



# **Extractor y Validador de Datos de Documentos de Identidad.**

Grado en Técnicas de Desarrollo de Aplicaciones Web y Móviles.

**Autor:** Roque Otero Freiría

**Tutor:** Carlos Molina Clemente

**Fecha:** Mayo/2022

## **Agradecimientos**

Querría aprovechar estas primeras líneas para expresar mi más sincero agradecimiento a la empresa AppToIn por darme la oportunidad de poder realizar el trabajo de fin de grado en su empresa, más concretamente a Castor Agra por la tutorización y por el apoyo durante la realización de este proyecto.

Querría agradecer también especialmente a mi novia, a mis amigos y a mi familia ya que gracias a su apoyo han hecho posible graduarme como desarrollador de aplicaciones web y móviles.

## **Resumen**

La empresa AppToIn, se dedica a la gestión de cerraduras electrónicas. Uno de sus principales clientes son los propietarios que utilizan las cerraduras para pisos vacacionales. Por esto, la propia aplicación permite al cliente de esos pisos hacer el check-in y registrarse para poder acceder al apartamento y así tener unas claves temporales de la cerradura. Gracias a esta utilidad el propietario se despreocupa de tener que recoger los datos del cliente y tener que gestionarlos ya que se encarga de todo la aplicación.

El objetivo de este Trabajo de Fin de Grado será la creación de un API que extraiga y valide los datos del usuario a través de una foto de su documento de identidad y la creación de la ventana en la cual el usuario interaccionará para cargar la imagen e introducir los datos.

La API y la ventana del usuario, además de la aplicación web de la empresa están hechas con el framework Symfony y en php.

## **Resum**

L'empresa AppToIn, es dedica a la gestió de panys electrònics. Un dels seus principals clients són els propietaris que utilitzen els panys per a apartaments de vacances. Per això, la mateixa aplicació permet al client d'aquests apartaments fer el check-in i registrar-se per poder accedir a l'apartament i així tindre unes claus temporals del pany. Gràcies a aquesta utilitat el propietari es despreocupa d'haver de recollir les dades del client i haver de gestionar-les, ja que s'encarrega de tot l'aplicació.

L'objectiu d'aquest Treball de Fi de Grau serà la creació d'un API que extragui i validi les dades de l'usuari mitjançant una foto del seu document d'identitat i la creació de la finestra en la qual l'usuari interaccionarà per carregar la imatge i introduir les dades.

L'API i la finestra de l'usuari, a més de l'aplicació web de l'empresa, estan fetes amb el framework Symfony i en php.

## **Abstract**

AppToIn is a company that deals with the management of electronic locks. One of its main clients is the locking system for seasonal flats. For this reason, the application itself allows the client of these flats to check in and register in order to gain access to the flat and thus have a temporary key to the lock. Thanks to this utility, the owner does not have to worry about having to collect the client's data and manage them, as the application takes care of everything.

The objective of this Final Degree Project will be the creation of an API that extracts and validates the user's data through a photo of their identity document and the creation of the frontend where the user will interact to load the image and enter the data.

The API and the frontend, in addition to the company's web application are made with the Symfony framework and are programmed in php.

# Índice

<b>1. Introducción</b>	<b>5</b>
1.1. Turismo	6
1.1.1. Registro de viajeros	7
1.1.2. Ejecución del registro.	8
1.1.3. Herramientos actuales para el check-in.	8
1.2. Empresa	9
1.3. Tecnología y automatización.	10
1.4. API.	12
1.5. Código MRZ	13
<b>2. Objetivos</b>	<b>14</b>
<b>3. Especificaciones.</b>	<b>16</b>
3.1. Base de la que partimos.	16
3.2. Especificaciones API.	16
3.3. Especificaciones parte del usuario.	16
<b>4. Diseño.</b>	<b>18</b>
4.1. Parte Visual	18
5.2. Parte API.	22
<b>5. Implementación.</b>	<b>23</b>
5.1. Tecnologías utilizadas	23
5.1.1. Symfony	23
5.1.2. Php	24
5.1.3. Visual Studio Code	25
5.1.4. Base 64	25
5.1.5. TESSERACT OCR	25
5.1.6. Javascript	26
5.1.7. HTML.TWIG	26
5.1.8. AJax	26
5.1.9. CORS	27
5.2. Programación	28
5.2.1. Microservicio API	28
5.2.2. Parte usuario.	30
<b>6. Pruebas</b>	<b>34</b>
6.1. Pruebas a la API.	34
6.2. Pruebas en la ventana de la aplicación.	35
<b>7. Legislación</b>	<b>36</b>
<b>8. Conclusión</b>	<b>40</b>
<b>9. Planificación Temporal.</b>	<b>42</b>
<b>10. Bibliografía</b>	<b>44</b>

<b>11. Anexos</b>	<b>46</b>
11.1. Anexo 1.	46
11.2. Anexo 2.	47
11.3. Anexo 3.	49

# 1. Introducción

## 1.1. Turismo

El turismo comprende las actividades que realizan las personas durante sus viajes y estancias en lugares distintos a su entorno habitual, por un período de tiempo consecutivo inferior a un año, con fines de ocio, por negocios y por otros motivos turísticos, siempre y cuando no sea desarrollar una actividad remunerada en el lugar visitado.

El turismo tal y como se conoce en la actualidad nace en el siglo XIX, como consecuencia de la Revolución Industrial. Aunque es cierto que en la Antigüedad también se hacían viajes, estos eran motivados por guerras, movimientos migratorios, conquista y comercio, no con la finalidad de ocio, descanso o cultura.

Actualmente, es una actividad cuyo crecimiento progresivo tiene impactos cada vez más importantes en el ámbito cultural, social y económico de un país. Esto es así porque produce efectos en la balanza de pagos, en las inversiones, en la construcción, y en la mejora del transporte, lo que a su vez repercute en el empleo y, en definitiva, en el bienestar de los miembros de una comunidad.

Tanto el turismo internacional receptivo como el interno generan ingresos genuinos ya sea directos: por pago de servicios utilizados, tales como alojamientos, empresas de viajes, restaurantes, industrias culturales, comercios de artesanías, transporte, comunicaciones...; como indirectos, el gasto turístico motoriza sucesivas cadenas de pago a proveedores y personal ocupado y, en definitiva, impacta en los ingresos de todos los sectores relacionados con la actividad.

En el siguiente gráfico (Figura 1) podemos observar que el número de turistas extranjeros que vienen a España casi se ha duplicado en los últimos diez años. Como hemos comentado previamente, el turismo genera ingresos, en 2019 hubo unos ingresos de 92 mil millones de euros, lo cual es una cifra bastante significativa.



**Figura 1.** Gráfica de distintos datos sobre el turismo en España.

El turismo tiene tanto ventajas como desventajas. Como ventajas se han hablado de algunas anteriormente, pero también añadiría que proporciona oportunidades de empleo, ayuda a diversificar la economía, aumenta los ingresos públicos, expande el desarrollo económico y social, etc. Como desventajas comentaría que crea un exceso en la demanda de recursos naturales y culturales, puede resultar en un desarrollo económico desequilibrado, genera contaminación, aumenta la vulnerabilidad a los cambios económicos y políticos, degrada el ambiente cultural, etc. Personalmente, creo que el turismo es bueno mientras no sea un turismo masificado, ya que eso es lo que provoca la mayoría de las desventajas previamente descritas.

### 1.1.1. Registro de viajeros

El registro de viajeros es un libro recopilatorio de los datos de identificación de cada huésped que se aloja en un establecimiento turístico. El conjunto de datos generados por todos los viajeros de una misma reserva se incluyen en un parte de entrada, que se va adjuntando y conservando en el libro de visitas.

La ley obliga a que: “Los establecimientos de hostelería, así como aquellos dedicados de modo profesional o habitual mediante precio, a proporcionar alojamiento a las personas, con o sin otros servicios de carácter complementarios, quedan obligados mediante la Orden INT/1922/2003, de 3 de Julio a cumplimentar el libro de registro de viajeros, así como a

presentar o remitir el parte de entrada de viajeros a las Comisarías de Policía o Puestos de la Guardia Civil, en su caso.”

Como leemos previamente, es obligatorio mantener y notificar un registro de viajeros, durante al menos 3 años. Para ello se necesita cubrir algunos datos del viajero, como por ejemplo, nombre y apellidos, dni, fecha de nacimiento... y posteriormente comunicarlo a las autoridades pertinentes, o también se puede presentar de forma online.

#### 1.1.2. Ejecución del registro.

Anteriormente, este registro se tenía que hacer de forma manual y presencial. Es decir, cuando el huésped llegaba al alojamiento el propietario cubría los datos en el registro y le entregaba las llaves. Hoy en día se puede automatizar todo, de tal forma que no es necesario que tengan que quedar presencialmente para hacer la entrega de las llaves, como ofrece AppToIn. El usuario se registra, hace el check-in por la web, y automáticamente se le envían unas claves para poder tener acceso a la vivienda en las fechas señaladas. El libro de registros se llevaría de forma online, rellenándose automáticamente con los datos que proporciona el usuario en el check-in y se irán comunicando los huéspedes directamente por la web que ofrece la guardia civil o la policía nacional.

#### 1.1.3. Herramientas actuales para el check-in.

Actualmente, existen varias aplicaciones que te permiten hacer el check-in del usuario y el envío de los datos a las autoridades correspondientes. También incluyen algunas funcionalidades extra, como una página web para poder aceptar reservas. El nombre de algunas de estas aplicaciones sería cheKin o Klippa.

Utilizar alguna de estas aplicaciones supondría un coste añadido para la empresa, primero en lo monetario, y segundo que habría que configurarlo y estar pendiente de la aplicación y sus requisitos. Como en la empresa también tiene que gestionar las cerraduras, decidieron incluir en su aplicación la parte de check-in, para así ofrecer un servicio más completo, tener todas las partes más centralizadas y abaratar costes.

## 1.2. Empresa

AppToIn se dedica a la instalación y a la gestión de cerraduras electrónicas. Permite la apertura de puertas con el móvil y la administración de diversas cerraduras, indistintamente de la marca, desde su aplicación.

La mayoría de su clientela son apartamentos turísticos, o personas que disponen de varias viviendas para hospedar. Una de las principales ventajas que ofrece AppToIn es poder tener distintas cerraduras electrónicas en los diferentes pisos, ya sean de distinta marca o de distinto tipo, y poder controlarlas todas desde la misma aplicación. También tienen usuarios particulares.

Como sus clientes son apartamentos turísticos, la aplicación permite al propietario o gestor de la cerradura poder registrar a los usuarios y asignarle permisos, gestionar los horarios, el calendario y ver los accesos realizados. Con esta utilidad, el propietario puede permitir que el cliente entre en el piso sin la necesidad de quedar para darle las llaves y a la salida del mismo también.

Para registrarse como cliente de un piso que disponga de estas cerraduras primero le llega una invitación del propietario, se registra en la aplicación, hace el check-in (en el check-in es donde se le va a dar uso al extractor de datos) para poder tener sus datos y gestionar el apartado legal y finalmente ya tendría acceso a esa vivienda en las fechas estipuladas.

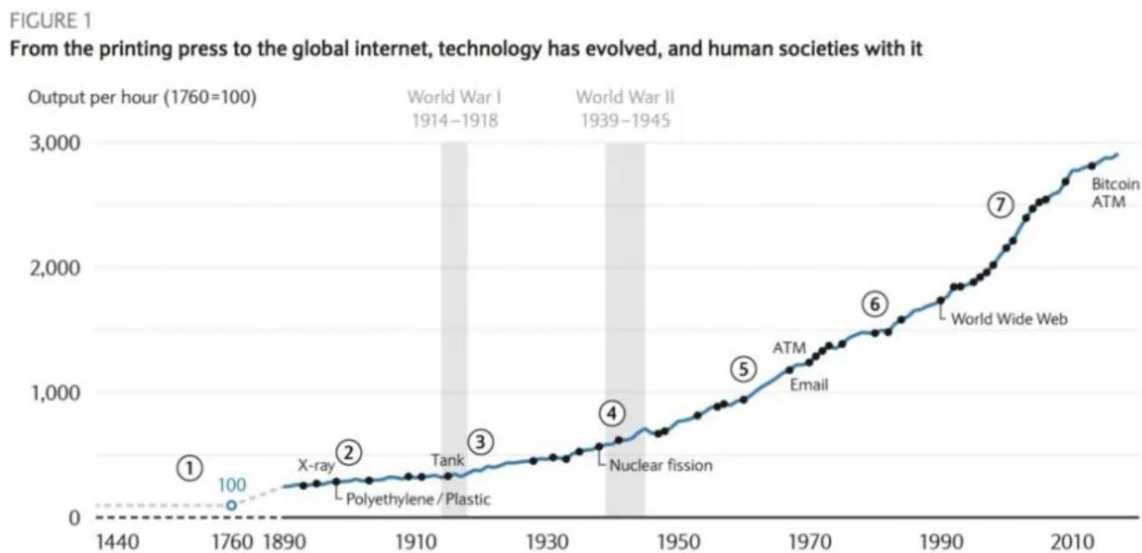


**Figura 2.** Logo de la empresa AppToIn

### 1.3. Tecnología y automatización.

Hoy en día, podemos estar todos de acuerdo en que la tecnología y la automatización forman ya parte de nuestras vidas, no hay momento en el que no estemos en contacto con un proceso automatizado o con un avance moderno de la tecnología. Desde que nos despertamos, utilizando la alarma del móvil; yendo al trabajo, fichando con una tarjeta y quedando grabado en una base de datos nuestro registro; volviendo a casa en coche, usando los sistemas de seguridad moderna de los coches o al desbloquear el móvil usando el reconocimiento facial. Uno de los principales usos de la tecnología es mejorar nuestro día a día.

En la siguiente gráfica (Figura 3) se muestra una relación entre algunos de los avances tecnológicos más destacables y la productividad. Se puede observar que la gráfica asciende exponencialmente, es decir aumentando a una velocidad mayor a medida que se inventan nuevas tecnologías. Se puede apreciar también que el avance que más facilitó, debido a que la inclinación es aún mayor que en el resto, es la llegada de Internet.



**Figura 3.** Gráfica que relaciona avances tecnológicos con la productividad.

Gracias a la tecnología, las empresas pueden mejorar la productividad, reducir costes, reducir trámites y permitir que los equipos de personas liberen tiempo para centrarse en otras tareas de valor añadido, como la relación con el cliente o el seguimiento de situaciones complejas. Las empresas utilizan la automatización y la inteligencia artificial (IA) para aumentar sus

fortalezas, compensar las debilidades y capacitar a las personas para que se concentren en lo que importa.

En los negocios, los avances en tecnología y automatización, permiten tener un mayor control y seguridad sobre ellos, por ejemplo, pudiendo controlar las cámaras de seguridad desde el móvil, o mismo abrir las puertas del negocio a distancia, cuando ellos quieran sin la necesidad de estar allí físicamente.

Otro gran avance, a mi parecer, son las redes sociales, las cuales nos permiten estar completamente conectados con nuestro entorno, y con cualquier parte del mundo al instante. Mirando nuevamente desde el punto de los negocios, permite llegar a una mayor audiencia sin la necesidad de tener que invertir una gran cantidad de dinero, pudiendo así crecer a niveles que sin ellas sería imposible.

Se estima que entre el 0% y el 30% de las horas trabajadas en todo el mundo podrían automatizarse para 2030, dependiendo de la velocidad de adopción de la tecnología. Sin embargo, es importante señalar que aunque algunas tareas estén automatizadas, el empleo en esas ocupaciones no disminuirá, sino que los trabajadores podrán realizar nuevas tareas.

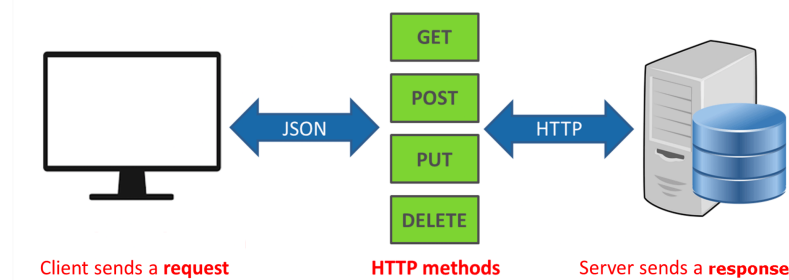
La automatización tendrá un efecto menor en los trabajos que involucran la gestión de personas, la aplicación de conocimientos e interacciones sociales, donde las máquinas no pueden igualar el rendimiento humano. Se tiene esta creencia, de que la automatización y la tecnología va a reducir el número de empleos donde son necesarias personas pero esto se ha demostrado que no es así. Por ejemplo en Estados Unidos con la introducción de los cajeros automáticos en los bancos también había esta creencia, pero contrariamente a lo que se había podido temer, la eliminación de puestos de trabajo no tuvo lugar, e incluso aumentó ligeramente a pesar del rápido despliegue de los cajeros automáticos. En otros casos, aunque se puedan eliminar empleos, el avance puede que ayude a generar nuevos puestos de trabajo. En Francia, se cree que, después de quince años de su introducción, Internet ha destruido unos 500.000 empleos; sin embargo, al mismo tiempo han creado 1,2 millones de nuevos puestos de trabajo.

## 1.4. API.

Una **API** (Application Programming Interfaces) se trata de un conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas. Se podría definir también como conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación. El uso de ellas deriva en procesos más sencillos, ágiles y automatizados con los sistemas.

Las APIs se pueden utilizar de diferentes formas: integrando diferentes sistemas para una mayor eficiencia, y tienen un papel estratégico en la rutina de las empresas. Algunos sistemas y aplicaciones que se utilizan en una empresa interactúan con otro software, a través de las APIs. Hoy en día la mayoría de las empresas utilizan APIs para crear servicios. Esto se debe a que es un estándar lógico y eficiente para la creación de servicios web. Por poner algún ejemplo tenemos los sistemas de identificación de Facebook o Google Maps, que nos permite incrustar mapas en nuestro propio sitio web.

La forma de utilizar una API es sencilla, el cliente hace una petición, utilizando uno de los distintos tipos que hay, esta petición se procesa en el servidor, y se devuelve una respuesta.



**Figura 4.** Esquema de uso de una API

En mi opinión, las APIs son una de las herramientas más útiles a la hora de implementar la comunicación con datos de una aplicación. Porque son muy funcionales y permiten un increíble ahorro de tiempo y de recursos en el sector industrial al facilitar el uso de infraestructuras, funciones y software ya existentes en nuevas plataformas, sin necesidad de volver a diseñarlas. De hecho, según estudios de empresas como MarketsandMarkets, las APIs pasarán de mover 1.200 millones (cifra de 2018) a 5.100 millones de dólares en 2023.



## 2. Objetivos

El objetivo principal de este trabajo es crear un microservicio (API) que permita extraer los datos de un documento de identidad y validarlo, y la creación también de la parte de la aplicación con la que interactuaría el cliente. Esto permite ahorrar tiempo y ayuda al usuario a cometer menos errores, ya que hay algunos campos como la fecha de expedición del documento que son difíciles de encontrar.

Es cierto que ya existen algunas aplicaciones que permiten hacer esto, pero en la empresa pensamos que podría ser interesante y útil desarrollar un extractor nuestro, para así ahorrar un gasto y beneficiar más al cliente.

En el plano más personal, otro de los principales objetivos, es también mejorar mis habilidades desarrollando la parte del frontend y también mejorar en la programación de APIs Rest, ya que creo que es una parte muy importante y útil en el desarrollo de aplicaciones tanto web como móviles o al crear algún servicio.

Como objetivos secundarios de este proyecto serían desarrollar una pantalla útil y funcional para el cliente y agilizar el proceso de check-in a los usuarios, ya que tener que sacar varias fotografías e introducir los datos manualmente puede resultar un proceso largo.



## 3. Especificaciones.

En este apartado veremos los principales requisitos de este servicio. Tenemos partes con especificaciones distintas, por un lado tenemos la API Rest y por otra, la parte del usuario.

### 3.1. Base de la que partimos.

A la hora de desarrollar este proyecto, se partía de una base previamente desarrollada por la empresa. Ya existía el servicio de check-in, es decir, la base sobre la que está desarrollada la parte del usuario ya estaba hecha. Se contaba con ficheros html.twig previamente creados y los controllers de las ventanas ya existían. Mi trabajo en esa parte consistió en revisar y entender cómo estaban hechos y cómo funcionaban y partiendo de ellos creé mi propio controller y mi propio fichero html.twig, con la vista y los determinados parámetros que se le pasa.

### 3.2. Especificaciones API.

Se necesitaba también que la parte de la API Rest fuese rápida, siendo la parte del recorte de imagen ejecutada por el usuario en local; que sea funcional, es decir que contemple todos los posibles casos, como por ejemplo que la imagen no contenga texto o que la imagen sea poco legible; y por último que sea a prueba de errores, por ejemplo que intenten pasar un script en vez de una imagen para un posible ataque.

La API tiene que poder recibir una imagen codificada, en base64, y ser capaz de descodificarla. La respuesta tiene que ser un JSON, y en un campo tiene que venir la información sobre si la extracción de datos funcionó correctamente o no, y en los otros campos la información necesaria.

### 3.3. Especificaciones parte del usuario.

En esta parte, lo principal es poder recoger los datos. En la ventana del usuario tiene que poder subirse una imagen. Para una mayor precisión, la imagen se podrá recortar; para facilitar el recorte habrá unos botones que servirán para cambiar el ángulo de la imagen, y así aunque la imagen esté girada, podrá encajar en el rectángulo que se recortará. Si la manera principal, que es en la que está enfocada este trabajo, no funciona correctamente, el usuario podrá introducir los datos de forma manual. Otro requisito de esta parte es que sea sencillo e intuitivo, teniendo pocos botones, y de claro uso.

## 4. Diseño.

### 4.1. Parte Visual

Este trabajo solo consta de una ventana con la que interaccionará el usuario, que sigue el diseño de la aplicación ya existente.

Esta pantalla es simple pero eficiente (Figura 6). Consta de una parte para la imagen, que a priori será una de ejemplo, y otra para un botón que permite sacar una fotografía o seleccionar una ya existente. En la parte inferior hay dos botones, uno para volver a la pantalla anterior y otro para avanzar, que se encuentra oculto hasta que estén los datos ya escritos.



**Figura 6.** Diseño de ventana, al aparecer la pantalla para extraer los datos.

Una vez introducida la imagen, se cambian los botones, desapareciendo el botón de “Foto”, y apareciendo uno de para eliminar la foto, además de 4 botones para poder ajustar el ángulo de la imagen, y un último botón “Escanear” para extraer los datos de la foto y enviarlos a la API (Figura 7).



**Figura 7.** Diseño de ventana, al subir una imagen para escanear.

Como podemos apreciar en la imagen (Figura 7) se aprecia un rectángulo (sombreado), este permite ajustar los bordes del documento para una mayor precisión a la hora de analizarlo.

Una vez extraídos los datos del documento, la ventana adquiere un nuevo diseño (Figura 8). Se muestra en la parte de arriba la imagen anterior, y debajo los datos del usuario extraídos.

The image shows a web interface with a light gray background. At the top, there is a large rectangular area labeled "Foto Documento". Below this, there are nine rows, each consisting of a label on the left and a light gray rectangular input field on the right. The labels are "Campo 1" through "Campo 9". At the bottom of the form, there are two blue rectangular buttons: "Volver" on the left and "Siguiente" on the right.

**Figura 8.** Diseño ventana, campos extraídos mediante API.

Si no se extraen los datos correctamente, el usuario debe introducirlos manualmente. Para ello, en la ventana (Figura 9) aparecerán todos los campos a cubrir.

Campo 1

Campo 2

Campo 3

Campo 4

Campo 5

Campo 6

Campo 7

Campo 8

Campo 9

**Figura 9.** Diseño ventana, rellenar datos manualmente.

## 5.2. Parte API.

Para diseñar una API se necesitan principalmente 3 componentes. El primero serían las rutas de las llamadas, es decir, las direcciones a las que hay que llamar y que tipo de peticiones acepta cada ruta. También es necesario un Controller, esta parte se encarga de gestionar cada llamada, e invoca a las funciones necesarias. Y por último, la parte de las funciones, es donde está programada la funcionalidad de la API y todas las tareas que ejecute. Si se accediera a una base de datos también podría tener una parte que fuera DAO(Data Acces Object).

En este caso, la API constará de un Controller, en el cual también se indican las rutas, y de un fichero principal, en el cual se programa todo el ciclo de vida de la API, que va llamando a subfunciones que están distribuidas en otros ficheros, según su funcionalidad.

Recibirá un parámetro en el body, que será la imagen codificada en base64, que más adelante se descodificará. Para la respuesta, el cliente recibirá un JSON, este estará formado siempre por un campo que tendrá la información sobre si la extracción de datos funcionó correctamente. Finalmente, dependiendo de si salió bien o no, aparecerán campos con informaciones distintas: si no funciona, un mensaje con el error y, si funciona, serán los datos extraídos.

## 5. Implementación.

Para poder realizar este proyecto se han utilizado diversas herramientas y funciones tales como lenguajes, librerías y programas los cuales en conjunto han hecho posible hacer el proyecto. A continuación, comentaré y explicaré los recursos utilizados.

### 5.1. Tecnologías utilizadas

#### 5.1.1. Symfony

Symfony es un framework diseñado para desarrollar aplicaciones web basado en el patrón Modelo Vista Controlador. Funciona de la siguiente manera, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

Symfony está desarrollado completamente en PHP, al cual dedicaremos un apartado más adelante. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows.

En Symfony está desarrollada la aplicación web de la empresa, y también he usado symfony para crear la API, donde va alojado el script extractor de datos.

#### - Annotations de Symfony

Específicamente en este framework y en este lenguaje, las anotaciones son comentarios en donde especifican cosas, es decir, son unos comentarios especiales o con un formato específico, a los que el framework le dará uso a la hora de ejecutarse.

En nuestro caso vamos a usar estas annotations para definir el routing de nuestra aplicación. Cuando una aplicación recibe una petición, esta llama a una acción del controller para

generar respuesta, la configuración del routing define qué acción ejecutar para cada URL entrante.

- Composer

Composer es un sistema de gestión de paquetes para programar en PHP el cual provee los formatos estándar necesarios para manejar dependencias y librerías de PHP.

Se usa el Composer para instalar el paquete Annotations de Symfony mediante el comando:

```
$ composer require doctrine/annotations
```

**Figura 10.** Comando instalar Annotations.

### 5.1.2. Php

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Una de las características de PHP, que lo distingue de algo como JavaScript del lado del cliente es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente.

Una de las ventajas de php es su facilidad de uso y también la cantidad de métodos que ya tiene integrado de forma propia, que con cada versión se van mejorando y añadiendo más. Otra característica resaltable es su disponibilidad, ya que lo encontramos en la mayoría de alojamientos web de todo tipo.

Uso el php para desarrollar todo el código de la aplicación, tanto la API, como el backend de la aplicación web.



**Figura 11.** Logo php.

### 5.1.3. Visual Studio Code

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE). Es compatible con múltiples lenguajes de programación. Es un entorno muy completo ya que permite instalar en él una gran cantidad de plugins, como el resaltado inteligente del código, aviso de errores de sintaxis, autocompletado de código.

Este es el editor de código escogido para desarrollar este proyecto, por todas las ventajas comentadas anteriormente, pero sobre todo por la facilidad que aporta para trabajar en remoto con un servidor mediante una conexión por SSH, pudiendo arrastrar ficheros desde tu explorador de documentos hasta la ventana de Visual Studio y subiéndose instantáneamente.

### 5.1.4. Base 64

Base64 es un grupo de esquemas de codificación de binario a texto que representa los datos binarios mediante una cadena ASCII, traduciéndolos en una representación radix-64. Se representan datos binarios en secuencias de 24 bits que pueden ser representados por cuatro dígitos Base64 de 6 bits.

Se usa base64 para pasar la imagen recortada en el cuerpo de la llamada a la API, luego en la API se descrypta y se usa como imagen png.

### 5.1.5. TESSERACT OCR

Tesseract es un motor de reconocimiento óptico de caracteres para varios sistemas operativos. Es software libre, liberado bajo la licencia Apache, Versión 2.02 y su desarrollo es financiado por Google desde el 2006.

Tesseract fue considerado en 2006 como uno de los motores de OCR de código abierto más precisos disponibles. En nuestro caso lo uso para poder leer el texto de la imagen del documento de identidad recortado.

### 5.1.6. Javascript

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, que es el uso que se le da en esta práctica.



**Figura 12.** Logo Javascript.

### 5.1.7. HTML.TWIG

Twig es un motor para generar templates basado en PHP, a través del cual se crean los templates o archivos reutilizables de un theme. Se usa generando código backend haciendo posible que se pueda consumir a través de objetos, variables y métodos en el front end.

En la aplicación se usa para generar la ventana donde introducir los datos, estos datos se generan en el backend, creando un formulario que el twig interpreta y muestra en la parte de frontend.

### 5.1.8. AJax

Las siglas de Ajax responden a Asynchronous JavaScript y XML, o Javascript y XML Asíncrono. Ajax es un lenguaje de programación que permite la actualización de secciones específicas de contenido en una página web, sin necesidad de recargar completamente la página.

Con Ajax, las aplicaciones web pueden enviar datos desde un servidor y recuperarlos de forma asíncrona (cargando de fondo), sin interferir con la visualización y el comportamiento de la página existente. Con Ajax se hace la llamada a la API, pasándole como argumento el código en base64 de la imagen recortada.

### 5.1.9. CORS

El intercambio de recursos de origen cruzado o CORS (Cross-origin resource sharing, en sus siglas en inglés) es un mecanismo que permite que se puedan solicitar recursos restringidos (cuando se abre una página web, cargar datos de servidores ajenos está, en teoría, estrictamente prohibido) en una página web desde un dominio diferente del dominio que sirvió el primer recurso. Está restringido porque una página web podría libremente incrustar imágenes, hojas de estilo, scripts, iframes y vídeos de origen cruzado. Ciertas peticiones de origen cruzado, más notablemente peticiones Ajax, están prohibidas por defecto por la política de seguridad del mismo origen (en inglés, same-origin security policy).

CORS define una forma en la cual el navegador y el servidor pueden interactuar para determinar si es seguro permitir una petición de origen cruzado.

Al querer hacer una llamada desde el frontend a la API aparece este problema, para solucionarlo hemos tenido que escribir unas líneas de código en el servidor al cual se le hace la petición, en estas líneas se ordena que se permita “entrar” todas las peticiones, y si el tipo de petición es de tipo OPTIONS, se mata esa petición. El código quedaría así:

```
<?php
use App\Kernel;

require_once dirname(__DIR__) . '/mrz/vendor/autoload_runtime.php';

header('Access-Control-Allow-Origin: *');
header("Access-Control-Allow-Headers: X-API-KEY, Origin, X-Requested-With, Content-Type, Accept, Access-Control-Request-Method");
header("Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE");
header("Allow: GET, POST, OPTIONS, PUT, DELETE");
$method = $_SERVER['REQUEST_METHOD'];
if($method == "OPTIONS") {
    die();
}

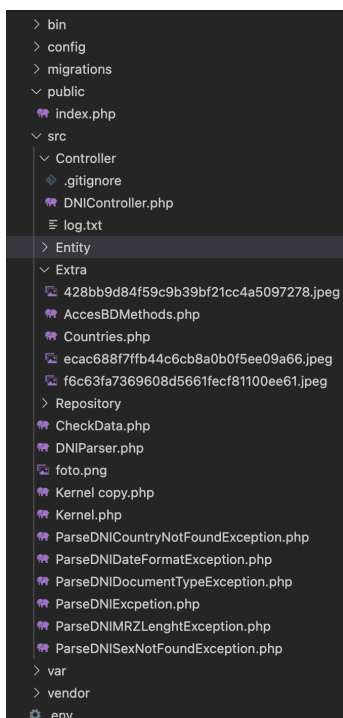
return function (array $context) {
    return new Kernel($context['APP_ENV'], (bool) $context['APP_DEBUG']);
};
```

**Figura 13.** Código para evitar problema de CORS.

## 5.2. Programación

### 5.2.1. Microservicio API

La api está estructurada de la siguiente manera:



**Figura 14.** Estructura API.

En la carpeta Controller se encuentra el fichero que se encarga del routing de la aplicación (DNIController.php), es decir maneja todas las peticiones que llegan. Desde aquí se llama al método parseMRZ (\$base64), de la clase parseDNI.php. Luego en la carpeta Extra se encuentran todas las clases de apoyo extras, como las Excepciones o clases con métodos concretos, que no están relacionados con el método principal.

Este método parseMRZ funciona de la siguiente manera, llega el código en base64 del controller, luego se descodifica (base64\_to\_jpeg()) y se guarda en imagen. Después se le aplica un filtro gris para pasar la imagen a blanco y negro, para que así se resalten mejor el texto del documento. Se extrae mediante, la librería TESSERACT OCR, el texto de la imagen en blanco y negro. Con este texto ahora buscamos el código mrz mediante una expresión regular ("/{2,}<{2,}/i"). Para poder saber que tipo de documento es necesitamos saber el primer carácter del código mrz, P para pasaporte e I para DNI. Según esto se llama a un método

concreto pero su funcionamiento es similar, se selecciona los datos del código por su determinada posición, según sea DNI o pasaporte los datos estarán colocados de manera distinta, pero la posición de los datos nunca varía.

La respuesta de la API está formada por un JSON de un campo ok, que es un booleano que indica si la extracción de los datos funcionó correctamente y depende de si el campo ok es true: vienen todos los datos extraídos, un dato por campo; y si es false, viene un campo mensaje, que es el mensaje de error de lo que ocurrió.

### 5.2.2. Parte usuario.

Para implementar la parte con la que interacciona el usuario, como ya he comentado anteriormente, parto de una estructura ya hecha. Principalmente se compone de dos ficheros el `InvitacionController.php` y `fotoCheckMRZ.html.twig`.

- `InvitacionController.php`

En este fichero se controla el hilo de ejecución del proceso de check-in, desde aquí se llama a las distintas pantallas y se reciben los formularios del `html.twig`.

El método que va a controlar esta ventana se llama `huespedFotoReversoMrzAction()`. En este método primero hacemos unas comprobaciones para ver si ya hay datos de ese usuario. Luego creamos el formulario, que se creará dinámicamente en el frontend, se escriben todos los campos y sus características. Se le añade un campo `HiddenType` que servirá posteriormente para saber si el usuario se registró a través del microservicio o tuvo que introducir manualmente los datos. Si los introdujo manualmente tendrá que insertar dos fotos más para poder validar correctamente el usuario. Después de crear el formulario, hacemos la parte para tratar la respuesta de ese formulario, para saber si el formulario está subido se hace un `if` llamando al método `isSubmitted()`. Se guarda la foto recortada del documento y se hace una inserción en la base de datos de ese usuario y posteriormente según la manera en que se haya registrado se le dirige a una ventana u otra. Por último, se hace la llamada al `fotoCheckMRZ.html.twig` pasándole algunos parámetros como el formulario, la foto (si ya se ha subido previamente), la foto del reverso de muestra...

- `FotoCheckMRZ.html.twig`

En el frontend se muestran los datos del formulario, que vienen del servidor, mediante los métodos que implementa el twig ( `{{ form_row() }}` ). Después usando javascript, se recorta la imagen, y usando Ajax se hace la llamada a la API.

Luego mediante el botón siguiente, se hace `submit` del formulario y dependiendo de si los datos fueron extraídos o no de la foto, se envía a una pantalla, donde tendrá que aportar más información, o a una pantalla donde introducirá su firma y habrá acabado con el check-in.

A continuación muestro como quedaría esta parte implementada, siguiendo el diseño comentado en el apartado de diseño:

- Ventana inicial:



Figura 15. Ventana usuario al iniciar.

- Ventana tras introducir imagen:

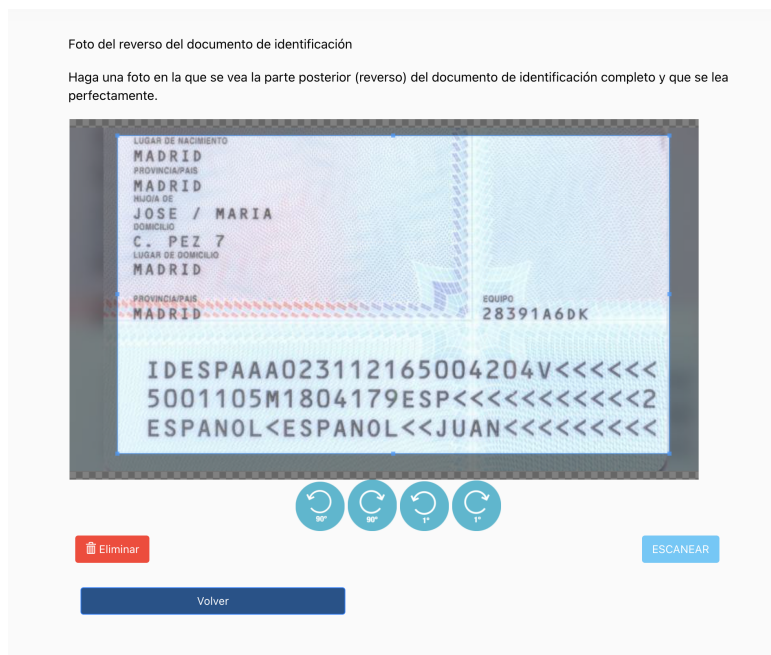


Figura 16. Ventana al subir foto.



-Ventana si es necesario que introduzca los datos manualmente:

Foto del reverso del documento de identificación

Haga una foto en la que se vea la parte posterior (reverso) del documento de identificación completo y que se lea perfectamente.

**Tipo documento**

**Documento**

**Fecha expedición**

**Sexo**

**Apellido 1**

**Apellido 2**

**Nombre**

**Fecha nacimiento**

**Nacionalidad**

**Figura 18.** Ventana para introducir los datos manualmente.

## 6. Pruebas

Para hacer las pruebas de este proyecto, las he dividido en dos partes. Primero en pruebas haciendo llamadas directamente a la API, desde PostMan y una pequeña página creada por mí en php, que permitía seleccionar una foto y recortarla, y las pruebas directamente a la ventana de la aplicación.

### 6.1. Pruebas a la API.

-Pasando en la llamada, un archivo en base64 que no fuera una imagen.

Con esta prueba se pretendía comprobar y corregir por si en vez de una imagen pasaban un archivo distinto o con código que este no se abra para evitar así errores de la aplicación. Cuando esto ocurre salta una excepción que recogemos y tratamos adecuadamente.

-Pasando en la llamada un archivo muy pesado.

Con esta prueba se pretende comprobar que aunque se pase un archivo muy pesado la aplicación no se detenga. Cuando esto ocurre salta una excepción que recogemos y tratamos adecuadamente.

-Pasando una imagen donde no se lea bien.

Con esta prueba se pretende comprobar que ocurre al recibir una imagen donde el código mrz no está borroso. El programa no lo detecta correctamente por tanto devuelve una excepción y que también tratamos adecuadamente.

-Pasando una imagen donde no esté el código mrz.

Con esta prueba se pretende comprobar qué ocurre cuando el microservicio recibe una imagen donde no se encuentra el código mrz. Si el programa no lo detecta devuelve una excepción que se trata correctamente.

-Pasando una imagen donde el código mrz esté incompleto.

Con esta prueba se pretende comprobar qué ocurre si sólo se detecta una parte del código mrz. Si el programa solo detecta una parte devuelve una excepción que se trata correctamente.

## 6.2. Pruebas en la ventana de la aplicación.

-Introducir una vez incorrectamente y la siguiente bien.

Con esta prueba se pretende comprobar qué ocurre cuando el usuario introduce una vez erróneamente la imagen. Si esto ocurre el usuario tiene dos oportunidades más de introducir correctamente la imagen.

-Introduciendo tres veces erróneamente la imagen.

Con esta prueba se pretende comprobar qué ocurre si el usuario introduce tres veces la imagen y en 3 ocasiones no se pueden extraer los datos. Cuando esto ocurre se quita la opción de introducir la imagen y se muestran los campos de los datos para introducirlos de forma manual.

-Al revisar los datos, cambiar algún campo poniéndolo en el formato incorrecto.

Con esta prueba se pretende comprobar qué ocurre si el usuario modifica los datos o los introduce de forma incorrecta. Si esto ocurre saldrá una alerta indicando que campo es incorrecto.

## 7. Legislación

Cuando se reciben clientes en un apartamento turístico, el propietario está obligado a declarar cuántos clientes recibe y sus datos, a la Policía Nacional o a la Guardia Civil.

La obligación del registro de huéspedes no es una ocurrencia moderna, lleva codificada en la legislación española desde 1959, hace ya más de 60 años. No obstante, el cambio más relevante para los propietarios de alquileres vacacionales se produjo en 2013, con la aprobación de la Ley 4/2013, de 4 de junio(Anexo 1), la cual modificaba la LAU(Ley de Arrendamientos Urbanos) con el objetivo de equiparar las obligaciones de las viviendas turísticas con las del resto de alojamientos.

Desde entonces, la normativa no ha hecho más que endurecerse. Actualmente, el registro de viajeros está regulado por la Ley de Seguridad Ciudadana 4/2015, de 30 de marzo (Anexo 2), pero se trata de una situación excepcional y con fecha de caducidad.

Actualmente, los datos necesarios son los siguientes:

Datos del establecimiento:

- Parte n.º.
- Nombre del establecimiento.
- NIF
- Municipio.
- Provincia.

Datos del viajero:

- Núm. de documento de identidad.
- Tipo de documento.
- Fecha expedición del documento.
- Primer apellido.
- Segundo apellido.
- Nombre.
- Sexo.
- Fecha de nacimiento.
- País de nacionalidad.
- Fecha de entrada.

Pero como hemos comentado previamente, los requisitos de registro de huéspedes van a cambiar. Se van a regir por el Real Decreto 933/2021 (Anexo 3) que, entre otras cosas, va a hacer que se aumente la cantidad de datos a recoger de 9 a 31 y la cuantía de las multas hasta los 30.000 €.

Estos cambios tenían que haber entrado en vigor el 27 de abril de este año, pero debido a las dificultades del sector para aplicar la nueva normativa han hecho que la Secretaría de Estado decidiera posponer su aplicación hasta el 1 de enero de 2023.

Los nuevos datos requeridos son:

-Datos de los viajeros:

- Nombre.
- Primer apellido.
- Segundo apellido.
- Sexo.
- Número de documento de identidad.
- Tipo de documento (DNI, pasaporte, TIE).
- Nacionalidad.
- Fecha de nacimiento.
- Lugar de residencia habitual.
- Dirección completa.
- Localidad.
- País.
- Teléfono fijo.
- Teléfono móvil.
- Correo electrónico.
- Número de viajeros.
- Relación de parentesco entre los viajeros (sólo si hay algún menor de edad).

-Datos de la estancia:

- Número de referencia del contrato.
- Fecha del contrato.
- Firma de las partes.

- Fecha de y hora de entrada.
- Fecha y hora de salida.
- Medio de pago (efectivo, tarjeta, transferencia, etc.).
- Identificación del medio de pago (número de tarjeta, IBAN, etc.).
- Titular del medio de pago.
- Fecha de caducidad (sólo tarjetas).
- Fecha del pago.



## 8. Conclusión

Desde siempre me ha llamado mucho la atención la tecnología, es por eso que he escogido esta empresa para poder hacer el TFG, la cual combina el software (aplicación) con el hardware (cerradura). Uno de mis principales objetivos era intentar familiarizarme con esta fusión, y aunque en este proyecto no es necesario el uso de un hardware específico, a mayores de un móvil o un ordenador, una parte de la aplicación sí que suplente el trabajo del lector de documentos.

Otro de los objetivos era poder ampliar los conocimientos desarrollando proyectos de una mayor profesionalidad, en los cuales hay que tener en cuenta todos los detalles, y en donde no tienes la tranquilidad de que aunque hagas un mal trabajo, no tiene una repercusión mayor. He podido ampliar también mis conocimientos en la parte de frontend.

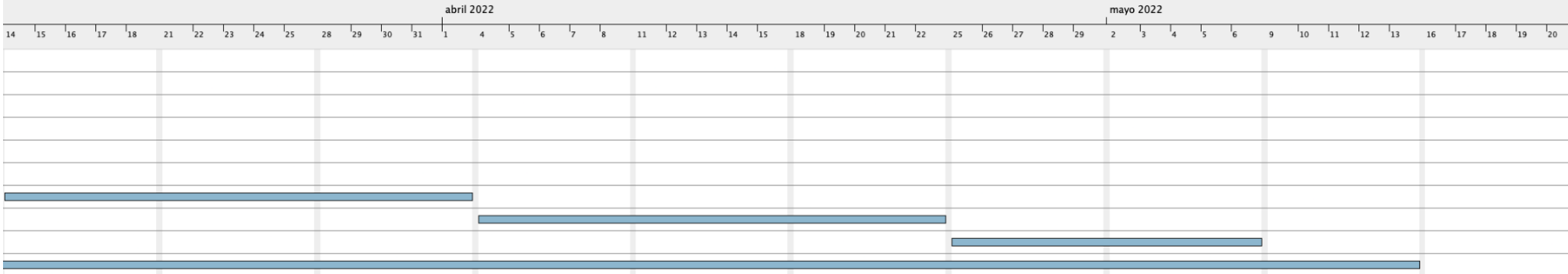
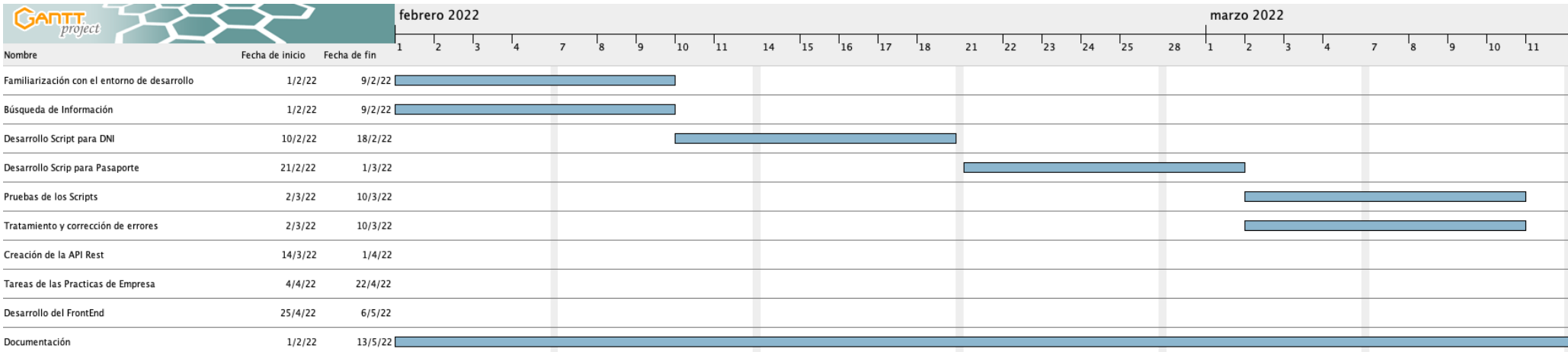
Realizando el trabajo he aprendido a trabajar sobre código de otra persona, y a aprender a entenderlo, reutilizarlo e incluso mejorar el ya existente. Considero que poder poner en práctica esto es una gran ayuda de cara a lo que nos encontraremos en el futuro entorno laboral.

Actualmente el proyecto está en fase de test, ajustando los últimos detalles para poder lanzar esta mejora a la aplicación principal. Este proyecto tiene algunas mejoras, una de ellas y en la que se espera poder trabajar pronto, es cambiar el recorte de la imagen, para que pase de ser manual a ser automático, es decir, que al subir la imagen, se recorten directamente los bordes para que solo quede el documento.

En resumen, la realización de este proyecto me ha permitido poner en práctica los conocimientos adquiridos durante la carrera, y poder aunarlos en una sola aplicación, sin tener que trabajar en cada uno de ellos de manera aislada.



# 9. Planificación Temporal.





## 10. Bibliografía

- [1] <https://startupxplore.com/es/startups/apptoin>
- [2] <https://www.elcorreogallego.es/hemeroteca/abrir-puerta-movil-puede-salvar-vidas-DBCG1194403>
- [3] <https://support.chekin.com/es/knowledge/c%C3%B3digo-mrz>
- [4] [https://www.indec.gob.ar/ftp/cuadros/economia/turismo\\_cyd.pdf](https://www.indec.gob.ar/ftp/cuadros/economia/turismo_cyd.pdf)
- [5] [https://www.lodgify.com/blog/es/registrar-viajeros-guardia-civil/?utm\\_campaign=ES%20-%20Content%20Newsletter&utm\\_medium=email&\\_hsmt=212274640&utm\\_content=212248972&utm\\_source=hs\\_email](https://www.lodgify.com/blog/es/registrar-viajeros-guardia-civil/?utm_campaign=ES%20-%20Content%20Newsletter&utm_medium=email&_hsmt=212274640&utm_content=212248972&utm_source=hs_email)
- [6] [https://www.indec.gob.ar/ftp/cuadros/economia/turismo\\_cyd.pdf](https://www.indec.gob.ar/ftp/cuadros/economia/turismo_cyd.pdf)
- [7] <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- [8] <https://aws.amazon.com/es/what-is/api/>
- [9] <https://www.zenvia.com/es/blog/que-es-api/>
- [10] <https://www.klippa.com/es/reconocimiento-de-texto-ocr/ocr-para-documentos-identidad/mrz-escaner/>
- [11] <https://es.statista.com/grafico/18881/turistas-internacionales-en-espana/>
- [12] [https://cincodias.elpais.com/cincodias/2020/01/20/economia/1579518415\\_556581.html](https://cincodias.elpais.com/cincodias/2020/01/20/economia/1579518415_556581.html)



# 11. Anexos

## 11.1. Anexo 1.



– La recuperación del inmueble por el arrendador, para destinarlo a vivienda permanente en determinados supuestos, que requiere que hubiera transcurrido al menos el primer año de duración del contrato, sin necesidad de previsión expresa en el mismo, dotando de mayor flexibilidad al arrendamiento.

– La previsión de que el arrendatario pueda desistir del contrato en cualquier momento, una vez que hayan transcurrido al menos seis meses y lo comunique al arrendador con una antelación mínima de treinta días. Se reconoce la posibilidad de que las partes puedan pactar una indemnización para el caso de desistimiento.

Asimismo, es preciso normalizar el régimen jurídico del arrendamiento de viviendas para que la protección de los derechos, tanto del arrendador como del arrendatario, no se consiga a costa de la seguridad del tráfico jurídico, como sucede en la actualidad.

La consecución de esta finalidad exige que el arrendamiento de viviendas regulado por la Ley 29/1994, de 24 de noviembre, de Arrendamientos Urbanos, se someta al régimen general establecido por nuestro sistema de seguridad del tráfico jurídico inmobiliario y, en consecuencia, en primer lugar, que los arrendamientos no inscritos sobre fincas urbanas no puedan surtir efectos frente a terceros adquirentes que inscriban su derecho y, en segundo lugar, que el tercero adquirente de una vivienda que reúna los requisitos exigidos por el artículo 34 de la Ley Hipotecaria, no pueda resultar perjudicado por la existencia de un arrendamiento no inscrito. Todo ello, sin mengua alguna de los derechos ni del arrendador, ni del arrendatario.

Por último, en los últimos años se viene produciendo un aumento cada vez más significativo del uso del alojamiento privado para el turismo, que podría estar dando cobertura a situaciones de intrusismo y competencia desleal, que van en contra de la calidad de los destinos turísticos; de ahí que la reforma de la Ley propuesta los excluya específicamente para que queden regulados por la normativa sectorial específica o, en su defecto, se les aplique el régimen de los arrendamientos de temporada, que no sufre modificación.

### III

Esta Ley opera una reforma de determinados preceptos de la Ley 1/2000, de 7 de enero, de Enjuiciamiento Civil, al objeto de abordar las cuestiones más puntuales que están generando problemas en los procesos de desahucio después de las últimas reformas de la anterior Legislatura. En concreto, se vincula el lanzamiento a la falta de oposición del demandado, de tal modo que si éste no atendiere el requerimiento de pago o no compareciere para oponerse o allanarse, el secretario judicial dictará decreto dando por terminado el juicio y producirá el lanzamiento, frente al sistema actual que impide señalar el lanzamiento hasta que no se sepa si la vista se ha celebrado o no. Estas modificaciones obligan a efectuar el ajuste de otros artículos de la Ley de Enjuiciamiento Civil.

**Artículo primero.** *Modificación de la Ley 29/1994, de 24 de noviembre, de Arrendamientos Urbanos.*

Los artículos 4, 5, 7, 9 a 11, 13 a 20, 23 a 25, 27, 35 y 36 de la Ley 29/1994, de 24 de noviembre, de Arrendamientos Urbanos, quedan modificados como sigue:

Uno. El artículo 4 queda redactado en los siguientes términos:

«Artículo 4. *Régimen aplicable.*

1. Los arrendamientos regulados en la presente Ley se someterán de forma imperativa a lo dispuesto en los títulos I y IV de la misma y a lo dispuesto en los apartados siguientes de este artículo.

2. Respetando lo establecido en el apartado anterior, los arrendamientos de vivienda se regirán por los pactos, cláusulas y condiciones determinados por la voluntad de las partes, en el marco de lo establecido en el título II de la presente ley y, supletoriamente, por lo dispuesto en el Código Civil.

## 11.2. Anexo 2.



### ANEXO

#### Modelo de parte de entrada de viajeros

*Hoja-registro*

(Rellenar con mayúsculas)

Datos del establecimiento

Parte n.º:  
Nombre del establecimiento:  
NIF:  
Municipio:  
Provincia:

Sello del establecimiento<sup>(8)</sup>

Datos del viajero:

Núm. de documento de identidad:  
Tipo de documento:<sup>(1)</sup>  
Fecha expedición del documento:<sup>(5)</sup>  
Primer apellido:  
Segundo apellido:  
Nombre:  
Sexo:<sup>(2)</sup>  
Fecha de nacimiento:<sup>(3)</sup>  
País de nacionalidad:<sup>(4)</sup>  
Fecha de entrada:<sup>(6)</sup>

....., de ..... de .....

Firma del viajero<sup>(7)</sup>

La recogida y tratamiento de estos datos se hará de acuerdo con la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales, y en lo que resulte de aplicación, la relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales por parte de autoridades competentes para fines de prevención, investigación, detección o enjuiciamiento de infracciones penales, y al amparo Ley Orgánica 4/2015, de 30 de marzo, de Protección de la Seguridad Ciudadana, artículo 25.1.

#### Instrucciones de confección del impreso

<sup>(1)</sup> Se admiten:

- Para españoles: DNI, pasaporte o permiso de conducir.
- Para extranjeros: Pasaporte, carta o documento de identidad (para ciudadanos de la Unión Europea, Andorra, Islandia, Suiza, Noruega, Malta, Mónaco y San Marino). Permiso de residencia español en vigor para los extranjeros residentes en España.

Se cumplimentará: D = DNI, P = Pasaporte, C = Permiso de conducir, I = Carta o documento de identidad, N = Permiso de residencia español, X = Permiso de residencia de otro Estado miembro de la Unión Europea.

<sup>(2)</sup> F = Sexo femenino, y M = Sexo masculino.

<sup>(3)</sup> En el formato AAAAAMDD, al menos se cumplimentará el año de nacimiento. El mes y el día, si se desconocen, irán como 0000.

- (4) Nombre del país de nacionalidad.
- (5) En el formato AAAAMMDD.
- (6) Únicamente para aquellos establecimientos que no lo cumplimenten en soporte digital.
- (7) Pudiendo recogerse la firma en papel o en un soporte digital.

## ANEXO I

**Datos a facilitar en el ejercicio de la actividad de hospedaje**A) *Datos a facilitar en el supuesto de ejercicio profesional de la actividad*

## 1. Datos de la empresa arrendadora

- a) Nombre o razón social del titular.
- b) CIF o NIF.
- c) Municipio.
- d) Provincia.
- e) Teléfono fijo y/o móvil.
- f) Dirección de correo electrónico.
- g) Web de la empresa.
- h) Url para identificar el anuncio.

## 2. Datos del establecimiento

- a) Tipo de establecimiento.
- b) Denominación.
- c) Dirección completa.
- d) Código postal.
- e) Localidad y provincia.

## 3. Datos de los viajeros

- a) Nombre.
  - b) Primer apellido.
  - c) Segundo apellido.
  - d) Sexo.
  - e) Número de documento de identidad.
  - f) Número de soporte del documento.
  - g) Tipo de documento (DNI, pasaporte, TIE).
  - h) Nacionalidad.
  - i) Fecha de nacimiento.
  - j) Lugar de residencia habitual.
- Dirección completa.
  - Localidad.
  - País.
- k) Teléfono fijo.
  - l) Teléfono móvil.
  - m) Correo electrónico.
  - n) Número de viajeros.
  - o) Relación de parentesco entre los viajeros (en el caso de que alguno sea menor de edad).

## 4. Datos de la transacción

- a) Datos del contrato.
- Número de referencia.
- Fecha.
- Firmas.

- b) Datos de la ejecución del contrato.
  - Fecha y hora de entrada.
  - Fecha y hora de salida.
- c) Datos del inmueble.
  - Dirección completa.
  - Número de habitaciones.
  - Conexión a Internet (sí/no).
- d) Datos del pago.
  - Tipo (efectivo, tarjeta de crédito, plataforma de pago, transferencia...).
  - Identificación del medio de pago: tipo de tarjeta y número, IBAN cuenta bancaria, solución de pago por móvil, otros.
  - Titular del medio de pago
  - Fecha de caducidad de la tarjeta.
  - Fecha del pago.

*B) Datos a facilitar en el supuesto de ejercicio no profesional*

1. Datos del titular del inmueble

- a) Nombre.
- b) Primer apellido.
- c) Segundo apellido.
- d) Sexo.
- e) Número de documento de identidad.
- f) Tipo de documento (DNI, pasaporte, TIE).
- g) Nacionalidad.
- h) Fecha de nacimiento.
- i) Teléfono fijo y/o móvil.
- j) Correo electrónico.

2. Datos del inmueble

- a) Dirección completa y código postal.
- b) Localidad.
- c) País.
- d) Número de habitaciones.
- e) Conexión a Internet (sí/no).

3. Datos de los viajeros

- a) Nombre.
  - b) Primer apellido.
  - c) Segundo apellido.
  - d) Sexo.
  - e) Número de documento de identidad.
  - f) Tipo de documento (DNI, pasaporte, TIE).
  - g) Nacionalidad.
  - h) Fecha de nacimiento.
  - i) Lugar de residencia habitual.
- Dirección completa.
  - Localidad.
  - País.

- j) Teléfono fijo.
- k) Teléfono móvil.
- l) Correo electrónico.
- m) Número de viajeros.
- n) Relación de parentesco entre los viajeros (en el caso de que alguno sea menor de edad).

#### 4. Datos de la transacción

- a) Datos del contrato.
  - Número de referencia.
  - Fecha.
  - Firmas.
- b) Datos de la ejecución del contrato.
  - Fecha de y hora de entrada.
  - Fecha y hora de salida.
- c) Datos del pago.
  - Tipo (efectivo, tarjeta de crédito, plataforma de pago, transferencia...).
  - Identificación del medio de pago: tipo de tarjeta y número, IBAN cuenta bancaria, solución de pago por móvil, otros.
  - Titular del medio de pago.
  - Fecha de caducidad de la tarjeta.
  - Fecha del pago.

### ANEXO II

#### Datos a aportar en el ejercicio de la actividad de alquiler de vehículos

##### 1. Datos del arrendador/a

- a) NIF/CIF.
- b) Nombre de la compañía.
- c) Municipio.
- d) Provincia.
- e) País.
- f) Dirección completa del arrendador/a.
- g) Teléfono fijo y/o móvil.
- h) Web de la empresa.
- i) E-mail.

##### 2. Datos del arrendatario

- a) Nombre o razón social.
- b) Primer apellido.
- c) Segundo apellido.
- d) Sexo.
- e) Numero de documento de identidad.
- f) Tipo de documento (DNI, pasaporte, TIE, NIF).
- g) Nacionalidad.
- h) Fecha de nacimiento.