Guía para el desarrollo de TFG

Grado en Ingeniería Multimedia





Trabajo Fin de Grado

Autor:

Pablo García Carbonell

Tutor/es:

Elena Lloret

agosto 2022



# Resumen

# Motivación, justificación y objetivo general

Viajar es una de las mayores aficiones a lo largo del mundo, a todo el mundo le gusta visitar y conocer nuevos destinos y culturas. Sin embargo, antes de realizar cualquier viaje es necesario investigar sobre el lugar que vamos a visitar, viendo cuáles son las zonas de interés de la ciudad para cada usuario y eso no siempre es divertido. Perdemos mucho tiempo viendo qué puede ser lo más interesante para cada uno, leyendo comentarios y opiniones en páginas como *Tripadvisor[[1]](#footnote-1)* o *Trivago[[2]](#footnote-2)*.

Por ello, comencé a investigar un poco sobre el tema de *apps* y *webs* turísticas, llegando a la conclusión de que existen dos opciones a la hora de realizar el estudio previo al viaje: por una parte, contratar una agencia de viajes y que ellos se encarguen de organizarlo por ti; y por otra parte buscar tú por tu propia cuenta esas zonas que te puedan llamar la atención del lugar que vas a visitar. Cada opción tiene sus ventajas e inconvenientes: por ejemplo, al contratar una agencia no pierdes tiempo buscando e investigando, pero tienes que pagar a esta para que haga el trabajo por ti; y la otra forma es gratuita, pero pierdes tiempo indagando sobre el viaje que vas a hacer.

Esta decisión que tiene que hacer el usuario antes de viajar para planificar fue la principal razón que me motivó a elegir este trabajo de final de grado (TFG), queriendo realizar una combinación que recoja las ventajas de cada opción y elimine los inconvenientes, es decir, realizar un servicio gratuito que realice el trabajo de búsqueda por ti. Además, a nivel personal, viajar es uno de mis principales *hobbies*.

Por otra parte, el afán por aprender nuevas tecnologías es otro añadido por el que escogí realizar este trabajo de final de grado. En el caso concreto del elegido, deberé utilizar técnicas de procesamiento de lenguaje natural (PLN), que se suele implementar utilizando el lenguaje de programación Python, el cual no he estudiado en profundidad a lo largo del grado, y por tanto me va a permitir adquirir nuevos conocimientos y destrezas con este lenguaje.

Por tanto, utilizando técnicas de procesamiento de lenguaje natural, el objetivo de este TFG es proponer y desarrollar una aplicación web gratuita que ejerza una búsqueda de lugares de interés adaptado a las características del usuario (edad, objetivo por el que viaja), reduciendo así de manera considerable el tiempo que pasaría buscando e investigando sobre la ciudad destino.

# Agradecimientos

Esta sección se utiliza para escribir unos agradecimientos breves a todas aquellas personas, organizaciones, empresas, etc., que han contribuido a que el alumno o alumna haya alcanzado el final de su grado o que han colaborado en su trabajo: padres, familiares, pareja, tutor, amigos... Es voluntaria y completamente personal.

Este apartado no suele contener 1 página.

# Citas

Incluir una o varias citas que demuestren el carácter de tu trabajo, de tu personalidad y actitud en la vida.

*Cuando tenía 17 años, leí una cita que decía algo como: “Si vives cada día como si fuera el último, algún día sin duda estarás en lo cierto”. Causó una honda impresión en mí, y desde entonces, por los últimos 33 años, me he mirado en el espejo cada mañana y me pregunto: “Si hoy fuera el último día de mi vida, ¿querría hacer lo que voy a hacer hoy?” Y si la respuesta es no por demasiados días seguidos, sé que necesito cambiar algo.*

*Steve Jobs*

*Si algo se vuelve demasiado complicado, se atasca o no te convence: reinicia.*

*José Vicente Berná*

# Dedicatoria

También es posible añadir una sección dedicatoria, donde se deja constancia de a quién va dedicado el trabajo. Suele ser una simple frase o texto como el siguiente. El estilo es libre.

*A mi padre*

# Índice de contenidos

[Resumen 1](#_Toc135611006)

[Motivación, justificación y objetivo general 2](#_Toc135611007)

[Agradecimientos 4](#_Toc135611008)

[Citas 5](#_Toc135611009)

[Dedicatoria 6](#_Toc135611010)

[Índice de contenidos 7](#_Toc135611011)

[Índice de figuras 10](#_Toc135611012)

[Índice de tablas 11](#_Toc135611013)

[1. Introducción 12](#_Toc135611014)

[2. Estudio de viabilidad 14](#_Toc135611015)

[2.1. Análisis DAFO 14](#_Toc135611016)

[2.2. Análisis de riesgos 17](#_Toc135611017)

[3. Planificación 20](#_Toc135611018)

[4. Estado del arte. 21](#_Toc135611019)

[4.1 Procesamiento del Lenguaje Natural (PLN) 21](#_Toc135611020)

[4.1.1 Modelos para el PLN 21](#_Toc135611021)

[4.1.2 Niveles/Componentes del PLN 22](#_Toc135611022)

[4.1.3 Aplicaciones del PLN 23](#_Toc135611023)

[4.2 Estudio de la competencia 23](#_Toc135611024)

[4.2.1. Tripadvisor 23](#_Toc135611025)

[4.2.2 Triposo 25](#_Toc135611026)

[4.2.3 FourSquare 25](#_Toc135611027)

[4.2.4 Agencias de viaje 27](#_Toc135611028)

[4.2.5 ChatGPT 27](#_Toc135611029)

[4.2.6 Comparación de la competencia 30](#_Toc135611030)

[5. Objetivos 31](#_Toc135611031)

[6. Metodología 33](#_Toc135611032)

[6.1 Metodología *Scrum* 33](#_Toc135611033)

[6.2 *Kanban* 34](#_Toc135611034)

[7. Análisis y especificación 36](#_Toc135611035)

[7.1 Requisitos funcionales 36](#_Toc135611036)

[7.2 Requisitos no funcionales 38](#_Toc135611037)

[8. Diseño 40](#_Toc135611038)

[8.1. Diseño arquitectura conceptual 40](#_Toc135611039)

[8.1.1 Diferenciación de la ciudad en el formulario 41](#_Toc135611040)

[8.1.2 Filtro de lugares por usuario 41](#_Toc135611041)

[8.1.3 Creación del resumen 42](#_Toc135611042)

[8.2. Diseño API Rest 43](#_Toc135611043)

[8.3. Diseño arquitectura tecnológica Front/Backend 46](#_Toc135611044)

[8.3.1 Frontend 47](#_Toc135611045)

[8.3.2 Backend 47](#_Toc135611046)

[8.3.3 Traza frontend-backend 48](#_Toc135611047)

[8.4. Diseño Interfaces – UI 48](#_Toc135611048)

[8.4.1 Interfaz inicio 49](#_Toc135611049)

[8.4.2 Interfaz resultado 49](#_Toc135611050)

[8.5. Guías de estilos 50](#_Toc135611051)

[8.6. Diseño de pruebas y validación 50](#_Toc135611052)

[9. Implementación 51](#_Toc135611053)

[10. Pruebas y validación 53](#_Toc135611054)

[11. Resultados 54](#_Toc135611055)

[12. Conclusiones y trabajo futuro 55](#_Toc135611056)

[Referencias 57](#_Toc135611057)

[Apéndice I 59](#_Toc135611058)

# Índice de figuras

[Figura 1. Ingreso por año en España por el turismo. 12](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611085)

[Figura 2. Esquema de un análisis DAFO 14](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611086)

[Figura 3. Página de inicio de Tripadvisor. 24](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611087)

[Figura 4. Resultado de búsqueda de Madrid en Tripadvisor 24](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611088)

[Figura 5. Página principal de FourSquare 26](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611089)

[Figura 6. Resultado búsqueda FourSquare 26](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611090)

[Figura 7 Respuesta “lugares de interés para pareja de 50 años en Madrid” por el ChatGPT. 28](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611091)

[Figura 8. Respuesta “lugares de interés para jóvenes de 20 años en Madrid” por el ChatGPT. 28](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611092)

[Figura 9. Resultado a comida para jóvenes de 20 años por ChatGPT. 29](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611093)

[Figura 10. Metodología ágil Scrum. 33](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611094)

[Figura 11. Lienzo Kanban. 34](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611095)

[Figura 12. Kanban del proyecto desde Trello. 35](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611096)

[Figura 13. Diagrama de flujo de la app 40](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611097)

[Figura 14. Método get\_ciudad() 41](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611098)

[Figura 15. Conexión con el API 42](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611099)

[Figura 16. Búsqueda en Wikipedia 42](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611100)

[Figura 17. Creación del resumen 43](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611101)

[Figura 18. Referencias en Wikipedia 43](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611102)

[Figura 19. Respuesta de la ruta /api/envio 45](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611103)

[Figura 20. Entrada de la ruta /api/envio 45](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611104)

[Figura 21. Entrada de la ruta /api/resumen 45](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611105)

[Figura 22. Respuesta de la ruta /api/resumen 46](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611106)

[Figura 23. Arquitectura frontend backend 46](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611107)

[Figura 24. Diseño interfaz inicio 49](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611108)

[Figura 25. Diseño interfaz resultado 50](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611109)

[Figura 26. Primer boceto de la aplicación. 59](#_Toc135611110)

# Índice de tablas

[Tabla 1. Análisis DAFO 15](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611111)

[Tabla 2. Análisis de riesgos 17](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611112)

[Tabla 3. Análisis de la competencia 30](#_Toc135611113)

[Tabla 4. Requisitos funcionales 36](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611114)

[Tabla 5. Requisitos no funcionales 38](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611115)

[Tabla 6. Endpoints del API 44](file:///C:\Users\pablo\Desktop\TFG\Memoria%20TFG%20Pablo%20García%20Carbonell%20.docx#_Toc135611116)

# Introducción

Viajar es un estilo de vida, conocer otras culturas, otros comportamientos. El ser humano es un ser social y le gusta viajar [1]. Existen otros tipos de viaje, como puede ser viajar por trabajo o viajar para visitar a algún familiar o conocido. Sin embargo, también existen otros tipos de viaje, los que se realizan por diversión, como podrían ser viajes como forma de bienestar, viajes como capricho, viajes en familia o con amigos. Pero todos estos últimos tienen algo en común: viajar por turismo y ocio.

Los últimos años los viajes han sufrido un descenso considerable por una razón obvia, la pandemia causada por el COVID-19 [2], dando como ejemplo España, que se ve cómo ha perdido hasta 50 mil millones de euros en turismo a causa del coronavirus, como se puede ver en la Figura 1.

Gráfico, Gráfico de barras

Descripción generada automáticamente

Figura . Ingreso por año en España por el turismo.

(en asterisco (\*) se marcan los años con COVID)

<https://es.statista.com/estadisticas/673456/ingresos-anuales-por-turismo-extranjero-espana/>

Sin embargo, actualmente se podría decir que se ha superado la pandemia y convivimos con dicho virus sin problema. Por tanto, hoy en día la mayoría de gente aprovecha cualquier hueco para viajar, normalmente siendo escapadas de un par de días, fin de semana la mayoría de las veces [3].

Los amantes de los viajes, al querer visitar un país y otro sin parar, intentan planificar su viaje rápidamente, habiendo dos formas principales: la contratación de agencias de viaje como Viajes el Corte Inglés[[3]](#footnote-3) o *webs* como *Tripadvisor[[4]](#footnote-4)*, cada una con sus ventajas e inconvenientes. Al contratar una agencia no se pierde tiempo investigando, pero es necesario pagar a esta. De la otra forma, se realiza gratuitamente, pero es necesario realizar la búsqueda de forma personal y no por algún agente externo, perdiendo así bastante tiempo.

Entonces, ¿por qué no crear un servicio que recoja estas ventajas, pero suprima los inconvenientes? Es decir, una página web que realice la búsqueda de los lugares de interés por el usuario y de forma gratuita. Este es el principal objetivo de TouristApp, el trabajo de fin de grado que se recogerá en este documento.

Es posible que exista algún servicio parecido, pero el resultado suele ser muy general. Sin embargo, con esta propuesta se busca un resultado más concreto y orientado al usuario. Por ejemplo, en el caso de una escapada a Barcelona con el objetivo de ocio, no es lo mismo que realice el viaje un matrimonio de 50 años que un grupo de adolescentes.

Cogiendo como idea las 5 W del periodismo [4], la búsqueda de los lugares de interés se realizará mediante cuatro preguntas clave: dónde vas a viajar, cuándo es el viaje (fecha de inicio y fecha de fin), quién va a realizar el viaje (que cantidad de gente y rango de edad) y para qué viaja (con que objetivo). Respondiendo estas 4 preguntas se podrá dar un resultado muy concreto y preciso para el usuario, resolviendo así el problema comentado anteriormente de tener un resultado muy general.

Por tanto, con TouristApp se busca realizar un algoritmo de búsqueda de rutas turísticas adaptado a las condiciones, gustos y características de cada usuario basado en herramientas de procesamiento de lenguaje natural (PLN).

# Estudio de viabilidad

Antes de ponernos manos a la obra con el proyecto se realizará un previo estudio de viabilidad con el fin de tener una mejor perspectiva del proyecto a desarrollar, estudiando y analizando todas sus características y factores externos. Este estudio de factores internos y externos del producto se realizará mediante un análisis DAFO.

Una vez realizado este punto, se realizará un análisis de riesgos para poder ver los percances que pueden surgir a lo largo del desarrollo del TFG, y con ello preparar planes para poder evitar o solucionar todos los problemas que puedan aparecer.

## Análisis DAFO

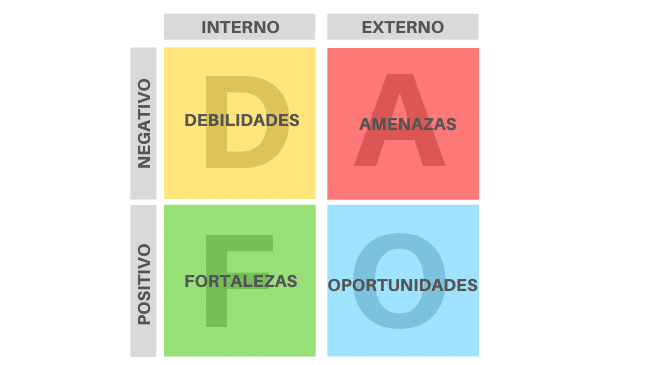
El análisis DAFO consiste en un esquema de trabajo utilizado para el estudio de un proyecto. Este esquema se divide en dos partes, factores externos e internos, las cuales se dividen en otras dos partes nuevas. La parte de factores externos se divide en **amenazas** y **oportunidades**, y la parte de factores internos en **debilidades** y **fortalezas**. De estas cuatro iniciales es de donde proviene el nombre este esquema: debilidades, amenazas, fortalezas y oportunidades, DAFO.

Figura . Esquema de un análisis DAFO

*Fuente*: <https://dircomfidencial.com/diccionario/analisis-dafo-20161113-1643/>

Como se observa en el esquema y se ha comentado anteriormente, el análisis DAFO se divide en dos secciones, las cuales se dividen en otras dos subsecciones. Los agentes internos, los cuales se pueden cambiar por el propio desarrollador del proyecto; y los agentes externos, los cuales afectan al proyecto, pero no dependen de él mismo, es decir, no son causados por el desarrollador. Sin embargo, aunque estos agentes externos no dependen del productor del proyecto, puede intentar cambiarlos evitando las amenazas o buscando las oportunidades.

En la Tabla 1 se muestra el análisis DAFO del proyecto para poder saber cuáles son los factores que mejorar y sobresalir entre las diferentes alternativas existentes en el mercado.

Tabla . Análisis DAFO

|  |  |  |
| --- | --- | --- |
| Factores externos | AMENAZAS  - Competencia con otras *apps* más conocidas. | OPORTUNIDADES  - Aprender a utilizar herramientas de PLN.  - Aprender *Python.*  - Conseguir experiencia.  - Pocas aplicaciones *web* con el mismo objetivo que el proyecto. |
| Factores internos | DEBILIDADES  - Falta de experiencia en proyectos de tales dimensiones.  - Creación y desarrollo de un proyecto en solitario.  - Poco conocimiento de Python y PLN. | FORTALEZAS  - Ambición tanto a la hora de afrontar el proyecto como de aprender nuevas tecnologías.  - Conocimiento en creación de aplicaciones *web*.  - A nivel personal, viajar es uno de mis mayores hobbies. |

En la Tabla 1 se ha realizado el análisis DAFO del proyecto, y a continuación se realizará de una forma más extendida:

**Factores externos:** agentes positivos y negativos externos al desarrollador del proyecto

* Amenazas: a la hora de hablar de las amenazas del proyecto no encontramos demasiadas más allá de la más obvia, que es la competencia con otras *apps* más conocidas, ya que existen alternativas a crear rutas turísticas o ver lugares de interés de una ciudad que visites.
* Oportunidades: a nivel personal y profesional, existe la oportunidad de aprender nuevas técnicas como son el uso de herramientas de procesamiento de lenguaje natural (PLN) y el lenguaje de programación de *Python*, con el que se implementarán la mayoría de los algoritmos de PLN. Por la parte más relacionada con el proyecto, la oportunidad de crear una *app web* como ninguna otra, aunque en amenazas se haya comentado que hay alternativas, existe la oportunidad de realizar un proyecto único como ningún otro. Además, todo esto aportará experiencia al desarrollador a la hora de realizar un proyecto de dicha magnitud.

**Factores internos:** agentes positivos y negativos del desarrollador del proyecto.

* Debilidades: La principal debilidad sería la falta de experiencia a la hora de realizar aplicaciones *web.* A lo largo de los cuatro años del grado se han realizado bastantes proyectos, pero siempre han sido de forma guiada, con pautas y ayudas. Sin embargo, ahora el proyecto se realizará desde cero y prácticamente sin guías ni ayudas. Además, realizar este proyecto de tal envergadura en solitario es otro problema a tener en cuenta, teniendo hacer todas las subtareas del proyecto por mí mismo. Por último, y hablando más de la parte del desarrollo en sí, el poco uso tanto de *Python* como de PLN a lo largo de la carrera es una gran debilidad, ya que tocará aprender un nuevo lenguaje de programación.
* Fortalezas: Ante las debilidades que han aparecido, la principal fortaleza es la ambición y las ganas, tanto de aprender nuevas tecnologías como *Python* o herramientas de PLN, como de sacar adelante un proyecto de tales dimensiones como es el TFG. Viajar es uno de los mayores hobbies a nivel personal, cosa que es un inventivo a la hora de querer afrontar este proyecto. Además, hay bastante conocimiento previo en la creación de aplicaciones *web*.

## Análisis de riesgos

Los riesgos de un proyecto son los distintos problemas e imprevistos que pueden surgir a lo largo del desarrollo del producto los cuales pueden llegar a parar o frenar el transcurso normal de este.

Por todo esto, se realizará un análisis de riesgos del proyecto con el objetivo de intentar evitarlos, y en el caso de sufrir un problema, poder sobrepasarlos de la manera más eficiente posible y menos perjudicial para el proyecto.

De cada riesgo se estudiarán los siguientes aspectos:

* **Riesgo:** descripción del riesgo que se puede sufrir.
* **Probabilidad:**posibilidad de que ocurra el riesgo. Determinaremos distintos niveles de probabilidad:
  + Improbable
  + Poco probable
  + Probable
  + Muy probable
* **Gravedad:** consecuencias que hay si el riesgo ocurre. Dentro de este aspecto se ha realizado una división para diferenciar la gravedad de cada riesgo:
  + **1:** Muy leve, retraso de un par de días, una semana como mucho**.**
  + **2:** Leve, retraso de semanas.
  + **3:** Grave, retraso de un mes.
  + **4:** Muy grave, tener que empezar el proyecto de nuevo.
* **Plan de contingencia:** método para solucionar un riesgo que ya ha sucedido de la manera más rápida y eficiente posible.
* **Plan de prevención:** método para prevenir que suceda un riesgo.

En la Tabla 2 se muestra el análisis de riesgos del proyecto:

Tabla . Análisis de riesgos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Riesgo | Probabilidad | Gravedad | Plan de contingencia | Plan de prevención |
| Enfermedad del desarrollador | Probable | 2 | Descansar y seguir las pautas del médico | Intentar cuidar la salud tanto física como mental |
| Pérdida del código del proyecto | Poco probable | 4 | Realizar el código de nuevo con todo lo aprendido anteriormente | Copias de seguridad en la nube y subir código en *GitHub[[5]](#footnote-5)* |
| Avería del ordenador | Probable | 3 | Arreglarlo de la mejor forma posible mientras se utiliza otro para no perder el hilo de trabajo | Tener todo ordenador y no descargar nada extraño de Internet |
| Mala planificación del proyecto | Poco probable | 3 | Retrasar las fechas de las tareas o realizar una nueva planificación con nuevas estrategias | Realizar una buena planificación mediante las estrategias y metodologías correctas. Trabajar de forma eficiente |
| No terminar a tiempo el proyecto | Poco probable | 3 | Intentar tener una buena planificación | Organizarse bien y trabajar eficientemente |
| Pérdida de la memoria del TFG | Poco probable | 4 | Al igual que con el código, volver a realizarla e intentar que quede lo más similar posible | Realizar guardados automáticos tanto en local como en la nube |
| Problemas con el lenguaje de Python | Probable | 1 | Seguir investigando y buscando información sobre el lenguaje y ver similitudes con otros conocidos | Realizar un estudio e informarse sobre el lenguaje para tener una buena base |
| Problemas con las herramientas de PLN | Probable | 1 | Investigar e indagar más sobre el tema además de buscar otras alternativas | Realizar un estudio previo sobre PLN y las librerías necesarias para la realización del proyecto |
| Elección incorrecta de las tecnologías | Poco probable | 2 | Ver las razones por las qué no ha funcionado una tecnología y ver qué otra se puede usar | Estudiar y documentarse bien sobre las tecnologías que vamos a usar y ver cuáles son las óptimas |
| Saturación entre el TFG y ABP[[6]](#footnote-6) | Probable | 2 | Reorganizar de forma correcta para poder compaginar los dos proyectos | Tener una buena organización e intentar solventar el ABP lo antes posible para poder centrar el 100% del trabajo al TFG |

# Planificación

Antes de ponernos manos a la obra con el desarrollo del proyecto, es necesario realizar una buena planificación del proyecto

# Estado del arte.

En este apartado se realizará una investigación sobre el tema que se tratará en el TFG y el entorno al que pertenece el desarrollo de este, es decir, a aplicaciones *webs* destinadas a crear rutas turísticas, las características que podemos encontrar y las tecnologías con la que son desarrolladas. Esto permitirá conocer más sobre el tema en el que nos vamos a manejar y así poder aplicar soluciones correctas en nuestro producto final.

Por una parte, se estudiará e investigará sobre el tema principal a tratar en este proyecto, el procesamiento del lenguaje natural, y por otra parte la generación de rutas turísticas, estudiando otras opciones existentes dentro del mercado.

## 4.1 Procesamiento del Lenguaje Natural (PLN)

El procesamiento del lenguaje natural [5] es el campo de conocimiento de la Inteligencia Artificial, computación y de la lingüística que estudia el comportamiento entre el humano y la máquina, es decir, estudia y hace posible las interacciones entre ambos. Desde que la máquina fue creada, el humano ha tenido la necesidad de comunicarse con ella. Hoy en día, la relación humano-máquina está aumentando de forma exponencial, no solo en comunicarse con la máquina sino también a través de ella, como ocurre con las redes sociales. Por tanto, el PLN busca permitir, facilitar y suavizar esta comunicación, por lo que este sistema debe conocer el lenguaje, como puede ser las propias palabras, cómo realizar enunciados con sentido, saber el contexto de estos, etc.

### 4.1.1 Modelos para el PLN

Tratar computacionalmente una lengua del mundo significa un proceso de modelización matemática. Normalmente, los mensajes son codificados en los lenguajes de programación C, Java o Python. Por tanto, hay que realizar una preparación para que el programador lo implemente en un código eficiente y funcional. Existen dos modelos principales:

* **Modelos lógicos** (gramáticas): Los lingüistas escriben reglas de reconocimiento de patrones estructurales, empleando un formalismo gramatical concreto. Estas reglas, en combinación con la información almacenada en diccionarios computacionales, definen los patrones que hay que reconocer para resolver la tarea (buscar información, traducir, etc.).
* **Modelos probabilísticos del lenguaje** (basados en ejemplos): La aproximación es a la inversa: los lingüistas recogen colecciones de ejemplos y datos y a partir de ellos se calculan las frecuencias de diferentes unidades lingüísticas y su probabilidad de aparecer en un contexto determinado. Calculando esta probabilidad, se puede predecir cuál será la siguiente unidad en un contexto dado, sin necesidad de recurrir a reglas gramaticales explícitas. Es el paradigma de “aprendizaje automático” que se ha impuesto en las últimas décadas en Inteligencia Artificial: los algoritmos infieren las posibles respuestas a partir de los datos observados anteriormente.

### 4.1.2 Niveles/Componentes del PLN

La arquitectura de un sistema de PLN se basa en una definición de lenguaje natural por niveles, los cuales son los siguientes cinco:

- **Nivel fonológico**: trata de cómo las palabras se relacionan con los sonidos que representan

- **Nivel morfológico**: trata de cómo las palabras se construyen a partir de unas unidades de significado más pequeñas llamadas morfemas

- **Nivel sintáctico**: trata de cómo las palabras pueden unirse para formar oraciones, fijando el papel estructural que cada palabra juega en la oración y qué sintagmas son parte de otros sintagmas

- **Nivel semántico**: trata del significado de las palabras, y de cómo los significados se unen para dar significado a una oración, también se refiere al significado independiente del contexto, es decir, de la oración aislada.

- **Nivel pragmático**: trata de cómo las oraciones se usan en distintas situaciones y de cómo el uso afecta al significado de las oraciones.

Una vez vistos los niveles podemos ver cómo sería el algoritmo que realiza la máquina una vez se le proporcione el mensaje:

1. El usuario expresa qué es lo que desea hacer

2. La computadora analiza las oraciones proporcionadas, a nivel morfológico y sintáctico.

3. El siguiente paso es analizar las oraciones semánticamente, es decir, saber cuál es el significado de cada oración.

4. Una vez realizado el paso anterior, podemos hacer el análisis pragmático de la instrucción, es decir, una vez analizadas las oraciones, ahora se analizan todas juntas, tomando en cuenta la situación de cada oración. Una vez realizado este paso ya tiene la expresión final.

5. Una vez obtenida la expresión final, el siguiente paso es la ejecución de ésta, para obtener así el resultado y poder proporcionárselo al usuario.

### 4.1.3 Aplicaciones del PLN

Una vez profundizado en el tema del PLN podemos ver qué tipo de aplicaciones lo utilizan. Las aplicaciones más comunes o conocidas son el análisis y síntesis de voz, la traducción automática, la recuperación de la información, respuesta a preguntas, extracción de la información, reconocimiento del habla o creación de resúmenes, entre otras.

En el caso de *TouristApp* utilizaremos herramientas de extracción de información para saber que ciudad quiere visitar el usuario, recuperación de información de cada lugar de interés y generación de resúmenes de toda la información recogida anteriormente.

## 4.2 Estudio de la competencia

Tras un estudio sobre aplicaciones *web* relacionadas con el turismo, se han encontrado algunos resultados cuyo objetivo es parecido al del proyecto, es decir, crear rutas turísticas o aconsejar sobre los sitios de importancia a nivel de turismo de una ciudad. Además de forma adicional, se pedirá al ChatGPT que realice una ruta turística, ya que viendo últimamente la cantidad de usuarios que hacen uso de este, por qué no probarlo con nuestro mismo objetivo.

### 4.2.1. Tripadvisor

Posiblemente a nivel de consejos a la hora de viajar, Tripadvisor[[7]](#footnote-7) sea la página *web* más conocida por la mayoría de los usuarios. Esta página recoge las reseñas de usuarios de distintos aspectos del turismo, como pueden ser los hoteles, restaurantes, cosas que hacer en el destino que visites. Su página principal es muy sencilla: consta de un simple buscador en el cual puedes poner tu ciudad destino, como se puede ver en la Figura 2.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura . Página de inicio de Tripadvisor.

Fuente: <https://www.tripadvsor.es/>

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamenteUna vez realices una búsqueda, te mostrará una lista sobre las entradas con mejores opiniones de los usuarios, ordenando estas entradas en tres grupos: qué cosas hacer, alójate y come. Además, la propia página proporciona una *review* sobre cada entrada. A continuación, en la Figura 3 se mostrará el resultado al buscar Madrid.

Figura . Resultado de búsqueda de Madrid en Tripadvisor

<https://www.tripadvisor.es/Tourism-g187514-Madrid-Vacations.html>

Esta opción de mercado a parte de recoger valoraciones, comentarios y opiniones de otros usuarios es gratuita y en la mayoría de casos te da la opción de comprar o reservar, ya sea una entrada para algún museo, un hotel, etc. Sin embargo, te da una lista de resultados muy general. Por ejemplo, al buscar Madrid, como se ha hecho anteriormente ha salido una lista de resultados de más de mil opciones en cada una de las secciones, es decir, ha dado como resultado mil entradas sobre restaurantes, mil entradas sobre hoteles y mil entradas sobre lugares de interés. Al ser un resultado tan amplio no se está adaptando realmente a las características del usuario, es decir, Tripadvisor daría el mismo resultado tanto a un hombre de 50 años que a una chica de 19.

### 4.2.2 Triposo

Triposo es una aplicación para móviles completamente gratuita que aconseja al usuario sobre dónde puede ir de viaje según sus gustos. La *app* conoce los gustos del usuario ya que este se los habrá proporcionado antes o los habrá extraído de Facebook[[8]](#footnote-8), al haber iniciado sesión con él anteriormente. Además, mientras el usuario esté de viaje, esta *app* recomendará lugares de interés al instante. Sin embargo, la aplicación solo está disponible en inglés, lo que dificulta el uso a una cantidad de usuarios, además de presentar el problema de la mayoría de *webs* de este ámbito: presenta un resultado muy general.

### 4.2.3 FourSquare

FourSquare [[9]](#footnote-9) es una aplicación tanto web como para móvil que, dada una ciudad y un objetivo, es decir, qué es lo que buscas, te da una lista de resultados. Como se puede ver en la Figura 4, la interfaz de su página de búsqueda es muy sencilla.

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Figura . Página principal de FourSquare

<https://es.foursquare.com/city-guide>

Mapa

Descripción generada automáticamenteA continuación, realizaremos una búsqueda para comprobar los resultados que proporciona, en este caso buscaremos Madrid, con el objetivo ‘Diversión’. Como se puede ver en la Figura 5, el resultado de la búsqueda se muestra de dos formas: una en forma de entrada, a la cual puedes acceder y ver las opiniones y valoraciones del usuario, y otra en forma de ubicación dentro del mapa de la ciudad que has indicado en el buscador

Figura . Resultado búsqueda FourSquare

<https://es.foursquare.com/explore?mode=url&ne=40.517192%2C-3.526096&q=Diversi%C3%B3n&sw=40.347591%2C-3.813801>

Foursquare da un resultado más preciso de lo que se quiere, es decir, se puede indicar si se busca un sitio para comer o un lugar de diversión. Sin embargo, se vuelve al mismo inconveniente que Tripadvisor, y es que el resultado es muy general, es decir, el mismo resultado saldrá tanto para un hombre de 50 años como para una chica de 19.

### 4.2.4 Agencias de viaje

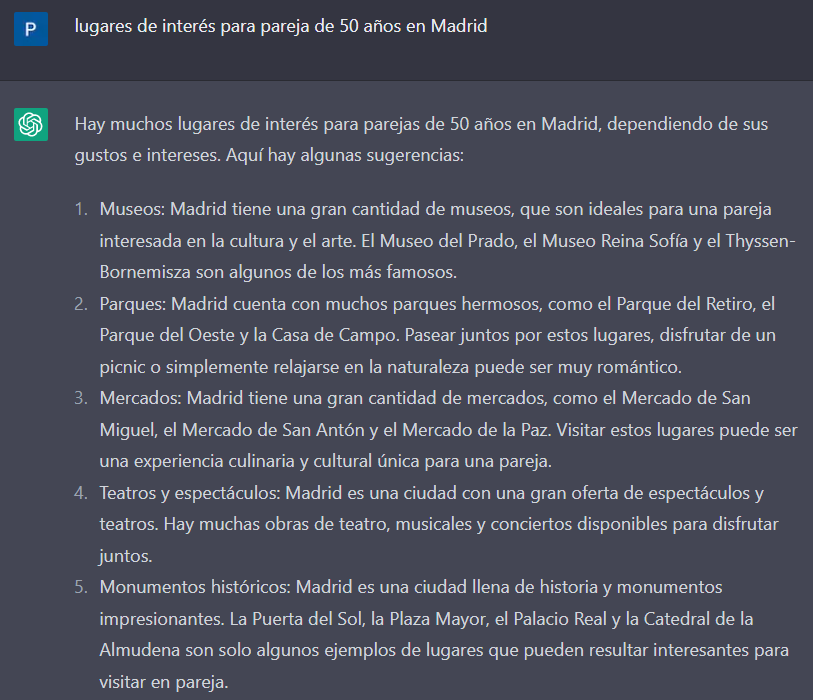
Otra opción puede ser contratar una agencia de viaje, como puede ser el caso de Viajes el Corte Inglés[[10]](#footnote-10), uno de los más conocidos a nivel nacional. Sin embargo, estas agencias se centran más en las partes de hospedaje y el medio de transporte por el que se viaja, es decir, se encargan de encontrar varias opciones de hotel y de medios de transporte. Una vez encontradas las mejores y dadas las mejores opciones al usuario, este tendrá que elegir cuál es la que más le gusta o la que más le conviene. Este no es principalmente el objetivo del proyecto. Las agencias se centran en la parte de llegar al lugar y hospedarse. Sin embargo, nosotros buscamos más la parte de turismo, qué zonas le interesan al usuario dependiendo de sus gustos. Además, las agencias no son gratuitos, se debe pagar por sus servicios.

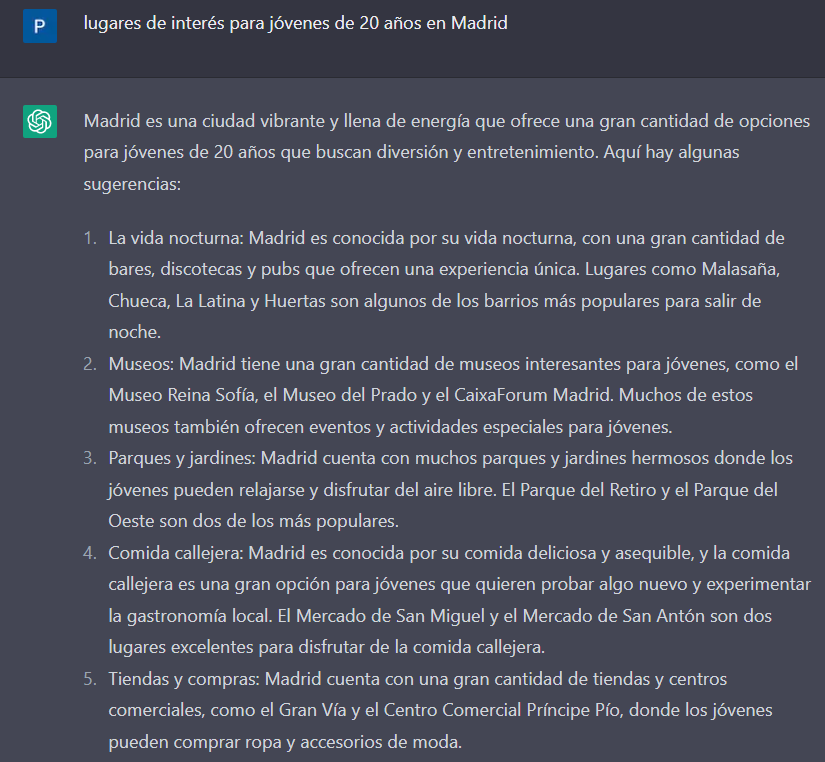
Este no es principalmente el objetivo del proyecto. Las agencias se centran en la parte de llegar al lugar y hospedarse. Sin embargo, nosotros buscamos más la parte de turismo, qué zonas le interesan al usuario dependiendo de sus gustos. Además, las agencias no son gratuitos, se debe pagar por sus servicios.

### 4.2.5 ChatGPT

El ChatGPT es [[11]](#footnote-11) una inteligencia artificial (IA) que en los en los últimos tiempos está de moda, ya que ha sido entrenada para poder tener conversación con cualquier persona. Lo más sorprendente de esta IA concreto es que es capaz de dar unas respuestas muy acertadas y completas, incluso de varios párrafos. Además, en estas respuestas es capaz de expresarte de manera natural y con información muy exacta, lo que hace muy complicado distinguir que el texto ha sido generado por IA o por un humano [6]. ChatGPT nos interesa porque realiza todo este tipo de búsquedas mediante herramientas de procesamiento natural.

Por probar cuál puede ser el resultado, se le pedirá al ChatGPT que nos realice una ruta turística por Madrid, primero para una pareja de 50 años, y después para un grupo de 20 años, y así se estudiarán los dos resultados, viendo si es capaz de adaptarse a las características del usuario.

En la Figura 6 podemos ver el resultado que daría a lugares de interés para el matrimonio de 50 años, y en la Figura 7 los lugares de interés para un grupo de 20 años, ambos viajando a Madrid.

Como se puede observar, el ChatGPT sí realiza una selección de lugares de interés en función de las características del usuario, diferenciando los lugares de ocio entre los dos casos dados. Además, puede llegar a ser más específico, como por ejemplo dando solo restaurantes y bares Texto

Descripción generada automáticamentesi se le solicita, como se puede observar en la Figura 8.

Figura Respuesta “lugares de interés para pareja de 50 años en Madrid” por el ChatGPT.

<https://chat.openai.com/chat>

Figura . Respuesta “lugares de interés para jóvenes de 20 años en Madrid” por el ChatGPT.

<https://chat.openai.com/chat>

Figura . Resultado a comida para jóvenes de 20 años por ChatGPT.

<https://chat.openai.com/chat>

Por tanto, el ChatGPT sí que da un resultado concreto y adaptado a las características del usuario.

En este proyecto se busca un resultado parecido al dado por ChatGPT, pero siendo un servicio solo de turismo y viajes, y no tan general como ChatGPT, que aborda todos los temas posibles. Además, si es posible, se apoyará la respuesta con algún tipo de elemento gráfico, como puede ser imágenes o algún mapa.

|  |  |  |
| --- | --- | --- |
|  | Ventajas | Inconvenientes |
| Tripadvisor | -Recoge opiniones variadas de distintos usuarios sobre cada entrada.  - Divide los resultados de búsqueda en ‘Come’, ‘Alójate’ y ‘Haz cosas’.  - Es gratuito.  - Interfaz sencilla e intuitiva. | - Resultado de búsqueda muy general y que no está adaptado a las características del usuario.  - Lista muy amplia de resultados (mil entradas aproximadamente por búsqueda). |
| Triposo | - Consejos sobre el viaje adaptados a los gustos y características del usuario.  - Es gratuito. | - Solo disponible en inglés.  - Resultado de búsqueda muy amplio (hasta 500 entradas por búsqueda). |
| FourSquare | - Disponible en *app web* y *app* móvil  - Interfaz sencilla e intuitiva.  - Doble resultado de búsqueda: tanto tipo entrada como ubicación dentro del mapa (apoyo gráfico).  - Se puede indicar el objetivo de la búsqueda (comida, ocio, deporte…)  - Es gratuito | - Lista muy amplia de resultados (hasta 700 entradas por búsqueda).  - Resultado de búsqueda muy general.  - Resultado no adaptado al usuario (mismo resultado a un usuario de 50 años que a uno de 20). |
| Agencias de viaje | - Se centra en los gustos del usuario  - No pierdes tiempo previo para estudiar las zonas de interés. | - Es de pago  - Se centra mayoritariamente en el hospedaje y en el medio de transporte |
| ChatGPT | - Es gratuito.  - Se centra en los gustos y características del usuario.  - No da una lista muy amplia de resultados (entre 5 y 10) | - No es una página centrada en el turismo.  - No se acompaña con elementos gráficos como imágenes. |

Tabla . Análisis de la competencia

### 4.2.6 Comparación de la competencia

Tras estudiar la competencia llegamos a la conclusión de que la mayoría dan un resultado muy amplio o un resultado no adaptado al usuario. Por tanto, buscaremos realizar un servicio que dé un resultado con un rango de 5 a 10 entradas, adaptado a las características dadas por el usuario y, si es posible, acompañado con algún elemento gráfico como imágenes.

# Objetivos

En este apartado se definirá el objetivo principal a alcanzar con el desarrollo del proyecto y los subobjetivos a desarrollar para llevar a cabo el objetivo principal.

Para ello se hará uso de la estrategia SMART (*Specific, Measurable, Attainable, Realistic and Timely*) [7], con la que podremos basar nuestros objetivos en unas características imprescindibles para dar valor a nuestro proyecto.

El objetivo principal del proyecto es proponer y desarrollar una aplicación *web* accesible y gratuita que proporcione al usuario una lista de los lugares que más le puedan interesar de una ciudad sabiendo las características y gustos de este, como la edad y el objetivo por el que viaja. Cada respuesta se acompañará tanto con elementos gráficos como por una pequeña reseña, incluso algún enlace directo a otra página externa, como a la página de un teatro para poder comprar entradas o algo por el estilo. La búsqueda de estos lugares de interés se desarrollará utilizando herramienta de procesamiento de lenguaje natural.

Además, surgen los siguientes subobjetivos:

* Diseñar de una interfaz sencilla e intuitiva.
* Elaborar un estudio de mercado de forma completa analizando las alternativas existentes a nuestro proyecto.
* Analizar y decidir las tecnologías más apropiadas para desarrollar el trabajo.
* Permitir a cualquier usuario acceder a la aplicación.
* Planificar el desarrollo de la aplicación mediante las herramientas de gestión de proyectos correspondiente.
* Evaluar la utilidad y la efectividad de la aplicación desarrollada.
* Aprender nociones básicas sobre el procesamiento del lenguaje natural y de las tecnologías del lenguaje humano, orientadas a la comprensión y generación de información.
* Aprender y utilizar herramientas y librerías específicas de Procesamiento de Lenguaje Natural, así como *frameworks* para diseñar arquitecturas.

A parte de estos subobjetivos a nivel profesional, surgen otros subobjetivos más enfocados a lo personal:

* Utilizar un lenguaje de programación nuevo como Python.
* Realizar un documento de las dimensiones de la memoria de un TFG.
* Aprender a realizar completamente una aplicación pasando por todas sus fases: desde investigar a fondo lo que existe ahora en el mercado, pensar qué se necesita – tecnologías, etc. - y cómo añadirlo en el proyecto, diseñar sus interfaces y, desarrollarlo hasta que esté terminada.

# Metodología

## 6.1 Metodología *Scrum*

La metodología que se usará a lo largo del proyecto será la metodología ágil de *Scrum*. La metodología Scrum permite abordar proyectos complejos desarrollados en entornos dinámicos y cambiantes de un modo flexible. Está basada en entregas parciales y regulares del producto final en base al valor que ofrecen a los clientes [8].

Además, en la metodología *Scrum* existen tres roles diferentes [9]:

* *Scrum Máster:* gestionar el correcto funcionamiento del proyecto y eliminar impedimentos
* *Product Owner:* encargado de optimizar y maximizar el valor del producto
* Equipo de desarrollo: desarrollar el producto, autoorganizándose y autogestionándose para conseguir entregar un incremento de software al final del ciclo de desarrollo.

En nuestro caso, al ser el desarrollo de un trabajo en individual no habrá reuniones de equipo, sino que serán con la tutora del TFG. En el caso de este proyecto, serán reuniones semanales. Y aunque no nos centremos en la parte de mejorar el trabajo en equipo, sí que usaremos esta metodología por la parte de las iteraciones, con límite de tiempo cada una, de modo de a cada una se le establecen unas tareas y funcionalidades.

En la Figura 10 se muestra una representación gráfica de la metodología *Scrum*. 

Figura . Metodología ágil Scrum.

<https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum>

## 6.2 *Kanban*

El método de *Kanban* es una forma de gestionar los proyectos basados en metodología ágil. *Kanban* es un método *Lean*, muy popular, de gestión del flujo de trabajo para definir, gestionar y mejorar los servicios que proporciona el trabajo de conocimiento. Te ayuda a visualizar el trabajo, maximizar la eficiencia y mejorar continuamente. El trabajo se representa en tableros Kanban, lo que te permite optimizar la entrega de trabajo a través de múltiples equipos y manejar, incluso los proyectos más complejos en un solo entorno [10].

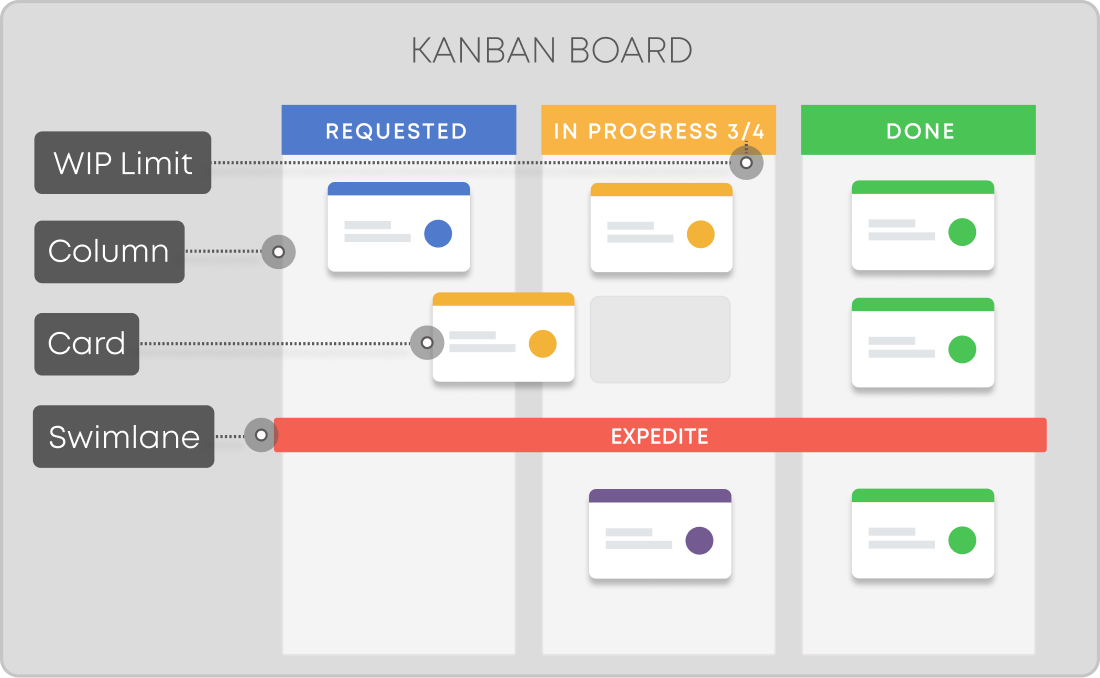


Figura . Lienzo Kanban.

<https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>

El lienzo *Kanban* consta de un tablero dividido en cuatro secciones principalmente:

* Lista de tareas: conjunto de todas las tareas a realizar
* *To do:* conjunto de tareas con más prioridad para hacer
* En proceso: conjunto de tareas que están realizándose en ese momento
* Hecho: conjunto de tareas que ya están acabadas

Para aplicar el método *Kanban* se ha utilizado la herramienta online de *Trello[[12]](#footnote-12)*, la cual te permite crear un lienzo y poder mover las secciones entre las diferentes columnas de forma sencilla. En la Figura 12 se muestra nuestro *Kanban* desde *Trello*.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura . Kanban del proyecto desde Trello.

*Fuente propia*

Fuen

Como se puede apreciar, he realizado una etiqueta por cada uno de los apartados en los que está dividida la memoria del TFG, asignando una tarea en *Trello* para cada uno de ellos. Además, de forma simultánea se ha realizado el desarrollo del proyecto, haciendo a la vez tanto este documento, es decir, la memoria del TFG, y la implementación del código del trabajo.

# Análisis y especificación

En la siguiente sección se diseñarán y definirán los requisitos a implementar para conseguir los objetivos del proyecto funcionales. Para realizar esta definición de forma correcta, se utilizará el estándar IEEE 830 [11], estándar que se ha utilizado además en otras asignaturas a lo largo de la carrera.

Este estándar IEEE 830 divide los requisitos en dos tipos, funcionales y no funcionales. Se usará la siguiente nomenclatura: RF – N para los requisitos funcionales y RNF – N para los no funcionales, siendo N el número de requisito. Además del identificador y el nombre del requisito, se definirá la prioridad y una pequeña definición de cada uno.

## 7.1 Requisitos funcionales

Los requisitos funcionales son definiciones de las funcionalidades que se van a desarrollar a lo largo del proyecto. En otras palabras, muestran qué es lo que se debe hacer para que el proyecto tenga un resultado óptimo y satisfaga las necesidades y expectativas de los usuarios. Por dar un ejemplo, una *web* de una tienda de ropa tiene como un requisito funcional tener sus productos actualizados a la última moda. A continuación, en la Tabla 4. se muestran los requisitos funcionales del proyecto.

Tabla . Requisitos funcionales

|  |  |
| --- | --- |
| Nombre | Realizar formulario con la información. |
| Identificador | RF - 01 |
| Descripción | Realizar formulario donde los usuarios puedan rellenar toda la información necesaria: dónde van, qué edad tienen, cuándo viajan y con qué objetivo |
| Prioridad | Prioritario |

|  |  |
| --- | --- |
| Nombre | Reconocer la ciudad dada por el usuario |
| Identificador | RF - 02 |
| Descripción | Diferenciar dentro de la información que ha dado el usuario la ciudad donde quiere viajar |
| Prioridad | Prioritario |

|  |  |
| --- | --- |
| Nombre | Reconocer la edad del usuario |
| Identificador | RF - 03 |
| Descripción | Diferenciar dentro de la información que ha dado el usuario la edad que tiene |
| Prioridad | Prioritario |

|  |  |
| --- | --- |
| Nombre | Reconocer el objetivo del viaje del usuario |
| Identificador | RF - 04 |
| Descripción | Diferenciar dentro de la información que ha dado el usuario el objetivo por el que viaja |
| Prioridad | Prioritario |

|  |  |
| --- | --- |
| Nombre | Reconocer el tiempo del viaje |
| Identificador | RF - 05 |
| Descripción | Diferenciar dentro de la información que ha dado el usuario cuándo va a viajar |
| Prioridad | Prioritario |

|  |  |
| --- | --- |
| Nombre | Proporcionar un resultado adaptado |
| Identificador | RF - 06 |
| Descripción | Con toda la información dada por el usuario, proporcionar un resultado adecuado a las características de cada uno |
| Prioridad | Prioritario |

|  |  |
| --- | --- |
| Nombre | Proporcionar información de cada resultado |
| Identificador | RF - 07 |
| Descripción | Realizar un pequeño resumen de cada uno de los lugares para acompañar el resultado final |
| Prioridad | Prioritario |
| Nombre | Proporcionar imagen de cada resultado |
| Identificador | RF - 08 |
| Descripción | Acompañar cada resultado con una imagen para dar más información sobre el lugar |
| Prioridad | Medio |

|  |  |
| --- | --- |
| Nombre | Realizar interfaz intuitiva |
| Identificador | RF - 09 |
| Descripción | Realizar interfaz en la que el usuario pueda desenvolverse sin ningún tipo de duda ni problema |
| Prioridad | Medio |

## 7.2 Requisitos no funcionales

Los requisitos no funcionales son aspectos del proyecto ajenos a las funcionalidades, como la accesibilidad, escalabilidad o usabilidad entre otros, es decir, aquellos requisitos que definen características del funcionamiento. Por tanto, el principal objetivo de los requisitos no funcionales es satisfacer las necesidades del cliente en aspectos de experiencia de usuario, sin incumplir ninguna restricción[[13]](#footnote-13). A continuación, en la se muestran los requisitos no funcionales del proyecto.

Tabla . Requisitos no funcionales

|  |  |
| --- | --- |
| Nombre | Usabilidad |
| Identificador | RNF - 01 |
| Descripción | La aplicación debe ser fácil de usar y de sencilla comprensión para el usuario. Realizando interfaces sencillas se conseguirá |

|  |  |
| --- | --- |
| Nombre | Accesibilidad |
| Identificador | RNF - 02 |
| Descripción | La aplicación debe permitir a cualquier usuario hacer uso de esta misma |

|  |  |
| --- | --- |
| Nombre | Disponibilidad |
| Identificador | RNF - 03 |
| Descripción | La aplicación debe estar disponible para cualquier tipo de usuario en cualquier momento |

|  |  |
| --- | --- |
| Nombre | Mantenibilidad |
| Identificador | RNF - 04 |
| Descripción | El código se programará de manera limpia para conseguir que sea reutilizable |

|  |  |
| --- | --- |
| Nombre | Fiabilidad |
| Identificador | RNF - 05 |
| Descripción | La aplicación estará preparada para manejar todas las acciones del usuario |

# Diseño

El siguiente apartado es el más importante dentro del TFG, ya que es la implementación de los requisitos tanto funcionales como no funcionales comentados anteriormente, con la meta de cumplir los objetivos marcados.

## Diseño arquitectura conceptual

La aplicación se ha dividido en tres partes fundamentales, por tanto, diremos que la arquitectura conceptual del proyecto es el conjunto de estos tres procedimientos.

Por una parte, tenemos la recogida de la información que nos da el usuario en el formulario, la cual usaremos para diferenciar sobre qué ciudad vamos a buscar información. Por otra parte, tenemos la búsqueda y filtro de los lugares que pueden interesar a cada usuario. Y, por último, la creación de un resumen con la información de los lugares que hemos determinado en el paso anterior, para finalmente mostrárselo al usuario junto con una imagen. Para que quede claro de manera más gráfica, en la Figura 13 se muestra un diagrama de flujo de la aplicación.

Diagrama

Descripción generada automáticamente

Figura . Diagrama de flujo de la app

### 8.1.1 Diferenciación de la ciudad en el formulario

El primero de los pasos es recopilar toda la información necesaria en el frontend y enviarlo al backend. Realizando un formulario en la página de inicio del proyecto hemos diferenciado tres principales campos: texto, objetivo y edad.

El primero, texto, es un input donde el usuario pone el lugar dónde quiere ir. Gracias a herramientas de PLN, se puede diferenciar cualquier ciudad dentro de un texto. Por tanto, si el usuario rellenase este campo escribiendo una frase como ‘Quiero viajar a Madrid’, nuestro backend diferenciaría que la ciudad es Madrid. En la Figura 14 se muestra el método *get\_ciudad()*, en el cual encontramos qué palabra indica una ciudad.

Texto

Descripción generada automáticamente

Figura . Método get\_ciudad()

En este método se le pasa por parámetro una variable llamada *span*, cuyo valor es el texto recogido del frontend. Gracias a la librería de *Spacy[[14]](#footnote-14)y* y a uno de sus métodos, *label\_,* podemos diferenciar qué tipo de palabra es cada una de las que conforma el texto. En este caso hemos filtrado por las palabras del tipo *LOC*, las cuales son localizaciones, por tanto, todas las ciudades se recogerán y se devolverán.

### 8.1.2 Filtro de lugares por usuario

Una vez tenemos la ciudad, el siguiente paso es encontrar cuáles son los lugares que puedan interesar al usuario dependiendo tanto de la edad como del objetivo por el que viaja el usuario. Este filtro de lugares se ha realizado en el backend, más concretamente en el método *create\_resource()*. En este método nos hemos ayudado de la inteligencia artificial para que realice el filtro de dichos lugares. En la *web* de RapidApi[[15]](#footnote-15) hemos encontrado varias opciones, eligiendo finalmente el api de inteligencia artificial llamada Glavier/chatGPT[[16]](#footnote-16). En esta función realizamos la conexión al api mediante código. En la Figura 15 se ve como se realiza la conexión.

Texto

Descripción generada automáticamente

Figura . Conexión con el API

Enviamos un *json* formado por dos partes: *payload* y *headers*. En la parte de *payload* es en la que escribimos el mensaje. El mensaje que enviamos es ‘Lugares de interés de **ciudad** para personas de **edad** años con el objetivo de **objetivo**’, siendo ciudad la palabra encontrada en el método comentado anteriormente, y edad y objetivo la información enviada al backend con los campos rellenados en el frontend.

### 8.1.3 Creación del resumen

Texto

Descripción generada automáticamentePara el tercer y último paso se ha creado el método *create\_summary()*. En este método se realiza una búsqueda en Wikipedia[[17]](#footnote-17) para sacar la información gracias a la librería de *Python wikipieda*-*api*, y una vez la tenemos, se ha realizado el resumen gracias a la librería *sumy* de *Python,* concretamente con las herramientas *LenRankSummarizer* y *Tokenizer.* En la Figura 16 se muestra cómo se realiza la búsqueda en Wikipedia.

Figura . Búsqueda en Wikipedia

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamenteRealmente se han realizado dos búsquedas, primero se busca el lugar en el buscador de Wikipedia y de todos lo resultados, cogemos el primero y nos guardamos el título de esa página. Una vez tenemos el título guardado realizamos la búsqueda con el api de Wikipedia gracias a la librería comentada anteriormente. Esta doble búsqueda se ha hecho para evitar fallos del siguiente tipo: tenemos como lugar de interés ‘El museo del prado’. Sin embargo, para que Wikipedia nos diese una página que podamos usar óptimamente deberíamos haber buscado ‘museo del prado’ sin el artículo **de**. Para evitar esta ausencia de página, realizamos una búsqueda previa para evitar esto. En la se muestra dentro de Wikipedia como salen los resultados de búsqueda en su buscador.

Figura . Resultado búsqueda Wikipedia

Como se ve, hay más de un resultado. Por ello cogemos la primera página, y ya dentro de esta nos guardamos su título para que no hay ningún tipo de fallo.

Una vez realizada la búsqueda, se almacena la información en la variable resumen. En la Figura 18 se muestra cómo se realiza el resumen.

Texto

Descripción generada automáticamente

Figura . Creación del resumen

Primero de todo, se hace un primer filtro para quitar todas las referencias de Wikipedia, en la Figura 19 se muestra a qué nos referimos con las referencias.

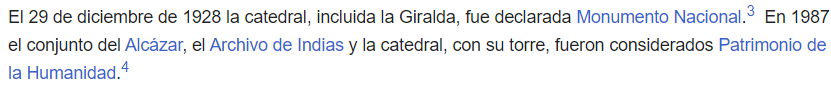


Figura . Referencias en Wikipedia

Estas referencias, una vez se ha procesado el texto con la librería se muestras de la siguiente manera: [1], por tanto, lo que realizamos es sustituir este tipo de cadena por una cadena vacía.

Una vez hemos filtrado para quitar las referencias, realizamos tres pasos para hacer el resumen: primero, cambia el texto al formato correcto para hacer el resumen con la herramienta *Tokenizer*. A continuación, se realiza un *ranking* de las palabras más importantes y de la más relevantes. Por último, realiza el resumen acorde con el número de oraciones que queramos, en nuestro caso solo una para que sea algo corto y sencillo de leer.

## Diseño API Rest

API es el acrónimo de interfaz de programación de aplicaciones (*application programming interface* en inglés). Es un conjunto de reglas bien definidas que se utilizan para especificar formalmente la comunicación entre dos componentes de software [13].

Las API son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. Suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor (la solicitud) y el que requiere el productor (la respuesta).

Por ejemplo, el diseño de una API de servicio meteorológico podría requerir que el usuario escribiera un código postal y que el productor diera una respuesta en dos partes: la primera sería la temperatura máxima y la segunda, la mínima.

En otras palabras, las API le permiten interactuar con una computadora o un sistema para obtener datos o ejecutar una función, de manera que el sistema comprenda la solicitud y la cumpla [14].

Para poder realizar el proyecto de manera correcta ha sido necesario realizar una pequeña API. Gracias a la herramienta de Postman[[18]](#footnote-18) ha sido fácil poder estudiar los endpoints, tanto la entrada del *body* como la respuesta de cada ruta.

|  |  |  |  |
| --- | --- | --- | --- |
| TIPO | ENDPOINT | BODY | RESPUESTA |
| POST | /api/envio | JSON que recoge del *input* del frontend | *Url* del lugar que corresponda |
| POST | /api/resumen | JSON que envía el lugar de interés que se va a buscar en Wikipedia | Resumen de la página con el algoritmo contado anteriomente |

Tabla . Endpoints del API

Como se puede ver hemos creado dos rutas para el desarrollo del proyecto. La primera se ha utilizado básicamente para enviar el texto del usuario desde el frontend al backend, analizarlo para diferenciar la ciudad y enviar en formato *JSON* la url que sea dependiendo de la ciudad que se haya escrito en el texto de entrada. La segunda se ha utilizado para enviar los lugares de interés y enviar el resumen de cada uno de ellos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteEn las Figura 20 y Figura 21 se muestra de manera más clara tanto la entrada como la respuesta de la llamada *POST* a /api/envio*.* Vamos a dar como ejemplo que el usuario escribiese “Me quiero ir a Madrid”.

Figura . Entrada de la ruta /api/envio

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Figura . Respuesta de la ruta /api/envio

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteEn las Figura 23 y Figura 22 se muestra más claramente tanto la solicitud como la salida de la llamada *POST*  a /api/resumen. Vamos a poner como ejemplo de lugar de interés ‘Museo del Prado’.

Figura . Entrada de la ruta /api/resumen

Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

Descripción generada automáticamente

Figura . Respuesta de la ruta /api/resumen

## Diseño arquitectura tecnológica Front/Backend

En el siguiente apartado se explicará detalladamente la arquitectura tecnológica que se utilizará para realizar nuestra aplicación *web*, explicando de forma concisa cada uno de los componentes que la conforman.

En este caso, la arquitectura escogida ha sido Frontend-Backend, ya que necesitaremos una parte de interfaz gráfica donde se recojan los datos que sean necesarios, y por otra parte una lógica de negocio que realice todos los métodos e implementaciones utilizando la información recogida en la interfaz para dar el resultado óptimo. Por esta principal razón se utilizará esta arquitectura, siendo el frontend la parte gráfica y el backend la lógica, teniendo que realizar una conexión entre ambas partes.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteEn la Figura 24 se muestra de manera gráfica como es una estructura básica de frontend-backend.

Figura . Arquitectura frontend backend

### 8.3.1 Frontend

El frontend sirve para realizar la interfaz de un sitio web, desde su estructura hasta los estilos, como pueden ser la definición de los colores, texturas, tipografías, secciones, entre otros. Su uso es determinante para que el usuario tenga una buena experiencia dentro del sitio o aplicación [12]. Por tanto, el frontend tiene una conexión fuerte con la experiencia de usuario y la interfaz de usuario (UX y UI).

En el caso de nuestro proyecto, para realizar el frontend se utilizará Angular[[19]](#footnote-19) , un *framework* para *apps web* desarrollado en TypeScript de código abierto que se utiliza. Angular se basa en clases tipo "Componentes", cuyas propiedades son las usadas para hacer el tratamiento de los datos. En dichas clases tenemos propiedades (variables) y métodos (funciones a llamar).

Cada componente está formado por tres ficheros fundamentales:

* **Componente.html**: este fichero es donde se estructura y se le da forma a la página.
* **Componente.css**: este archivo es donde se le da estilo al fichero componente.html.
* **Componente.ts**: este fichero se encarga de las distintas funcionalidades de las interfaces, como puede ser recoger información de un formulario y enviarla al servicio que corresponda.

Además, en los proyectos basados en Angular es necesario crear un nuevo fichero, normalmente conocido como servicio, el cual realiza la conexión entre el frontend y el backend.

Por dar un ejemplo más aplicado a nuestro proyecto, tenemos una interfaz llamada inicio.html donde hay un formulario, el cual le hemos dado estilo en inicio.css. Una vez se envíe la información, inicio.ts la recibe y la envía al servicio, enviar.service.ts, que hará una llamada a nuestra API, y por tanto, conectará con nuestro backend.

### 8.3.2 Backend

El backend son todos los códigos ocultos que sirven para que una página web o aplicación funcione correctamente. Además, de su estructura y organización depende la experiencia de usuario. De igual forma, el backend se encarga de optimizar otros elementos y recursos como la seguridad y privacidad en un sitio web o aplicación [12]. En nuestro caso, al no haber login ni nada del estilo no nos centraremos en esta segunda parte, sino que usaremos el backend para la lógica de negocio.

El backend se ha implementado en el lenguaje de *Python*, que como se comentó anteriormente, es el más apropiado para utilizar herramientas de PLN. Por una parte, se analiza el texto enviado desde el frontend gracias a la librería *Spacy[[20]](#footnote-20)*, la cual nos permite diferenciar los tipos de palabras que hay en el texto. Por lo que nos incumbe, hemos diferenciado qué palabras son ciudades.

### 8.3.3 Traza frontend-backend

Para hacer un resumen de cómo funciona esta conexión se va a hacer una pequeña traza con todos los pasos que se van realizando

1. El fichero del frontend inicio.component.ts recibe la información dada en el formulario creado en inicio.component.html
2. Inicio.component.ts envía la información al servicio creado de nombre envio.service.ts
3. Envio.service.ts hace una llamada a la url[*http://localhost:5000/api/envio*](http://localhost:5000/api/envio)*,* conectando así con el backend y pasando por parámetro la información recibida del formulario.
4. Ese *endpoint* analiza el texto que ha recibido, diferencia qué palabras representan una ciudad y devuelve una respuesta en formato *JSON* con la url de la ciudad.

## Diseño Interfaces – UI

Las interfaces cumplen un papel muy importante en las aplicaciones *web*. Para que una *web* sea efectiva, muchas veces necesita que sus interfaces sean fáciles de usar e intuitivas, ya que normalmente se busca que los servicios sean usados por el mayor número de personas posible. Por esta razón no es conveniente realizar una UI muy compleja y que pueda causar alguna contradicción o confusión.

En nuestro proyecto buscamos realizar una interfaz muy intuitiva, pero a la vez efectiva, que no dé lugar a ningún tipo de problema de entendimiento.

Para hacer los *mockups* de las interfaces de nuestro proyecto hemos utilizado la herramienta de *Balsamiq[[21]](#footnote-21).*

Nuestra aplicación *web* constará de dos interfaces, a las cuales llamaremos inicio y resultado para diferenciarlas.

### 8.4.1 Interfaz inicio

Interfaz de usuario gráfica

Descripción generada automáticamentePrimero hablaremos de la interfaz inicio, la cual será la principal, donde aparecerá el formulario que el usuario deberá rellenar con la información que vea necesaria. En la Figura 25 se puede ver cómo es la idea de interfaz de la página de inicio.

Figura . Diseño interfaz inicio

Como se puede apreciar, el formulario se divide en cuatro partes fundamentales: un cuadro de texto donde el usuario escribirá la ciudad que quiere visitar, una segunda parte donde el usuario indicará el objetivo por el que viaja, el siguiente para marcar las fechas de inicio y fin, y, para terminar, el rango de edad del usuario.

Se ha realizado la interfaz de esta forma para que no haya ningún tipo de confusión ni problema a la hora de rellenar la información necesaria.

### 8.4.2 Interfaz resultado

En la interfaz del resultado se mostrarán aquellos lugares de interés del usuario, tanto una imagen de cada lugar como un pequeño resumen o reseña de él. De manera adicional se podrán añadir algún tipo de enlace, como puede ser con un museo o un teatro que haya alguna exposición o concierto en los días que el usuario ha marcado que está de viaje.

En la Figura 26 se muestra como es la idea de la interfaz del resultado

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Figura . Diseño interfaz resultado

## Guías de estilos

### 8.5.1 Paleta de colores

La paleta de colores utilizada para *TouristApp*, además de negro y blanco para pintar la fuente, ha sido una gama de azules, como se puede apreciar en la Figura 27.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Figura . Paleta de colores

Se han escogido estos colores ya que el tema principal de nuestro proyecto es viajar, y el azul es un color que simboliza libertad, viaje, ya sea porque se le asocia con el color del cielo o con el tono del mar, cosas que simbolizan viajar. A la hora de hacer las hojas de estilos se ha utilizado el modelo de color, para poder jugar con la luminosidad del color azul. El modelo HSL está formado por tres componentes: matiz, saturación y luminosidad. El nombre son las iniciales de las tres componentes en inglés (*Hue, Saturarion, Luminosity*)*.*

### 8.5.2 Tipografía

La tipografía usada en el proyecto ha sido *TT-Travel*. Esta fuente tiene más de diez estilos, pero el elegido para ha sido *TT-Travel DemiBold.* En la Figura 28 se muestra como es esta tipografía.

Figura . Tipografía de TouristApp

## Diseño de pruebas y validación

Al tener una API para poder realizar el proyecto, es recomendable validar que todas las rutas funcionan de manera óptima. En el caso de TouristApp ha sido necesario crear una ruta para poder hacer la conexión de backend con frontend haciendo una llamada de tipo *POST*. Para confirmar que las llamadas a esta ruta han funcionado correctamente se ha utilizado la herramienta *Postman[[22]](#footnote-22).*

En esta herramienta se ha creado un proyecto llamado TFG donde hemos almacenado la petición.

# Implementación

En este apartado se documenta el proceso de desarrollo pero no se escribe el código. Es muy importante tener en cuenta que debe adecuarse a la metodología propuesta y exhibir evidencias de dicha metodología. Si se indicó por ejemplo una metodología iterativa, deberían indicarla las iteraciones realizadas y las funcionalidades que se incluyeron en cada una. Si la metodología es Scrum podrían mostrarse los sprints.

También se puede documentar aspectos como la preparación de los entornos en caso de que se hayan preparado entornos de desarrollo, integración, preproducción y producción, herramientas para el desarrollo y soporte (como por ejemplo gestión de versiones con Git).

Se indica la estructura del proyecto, la distribución de archivos, archivos de configuración a nivel de sistemas (por ejemplo, si es necesario modificar la configuración de nuestro servidor web Apache) y de aplicación (si nuestro aplicación por ejemplo tiene un idioma por defecto y se configura en alguna variable).

Se analizan los problemas ocurridos durante el desarrollo, las decisiones tomadas, los ajustes necesarios y su justificación (por ejemplo, descarte de funcionalidades por razones temporales).

Si se han desarrollado algún algoritmo o solución muy a destacar se puede mostrar el código comentado, pero se debe ser muy crítico con esto y es preferible mostrar pseudocódigo o algoritmos en vez de código puro. Se puede utilizar la sección de Anexos añadida al final para mostrar código, documentos de seguimiento, documentos manuales que han sido utilizados u otros elementos de interés.

Exhibir el producto final, sus interfaces destacables para comprobar que el resultado es similar al diseñado.

Habitualmente este apartado consistirá en varios subapartados, uno para cada esprint o fase del desarrollo. Se debería indicar claramente al inicio de cada subapartado el objetivo concreto de esa sección y luego dentro desarrollar los aspectos necesarios. Se suele mostrar cómo se hace una cierta acción por primera vez, por ejemplo, como se crea un componente en Angular, la técnica utilizada para el texturizado, el filtro aplicado en el vídeo. De esta forma cuando se vuelva a utilizar esto mismo simplemente se dice que se ha hecho igual que las veces anteriores. La idea es no pegar todo el código, pero sí ir dejando huellas de tu código o tu trabajo. Tal vez se aproxima a un “cuaderno de bitácora”. Ten en cuenta que aquí debería mostrarse como avanza el desarrollo. Por ejemplo, es habitual desarrollar primero las funcionalidades y en una fase final realizar la maquetación, por lo que se verían en las primeras capturas de pantalla las interfaces planas, sin colores y sin estilos gráficos, y en la fase de maquetación se verían como quedan definitivamente.

Es muy importante que desarrolles este capítulo al mismo tiempo que desarrollas porque si no, no podrás mostrar los pasos intermedios. Comienza siempre por una primera fase de preparación de entornos de trabajo, donde documentes perfectamente como has preparado tu entorno de trabajo, es decir, qué herramientas has instalado, que configuraciones has seguido, que versiones del software, si has utilizado un repositorio cómo lo has enlazado, etc.

Por último, muy importante, si has generado código, no olvides dejar un enlace claro y directo al repositorio donde el tribunal podrá ver tu proyecto. Y por supuesto, que este código quede en una rama final principal, o bien la *master* o bien una *release*. Deja un código limpio, comentado, coherente y estructurado. Y por supuesto, que sea público y accesible.

# Pruebas y validación

Una vez la proyecto ha sido finalizada deben incluirse las pruebas y validaciones diseñadas en el apartado anterior. La idea es demostrar que no solo se ha generado la herramienta esperada, sino que es fiable y robusta. Validar finalmente que resuelve el problema analizado y propuesto de una forma aceptable.

Es posible que muchas pruebas unitarias se hayan desarrollado durante la codificación, pero si por ejemplo se posee una batería de llamadas a la API en Postman, puede ser un buen momento comprobar que realmente todas pasan la ejecución.

Una de las pruebas que son muy deseables son las pruebas de campo con usuarios reales. Cada vez es más habitual que cuando termina el desarrollo, la herramienta, el videojuego, el corto de animación o la creación que se haya llevado a cabo, sea sometida al juicio de un conjunto de usuarios. Este proceso no es baladí, pues debe seleccionarse al conjunto de usuarios de prueba adecuado (si es un videojuego por ejemplo debe seleccionarse a personas que suelan jugar). Después diseñar qué es lo que queremos que el usuario haga con nuestro producto. Y finalmente facilitarles algún tipo de encuesta o test que recoja los resultados y nos permita analizar si el producto cumple con las expectativas.

Si se han definido indicadores es un buen momento para demostrar que las métricas son adecuadas, por ejemplo comprobar que el rendimiento cumple con unos mínimos o que algunas cifras son alcanzadas.

Si hay problemas en las pruebas se deben documentar y analizar a que son debidas proponiendo las posibles soluciones, estas pueden ser propuestas como trabajo futuro.

# Resultados

Analizar el producto final y mostrar los resultados. Si está en producción o ha sido probada por usuarios tester o reales indicar las valoraciones, el número de usuarios, si tiene contenidos, si está generando monetización o los indicadores de tráfico, que comentarios o valoraciones ha recibido nuestros resultados.

En este punto se trata de analizar cómo lo que se ha hecho ha mejorado el problema. Es decir, las interfaces y producto se han mostrado durante el diseño, aunque en esta sección se puede hacer un resumen de este. Pero estaría muy bien por ejemplo si hemos hecho una herramienta web, mostrar cómo de eficiente es el proceso que resuelve ahora. Si se trata de un corto con algún objetivo, pues analizar que impacto ha tenido, si hay noticias que hablan de él, etc. Si por ejemplo es un videojuego y se ha presentado en alguna gamejam, pues hablar de si ha ganado algún premio o a obtenido alguna mención. Si es una aplicación en producción y tiene por ejemplo algún plan de pagos o monetiza a través de publicidad, sería muy adecuado hablar de si ya ha generado ingresos.

Realizar también un análisis interno sobre los costes temporales planificados y los reales, analizar las diferencias entre ellos y justificarlas. Analizar también otros ajustes como los funcionales, si se deben a un exceso de funcionalidades, a otros problemas como posibles errores que producen retrasos o problemas con el hardware. Si has tomado tiempos a través de herramientas como Toggl se pueden mostrar gráficas y resúmenes de los tiempos generados.

También se puede hacer un recorrido por algunas asignaturas de la carrera enlazando los contenidos vistos con los desarrollados en el trabajo. De esta forma se pone en valor la puesta en práctica de las competencias desarrolladas durante la carrera. Es una buena reflexión que puede ayudar a hacer un análisis final sobre aquellas asignaturas más significativas para nuestro proyecto.

Esta sección no es para repetir cosas y dichas antes durante el desarrollo, sino para analizar qué resultado a producido al final nuestro trabajo. Recuerda que el TFG tiene una motivación, un problema de trasfondo y objetivo. ¿De verdad se ha logrado algo sobre este objetivo inicial?

# Conclusiones y trabajo futuro

Apartado de carácter obligatorio que todo trabajo debe contener. En este apartado se recogerán las conclusiones extraídas de nuestro trabajo. Cosas como recalcar el alcance de los objetivos, si se han logrado por completo o no, sobre los resultados obtenidos que repercusiones pueden tener en el entorno o comunidad destino, sobre la dificultad global del proyecto y tus propias impresiones personales.

Además, todo trabajo siempre está inacabado y por ello debemos finalizar las conclusiones con unas líneas de trabajo futuro indicando por ejemplo mejoras que deseamos realizar en posteriores versiones, una línea de trabajo para su puesta en producción, funcionalidades que no han podido ser implementadas e incluso si hay un plan de comercialización a la vista. También se pueden analizar costes futuros para un despliegue o necesidades de desarrollo más allá del alcance del TFG.

Sirva como ejemplo las propias conclusiones de la generación de este documento:

Tras varios años de experiencia donde se aprecia que los estudiantes tienen un gran problema en la estructuración de su TFG, tanto a nivel de contenidos como en planificación, y también en proporcionar un formato adecuado, se ha decidido solucionar el problema generando una guía que sirva tanto de plantilla y modelo de documento como de ejemplo y descripción de los posibles contenidos. Esta guía no es exhaustiva ni universal, pues está orientada a proyectos de tipo desarrollo software y pincela algunos otros tipos de proyectos, de ahí sus secciones tan orientadas a este proceso. Pero sin embargo puede servir como base a cualquier proyecto y TFG/TFM.

Se desprende de esta guía que el trabajo del ingeniero es precisamente el arte y técnica de aplicar los conocimientos científicos a la invención, diseño, perfeccionamiento y manejo de nuevos procedimientos en la industria y otros campos de aplicación científicos. En resumen, la creación e innovación. Por ello ha de poner su empeño no tanto en el desarrollo de la solución final sino en el proceso que sigue a lo largo de todo el proyecto y que le lleva hacia el producto final.

Este es el resultado de varios años de trabajo en la dirección de TFG/TFM y después de haber elaborado diversos recursos, talleres y seminarios, y creo que será un buen principio donde los alumnos podrán superar fácilmente la primera dificultad, el del documento vacío, y ponerse a trabajar siguiendo indicaciones claras y una hoja de ruta en forma de plantilla.

Como trabajo futuro inmediato queda por supuesto evaluar formal o informalmente el resultado de esta guía, recogiendo impresiones de mis propios alumnos de TFG/TFM junto con sus tasas de éxito, la calidad de los trabajos tras la evaluación del tribunal e incluso el impacto en otros TFG/TFM.

Como trabajo a largo plazo queda la mejora continua de dicha guía a partir de los comentarios de alumnos, profesores, compañeros y tribunales junto con los propios resultados de la evaluación.

# Referencias

1. Esta es la razón científica por la que alguien ama viajar (2016). <https://www.semana.com/hogar-y-familia/articulo/por-que-me-gusta-viajar/60709/>
2. Hinojosa, V. (2022). Cómo ha cambiado el COVID nuestra forma de viajar: ochos nuevos hábitos.<https://www.hosteltur.com/150165_como-la-covid-ha-cambiado-nuestra-forma-de-viajar-ocho-nuevos-habitos.html>
3. Turismo en España: las cifras que reflejan cómo los viajes y escapadas hicieron crecer la economía en 2022. (2023) <https://www.cronista.com/espana/economia-finanzas/turismo-en-espana-las-cifras-que-reflejan-como-los-viajes-y-escapadas-hicieron-crecer-la-economia-en-2022/>
4. Waisbord, S. (2019). The 5Ws and 1H of digital journalism. *Digital Journalism*, *7*(3), 351-358
5. Moreno, A (2022). Procesamiento del lenguaje natural, ¿Qué es?.<https://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural/>
6. Fernández, Y (2023). ChatGPT: qué es, cómo usarlo y qué puedes hacer con este chat de inteligencia artificial GPT-3. <https://www.xataka.com/basics/chatgpt-que-como-usarlo-que-puedes-hacer-este-chat-inteligencia-artificial>
7. Doran, G. T. There's a S.M.A.R.T. Way to Write Management's Goals and Objectives. Management Review, Vol. 70 (11), pp. 35-36. 1981
8. Hurtado Sánchez, J (2021). Cómo funciona la Metodología Scrum: Qué es y cómo utilizarla. <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>
9. Roche, J (2021). Scrum: roles y responsabilidades. Los 3 roles de la metodología Scrum. <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html>.
10. (2020). ¿Qué es Kanban? Explicación para principiantes. .<https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>.
11. IEEE (2008). Especificación de requisitos según el estándar de IEEE 830. <https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
12. (2022). Frontend y backend: qué son, en qué se diferencian y ejemplos. <https://blog.hubspot.es/website/frontend-y-backend>
13. (2022). Qué es una API REST, para qué sirve y ejemplos. <https://blog.hubspot.es/website/que-es-api-rest>
14. (2022). ¿Qué es una API de REST?. <https://www.redhat.com/es/topics/api/what-is-a-rest-api>

# Apéndice I

Se pueden añadir tantos apéndices como sean necesarios. Los apéndices recogen todo aquello que no es recomendable que esté en mitad del texto como por ejemplo código sobre el que se habla específicamente pero que es demasiado largo como para insertarlo en mitad de la memoria, documentos escaneados de originales y que han servido para extraer el análisis de datos, bocetos realizados a mano y que luego fueron digitalizados en las herramientas de diseño. Todo esto si se quiere dejar constancia puede ser insertado en estos apéndices.

Hay que tener en cuenta que todo apéndice tiene que estar referenciado siempre en la memoria, es decir, que en algún momento en el texto tiene que hacerse mención del apéndice y su contenido.

Además, el apéndice debe contener no solo imágenes sino también el texto suficiente para explicar su contenido pero sin repetir texto de la memoria. Por ejemplo, supongamos que se ha hecho un primer wireframe a papel que después ha sido digitalizado. En la sección de diseño correspondiente indicaríamos que:

*“… en el Apéndice I hemos dejado el primero boceto manual que aquí está digitalizado y que incluye los elementos a, b, c…”*

Mientras que en el Apéndice I indicaríamos lo siguiente, junto a la imagen bocetada:

*“A continuación se muestran los primeros bocetos hechos a mano alzada y que dieron paso a los posteriores diseños de la aplicación. Se incluyen aquí,* Figura 19*, para que pueda apreciarse como las primeras ideas son perfeccionadas posteriormente en los procesos digitales”.*



Figura . Primer boceto de la aplicación.

(Fuente SmartBlog <http://www.smartblog.es>)

Los apéndices no son secciones de relleno donde volcar contenidos con el objetivo de alargar la memoria. Es una sección complementaria de contenidos de cualquier tipo que deben ser valiosos para el trabajo.

1. <https://www.tripadvisor.es/> [↑](#footnote-ref-1)
2. <https://www.trivago.es/> [↑](#footnote-ref-2)
3. <https://www.viajeselcorteingles.es/> [↑](#footnote-ref-3)
4. <https://www.tripadvisor.es/> [↑](#footnote-ref-4)
5. <https://github.com/> [↑](#footnote-ref-5)
6. <https://eps.ua.es/es/ingenieria-multimedia/gestioncontenidos/que-es-abp.html> [↑](#footnote-ref-6)
7. <https://www.tripadvisor.es/> [↑](#footnote-ref-7)
8. <https://es-es.facebook.com/> [↑](#footnote-ref-8)
9. <https://es.foursquare.com/city-guide> [↑](#footnote-ref-9)
10. <https://www.viajeselcorteingles.es/> [↑](#footnote-ref-10)
11. <https://chat.openai.com/chat> [↑](#footnote-ref-11)
12. <https://trello.com/> [↑](#footnote-ref-12)
13. <https://visuresolutions.com/es/blog/non-functional-requirements/#:~:text=Los%20requisitos%20no%20funcionales%20(NFR,la%20confiabilidad%20y%20muchos%20m%C3%A1s.> [↑](#footnote-ref-13)
14. <https://spacy.io/> [↑](#footnote-ref-14)
15. <https://rapidapi.com/hub> [↑](#footnote-ref-15)
16. <https://rapidapi.com/Glavier/api/chatgpt53/> [↑](#footnote-ref-16)
17. <https://es.wikipedia.org/wiki/Wikipedia:Portada> [↑](#footnote-ref-17)
18. <https://www.postman.com/> [↑](#footnote-ref-18)
19. <https://angular.io/> [↑](#footnote-ref-19)
20. <https://spacy.io/> [↑](#footnote-ref-20)
21. <https://balsamiq.com/> [↑](#footnote-ref-21)
22. <https://www.postman.com/> [↑](#footnote-ref-22)