

PROYECTO DE COMPUTACION II

Memoria Final

SILENT LIFE

Dirigido por:

Alfonso Vilchez De Las Heras

Repositorio GitHub

<https://github.com/sandrokiller54/PC3-ANGULAR>

Realizado por:

Yoseph Daniel Sabbagh, N° EXP 21714038

Alberto González Fernández, N° EXP 21719077

David Vázquez Argibay, N° EXP 21839227

Sandro Santino Zanata Ortega, N° EXP 21838989

Introducción	3
Antecedentes	3
Estado del arte	4
Metodología de trabajo	5
Objetivos	5
Descripción de las fuentes de datos	6
Descripción del Front End	7
Descripción del Back End	8
Presupuesto	9
Diagrama de Gantt	10
Conclusiones	10
Manual de instalación	10
Bibliografía	21

Introducción

Tras la pandemia de la Covid_19, la mayoría de la población residente en madrid busca un entorno en el que teletrabajar fuera de la comunidad, permitiendo así un cambio de entorno y de ámbito por el que se pueda estar más relajado. No solo el proyecto se centra en el ámbito laboral, también tenemos en cuenta las vacaciones que tienen los empleados, que son desaprovechadas por culpa de no poder salir de la propia comunidad. El principal problema que nos encontramos es la cantidad de población que no disponen de una residencia fuera de dicha comunidad, sin tener un lugar de referencia al que viajar. Algunas de las soluciones que se han encontrado son turas y circuitos para diferentes pueblos abandonados, lo cual ofrece una variedad de actividades y recorridos, pero eso no valdría del todo porque la mayoría de la población necesita un sitio donde hospedarse y poder teletrabajar.

Con el nacimiento de *Silent Life*, se trata de informar al usuario sobre los diferentes lugares donde hay despoblación y, mediante un buscador, se tratara de encontrar el sitio al que el usuario desea visitar, pudiendo ver las diferentes casas rurales, parcelas, etc de la zona, de las que podremos ver sus características

También se le ofrece al usuario, si no tiene un destino claro, un formulario que podrá rellenar y así obtener de la página web recomendaciones con las especificaciones que ha introducido.

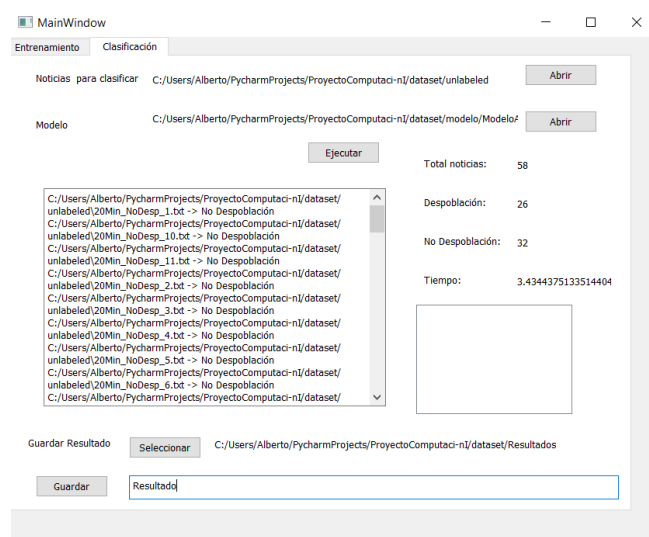
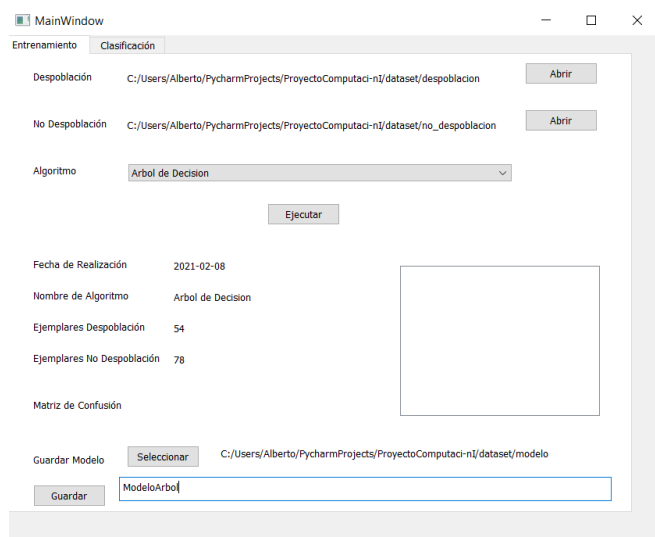
Antecedentes

Para esta segunda etapa del proyecto, tenemos lo ya desarrollado anteriormente, un clasificador de noticias de despoblación, que usaremos como una de las fuentes de datos para el producto final.

Dentro de nuestro clasificador tenemos incluidas noticias de despoblación y no despoblación para realizar el entrenamiento del modelo. Para conseguir este entrenamiento, hemos implementado seis algoritmos diferentes, Decision Tree, Naive Bayes, Random Forest, KNN, Logistic Regresion y SVC, de los cuales el usuario puede elegir cuál desea emplear.

Una vez entrenado, el usuario introduce en la aplicación las noticias que desea clasificar y recibe un documento txt con el nombre de la noticia y si ha sido clasificada como despoblación o no despoblación.

Sobre este funcionamiento, tenemos implementada una interfaz gráfica usando PyQt.



Estado del arte

Antes de decidir la idea final del proyecto se investigó el campo relacionado con guías turísticas online y páginas de alquileres de pisos.

Para centrarnos un poco en el ámbito en el que se va desarrollar el trabajo, podemos citar un documento publicado en la web ieeexplore.ieee.org el cual hace referencia a la clasificación de noticias mediante la utilización de Deep Learning.

Más concretamente, la clasificación la realiza en categorías (Deportes, Política, Economía) y en alcance (Local, Nacional e Internacional). Este clasificador emplea una Red Neuronal Artificial de múltiples clases con la cual consigue un porcentaje de éxito general tanto para categoría como para alcance del 83,75%

Otro de los ejemplos que podemos ver sería un tfg encontrado en la Universidad Politécnica de Valencia el cual basa sus esfuerzos en el desarrollo de un clasificador para el ámbito laboral, pudiendo así mejorar la eficiencia de la clasificación de artículos en las páginas webs de las empresas que lo empleen.

Dado que existen pocos proyectos relacionados con clasificadores de texto, al menos españoles, nos centraremos en “web scraping”.

Si nos damos cuenta, actualmente la mayoría de las páginas webs de internet no ofrecen la opción de rescatar datos de interés para poderlos guardar o reutilizar de manera fácil, quitando la típica manera de copiar y pegar.

Web Scraping automatiza este proceso y ayuda a almacenar todos los datos que necesites con facilidad y en un tiempo considerablemente aceptable, para ello muchos profesionales utilizan esta técnica.

Actualmente en todo lo relacionado con la ciencia de los datos, para hacer cualquier cosa es necesario tener mucha cantidad de datos a mano.

El web scraping recopila y clasifica todos estos datos en una ubicación accesible.

Ejemplos en los que se usa permanentemente web scraping:

- Marketing de contenidos: podemos desarrollar un programa que haga “scrapeo” de datos concretos de una web para posteriormente poder generar nuestro propio contenido a partir de la anterior web, datos mucho más precisos.
- Generar visibilidad y controlar nuestra marca en internet, a través del “scrapeo”, fácilmente podemos automatizar el posicionamiento de nuestra web en google o por ejemplo controlar la presencia de nuestra marca en determinados foros populares como puede ser forocoches.

Existen dos herramientas online para realizar el web scraping:

- Webscraper.io: plugin para el navegador de Google.
- Import.io: Construido sobre la base del software open source Chromium, el motor de Google Chrome

Metodología de trabajo

Para llevar a cabo este proyecto, se ha empleado una metodología de trabajo Scrum, ya que es la metodología que más se adapta a nuestra situación y forma de trabajo, siendo el cliente tanto Javier Sánchez Soriano, como Alfonso Vílchez De Las Heras, ya que son los dos profesores a los que tendremos que entregar el proyecto, cada uno con diferentes requisitos.

Se han empleado las horas de clase para hacer una breve reunión de no más de quince minutos para comprobar qué se ha hecho y qué está por hacer y se ha usado el resto del tiempo, a parte del trabajo fuera de las horas de clase, para desarrollar todo lo necesario del proyecto. A parte, los fines de semana también se han seguido estas reuniones, aunque con menos frecuencia, una cada dos semanas.

Las entregas con el cliente han sido diferentes entre Javier Sánchez Soriano y Alfonso Vílchez De Las Heras. Las entregas realizadas al primer cliente han sido tres, el anteproyecto para presentar la idea al cliente, y dos hitos que muestran el avance del proyecto antes de la entrega final, mientras que al segundo cliente solicitó un mayor número de entregas, el estado y planificación del proyecto en su fase inicial, el estudio de las fuentes de datos a explotar, los casos de uso de la aplicación junto a su planificación de desarrollo, los mockups y esquema de la base de datos a emplear y dos hitos de evolución del proyecto para comprobar el estado del desarrollo hasta su entrega final.

Objetivos

Objetivo General

El objetivo general del proyecto es la creación de una aplicación web informativa que facilite al usuario la búsqueda de la oferta tanto de ocio, sociocultural, vivienda, etc.. de las zonas afectadas por la despoblación en España.

Objetivos Específicos

1. Permitir realizar búsquedas a usuarios registrados y no registrados.
2. Facilitar un cuestionario de tal manera que ayude al usuario en la búsqueda de su destino.
3. Almacenar un histórico de búsquedas y favoritos para los usuarios registrados.
4. Facilitar un listado de lugares más valorados, búsquedas recientes...
5. Permitir la introducción de reseñas a cada una de las ofertas proporcionadas.
6. Habilitar una selección de filtros para los resultados generales de la primera búsqueda.
7. Facilitar medios de transporte disponibles a la localización solicitada.

Objetivos no Contemplados

1. No se pretende ser una página web de gestión de reservas, simplemente tenemos un carácter informativo.

Descripción de las fuentes de datos

Para obtener los datos necesarios para sustentar nuestra aplicación han sido tres, como se pedía de requisito por el cliente.

La página web <https://15mpedia.org/> se ha usado como fuente para obtener, mediante web scraping. De aquí se obtiene información valiosa para la aplicación, sobre todos los municipios de España, como el nombre del municipio, los nombres de la comarca, provincia y comunidad autónoma en la que se encuentra, y la población y densidad de este municipio.

La página web <https://www.spainhouses.net/> se ha usado como fuente para obtener, mediante web scraping, toda la información necesaria referente a los diferentes anuncios de alquileres a corto plazo, para poder mostrarselos al usuario desde la aplicación.

Se obtiene diferente información de la vivienda, la ciudad o pueblo en la que se encuentra, los metros cuadrados que abarca, una breve descripción escrita por el anunciante o propietario de la casa y su número de teléfono de contacto, el coste por metro cuadrado y mensual y, por último, una lista de las imágenes que haya proporcionado el anunciante.

La página web <https://www.escapadarural.com/> se ha usado como fuente para obtener datos mediante scraping. Muy similar a la página web anterior, *Escapada Rural* se usa como medio para obtener más anuncios, por lo que se obtienen los mismos datos, el lugar donde se encuentre la vivienda, la descripción proporcionada por el anunciante, el precio mensual del alquiler, las especificaciones de la propiedad y las imágenes que se encuentren en la plataforma.

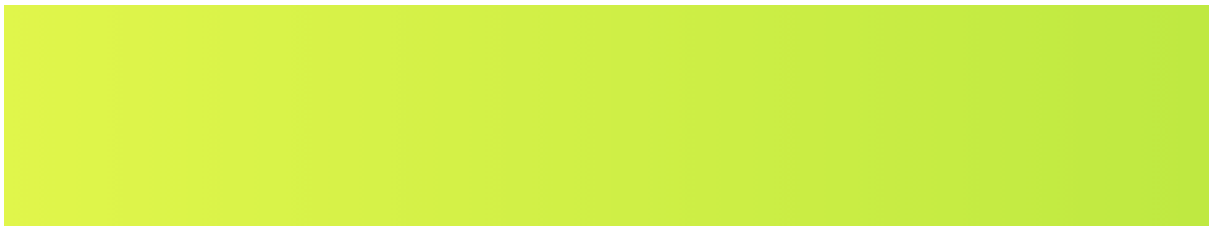
Descripción del Front End

Para llevar a cabo el desarrollo del Front End, se ha usado el framework de Angular. Se trata de un framework para desarrollo web de código abierto mantenido por Google que actualmente se encuentra en auge, ya que, actualmente, Angular se usa para la implementación global de la productividad y la infraestructura escalable que admite las aplicaciones más grandes de Google.

La principal ventaja de Angular es que se usa solo una pantalla para mostrar todo el contenido al usuario, agrupado en componentes, lo cual hace que el desarrollo sea muy ágil y haya una gran velocidad de carga para el usuario.

El web scraping para extraer la información se ha realizado mediante scripts de Python haciendo uso de varias librerías. La primera es la librería *requests*, empleada para realizar una petición HTTP y obtener en código HTML completo de la página web deseada. Una vez obtenido el HTML, se hace uso de la segunda librería, *BeautifulSoup*, usada para extraer la información necesaria para la web. El siguiente paso es hacer uso de la tercera librería, *re*, un paquete incorporado en Python para hacer uso de RegEx y así poder limpiar la información extraída. Y, por último, haciendo uso de la última librería *mysql.connector*, empleada para hacer una conexión con la base de datos de la página web, una vez lista la información, se introduce en la base de datos.

En toda la aplicación se ha usado un color degradado de amarillo a verde:



Y el código implementado en css de dicho degradado es:

```
background: linear-gradient(to right, #f9ff52 0%, #b8e740 79%, #b8e740 91%);
background: -moz-linear-gradient(left, #f9ff52 0%, #b8e740 79%, #b8e740 91%);
background: -webkit-gradient(left top, right top, color-stop(0%, #f9ff52), color-stop(79%, #b8e740), color-stop(91%, #b8e740));
background: -webkit-linear-gradient(left, #f9ff52 0%, #b8e740 79%, #b8e740 91%);
background: -o-linear-gradient(left, #f9ff52 0%, #b8e740 79%, #b8e740 91%);
background: -ms-linear-gradient(left, #f9ff52 0%, #b8e740 79%, #b8e740 91%);
background: linear-gradient(to right, #f9ff52 0%, #b8e740 79%, #b8e740 91%);
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#f9ff52', endColorstr='#b8e740', GradientType=1 );
```

Se han usado estos colores como base de la página web, ya que el color amarillo se relaciona siempre con la alegría, felicidad, positividad y optimismo, sensaciones que representa la plataforma, ya que son sentimientos que son comunes de experimentar cuando se está relajado de vacaciones. A su vez, también se logra tranquilidad y calma estando de vacaciones, sentimientos que transmite el color verde, de ahí la mezcla de los dos. También se ha usado un recurso de degradado, algo poco empleado en páginas web, de ahí que se use para descartar y sorprender al usuario.

La fuente empleada en toda la página web es *Roboto*, ya que, como la describe el propio Google, se trata de una fuente moderna pero accesible y emocional.

En la primera pantalla, la que encuentra el usuario al acceder a la web, se puede ver un formulario en la parte central, y, en el menú de la parte de arriba, puede visitar diferentes pestañas y registrarse si desea crear una cuenta o hacer login si ya tiene una cuenta creada. El funcionamiento de este registro y login se tratará más adelante.

La segunda pantalla, la de blog, será una pantalla informativa sobre nosotros y nuestro proyecto.

En la tercera pantalla, se le muestran al usuario los diferentes anuncios que hay registrados en la base de datos. También tiene la opción de rellenar un formulario si no tiene claro dónde ir. De esta forma se hará un filtro con las opciones que haya introducido.

Descripción del Back End

En la parte de Back End, el cliente solicitó que se desarrollará con el framework *Laravel*, a diferencia de la parte Front End, en la que se tenía libertad para emplear lo que deseara el equipo de desarrollo.

Laravel es un framework con un uso muy extenso basado en PHP que permite el desarrollo de aplicaciones web, y el principal motivo por el que el cliente nos pidió este framework tiene un sistema de organización por carpetas, evitando así un código confuso, y la facilidad que ofrece a la hora de crear APIs para consultas a la base de datos.

La base de datos final que se utiliza en la página web ha sufrido varias modificaciones desde el primer diseño hasta el presente, ya que se han ido cortando y ajustando funcionalidades.

Las tablas que se utilizan de esta base de datos son *users*, en la que se guardan toda la información de los usuarios registrados en la aplicación, la ID que le asigna la base de datos, el nombre, el correo electrónico, la fecha de verificación para comprobar si el usuario ha verificado su correo electrónico o no, la contraseña de acceso, el token para recuperar contraseña, la fecha en la que se ha creado y la fecha en la que se ha modificado.

La tabla *alquileres* almacena toda la información referente a los anuncios en la web, la ID que le da la base de datos, el nombre del anuncio y una descripción, la ubicación en la que se encuentra, las fotos que el anunciante haya subido, el coste y las especificaciones de la vivienda que se anuncia.

La tabla *localidades* almacena la ID que le da la base de datos a la localidad, el nombre del municipio, la comarca, provincia y comunidad autónoma a la que pertenece el municipio, la población que habita y la densidad de población.

La tabla migrations es generada y usada por Laravel para hacer una especie de control de versión de esta, si se hace una modificación a la base de datos, las migraciones de laravel ayudan a actualizar todos los sistemas que lo usan a esa nueva versión de la base de datos.

Por último, la tabla *usuario_alquileres* es usada para guardar la unión entre el usuario y el alquiler y almacenar esta información en la base de datos.

Con respecto a la API, localizadas dentro del directorio Routes, al principio del desarrollo se crearon más llamadas de las que se han acabado usando, por lo que se han limpiado y se han dejado las esenciales.

También se han implementado llamadas a la api para usuarios y alquileres específicos, filtrando por la ID del que se desea buscar.

Se han creado el resto de controladores de las llamadas de la API en su respectivo directorio App\Http\Controllers.

Por último, se han actualizado los modelos en el directorio App\Http\Models para ajustarse a los cambios y recortes que se han realizado dentro de la base de datos.

A la hora de hacer uso del registro y el login de usuarios, se ha empleado *JWT*, es un estándar abierto que define una forma compacta y autónoma la manera de transmitir información de forma segura entre las partes como un objeto JSON.

La web se comunica con el Back End que genera tokens, que pueden ser enviados al Back End como autenticación para comprobar si están o no autenticados en la plataforma.

Para satisfacer el requisito de los clientes de tratar el tema de la España vaciada, se ha seguido el mismo planteamiento que en la primera fase del proyecto.

Lo que se ha hecho ha sido, una vez el usuario ha seleccionado el municipio que desea visitar, se hace una consulta a la página web del periódico *20Minutos* introduciendo el nombre de dicho municipio. De esta consulta se recogen las noticias que salen y se limpian hasta tener solo el texto de la noticia.

Una vez llegado a este paso, se hace uso de un modelo ya entrenado para hacer la comparación con las noticias recogidas y ver si éstas tratan de despoblación o no.

Presupuesto

Integrantes	€/Hora	Fase Entrenamiento	Fase Proyecto	Fase Documentación	Coste Sin IVA	Coste Total (IVA 21%)
Dani	80	2.400,00 €	9.775,20 €	540,00 €	12.715,20 €	15.385,39 €
David	80	2.400,00 €	9.775,20 €	540,00 €	12.715,20 €	15.385,39 €
Sandro	80	2.400,00 €	9.775,20 €	540,00 €	12.715,20 €	15.385,39 €
Alberto	80	2.400,00 €	9.775,20 €	540,00 €	12.715,20 €	15.385,39 €
Todos	80				50.860,80 €	61.541,57 €

Diagrama de Gantt

Debido a que el Diagrama de Gantt es demasiado extenso para introducirlo en el documento, se puede ver haciendo click [aquí](#).

También se puede consultar en el fichero adjunto a esta documentación titulado Diagrama_Gantt_Grupo_4.

Conclusiones

La página web de *SilentLife* nace para facilitar información a los usuarios sobre posibles lugares en los que pueden ir a alejarse de la ciudad y, ya sea para ir a teletrabajar a una localización mucho más tranquila, o bien para ir de vacaciones, nuestra página web es idónea para ellos, ya que a parte de todo esto, otro de nuestros objetivos es frenar la incontrolable despoblación que sufren estos pueblos.

Como trabajos futuros, hay varias posibles mejoras en la plataforma. Se podría hacer el scrapeo de más páginas web para aumentar el número de anuncios almacenados en la base de datos.

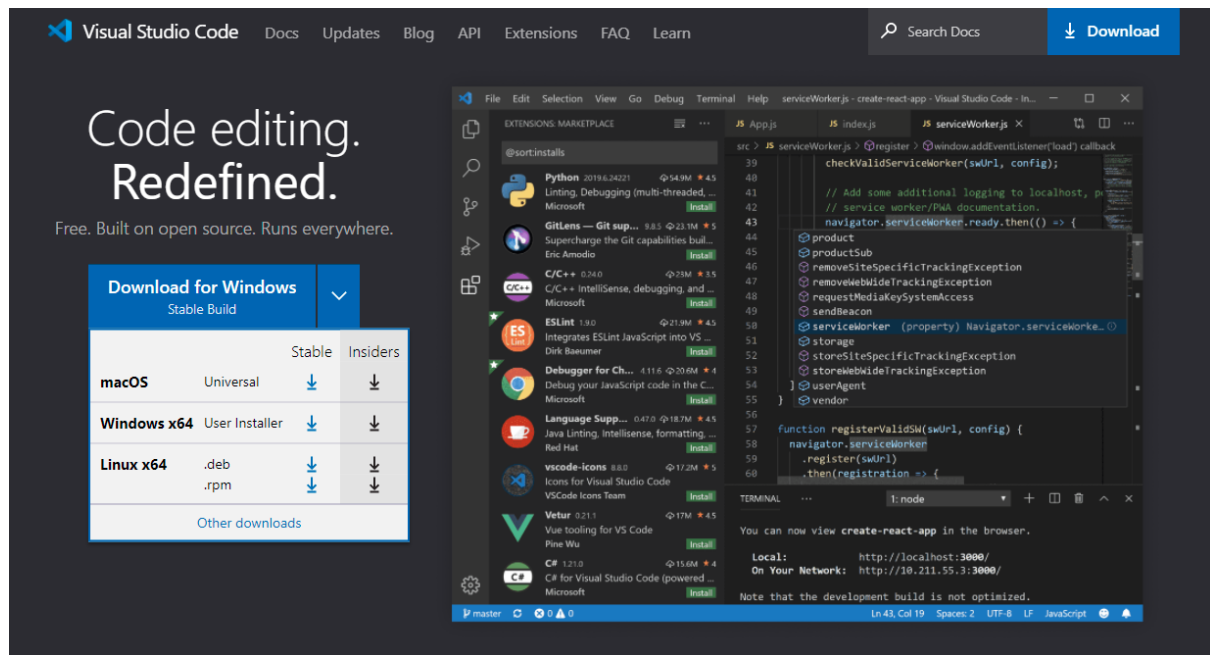
También se podría hacer el cambio de planteamiento de la web, ya que, aunque para la entrega con los clientes se ha acordado que el objetivo final es informar al cliente de los anuncios, en el futuro sería posible implementar un sistema de contacto con el anunciante y trámite de alquiler, para poder realizar todo en el mismo lugar, lo cual es siempre más cómodo para el usuario.

Manual de instalación

Descarga de entorno de desarrollo

El primer paso y más importante a la hora de realizar la instalación de nuestro proyecto es la descarga de un entorno de programación. El recomendado es Visual Studio Code, ya que con su terminal integrada, nos facilitará los próximos pasos.

Para ello, debemos ir a la página principal de [Visual Studio Code](#). Una vez dentro, lo primero que vemos es un botón azul que nos dice “*Descargar para Windows*”. Si no tenemos Windows, también lo tenemos disponible para Mac y Linux.



Siempre recomendamos descargar la versión estable, ya que es mucho más raro que tenga errores, a diferencia de la versión para iniciados, que se actualiza cada poco tiempo y puede contener bugs.

Una vez que descargamos el instalador que deseemos, lo ejecutamos, aceptamos los términos de licencia del software y pulsamos el botón siguiente.

En la ventana de tareas adicionales, además de dejar marcada la opción para agregar a path el visual studio code, marcaremos las dos acciones de "Agregar con code". La opción de crear un acceso directo en el escritorio la puede marcar si lo desea.

Una vez que hayamos hecho todas las selecciones, hacemos click en siguiente y en instalar en la próxima ventana para comenzar a instalar el entorno. Una vez haya acabado la instalación hacemos click en finalizar.

Para el último paso, daremos click izquierdo en el acceso directo a Visual Studio Code, iremos a propiedades, compatibilidad, y seleccionaremos la casilla de ejecutar este programa como administrador.

Seleccione las Tareas Adicionales

¿Qué tareas adicionales deben realizarse?



Seleccione las tareas adicionales que desea que se realicen durante la instalación de Visual Studio Code y haga clic en Siguiente.

Accesos directos adicionales:

☐ Crear un acceso directo en el escritorio

Otros:

☒ Agregar la acción "Abrir con Code" al menú contextual de archivo del Explorador de Windows

☒ Agregar la acción "Abrir con Code" al menú contextual de directorio del Explorador de Windows

☐ Registrar Code como editor para tipos de archivo admitidos

☒ Agregar a PATH (disponible después de reiniciar)

< Atrás

Siguiente >

Cancelar

Instalación de Xampp

Una vez instalado el entorno de programación, lo siguiente que instalaremos será la herramienta Xampp, la cual nos ayudará a gestionar un servidor local y la base de datos. Para esto, debemos ir a la página de [descarga](#) de la herramienta. Al entrar nos encontramos con que, como anteriormente, Xampp también tiene varios instaladores dependiendo del sistema operativo, por lo que debemos seleccionar del que se disponga.



XAMPP Apache + MariaDB + PHP + Perl

¿Qué es XAMPP?

XAMPP es el entorno más popular de desarrollo con PHP

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.



XAMPP

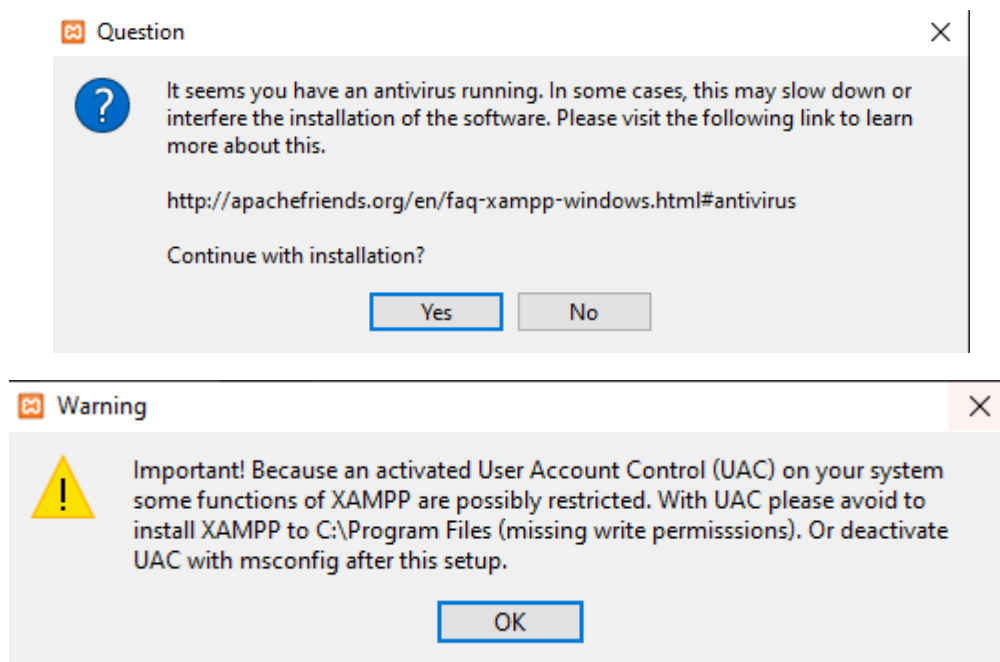
Descargar
 Pulsa aquí para otras versiones


XAMPP para Windows
 8.0.6 (PHP 8.0.6)


XAMPP para Linux
 8.0.6 (PHP 8.0.6)

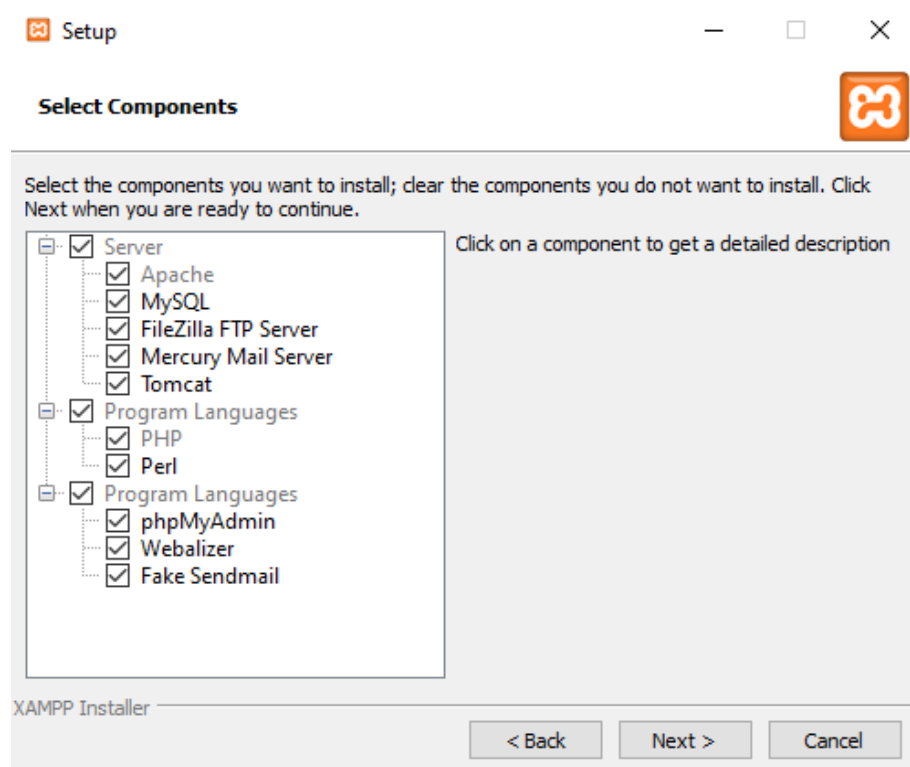

XAMPP para OS X
 8.0.6 (PHP 8.0.6)

Cuando tenemos el instalador descargado lo ejecutamos. Dependiendo del ordenador que se tenga y si se tiene un antivirus activo que no sea Windows Defender, Xampp puede avisar que la descarga podría ir más lenta si no se desactiva, pero nos da la opción de continuar sin hacer ningún cambio, por lo que le damos a “Yes” y a “OK”.



Si estos avisos no nos aparecen seguiremos la instalación de la misma forma.

Una vez hecho esto, nos aparecerá el instalador de Xampp, en la que daremos a “Next” dos veces, la primera para empezar la descarga y la segunda para elegir los componentes, pero para simplificarlo, se descarga todo lo que viene marcado. elegiremos dónde queremos descargarlo. Seguidos estos pasos, ya lo tendríamos correctamente instalado.



Instalación de Laravel

Laravel no tiene instalador como hemos visto antes, deberemos usar [Composer](#). Para descargarlo en Windows, hacemos click en el botón “Download”, que nos lleva a otra página donde podremos descargar Composer clicando en *Composer-Setup.exe*.



A Dependency Manager for PHP

Latest: **2.0.13** ([changelog](#))

Composer 2.0 is now available! [Read our announcement!](#)

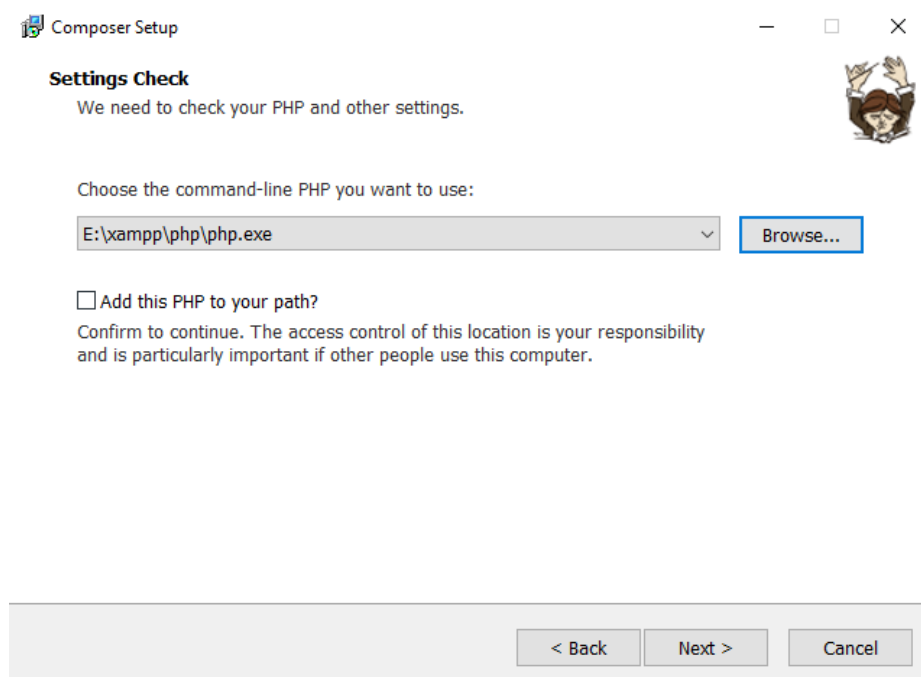
[Getting Started](#)[Download](#)[Documentation](#)[Browse Packages](#)[Issues](#)[GitHub](#)

Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

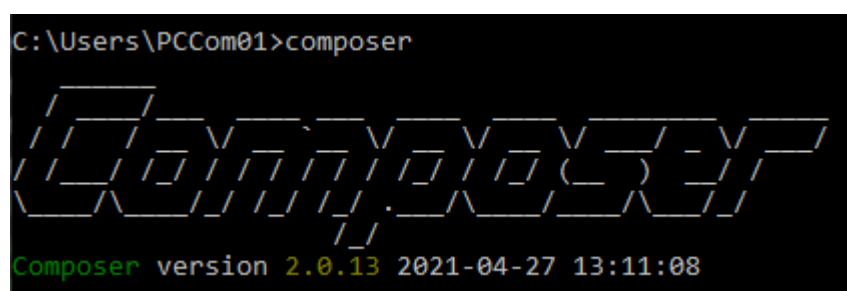
Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

Una vez descargado, al ejecutarlo, seleccionamos instalarlo para todos los usuarios, y a continuación, cuando nos da la opción de modo desarrollador, lo dejamos sin marcar y le damos a “Next”. Lo siguiente que nos pide es elegir el intérprete de PHP que queremos usar. En nuestro caso, debemos darle a “Browse”, y buscar el .exe del Xampp, hay que buscarlo en la carpeta en la que lo hemos descargado anteriormente. Si hemos usado la ruta por defecto, sería C:\xampp\php\php.exe.



Hacemos click en “Next” para continuar con la instalación. Nos preguntará por información del proxy, en el caso de una conexión normal no es necesario, por lo que pasamos a instalar Composer y finalizar el instalador.

El próximo paso, una vez instalado composer, sería abrir el símbolo del sistema y verificar que está correctamente instalado escribiendo *composer*. Si nos dice que no se reconoce como comando interno, la instalación del Composer no se ha realizado correctamente. De lo contrario, escribiremos *composer global require laravel/installer*. Nos saldrán por pantalla muchos *installing*, por lo que esperaremos a que termine. Si no nos da ningún error, ya tendríamos Laravel correctamente descargado.



Podemos comprobarlo de la misma forma que con el composer, si escribimos laravel y no nos dice que no se reconoce como comando interno, está instalado correctamente.

Instalación de Angular

Para instalar Angular, primero debemos tener instalado [Node.js](#) en nuestro equipo, por lo que tendremos que ir a la página de descarga. Para Windows podemos descargar la versión estable y la no estable. Nosotros recomendamos la estable por el mismo motivo que con Visual Studio Code, ya que la estable tiene mucha más fiabilidad.



Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome.

New security releases now available for 15.x, 14.x, 12.x and 10.x release lines

Descargar para Windows (x64)

14.16.1 LTS

Recomendado para la mayoría

16.1.0 Actual

Últimas características

[Otras Descargas](#) | [Cambios](#) | [Documentación del API](#) [Otras Descargas](#) | [Cambios](#) | [Documentación del API](#)

Una vez descargado y ejecutado el instalador, dejaremos todo por defecto, solo tendremos que aceptar los términos de la licencia y la ruta donde queramos instalar Node.js. Se nos abrirán varias ventanas de instalación, solo tendremos que esperar a que acabe el proceso. para comprobar la correcta instalación, podemos usar el comando `node -v` en el símbolo del sistema. Si nos dice que nos se reconoce como comando interno, igual que nos pasaba con las otras instalaciones, se ha producido algún error en el proceso.

```
C:\Users\PCCom01>node -v  
v14.15.1
```

Una vez concluida la instalación, tenemos que instalar el resto de componentes que nos faltan. Esto lo podemos hacer desde el símbolo del sistema de Windows, o desde la terminal de Visual Studio Code, no hay ningún cambio, se usarán los mismos comandos. Primero usaremos el comando `npm install -g @angular/cli` para instalar el cliente de angular en nuestro equipo. Tendremos que esperar a que se instale. Nos preguntará si queremos compartir los datos de cómo hacemos uso de Angular, esta opción depende de lo que quiera el usuario.

Una vez respondido, concluirá la instalación de Angular.

Obtener el programa

Una vez tengamos todo lo anterior, lo próximo será descargar nuestro programa para poder ejecutarlo. Haremos uso del programa Git, para poder hacer la descarga de nuestro repositorio desde Visual Studio Code.

Descarga Git

Para descargar [Git](#), iremos a su página web. Una vez dentro, seleccionaremos el sistema operativo que deseemos, en este caso Windows.



Una vez seleccionado windows, seleccionaremos dónde queremos guardar nuestro instalador. Una vez descargado, lo ejecutamos. Tendremos que aceptar terminos y condiciones de la licencia, seleccionar componentes, que dejaremos tal cual viene, elegir el editor por defecto, esta opción depende de cada usuario de elegir la que más le convenga, y seguiremos dando a next, ya que el resto de opciones deberemos dejarlas tal cual vienen por defecto. Una vez acabado, instalamos y tendríamos completado este paso.

Git Clone y preparación

Con el Git instalado, tendremos que abrir Visual Studio Code y pulsar a la vez las teclas CTRL, SHIFT y P, para abrir un diálogo emergente en el que escribiremos Git: Clone, pulsaremos la tecla ENTER, y nos pedirá la dirección del repositorio del que se quiere clonar. En este apartado se debe poner el link a nuestro [repositorio](#) y darle a ENTER otra vez.

Se nos pedirá elegir dónde queremos guardar el proyecto, esto depende de cada usuario. Cuando se seleccione la carpeta haremos click en elegir la ruta, Visual Studio Code nos pedirá hacer login con nuestra cuenta de Github, si no se dispone de una es necesario crearla. Una vez completado esto, se comenzará a descargar, esto puede tardar varios minutos dependiendo de la velocidad de internet de cada usuario. Nos saldrá una ventana emergente en la esquina inferior derecha, en la que nos pregunta si queremos abrir este nuevo proyecto, por lo que haremos click en "Open".

Una vez que tenemos el código descargado, tendremos que introducir unos comandos en la consola de Visual Studio Code ya que aunque subamos el proyecto completo a Github, hay carpetas que no se suben por seguridad y hay que migrarlas.

Lo primero será entrar en la carpeta de angular, llamada *“pc3”*, para ello, escribiremos en la consola *cd pc3* y una vez dentro, escribiremos *npm install*. Esto nos empezará a descargar y tardará un par de minutos.

Cuando acabe, saldremos de la carpeta de angular usando *cd..* y entraremos en la carpeta de laravel con *cd pc3-laravel*. Dentro de esta carpeta, nuestro siguiente comando será *composer update* para actualizar composer. Si este comando nos da error, usaremos *composer update --ignore-platform-reqs* para forzar la actualización de composer.

Una vez dentro, escribiremos *cp .env.example .env* para crear el archivo .env.

Lo siguiente es escribir *composer update* y esperar a que termine de actualizar. Si nos da error hay que usar el comando *composer install*.

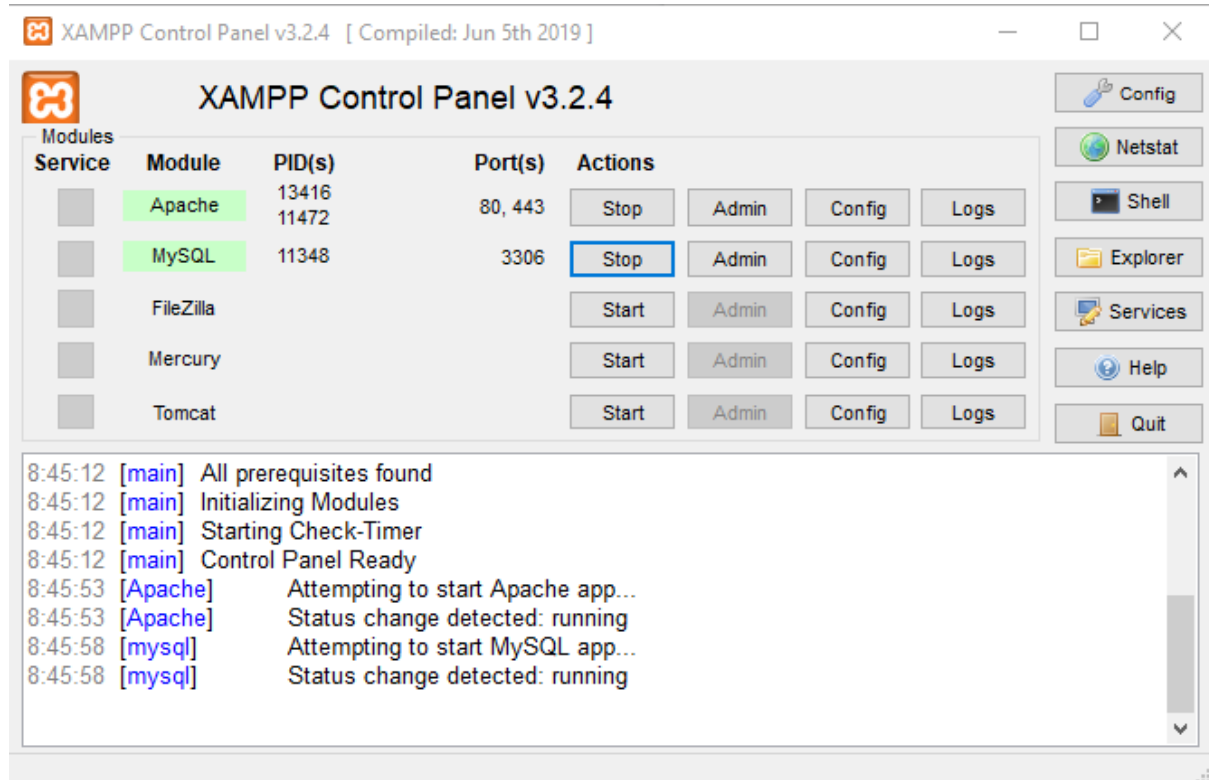
A continuación generamos la key de la API con el comando *php artisan key:generate*.

Una vez completados todos estos pasos, tendremos nuestro proyecto listo para ejecutarse.

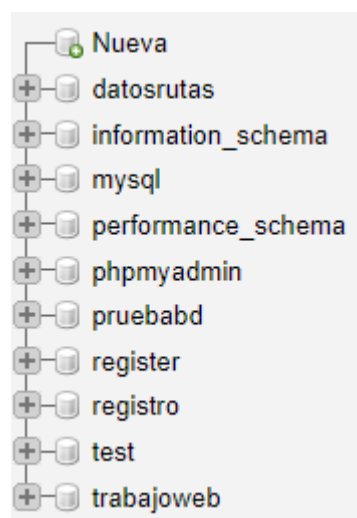
Ejecutar el programa

Para lograr la ejecución del programa debemos seguir unos pasos previos para su correcto funcionamiento.

Primero debemos abrir la herramienta Xampp, y hacer click en los botones “Start” en el módulo de Apache y de MySQL. Tendremos que esperar a que se pongan en verde los módulos, para saber que se han iniciado correctamente.





Una vez hecho esto, haremos click en el botón “Admin” del módulo de MySQL, para que nos lleve al administrador de bases de datos. Una vez dentro de esta página, veremos que a la izquierda nos sale una lista con varios nombres. Lo que deberemos hacer es hacer click en *Nueva*, para crear una nueva base de datos.



Dentro de esta opción, sólo tendremos que poner el nombre a la base de datos de “*silentlife*”. Tiene que ser ese nombre sin ninguna modificación, ya que es el nombre que se

le tiene configurado en nuestro proyecto para que sea la base de datos, cualquier otro nombre no funcionara.

Hacemos click en “*Crear*” y ya habremos creado nuestra base de datos.

 **Crear base de datos** 

<input type="text" value="silentlife"/>	<input type="text" value="utf8mb4_general_ci"/>	<input type="button" value="Crear"/>
---	---	--------------------------------------

Lo último que nos queda es cargar los servidores de Laravel y Angular. Para ello, usando Visual Studio Code, entraremos en el directorio donde tenemos nuestro proyecto, de ahí, usaremos *cd pc3-laravel* para entrar en la carpeta de laravel y *php artisan serve* para iniciar el servidor de Laravel.

Después, abriremos una nueva terminal y haremos lo mismo, entraremos en la carpeta de angular usando *cd pc3*, y con *ng serve -o*, cargaremos el servidor de Angular. Para acceder a este servidor, y por tanto a nuestra aplicación, tendremos que mantener el cursor por encima de la ip que se nos muestra y haciendo click sobre ella se nos redireccionará a la página principal.

Bibliografía

- [1]. (n.d.) IEEE Xplore. Se recuperó el abril 5, 2021 de <https://ieeexplore.ieee.org/Xplore/home.jsp>
- [2]. (n.d.) idealista — Casas y pisos, alquiler y venta. Anuncios gratis. Se recuperó el abril 3, 2021 de <https://www.idealista.com/>
- [3]. (n.d.) Fotocasa.es: Alquiler de pisos, compra y venta. Se recuperó el abril 3, 2021 de <https://www.fotocasa.es/es/>
- [4]. (n.d.) 15Mpedia. Se recuperó el abril 4, 2021 de <https://15mpedia.org/>
- [5]. (n.d.) Casas rurales en EscapadaRural.com. Se recuperó el abril 3, 2021 de <https://www.escapadarural.com/>
- [6]. (n.d.) Pisos y casas en venta y alquiler – SpainHouses.net. Se recuperó el abril 3, 2021 de <https://www.spainhouses.net/>
- [7]. (n.d.) Angular. Se recuperó el junio 8, 2021 de <https://angular.io/>
- [8]. (n.d.) ¿Qué Transmiten los Colores? | Helloprint. Se recuperó el junio 8, 2021 de <https://www.helloprint.es/blog/que-transmiten-los-colores/>
- [9]. (n.d.) Laravel - The PHP Framework For Web Artisans. Se recuperó el junio 8, 2021 de <https://laravel.com/docs/8.x>
- [10]. (n.d) Python requests. La librería para hacer peticiones http en Python. Se recuperó el junio 9, 2021 de <https://j2logo.com/python/python-requests-peticiones-http/>
- [11]. (n.d.) Web scraping con Python. Guía de inicio de BeautifulSoup. Se recuperó el junio 9, 2021 de <https://j2logo.com/python/web-scraping-con-python-guia-inicio-beautifulsoup/#scraping-beautifulsoup>
- [12]. (n.d.) Python RegEx. Se recuperó el junio 9, 2021 de https://www.w3schools.com/python/python_regex.asp
- [13]. (n.d.) re - Regular expression operations - Python 3.9.5 documentation. Se recuperó el junio 9, 2021 de <https://docs.python.org/3/library/re.html>
- [14]. (n.d.) MySQL :: MySQL Connector/Python Developer Guide. Se recuperó el junio 9, 2021 de <https://dev.mysql.com/doc/connector-python/en/>
- [15]. (n.d.) Laravel. Se recupero el junio 9, 2021 de <https://desarrolloweb.com/home/laravel>
- [16]. (n.d.) Roboto Font. Se recuperó el junio 9, 2021 de shorturl.at/uSVW7

[17]. (n.d.) JSON Web Token Introduction - jwt.io. Se recuperó el junio 9, 2021 de <https://jwt.io/introduction>

[18]. (n.d.) Angular 11 - JWT Authentication. Se recuperó el junio 9, 2021 de <https://www.section.io/engineering-education/jwt-auth-angular/>