': '',
        'access_token': ACCESS_TOKEN_FACEBOOK
    }

    r = requests.post(post_url, data=payload)
    result = json.loads(r.text)

    if 'id' in result:
        creation_id = result['id']
        second_url = 'https://graph.facebook.com/v10.0/17841405395759208/media_publish'
        second_payload = {
            'creation_id': creation_id,
            'access_token': ACCESS_TOKEN_FACEBOOK
        }
        r = requests.post(second_url, data=second_payload)
    else:
        print('Error al publicar en Instagram')
```

Ilustración 5.36 Procedimiento publicarInstagram

Con este procedimiento se publica en el perfil de Instagram “Polen Jaén” la plantilla de los niveles de polen, aunque esta vez sin ningún texto. A diferencia de los otros métodos, en esta ocasión la API de Instagram obliga que la imagen a publicar se encuentre alojada en algún servidor, y es por eso que se recibe la url de la imagen subida a Twitter en la anterior función y se utiliza para tal fin.

```
def publicarTelegram():
    with open('Images\FramUtil terminado.jpg', 'rb') as photo_file:
        bot.sendPhoto(chat_id=CHAT_ID_CANAL_PUBLICO,
                      photo=photo_file,
                      caption='Facebook: facebook.com/polenjaen\nInstagram: instagram.com/polenjaen\nTwitter: twitter.com/polen_jaen')
```

Ilustración 5.37 Procedimiento publicarTelegram

Este procedimiento logra publicar la plantilla terminada ubicada en el repositorio del proyecto junto con el texto deseado en el canal público de Telegram “Polen Jaén”.

Como es lógico, la ejecución de todas estas funciones y procedimientos son controladas por una función *main* que controla todo el flujo de trabajo del sistema tal como viene plasmado en el apartado “4.2 Workflow”.

5.3. Repositorio del proyecto

En esta sección se va a proceder a mostrar y exponer los diferentes archivos que componen el repositorio del proyecto Python desarrollado.

Nombre	Fecha de modificación	Tipo	Tamaño
Images	07/06/2021 14:29	Carpeta de archivos	
.env	09/07/2021 14:13	Archivo ENV	1 KB
.env	09/07/2021 18:00	Archivo SAMPLE	1 KB
Polen Jaén	10/08/2021 23:17	Python File	11 KB
polen_jaén.nja	23/04/2021 18:29	Archivo NJA	1 KB
URL_MP4	12/08/2021 18:35	Documento de te...	1 KB
Vídeo	12/08/2021 18:35	MP4 Video File (V...	11.834 KB

Ilustración 5.38 Repositorio del proyecto

- Images

Carpeta donde se van guardando las distintas imágenes tras las diferentes ediciones.

- .env

Fichero desde el que son leídos ciertos parámetros y variables por el código fuente. En concreto contiene la url de la web del Hospital de Jaén, el nombre del vídeo descargado de la web, el tiempo de refresco de comprobación de cambios y credenciales de conexión a las diferentes APIs. No se adjuntará al compartir el proyecto por motivos de seguridad.

- .env.sample

Fichero idéntico al anterior, pero con el valor de las variables de credenciales borrados. Será el que se adjunte al compartir el proyecto para ser completado por el usuario que lo requiera con sus credenciales correspondientes.

- Polen Jaén.py

Archivo Python con el código en dicho lenguaje preparado para ser ejecutado.

- polen_jaén.nja

Archivo creado por el editor IDE Ninja, que ha sido el utilizado para programar el proyecto.

- URL_MP4.txt

Fichero donde se almacena la url del vídeo en formato .mp4 extraído mediante web scrapping del código Html del Hospital de Jaén.

- Vídeo.mp4

Vídeo en formato .mp4 descargado de la web del Hospital de Jaén para posteriormente tratarlo convenientemente.

Nombre	Fecha	Tipo	Tamaño
FrameUtil limpios	08/05/2021 12:25	Carpeta de archivos	
Iconos	08/05/2021 11:48	Carpeta de archivos	
FrameUtil terminado	08/05/2021 14:23	Archivo JPG	128 KB
FrameUtil	28/04/2021 11:29	Archivo JPG	237 KB
Recorte con fecha	08/05/2021 14:09	Archivo JPG	61 KB
Recorte sin fecha	08/05/2021 11:55	Archivo JPG	207 KB
Recorte sin fecha	10/05/2021 13:14	Adobe Photoshop...	1.770 KB

Ilustración 5.39 Contenido de la carpeta 'Images' del repositorio del proyecto

- FrameUtil limpios.

Carpeta que contiene las imágenes 'FrameUtil limpio.jpg' y 'Recorte limpio.jpg', que son el frame capturado del vídeo y el recorte inferior izquierdo, ambos con los textos eliminados. No son utilizados durante la ejecución del programa, solo para posibles ediciones futuras.

- FrameUtil terminado.jpg

Imagen con la plantilla ya editada convenientemente y lista para enviar al canal privado de Telegram para la revisión por parte del usuario y, si se diera el visto bueno, para el correspondiente envío en las diferentes redes sociales.

- FrameUtil.jpg

Imagen con el frame original extraído del vídeo del Hospital de Jaén.

- Recorte con fecha.jpg

Imagen del recorte inferior derecho con la fecha ya superpuesta y preparada para combinar con 'FrameUtil.jpg'.

- Recorte sin fecha.jpg

Imagen del recorte inferior derecho con la publicidad correspondiente editada en Photoshop y preparada para superponerle la fecha.

- Recorte sin fecha.psd

Archivo de Photoshop correspondiente a la imagen anterior y preparada para hacer las ediciones y poder así publicitar a empresas cuando sea necesario.

5.4. Manual de despliegue

Para conseguir ejecutar el proyecto completo en otros equipos conforme a todo lo especificado y redactado en esta memoria se necesitan realizar una serie de pasos que se detallarán en esta sección.

Resaltar que este sistema ha sido desarrollado y probado desde equipos con Microsoft Windows, que sería el sistema operativo recomendado y se desconoce su funcionamiento en otros sistemas operativos.

1. Copiar el contenido del proyecto en el equipo correspondiente. Para su ejecución será necesario tener instalado Python 3, el cual puede ser descargado desde su web oficial (<https://www.python.org/downloads>). A

partir de la versión 3.4 de Python, la instalación viene incluida con PIP, que es un sistema de gestión de paquetes utilizado para instalar y administrar paquetes de software escritos en Python. En este caso se ejecutarán las siguientes instrucciones desde la consola de Windows con privilegios de administrador para instalar los correspondientes paquetes imprescindibles para la ejecución del código:

```
pip install beautifulsoup4
```

```
pip install requests
```

```
pip install opencv-contrib-python
```

```
pip install tweepy
```

```
pip install instabot
```

```
pip install facebook-sdk
```

```
pip install python-telegram-bot
```

```
pip install python-dotenv
```

2. Crear los canales privado y público de Telegram, la página en Facebook y los perfiles de Instagram y Twitter para la difusión de los niveles de polen, además de realizar las configuraciones previas descritas en el apartado 5.1. La del apartado 5.1.1 será opcional si se quiere ejecutar desde la nube de Amazon, pero también se podría ejecutar desde algún servicio equivalente o incluso desde algún equipo local.
3. Una vez realizadas dichas configuraciones, se obtendrán las credenciales de acceso de cada una de las redes sociales para la difusión de los niveles de polen y de comunicación con el sistema a través de Telegram. Estas credenciales deben de ser introducidas en las claves correspondientes del fichero *.env.sample* del repositorio del proyecto. Tras esto, el fichero debe convertirse a la extensión *.env*.

Aplicados todos estos pasos, el sistema estará listo para ser ejecutado correctamente, situación que se producirá manualmente, o de manera automática a la hora planificada si se ha configurado en el programador de tareas de Windows.

5.5. Prueba real

Una vez explicado detalladamente el desarrollo del sistema, voy a proceder a realizar una simulación de una prueba real mostrando la secuencia de pasos por los que transcurre todo el proceso, con la intención de plasmar de una manera mucho más nítida el funcionamiento del trabajo aquí desarrollado.

Al realizarse esta prueba con fecha 10 de agosto de 2021 y estar fuera del periodo de publicación de los niveles de polen se “forzará” al sistema para que realice todos los pasos. También se hará una publicación real en nuestros canales de difusión, aunque serán borrados con la mayor inmediatez posible para no confundir a los numerosos seguidores.

En un principio el sistema se encuentra pausado hasta que el programador de tareas de Windows lo activa a la hora configurada. Ahí es cuando comienza la ejecución del programa Python y se inicia la monitorización de la web del Hospital de Jaén. Es en ese momento cuando llega una notificación al canal privado para avisar al administrador de dicho comienzo y, a la vez, una consola de Windows en el servidor en la nube se inicia mostrando el proceso.

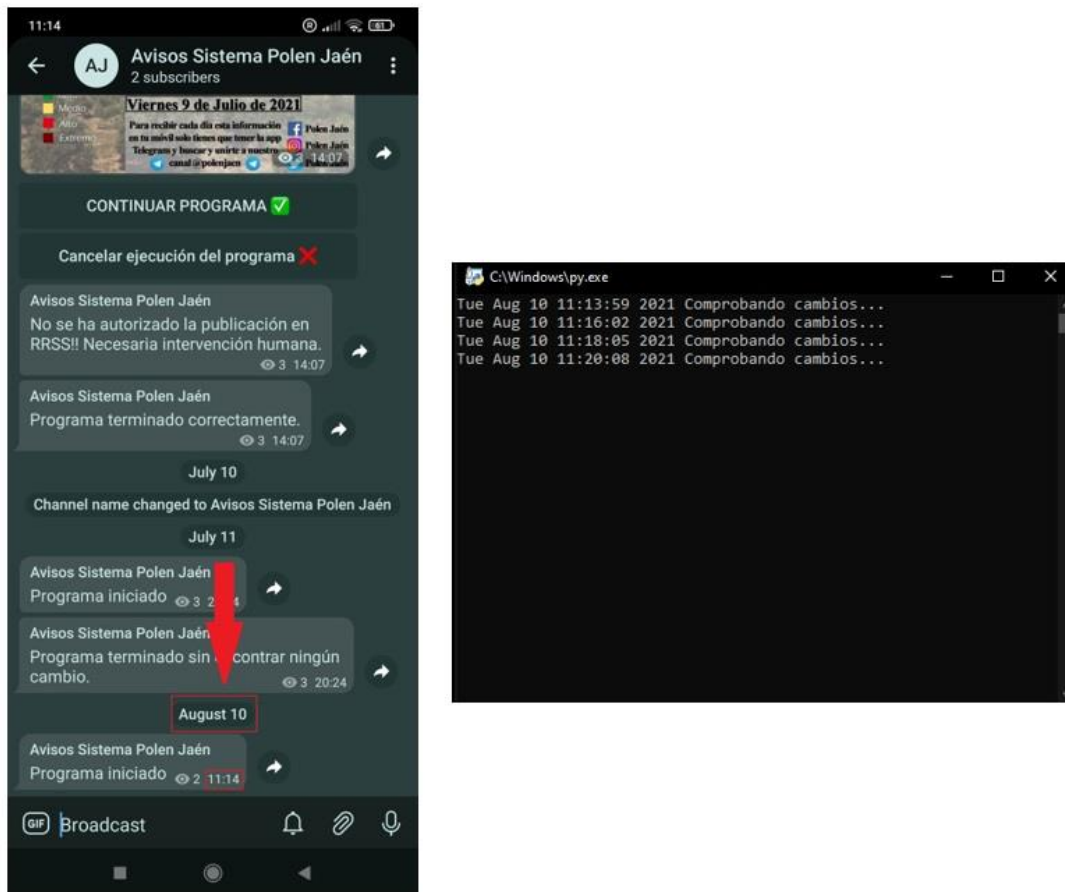


Ilustración 5.40 Comunicaciones al administrador de comienzo del sistema

Llega el momento en el que se producen cambios en la web del Hospital de Jaén y se actualiza con los datos nuevos del día. Es entonces cuando el sistema informa de tal acontecimiento por la consola de Windows y también de los pasos que le siguen: actualización de sus datos internos para el correcto funcionamiento de la monitorización al día siguiente, descarga del vídeo con los nuevos datos de los niveles de polen, captura del frame del vídeo, ediciones pertinentes sobre el frame y, por último, envío a través del canal privado de Telegram de la imagen final donde se da la posibilidad de autorizar publicación en redes sociales o de cancelar el proceso. El sistema se mantendrá en estado de espera hasta que el administrador elija la opción conveniente.

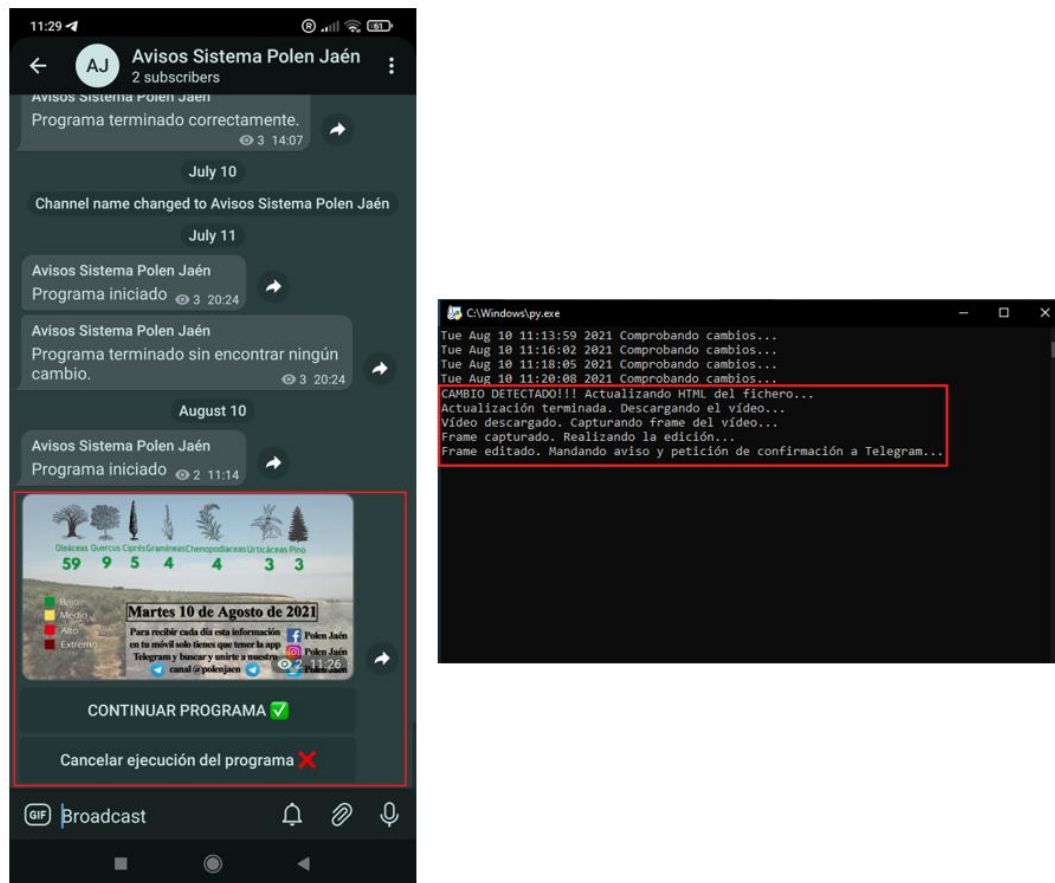


Ilustración 5.41 Comunicaciones al administrador de cambios detectados y petición de continuación

Ahora se produce el caso en el que el administrador ha notado algún error en la plantilla final y decide cancelar la ejecución del programa y no publicar en redes sociales. En las siguientes imágenes se muestran las correspondientes salidas tanto por consola como por el canal privado de Telegram. Tras esto, se requerirá intervención humana para subsanar el posible error que se haya producido. Destacar que esta situación no se ha dado durante los dos meses que ha estado funcionando el sistema durante esta primavera de 2021.

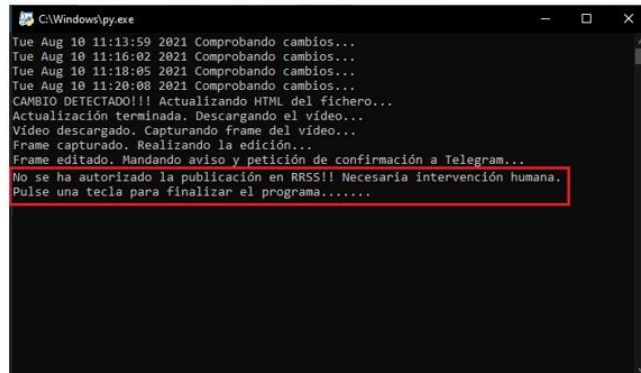


Ilustración 5.42 Comunicaciones al administrador de detección del sistema por orden recibida

Para el caso contrario, el más habitual y en el que todas las ediciones al frame se han realizado correctamente, el administrador pulsará en “CONTINUAR PROGRAMA” y se procederá a la publicación automática de la plantilla en cada una de las redes sociales. A continuación, se muestran las comunicaciones emitidas por el sistema tanto en el canal privado como en la salida por consola.

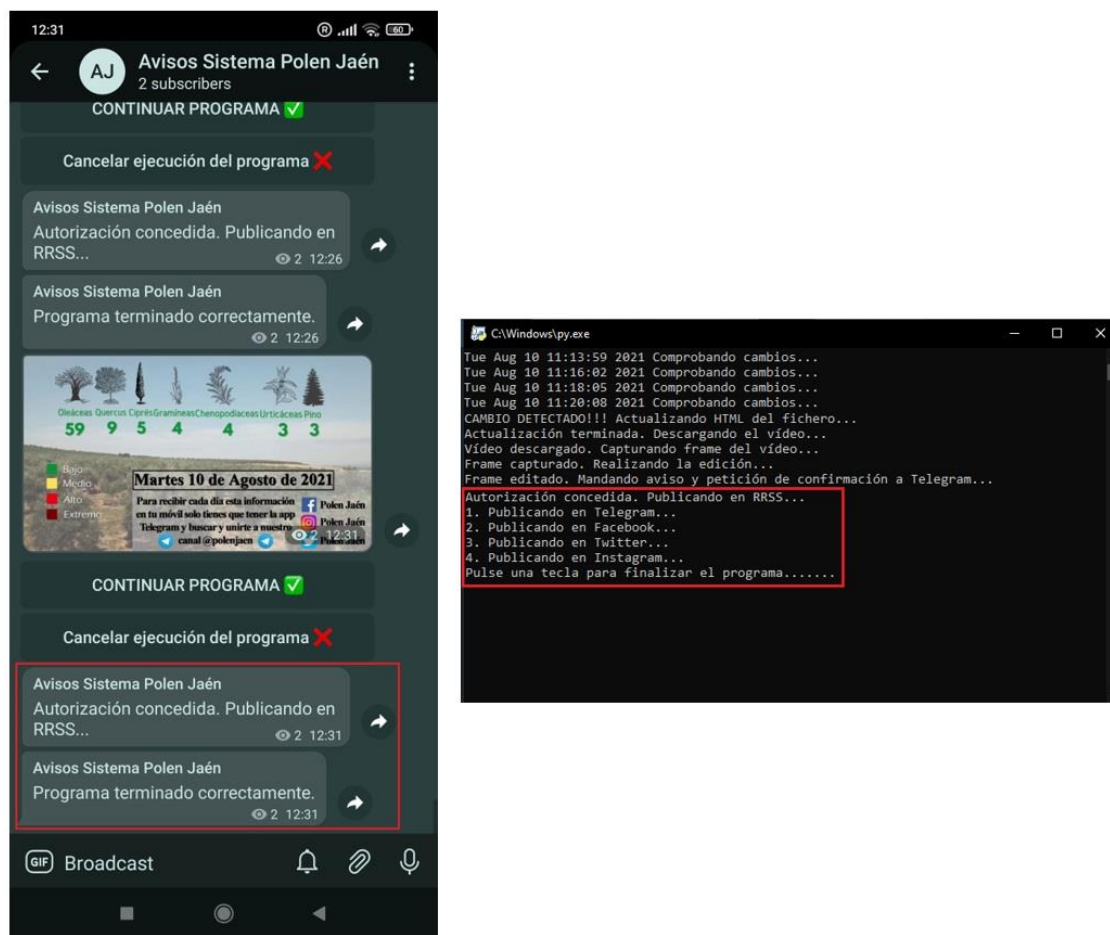


Ilustración 5.43 Comunicaciones al administrador de publicación en RRSS tras orden recibida

Por último, se muestran las capturas de los cuatro medios de difusión de los niveles de polen con la correspondiente publicación del día 10 de agosto en la que se está realizando esta prueba. Como se comentó anteriormente, estos datos publicados no corresponden a la realidad, pues publica los últimos datos actualizados de este año (15 de junio de 2021) con fecha del día actual (10 de agosto de 2021) y serán borrados tras realizar las capturas para evitar confusiones a los seguidores de las redes sociales.

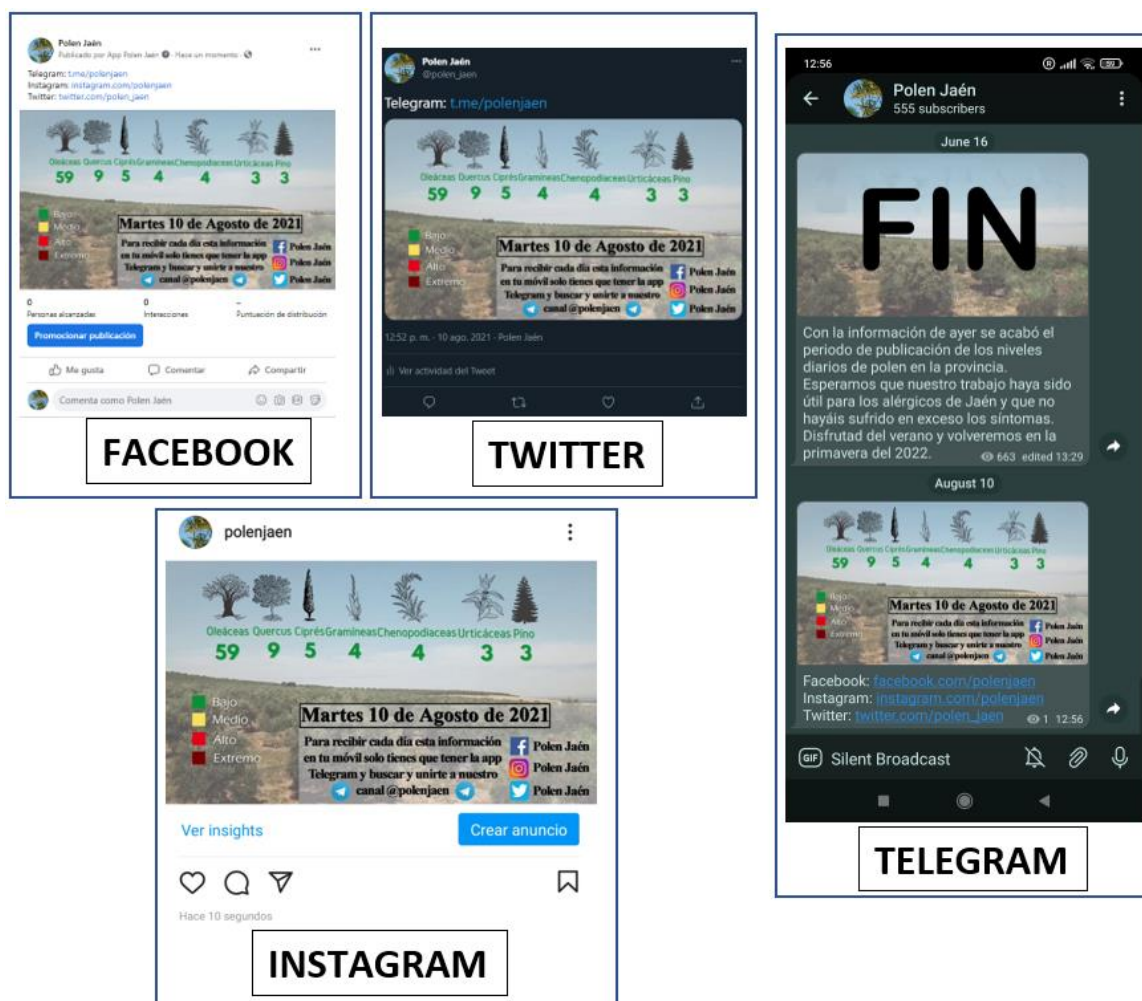


Ilustración 5.44 Publicación de los niveles de polen en Facebook, Instagram, Twitter y Telegram

La última posibilidad por plasmar en esta sección sería la de que estuviera intentando detectar cambios desde la web del Hospital de Jaén y llegaran las 7 de la tarde y éstos no se produjeran. En tal caso el sistema está programado para que se detenga. A continuación, se muestran las salidas en el canal privado de Telegram y en la consola de Windows.

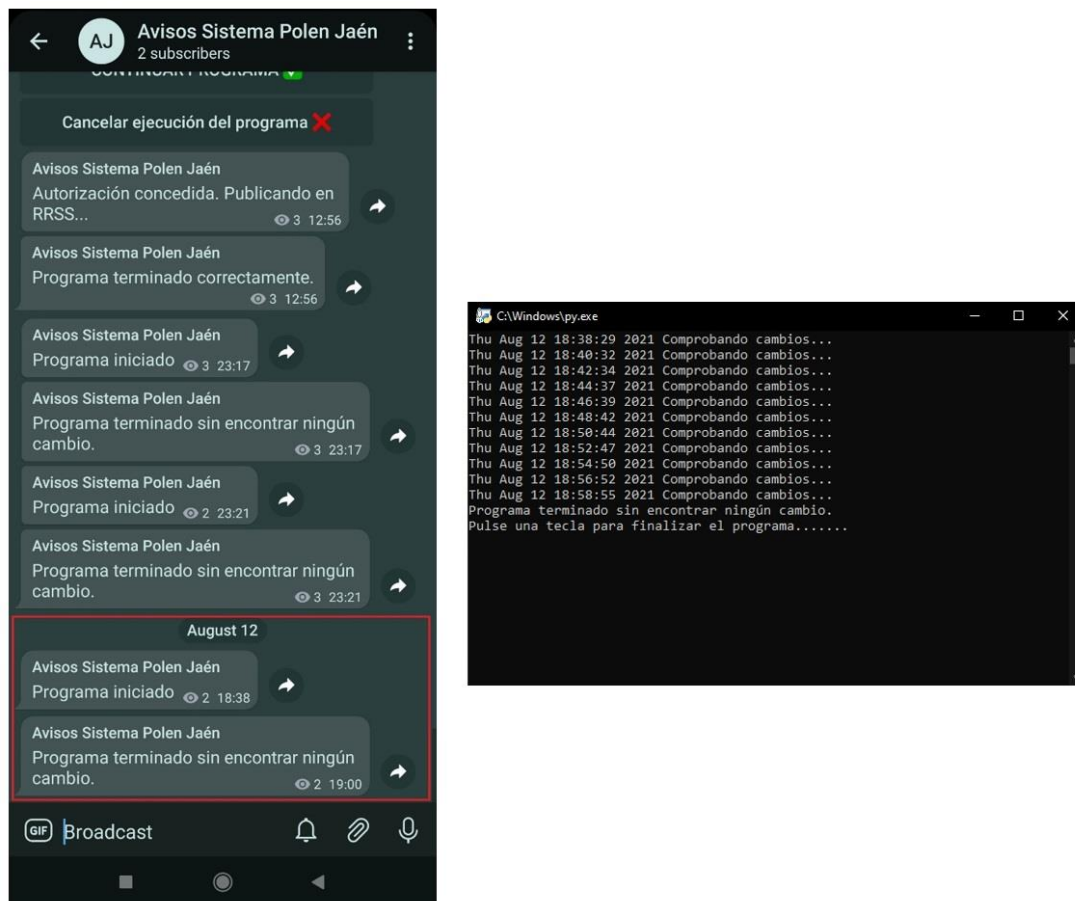


Ilustración 5.45 Comunicaciones al administrador de finalización del sistema por falta de cambios

Con esto se acaba la simulación de una prueba real del funcionamiento del sistema desarrollado en este TFG y donde se consiguen automatizar todo el proceso de detección, captura, edición y envío de los datos de los niveles de polen en la provincia de Jaén, alcanzando así los objetivos de este proyecto.

6. LÍNEAS DE FUTURO

Este capítulo se va a utilizar para explicar, desde el punto de vista subjetivo, las diferentes evoluciones futuras que puede tener el sistema creado y configurado en este proyecto.

Considero que son numerosas las posibilidades y los caminos a tomar para mejorar el sistema, además de todas las posibles adaptaciones que se podrían

tomar para ajustarse a ciertas novedades que puedan surgir en el contexto de este TFG.

6.1. Posible monetización del sistema

Del estudio sobre la utilidad real de la información difundida y del gran seguimiento de los usuarios de las diferentes redes sociales comentada en el capítulo 2.2, surge la idea de aprovechar ese seguimiento ofreciendo la posibilidad de que el boletín diario con los niveles de polen vaya acompañado de un pequeño espacio publicitario para empresas que lo consideren de utilidad.

Es probable que numerosas empresas vean con buenos ojos promocionarse en este tipo de boletines e incluirlo entre sus campañas de marketing. El público objetivo al que mayormente se llega con este tipo de boletines se podría considerar que es a personas alérgicas al polen primaveral y de la provincia de Jaén. También podríamos concluir que, al difundirse en cuatro redes sociales en las que encajan usuarios de diferentes generaciones, también está dirigido a personas de todas las edades.

Así pues, esta promoción sería interesante para empresas que vendan algún producto enfocado a las personas alérgicas o también para empresas de cualquier ámbito de la provincia Jaén que operen o busquen clientes en distintas ciudades de la comarca.

En cuanto al primer grupo que es el más específico, he buscado algunos ejemplos de empresas tipo que podrían guardar el perfil con las características comentadas. Algunos podrían ser:

Rino-Ebastel es un antihistamínico y descongestivo que está indicado en el alivio de los síntomas nasales y oculares asociados a la rinitis alérgica estacional con congestión nasal en adultos y adolescentes a partir de 12 años. Una sola toma al día, te ayuda a mantener la actividad diaria.



Ilustración 6.1 Medicamento Rino-Ebastel

Persianas Efalú, una empresa jiennense, ofrece mosquiteras Mosquipolen. La promocionan así: Los exclusivos tejidos Mosquipolen funcionan como filtros contra la entrada de todo tipo de polen. Con una efectividad del cien por cien en la gran mayoría de tipos de polen y hasta de un 90.9% en aquellos de mayor densidad, Mosquipolen es un tejido cien por cien poliéster que entre otras muchas cosas le ayudará a mejorar su calidad de vida.



Ilustración 6.2 Empresa Persianas Efalú

Grupo Aldes. Empresa francesa internacional de propiedad familiar cuya actividad consiste en mejorar la calidad de vida de las personas en el interior de los edificios mediante soluciones integrales e innovadoras en materia de ventilación, purificación, calefacción, refrigeración y agua caliente sanitaria. Entre ellas ofrecen purificadores para un aire interior libre de alérgenos.

**Ilustración 6.3 Grupo Aldes**

Clínica Haloterapia Dani Reig en Jaén. Se promociona así: La haloterapia puede ser una alternativa muy eficiente contra las alergias primaverales. Durante estos días, de alta polinización, son muchos los pacientes que han podido comprobar su mejoría con esta terapia, ya que la limpieza de las vías respiratorias disminuye sus síntomas.

**Ilustración 6.4 Clínica Fisioterapia, Haloterapia y Osteopatía**

Recalcar que se han elegido estas cuatro empresas porque en concreto han sido estas cuatro las que en estos últimos años ya se han puesto en contacto conmigo interesándose por algún tipo de publicidad que se pudiera hacer aprovechando el seguimiento de personas alérgicas en nuestros canales de difusión.

Siendo Jaén una ciudad altamente perjudicada para los alérgicos al polen, en especial a aquellos que lo son al polen de olivo, su población es un atractivo evidente para las empresas como las que acabamos de citar en los ejemplos.

Esta es una de las principales razones por las que personalizar la plantilla final que se envía por redes sociales era uno de los objetivos de este proyecto. Por supuesto, la idea es que, siempre y cuando no se disponga de ninguna empresa interesada, se personalizará dando publicidad a las diferentes redes sociales de este sistema por donde se difunden los niveles para así dar más a conocer el trabajo que aquí se realiza.

6.2. Posible adaptación a otras ciudades

Tras la elaboración de este sistema en exclusiva para la ciudad de Jaén, se ha estudiado la posibilidad de poder ampliarlo en un futuro a otros puntos geográficos del país. Después de observar la red de captadores de polen de la Sociedad Española de Alergología e Inmunología Clínica (SEAIC) se ha comprobado la periodicidad y calidad de los datos que se publican para otras regiones españolas.

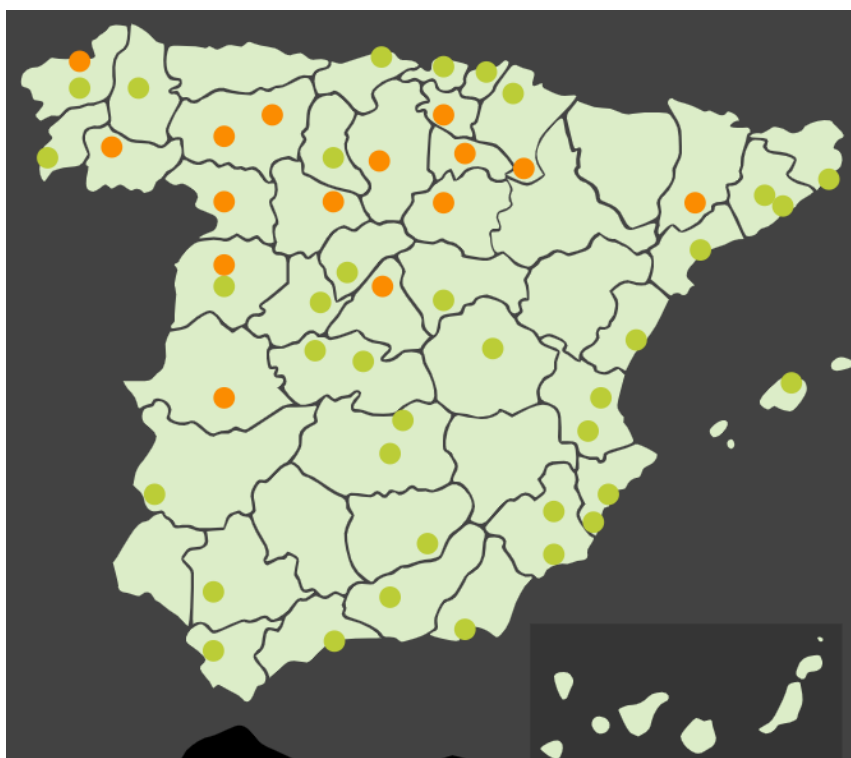


Ilustración 6.5 Mapa de captadores de SEAIC

Por desgracia, en la gran mayoría de ciudades que disponen de algún captador de polen, no se actualizan los datos de los niveles de polen diariamente y, además, en algunas ciudades simplemente publican los rangos sin cifras exactas (bajo, medio, alto). Aquí se muestran algunos casos de lo comentado.



INFORMACIÓN AEROBIOLÓGICA

Niveles de polen en la atmósfera:

Granada



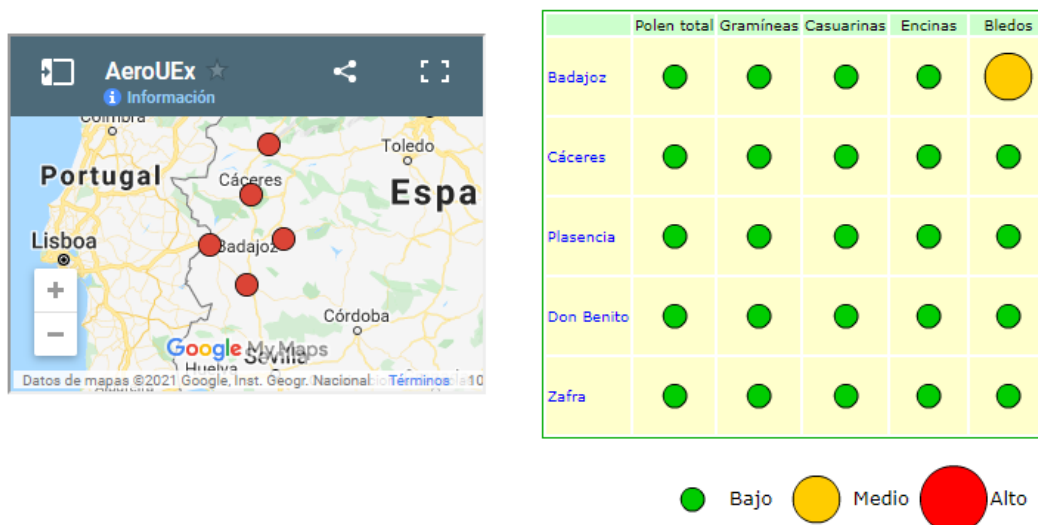
Acceso identificado

Semana	Artemisa	Casuarina	Cipreses	Encina	Gramíneas	Olivo	Palmera	Parietaria	Pinos	Plantagos	Platano de Sombra	Quenopodios
14 al 20 junio 2021	○	○	●	●	●	●	○	●	●	●	○	●
7 al 13 junio	○	○	●	●	●	●	○	●	●	●	○	●
31 mayo al 6 junio 2021	○	○	●	●	●	●	○	●	●	●	○	●
24 al 30 mayo 2021	○	○	●	●	●	●	○	●	●	●	○	●

Ilustración 6.6 Información del captador de Granada

		Niveles de polen facilitados por la Junta de Castilla y León Semana 17 al 24 de junio 2021	
GRAMÍNEAS		MODERADO	
CUPRESÁCEAS		BAJO	
OLIVO		BAJO	
LLANTENES		BAJO	
PLATANUS		BAJO	
ACEDERA		BAJO	
URTICACEAS		BAJO	

Ilustración 6.7 Información de los captadores de Castilla y León

Red Extremeña de Aerobiología**Ilustración 6.8 Información de los captadores de Extremadura****Ilustración 6.9 Información de los captadores de Castilla-La Mancha**

Dentro de las zonas que sí actualizan sus datos periódicamente destaca sobre todo la Comunidad de Madrid desde su web (www.comunidad.madrid/servicios/salud/polen). En esta comunidad disponen de 11 captadores adheridos a la red PALINOCAM (Red Palinológica de la Comunidad de

Madrid) repartidos en diferentes zonas. Los datos de las partículas recogidos por cada captador son estudiados diariamente, exceptuando los fines de semana y festivos, en sus respectivos laboratorios desde el 1 de enero hasta el 30 de junio. Tras este procedimiento, un boletín en formato pdf con el resumen diario es actualizado en su web, y además es enviado por correo electrónico a los usuarios que soliciten recibir esa información.

Información diaria por tipo de polen

RED PALINOCAM. ÚLTIMOS DATOS DISPONIBLES: 30 DE JUNIO 2021 (BOLETÍN DE PREDICCIÓN)

NIVELES BAJOS DE POLEN DE GRAMÍNEAS CON 20 GRANOS DE POLEN POR METRO CÚBICO DE AIRE

CON UN MÁXIMO DE 30 GRANOS DE POLEN EN LAS ROZAS

CON UN MÍNIMO DE 7 GRANOS DE POLEN EN ALCOBENDAS

NIVELES BAJOS DE POLEN DE DE OLIVO CON 2 GRANOS DE POLEN POR METRO CÚBICO DE AIRE

CON UN MÁXIMO DE 5 GRANOS DE POLEN EN ALCOBENDAS

CON UN MÍNIMO DE 0 GRANOS DE POLEN EN ALCALÁ DE HENARES

Las escalas para cada tipo de polen atienden únicamente a criterios aerobiológicos y son distintas para cada tipo de polen.

Ilustración 6.10 Información diaria de los captadores de la Comunidad de Madrid

Por tanto, en la Comunidad de Madrid sí tenemos inmediatez y exactitud en la publicación de los datos, aunque solo se pueden visualizar visitando el web o dándote de alta en las alertas por correo electrónico. Así pues, se podría estudiar el realizar un sistema como el desarrollado en este proyecto para esta Comunidad y que seguro sería de utilidad a muchos madrileños.

En Cataluña también disponen de una web donde consultar los datos de polen (<https://lap.uab.cat/aerobiologia/es>). Incluso disponen de una API (<https://lap.uab.cat/aerobiologia/es/api>) para solicitar y tratar dichos datos, sin embargo, éstos son actualizados semanalmente, por lo que carecen de utilidad real para los alérgicos.

6.3. Más redes sociales futuras

Ya se conoce que el mundo de las redes sociales está en constante evolución. Para este proyecto se ha optado por difundir los niveles de polen por las redes

sociales que más se cree que se adaptan a las necesidades del sistema y con las que se consigue mayor utilidad para las personas alérgicas. La acogida de estos años en las cuatro vías de comunicación ha sido más que aceptable por lo que se consideran elecciones acertadas.

Sin embargo, esto no tiene por qué ser siempre así. Con el tiempo puede aparecer alguna red social nueva que se adapte a lo que se busca, por lo que sería conveniente amoldar el sistema para que los niveles de polen se difundan también por ahí. A la misma vez, algunas de las redes escogidas actualmente pueden quedarse obsoletas y se podría decidir descartarla como vía de difusión del sistema.

6.4. Cambios en la web del Hospital de Jaén

Otra de las variaciones que pueden surgir en el futuro y que requerirían la readaptación del sistema sería la del cambio en la forma que el Hospital de Jaén publica los niveles en su web. De hecho, este ha sido uno de las grandes dificultades que me he encontrado durante el desarrollo del proyecto, pues en este último año la web donde se publicaban los datos y la forma de hacerlo se renovaron por completo. Esto se explicará más detalladamente en el siguiente capítulo.

7. CONCLUSIONES

Este último capítulo de la memoria va a servir para recapitular y razonar todos los resultados finales tras el análisis, diseño y desarrollo del sistema. Se contestará a preguntas tales como: ¿Se ha dado solución al problema de partida? ¿Se han cumplido las especificaciones del sistema? ¿Cuáles han sido las mayores dificultades encontradas durante el desarrollo?

7.1. Posibles errores que se pueden producir

Aunque se han empleado diferentes técnicas para detectar posibles errores durante la ejecución del sistema, soy consciente de que es inevitable y se escapa del control abarcado por este trabajo ciertos fallos producidos por agentes externos al sistema, en especial de la fuente origen de los datos: el Hospital de Jaén.

Se considera incontrolable pues, que en cualquier momento el Hospital de Jaén cambie de alguna forma su manera de publicar los datos, por lo que el sistema tendría que adaptar a los nuevos cambios su estrategia de captura.

Otro tipo de fallos que se pueden producir es a causa de las comunicaciones con las APIs de Facebook/Instagram, Twitter o Telegram. Al estar alojados en servidores externos, aunque es improbable, se pueden producir caídas y las conexiones con el sistema tendrían un resultado fallido.

Otro agente externo que interviene sería la nube de Amazon Web Services que, aunque también sea altamente improbable, no está exento a caídas de red o de disponibilidad.

Es por eso que, aunque la automatización de todo el proceso es la base del proyecto desarrollado, se requiere de un mínimo de intervención humana por parte del administrador para comprobar que todo funciona cada día como es debido, además de las herramientas de comunicación facilitadas por el sistema como son el canal privado de Telegram y las salidas por consola del ejecutable Python, que son de gran ayuda para el control y monitorización del sistema.

7.2. Dificultades encontradas

La gran dificultad que he tenido que afrontar durante el desarrollo del proyecto ha sido que, después de tener elaborada la parte de captura de los datos de niveles de polen por parte de la web del Hospital de Jaén, en abril de este año 2021 procedieron a cambiar y renovar la web por la que lo venían haciendo durante los últimos 10 años.



Ilustración 7.1 Comparativa de las webs de publicación de niveles de polen del Hospital de Jaén antigua y nueva

De hecho, a día de hoy aún se encuentra operativa la web antigua del Hospital de Jaén aunque sin ningún dato publicado, por lo que es posible visitarla o, incluso, echar un vistazo a su código fuente. También vuelvo a dejar aquí el enlace a la web actual para facilitar la comparativa entre ellas.

Web antigua (2011-2020):

<http://www.sspa.juntadeandalucia.es/servicioandaluzdesalud/chjtemas.php?id=1303>

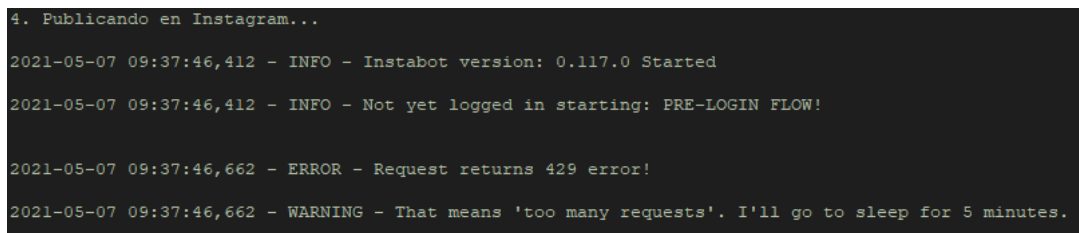
Web nueva (2021-actualidad):

<https://www.hospitaljaen.es/mediciones-de-los-niveles-de-polen-en-la-primavera-de-2021/>

Esto supuso un cambio radical a toda la estrategia diseñada y desarrollada para la captura de dichos datos y un retraso considerable a los plazos previstos con el sobreesfuerzo consiguiente requerido.

Otro problema al que me ha costado enfrentarme ha estado relacionado con la publicación de los niveles de polen a través del perfil de Instagram. Para ello utilizaba una librería Python llamada “instabot” que facilitaba el proceso considerablemente.

Durante las pruebas realizadas en mis equipos locales todo funcionaba perfectamente, pero me di cuenta que durante las pruebas en la instancia en la nube de AWS se producía una situación anómala que no conseguía solucionar.



```
4. Publicando en Instagram...
2021-05-07 09:37:46,412 - INFO - Instabot version: 0.117.0 Started
2021-05-07 09:37:46,412 - INFO - Not yet logged in starting: PRE-LOGIN FLOW!
2021-05-07 09:37:46,662 - ERROR - Request returns 429 error!
2021-05-07 09:37:46,662 - WARNING - That means 'too many requests'. I'll go to sleep for 5 minutes.
```

Ilustración 7.2 Salida por consola del error producido con la librería instabot en la instancia de AWS

El error consistía en una advertencia de que se realizaban demasiadas peticiones a la API de Facebook, cosa que no era cierta. Llegué a la conclusión de que la combinación del uso de dicha librería y la arquitectura de red provista por la instancia de AWS producía algún tipo de irregularidad incompatible con la API de Facebook.

La solución por la que opté fue desistir en el uso de la librería que ejercía de intermediaria 'instabot' y establecer mediante el código comunicación directa con la API Graph de Facebook. Opción que a la postre fue la acertada pues conllevó a la solución al problema.

Por supuesto, fueron otras muchas dificultades a las que me he tenido que enfrentar, pero se consideran menores y no tan relevantes como para tener que ser destacadas en este capítulo.

7.3. Conclusiones finales

Después de la finalización del análisis, diseño y desarrollo de este proyecto, se va a proceder a la redacción de unas breves líneas donde expongo algunas conclusiones finales desde una visión personal.

Se ha tratado de un TFG que me ha motivado bastante, ya que, cuando comencé en el año 2017 a publicar los niveles de polen de Jaén en una página de Facebook, en ningún momento pensé que eso sería la semilla que años después se convertiría en el fruto de este trabajo.

Aun así, me ha sido muy difícil compaginar su desarrollo con mis obligaciones laborales actuales, por lo que, como se esperaba, la planificación temporal plasmada en el correspondiente apartado de esta memoria ha quedado relegada a un segundo plano. Los resultados han sido semanas sin poder dedicarle tiempo, y otras, en las que ya se aproximaba la finalización de ciertos plazos, intensas en cuanto a su desarrollo.

Sin embargo, he acabado realmente satisfecho con los resultados obtenidos al finalizar este proyecto. Cumplidos todos y cada uno de los requisitos previos de una manera eficaz y creo que también eficiente. Todo se ha desarrollado tal y como lo tenía en mente cuando me decidí a realizar el TFG sobre esta temática.

Es por todas estas cosas que el sistema desarrollado cubre todas las necesidades que se le pretendían, y me ayudará considerablemente en los próximos años a seguir informando a los alérgicos de Jaén a estar informados de los niveles de polen en la provincia, fin último y principal de la realización de este proyecto.

He de decir también, que el hecho de enmarcar una idea propia dentro del contexto de un proyecto formal de ingeniería como este, desarrollando todas sus fases y su documentación, hace comprender un poco mejor la relevancia de la ingeniería en nuestro día a día y la importancia que tiene, y puede llegar a tener, en nuestras vidas.

Bibliografía

- [1] Wikipedia - *Desarrollo ágil de software*. (s.f.). Obtenido de [https://es.wikipedia.org/wiki/Desarrollo %C3%A1gil de software](https://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software)
- [2] Wikipedia - *Scrum (desarrollo de software)*. (s.f.). Obtenido de [https://es.wikipedia.org/wiki/Scrum \(desarrollo de software\)](https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software))
- [3] Radigan, D. (s.f.). *ATLASSIAN - Kanban*. Obtenido de <https://www.atlassian.com/es/agile/kanban>
- [4] Normon. (8 de Abril de 2016). Obtenido de <https://www.normon.es/articulo-blog/alergia-primaveral-sintomas-y-tratamientos>
- [5] Wikipedia - *Requisito funcional*. (s.f.). Obtenido de [https://es.wikipedia.org/wiki/Requisito funcional](https://es.wikipedia.org/wiki/Requisito_funcional)
- [6] Wikipedia - *Requisito no funcional*. (s.f.). Obtenido de [https://es.wikipedia.org/wiki/Requisito no funcional](https://es.wikipedia.org/wiki/Requisito_no_funcional)
- [7] Wikipedia - *Caso de uso*. (s.f.). Obtenido de [https://es.wikipedia.org/wiki/Caso de uso](https://es.wikipedia.org/wiki/Caso_de_uso)
- [8] Wikipedia - *Web scraping*. (s.f.). Obtenido de [https://es.wikipedia.org/wiki/Web scraping](https://es.wikipedia.org/wiki/Web_scraping)
- [9] Configuración para usar Amazon EC2. (s.f.). Obtenido de [https://docs.aws.amazon.com/es es/AWSEC2/latest/WindowsGuide/get-set-up-for-amazon-ec2.html#sign-up-for-aws](https://docs.aws.amazon.com/es_es/AWSEC2/latest/WindowsGuide/get-set-up-for-amazon-ec2.html#sign-up-for-aws)
- [10] Tutorial: Introducción a las instancias de Windows de Amazon EC2. (s.f.). Obtenido de [https://docs.aws.amazon.com/es es/AWSEC2/latest/WindowsGuide/EC2 GetStarted.html#ec2-getstarted-prereqs](https://docs.aws.amazon.com/es_es/AWSEC2/latest/WindowsGuide/EC2_GetStarted.html#ec2-getstarted-prereqs)
- [11] Stackoverflow - *¿Cómo ejecutar un script python a una determinada hora y fecha en Windows?* (s.f.). Obtenido de <https://es.stackoverflow.com/questions/60531/c%C3%B3mo-ejecutar-un-script-python-a-una-determinada-hora-y-fecha-en-windows>
- [12] lylinquiman. (s.f.). *Tutorialdeprogramacion - Creando bot de telegram con python*. Obtenido de <https://tutorialdeprogramacion.com/python-telegram-bot-cap-2-como-enviar-un-archivos-multimedia-con-un-bot-de-telegram/>
- [13] Learning, A. (14 de Febrero de 2021). *Youtube - Automate Facebook using python*. Obtenido de <https://www.youtube.com/watch?v=nPOqLZuuURs>
- [14] roja, v. (s.f.). *QA Stack - Token larga duración Facbook*. Obtenido de <https://qastack.mx/programming/12168452/long-lasting-fb-access-token-for-server-to-pull-fb-page-info>
- [15] Facebook for developers - *API Graph de Instagram*. (s.f.). Obtenido de <https://developers.facebook.com/docs/instagram-api/>

- [16] Mora, J. (19 de Junio de 2019). *Crear Aplicación en Twitter*. Obtenido de <https://jantoniomora.wordpress.com/2019/06/19/crear-aplicacion-en-twitter/>
- [17] Moya, R. (27 de Mayo de 2015). *Jarroba - Scraping en Python (BeautifulSoup)*. Obtenido de <https://jarroba.com/scraping-python-beautifulsoup-ejemplos/>
- [18] Shokeen, M. (4 de Abril de 2017). *Envatotuts+ - Raspando Páginas Web en Python Con BeautifulSoup*. Obtenido de <https://code.tutsplus.com/es/tutorials/scraping-webpages-in-python-with-beautiful-soup-the-basics--cms-28211>
- Bahit, E. (2012). Curso: Python para principiantes.
- F. Orlandi, B. Romano, M. Fornaciari, Effective pollination period estimation in olive (*Olea europaea* L.): a pollen monitoring application, *Scientia Horticulturae* 105 (2005) 313–318.
- A. Viney, J.F. Nicolás, N. Galindo, J. Fernández, V. Soriano-Gomis, M. Varea, Assessment of the external contribution to *Olea* pollen levels in southeastern Spain, *Atmospheric Environment* 257 (2021) 118481.
- GB, T. (6 de Octubre de 2019). Tony GB - Variables de entorno con Dotenv en Python. Obtenido de <https://medium.com/@tonybolanyo/variables-de-entorno-con-dotenv-en-python-4697540bf7e4>
- Git Connected Automating Instagram Posts with Python and Instagram Graph API. (s.f.). Obtenido de <https://levelup.gitconnected.com/automating-instagram-posts-with-python-and-instagram-graph-api-374f084b9f2b>
- Iglesias Morales, J. (19 de Junio de 2017). Youtube - Manipulación de Imágenes. Obtenido de <https://www.youtube.com/watch?v=zrjacprXmWY>
- Kuffo, L. (22 de Diciembre de 2019). Youtube - Python + Tweepy para extraer datos de Twitter. Obtenido de <https://www.youtube.com/watch?v=PUqPAM5Ect8>
- Mora, J. (28 de Junio de 2019). Jantoniomora - Publicando Tweets con Tweepy. Obtenido de <https://jantoniomora.wordpress.com/2019/06/28/publicando-tweets-con-tweepy/>
- Recursos Python - Añadir texto a una imagen con PIL / Pillow. (23 de Marzo de 2016). Obtenido de <https://recursospython.com/guias-y-manuales/anadir-texto-imagen-pillow/>