



TRABAJO FIN DE GRADO
GRADO EN INGENIERIA INFORMATICA

Covid-19 Reports

Visualización personalizada de datos del COVID-19

Autor

Jesús Miguel Jaldo Ruiz

Director

Juan Julián Merelo Guervós



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, Noviembre de 2020

Covid-19 Reports

Subtítulo

Jesús Miguel Jaldo Ruiz

Palabras clave: *software libre, covid-19, pandemia*

Resumen

Covid-19 Reports

Jesús Miguel Jaldo Ruiz

Keywords: *open source, covid-19, pandemic, floss*

Abstract

D. **Tutora/e(s)**, Profesor(a) del ...

Informo:

Que el presente trabajo, titulado **Chief**, ha sido realizado bajo mi supervisión por **Estudiante**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a Noviembre de 2020.

El/la director(a)/es:

(nombre completo tutor/a/es)

Jesús Miguel Jaldo Ruiz

Agradecimientos

Índice general

| | |
|---|-----------|
| 1. Introducción | 17 |
| 1.1. Motivación | 17 |
| 1.2. Alcance | 19 |
| 1.3. Objetivos generales | 19 |
| 2. Estado del arte | 21 |
| 2.1. Análisis de mercado | 21 |
| 2.1.1. Representación de datos Covid-19 en España | 23 |
| 2.1.2. Comparativa | 26 |
| 2.2. Recursos necesarios | 27 |
| 2.2.1. Telegram | 27 |
| 2.2.2. Python | 28 |
| 2.2.3. Git y GitHub | 31 |
| 2.2.4. Heroku | 32 |
| 3. Planificación | 33 |
| 3.1. Metodología utilizada | 33 |
| 3.1.1. Metodología Ágil | 33 |
| 3.1.2. Metodología Scrum | 35 |
| 3.2. Historias de usuario y tareas | 38 |
| 3.2.1. Historias de Usuario | 40 |

| | |
|--|-----------|
| 3.2.2. Tareas | 46 |
| 3.3. Temporización | 54 |
| 3.3.1. Sprint 0 | 54 |
| 3.3.2. Sprint 1 | 55 |
| 3.3.3. Sprint 2 | 58 |
| 3.3.4. Sprint 3 | 59 |
| 3.3.5. Sprint 4 | 60 |
| 3.4. Costes | 61 |
| 3.4.1. Costes de desarrollo | 61 |
| 3.4.2. Costes de producción | 62 |
| 3.4.3. Costes de operaciones | 62 |
| 3.4.4. Resumen de costes | 63 |
| 4. Análisis | 65 |
| 4.1. Requisitos del sistema | 65 |
| 4.1.1. Requisitos funcionales | 65 |
| 4.1.2. Requisitos no funcionales | 66 |
| 4.2. Diseño de la Interfaz | 67 |
| 4.2.1. Diseño del sitemap | 67 |
| 4.2.2. Diseño del wireframe | 68 |
| 5. Implementación | 81 |
| 6. Pruebas | 83 |
| 7. Conclusiones | 85 |
| 8. Trabajos futuros | 87 |

Índice de figuras

| | |
|--|----|
| 1.1. Preocupaciones de la población española. Fuente: [23] | 18 |
| 2.1. Página de inicio La Moncloa. | 23 |
| 2.2. Opciones de consulta de La Moncloa. | 24 |
| 2.3. Página de consulta del Covid-19 del Ministerio de Sanidad. | 24 |
| 2.4. Mapa con la incidencia acumulada por Comunidades Autónomas. | 25 |
| 2.5. Logotipo de Telegram. | 27 |
| 2.6. Tabla comparativa WhatsApp vs Telegram. Fuente: Xataka [21] | 27 |
| 2.7. Logotipo de Python. | 28 |
| 2.8. Logotipo de python-telegram-bot. | 29 |
| 2.9. Logotipo de pandas. | 30 |
| 2.10. Logotipo de Matplotlib. | 30 |
| 2.11. Logotipo de Git. | 31 |
| 2.12. Logotipo de GitHub. | 31 |
| 2.13. Logotipo de Heroku. | 32 |
| 3.1. Ciclo de entrega en proyectos ágiles. | 34 |
| 3.2. Ciclo de vida de Scrum. | 36 |
| 3.3. Estructura de la participación de los roles en el Scrum. | 38 |
| 4.1. Sitemap del Bot. | 67 |
| 4.2. Boceto de la pantalla principal. | 68 |

| | |
|--|----|
| 4.3. Boceto Menú principal de selección. | 69 |
| 4.4. Boceto Menú Comunidad Autónoma. | 70 |
| 4.5. Boceto Menús Incremento y Casos acumulados. | 71 |
| 4.6. Boceto Menús Fallecidos y Hospitalizaciones. | 72 |
| 4.7. Boceto Menú España. | 73 |
| 4.8. Boceto Menús Incremento y Casos por edad en España. | 74 |
| 4.9. Boceto Menús Casos acumulados y Fallecidos en España. | 75 |
| 4.10. Boceto Menús Casos por Región y Casos cada 100mil habitantes en España. | 76 |
| 4.11. Boceto Menús Casos cada 100mil hab. semana y Fallecidos cada 100mil habitantes en España. | 77 |
| 4.12. Boceto Menú Fallecidos cada 100mil hab. semana en España. | 78 |
| 4.13. Boceto Menús Ayuda e Información. | 79 |

Índice de tablas

| | |
|--|----|
| 3.1. Modelo Historia de Usuario. | 40 |
| 3.2. Historia de Usuario - Apariencia. | 41 |
| 3.3. Historia de Usuario - Funcionamiento. | 41 |
| 3.4. Historia de Usuario - Menú Principal. | 42 |
| 3.5. Historia de Usuario - Selección Comunidad Autónoma. | 42 |
| 3.6. Historia de Usuario - Selección datos. | 43 |
| 3.7. Historia de Usuario - Selección ayuda. | 44 |
| 3.8. Historia de Usuario - Selección información. | 44 |
| 3.9. Historia de Usuario - Despliegue. | 45 |
| 3.10. Historia de Usuario - Modularización. | 45 |
| 3.11. Modelo Tarea. | 46 |
| 3.12. Tarea 01. | 46 |
| 3.13. Tarea 02. | 46 |
| 3.14. Tarea 03. | 47 |
| 3.15. Tarea 04. | 47 |
| 3.16. Tarea 05. | 47 |
| 3.17. Tarea 06. | 47 |
| 3.18. Tarea 07. | 48 |
| 3.19. Tarea 08. | 48 |
| 3.20. Tarea 09. | 48 |
| 3.21. Tarea 10. | 48 |

| | |
|--|----|
| 3.22. Tarea 11. | 48 |
| 3.23. Tarea 12. | 49 |
| 3.24. Tarea 13. | 49 |
| 3.25. Tarea 14. | 49 |
| 3.26. Tarea 15. | 49 |
| 3.27. Tarea 16. | 49 |
| 3.28. Tarea 17. | 50 |
| 3.29. Tarea 18. | 50 |
| 3.30. Tarea 19. | 50 |
| 3.31. Tarea 20. | 50 |
| 3.32. Tarea 21. | 50 |
| 3.33. Tarea 22. | 51 |
| 3.34. Tarea 23. | 51 |
| 3.35. Tarea 24. | 51 |
| 3.36. Tarea 25. | 51 |
| 3.37. Tarea 26. | 51 |
| 3.38. Tarea 27. | 52 |
| 3.39. Tarea 28. | 52 |
| 3.40. Tarea 29. | 52 |
| 3.41. Tarea 30. | 52 |
| 3.42. Tarea 31. | 52 |
| 3.43. Tarea 32. | 53 |
| 3.44. Tarea 33. | 53 |
| 3.45. Tarea 34. | 53 |
| 3.46. Tarea 35. | 53 |
| 3.47. Tarea 36. | 53 |
| 3.48. Tarea 37. | 54 |
| 3.49. Capacidad de trabajo Sprint 0. | 55 |
| 3.51. Product Backlog. | 56 |
| 3.52. Capacidad de trabajo Sprint 1. | 57 |
| 3.53. Tiempo invertido en el Sprint 1. | 57 |

| | |
|--|----|
| 3.54. Capacidad de trabajo Sprint 2. | 58 |
| 3.55. Tiempo invertido en el Sprint 2. | 59 |
| 3.56. Capacidad de trabajo Sprint 3. | 59 |
| 3.57. Tiempo invertido en el Sprint 3. | 60 |
| 3.58. Capacidad de trabajo Sprint 4. | 60 |
| 3.59. Tiempo invertido en el Sprint 4. | 60 |
| 3.60. Resumen tiempo invertido | 61 |
| 3.61. Estimación coste de los empleados. | 62 |
| 3.62. Estimación costes del hardware. | 62 |
| 3.63. Resumen costes producción. | 62 |
| 3.64. Resumen costes producción. | 63 |
| 3.65. Resumen costes del proyecto. | 63 |

Capítulo 1

Introducción

Este proyecto es software libre, y está liberado con la licencia [\[12\]](#).

1.1. Motivación

Vivimos en tiempos extraños, desde diciembre de 2019 escuchamos hablar sobre la aparición de un brote epidemiológico de neumonía cuya causa era desconocida por aquel entonces en la ciudad china de Wuhan. Este brote tuvo lugar en el mercado de la ciudad y tras el aviso de las autoridades competentes de Wuhan a la OMS (Organización Mundial de la Salud) se empezó a investigar que podría estar provocando dicho brote. Esto hizo que tras varias investigaciones se conociera que la causa del mismo era la aparición de un nuevo coronavirus [\[17\]](#) de tipo zoonótico; al cual se le dio el nombre de Covid-19. Según declaraciones de la OMS, este virus presentaba un riesgo para la salud pública, bajo las regulaciones del Reglamento Sanitario Internacional [\[18\]](#) y posteriormente este sería considerado como una pandemia.

Poco a poco este nuevo virus fue extendiéndose, empezando por el continente asiático y por el resto del planeta. Todos y cada uno de los países afectados han aportado los datos de las diferentes incidencias que ha tenido el virus dentro de sus fronteras, aunque a pesar de ello seguimos sin conocer el alcance real de la pandemia ya que no existe un criterio común para la publicación de los datos, si no que cada país los calcula siguiendo los métodos que considera oportunos, por lo que la incidencia puede ser incluso mayor de lo mostrado por los datos, pero no vamos a centrarnos en como se calculan dichos datos.

Basándonos en esto, sabemos que existe una preocupación por parte de la población hacia el virus, a la cual le gusta estar informada. Hoy en día es importante estar bien informado, como nos muestra Gabriela Nova en su artículo

"Los beneficios de estar informado" [16]. Por ello, la mayor preocupación de la gente es conocer información a cerca del virus, de como está evolucionando, cuantas contagios, muertes y altas se han producido, entre otros datos. Podemos comprobar que actualmente el Covid-19 es una de las principales preocupaciones de la población, algunas de ellas que aparecen como consecuencia del propio virus. La Figura 1.1 refleja una tabla con las preocupaciones principales de la población, donde se puede apreciar que el Covid-19 es la segunda preocupación de los españoles.

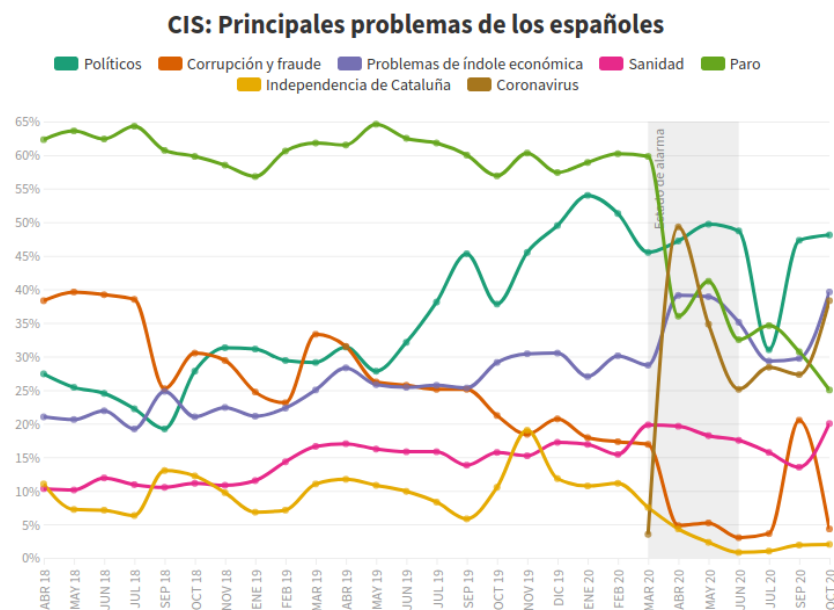


Figura 1.1: Preocupaciones de la población española. Fuente: [23]

Como hemos dicho, a la gente le gusta estar informada, y más ahora, por lo que la transparencia en una situación como la que se está viviendo es algo más que esencial. Esta información tiene como funcionalidad mantener a la gente tranquila, pudiendo aportarles datos de la evolución de la pandemia, pero no solo eso, si no que también sirven para hacer un análisis de cara a la implementación y toma de medidas por parte de los gobiernos de los países afectados. Sin embargo todos estos datos a pesar de ser accesibles para la población, presentan un problema, la difícil visualización de esto de una manera sencilla e intuitiva. En este punto podemos hacernos una pregunta: ¿cómo podemos ofrecer esa información de manera que pueda ser útil a toda la población que quiera consultar y hacer uso de ella?

La respuesta a la pregunta anterior podemos decir que puede llegar a ser obvia en los tiempos en los que vivimos, donde la tecnología se ha instalado para quedarse en nuestra vida, por lo que podríamos decir que la respuesta a nuestra pregunta es: hacer uso de la tecnología. Los smartphones es un producto que

hoy en día todo el mundo tiene, según la web Statista [10], en 2019 alrededor de los 2.650 millones de usuarios cuentan con uno. Para ser mas concretos el 96 % de los ciudadanos españoles mayores de 14 años que acceden a internet lo hacen desde su smartphone. De la misma forma un 92 % usa el móvil todos los días en una media de 3 horas diarias según el siguiente artículo [11].

1.2. Alcance

Este proyecto está pensado para poder llegar a una gran número de usuarios de todo el país. Lo que se pretende es que sean los propios usuarios los encargados de dar a conocer el proyecto. Sabemos que a veces el propio boca a boca es una de las maneras más eficaces de propagación, y en la época en la que nos situamos las personas suelen buscar recomendaciones sobre los productos que van a consumir o utilizar. Si tu como usuario estas contento con la aplicación se la recomendarás a más gente de tu entorno, lo que provocará no solo un aumento de usuarios, ya que estos se lo dirán a su vez a otros, con el fin de conseguir llegar al máximo número de personas posibles, proporcionándole información sobre la enfermedad.

Como hemos mencionado al inicio, toda la información puede ser consultada por la propia población, ¿entonces que sentido tiene este proyecto?. Entre otras cosas, cada Comunidad Autónoma tiene su propia manera de publicar los datos, a veces de una manera más sencilla y otras de una manera más compleja. Por ello lo que se pretende con **Covid-19 Reports** es poder agrupar toda esta información en un mismo lugar, facilitando el acceso a los usuarios y mostrando información adicional, por ejemplo, gráficos de la evolución de los datos seleccionados.

1.3. Objetivos generales

El objetivo principal de este proyecto es el desarrollo de un bot móvil orientado a informar a la población española para poder conocer los datos más relevantes sobre la pandemia del Covid-19 de una manera fiable. Este nos permitirá la consulta, tanto a nivel de todo el país como de las diferentes Comunidades Autónomas, de datos como el incremento de los casos desde el inicio de la pandemia, los casos registrados las últimas 24 horas, el número de fallecidos, los datos hospitalarios, etc.

Para llevar a cabo el proyecto se pondrá en práctica la metodología de desarrollo Scrum, la cual usaremos para la planificación y desarrollo del proyecto, con el fin de aprovechar las ventajas que esta nos ofrece.

Está claro, como hemos dicho antes, que una de las mayores preocupaciones de la población es el Covid-19, Por ello, **Covid-19 Reports** se ha llevado a cabo

con el fin de evitar una desinformación en cuanto a los datos y la evolución de esta pandemia se refieren, estando de forma libre y accesible para todos los usuarios que así quieran hacer uso de ella.

También es cierto, que estos mismos datos ya pueden ser consultados en diferentes lugares, como son las webs de los propios organismos de gobiernos, pero también lo es que cada organismo adapta esos datos y su acceso según ellos crean convenientes y adaptándose a estructuras en sus webs que ya disponían. Por todo ello, otro de los objetivos de este bot es la de no solo unificar, si no la de mostrar de una forma sencilla e idéntica los datos, evitando tener que aprender a como acceder a los datos o evitar la confusión del usuario si desea consultar los datos de otra comunidad, bien por ver la evolución, o bien para conocer la situación en todas las Comunidades Autónomas del país.

Capítulo 2

Estado del arte

En este apartado lo que pretende es mostrar como se encuentra el panorama en el cual vamos a llevar a cabo nuestro proyecto. Para ello, en primer lugar realizaremos un análisis del mercado, donde se presentarán las principales alternativas a nuestra idea de proyecto ya existentes, de forma que analicemos los requisitos de nuestro sistema comparándolo con los de aquellos que ya existen. En segundo lugar analizaremos punto por punto los recursos utilizados para poder llevar a cabo este proyecto.

2.1. Análisis de mercado

En el Capítulo 1 comentábamos que el acceso a los datos ya era algo posible para toda la población. Esto se debe a que son de dominio público y es el propio gobierno de España o las Comunidades Autónomas los que publican los datos en sus respectivas web. Dado que este proyecto está centrado en la visualización de datos, es muy importante poder realizar una comparación de los requisitos que queremos presentar en el mismo con las diversas funcionalidades de otras aplicaciones englobadas dentro del mismo género. Por desgracia, tras realizar una búsqueda de aplicaciones en la **Play Store** de Google a fecha de Noviembre de 2020, el resultado mostrado a sido bastante sorprendente. Según **Play Store**, no hay ningún tipo de aplicación similar a lo que se pretende desarrollar en este proyecto. Ejemplo de ello son las aplicaciones que nos sugiere:

- **Radar COVID:** Aplicación de alertas en caso de haber estado cerca de alguien que tiene COVID-19
- **GVA Coronavirus:** Aplicación de la Generalitat Valenciana para solicitar citas en caso de mostrar síntomas.

- **PassCOVIS.gal** Aplicación de la Xunta de Galicia para recibir avisos, información de las restricciones, recomendaciones y novedades.
- **COVIS-19.eus** Aplicación del Gobierno del País Vasco para realizar un auto-diagnóstico y avisar a tu círculo de personas.
- **CONFINAPP** Aplicación que pretende ser un acompañamiento y la puerta de entrada a la información y servicios de la Generalitat de Cataluña.

Visto que en el ámbito de las aplicaciones móviles no encontramos nada similar, tendremos que hacer una comparación con la principal fuente actual de información de la que disponemos, el **Gobierno de España**.

En este punto, antes que hacer el análisis, debemos conocer cuales son los requisitos que queremos tener en nuestra aplicación. Tras un estudio de los datos, siguiendo un criterio propio, he recogido las funcionalidades principales con las que debería contar una aplicación como la que se va a desarrollar y que permitan al propio usuario tener acceso a la misma sin ningún conocimiento previo. Estas funcionalidades por consiguiente sí están presentes en **Covid-19 Reports** y son las siguientes:

1. Seleccionar la Comunidad Autónoma sobre la que se quieren conocer los datos.
2. Visualización del incremento de casos, casos acumulados, media de la última semana y últimas 24h.
3. Visualización de los fallecimientos acumulados, media de la última semana y últimas 24 horas.
4. Visualización de los datos de hospitalización, camas y camas UCI ocupadas, porcentaje del total de camas, altas e ingresos.
5. Visualización de casos y muertes según la edad, solo al consultar datos de España.
6. Visualización de los casos por cada 100mil habitantes, solo al consultar datos de España.
7. Visualización de los casos por cada 100mil habitantes en la última semana, solo al consultar datos de España.
8. Visualización de los fallecidos por cada 100mil habitantes, solo al consultar datos de España.
9. Visualización de los fallecidos por cada 100mil habitantes en la última semana, solo al consultar datos de España.

10. Visualización ordenada y clara.
11. Toda la información es gratuita.
12. No tiene ampliaciones de pago.
13. No tiene anuncios.
14. Plataforma móvil donde se ofrece

2.1.1. Representación de datos Covid-19 en España

Como hemos dicho antes, llevaremos a cabo un análisis de como se representan los datos por parte del **Gobierno de España**. Para ello tendremos que acceder a la página web de **Gobierno de España**, donde podemos consultar los datos del conjunto del territorio español. Teniendo en cuenta que no se trata de una aplicación si no de una web, se ha creído conveniente que los puntos 11-14 no es necesario analizarlos.

A primera vista el acceso a la información desde el navegador aparenta ser sencillo. El usuario tiene que acceder a la web de La Moncloa [3], desde ahí se accederá a una pequeña pestaña llamada *Covid-19*, la cual mostrará una serie de opciones, como se puede ver en la Figura 2.1 y cuya opción a seleccionar será *Cifras de la situación*.

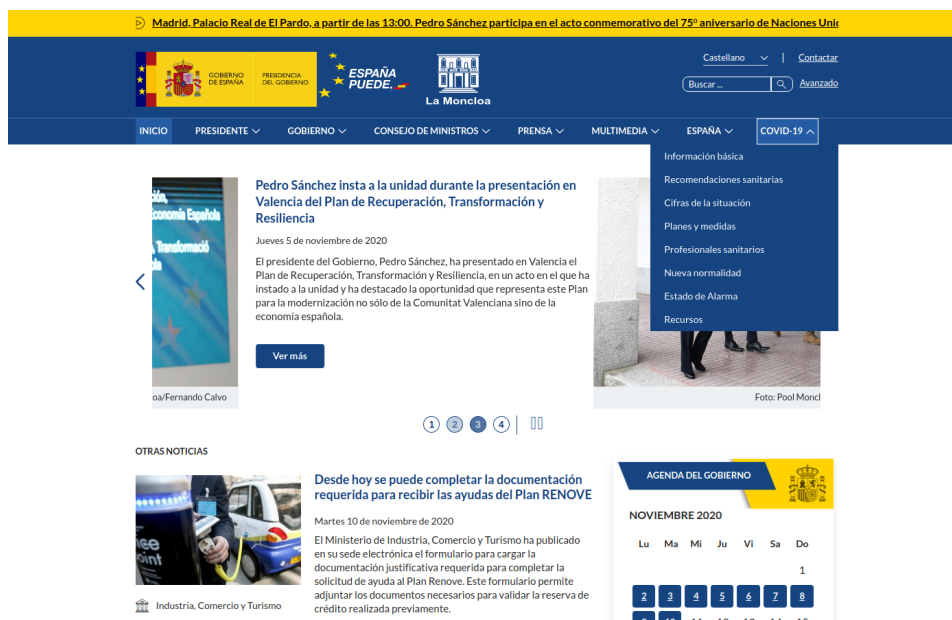


Figura 2.1: Página de inicio La Moncloa.

Una vez se ha redirigido al usuario, se le dará la opción de consultar los datos de España o los datos globales, junto con la opción de visualizar un video del proceso de recogida los datos en España, como se muestra en la Figura 2.2. Para este caso, la opción sobre la que nos centraremos es concretamente la que nos permite acceder a los datos a nivel de España, el usuario la seleccionará y será redirigido a una web del Ministerio de Sanidad [5], la cual vemos en la Figura 2.3.

Cifras de la situación

[España: datos de la situación actual del Ministerio de Sanidad, Consumo y Bienestar Social](#) - Actualización diaria

[Cifras globales y casos de COVID-19 por país de la Organización Mundial de la Salud](#) - En inglés

Video: [¿Cómo se recogen los datos epidemiológicos en España?](#) - 20/6/2020

Última actualización: 23 de junio de 2020.

Figura 2.2: Opciones de consulta de La Moncloa.

Situación actual



1.381.218
casos confirmados en España

12.715.902
casos confirmados en Europa

49.578.590
casos confirmados en el mundo

ATENCIÓN: Dada la evolución epidemiológica en nuestro país, la siguiente publicación de datos se producirá el lunes. Durante el fin de semana seguirán activos todos los sistemas de alerta de los servicios de Salud Pública de las CCAA para mantener la vigilancia y el control en todo el territorio español.

La actualización de los datos del fin de semana se realizará, por tanto, los lunes salvo que sea necesario adelantar esta comunicación.

Resumen de la situación

Esta información está en continua revisión.

- > Situación de COVID-19 en España [🔗](#)
- > Actualización nº246: enfermedad por SARS-CoV-2 (COVID-19) 09.11.2020 [🔗](#) [Escuchar](#)
- > Información inicial de la alerta en China 31.01.2020 [🔗](#) [Escuchar](#)
- > Análisis epidemiológico COVID-19 [🔗](#)
- > Pruebas de laboratorio [🔗](#) [Escuchar](#)

Figura 2.3: Página de consulta del Covid-19 del Ministerio de Sanidad.

Dentro de la página del Ministerio de Sanidad lo primero que llama la atención y más resalta es el total de casos confirmados a nivel de España, de Europa y del Mundo. Además se indica al usuario que los datos, debido a la situación excepcional, se publicarán los lunes, de manera que la evolución de estos se verá semanalmente. Dentro de las opciones que se permiten seleccionar, el usuario tendrá seleccionar dos de ellas, en primer lugar *Situación de COVID-19 en España*, la cual redirigirá al usuario a un mapa del territorio español interactivo donde lo único que se podrá consultar será la incidencia acumulada de las Comunidades Autónomas en los últimos 7 y 14 días como se muestra en la Figura 2.4.

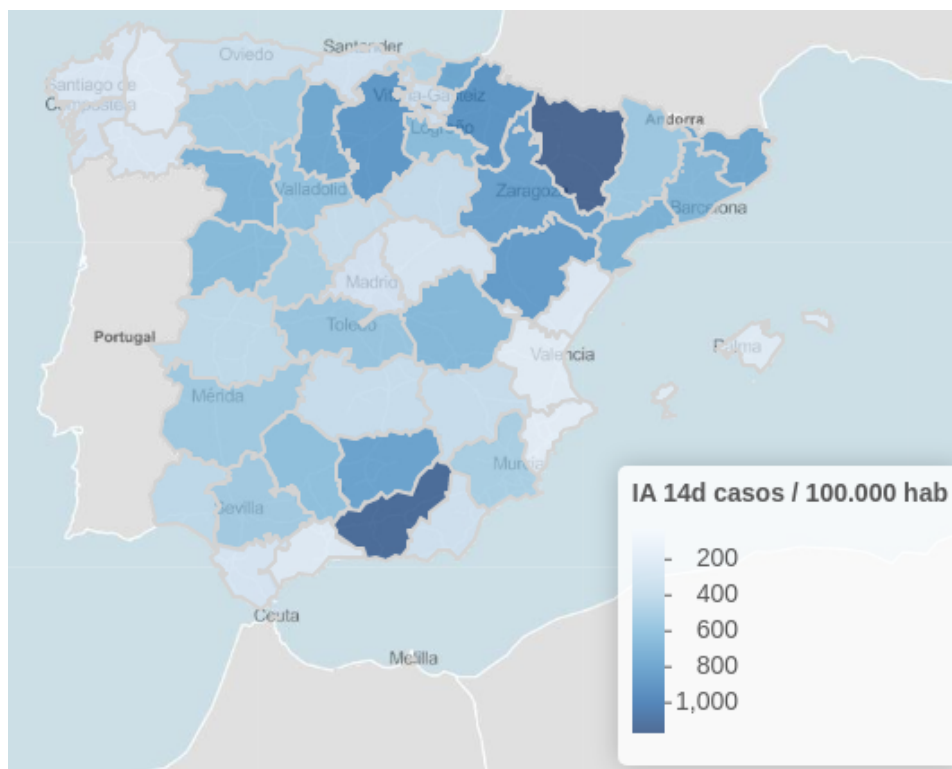


Figura 2.4: Mapa con la incidencia acumulada por Comunidades Autónomas.

Otras de las opciones que el usuario puede seleccionar es la *Actualización nº X*, que le permitirá visionar un pdf donde se encuentra toda la información unificada en tablas y posteriormente una serie de gráficas asociadas a las mismas. Dentro de toda la información que se nos proporciona, podemos ver la siguiente repartida en diversas tablas:

- Casos de COVID-19 confirmados totales, diagnosticados el día previo y diagnosticados o con fecha de inicio de síntomas en los últimos 14 y 7 días.
- Casos de COVID-19 que han precisado hospitalización e ingreso en UCI.
- Situación capacidad asistencial y actividad Covid-19 en hospitales.
- Total de Pruebas diagnósticas realizadas.
- Casos de COVID-19 que han fallecido (total y con fecha de fallecimiento en los últimos 7 días).
- Nº de casos COVID-19 importados de otro país.

- Detalles de los quince países con más casos confirmados de Europa
- Casos confirmados de COVID-19 fuera de Europa.
- Detalles de los quince países con más casos confirmados fuera de Europa

Si se quiere ver el modelo con el que se muestra esta información podemos acceder desde aquí [\[1\]](#).

Como se ha podido de ver en el análisis, el acceso a los datos que nos proporciona el **Gobierno de España** es más o menos sencillo, pero para la gente que apenas usa esta web, mostrar un archivo pdf donde se representen enormes tablas con diversos datos que, aunque relacionados, cuando superan cierta cantidad pueden llegar a ser confusos, y gráficas separadas por varias páginas de estas tablas de datos, no es la mejor práctica posible.

2.1.2. Comparativa

Una vez hecho el análisis de la única opción conocida que puede llegar a aportar datos similares a los que queremos mostrar con nuestra aplicación, procederemos a hacer una comparativa con los diferentes requisitos que definimos con anterioridad.

Como hemos dicho antes, no compararemos los puntos 11-14 porque estos se han considerado que están asociados a un aplicación móvil y no a una página web como es el caso. En cuanto a los requisitos restantes, en referencia al 1, es cierto que podemos ver los datos de las diferentes Comunidades Autónomas, pero al tratarse de un simple documento pdf es imposible realizar una selección de las mismas.

En cuanto al resto de requisitos (2-9), todos ellos son visibles dentro del documento que nos proporciona la página web. En cuanto al requisito 10, queda claro, como se ha mencionado antes, que la visualización tanto de datos y gráficas no es ordenada y mucho menos clara, sobre todo cuando se acumulan varios tipos de información.

Es cierto que en la información proporcionada por el **Gobierno de España** es muy completa, como debe esperarse de la máxima autoridad del país, pero eso no hace que sea perfecta. La principal idea de la aplicación que se va a desarrollar es poder facilitar el acceso a estos datos al usuario, permitiéndole acceder de manera sencilla e intuitiva a los mismos y seleccionar la información que desea utilizar. Es decir, lo que se pretende es una aplicación sencilla e interactiva, cosa que no se nos proporciona por parte del **Gobierno de España**.

2.2. Recursos necesarios

2.2.1. Telegram



Figura 2.5: Logotipo de Telegram.

Covid-19 Reports está concebido para ser un chatbot, por lo que la primera decisión que se ha tenido que tomar ha sido la plataforma donde se implementará. Hay muchas aplicaciones sobre las que se podría llevar a cabo el desarrollo de este chatbot, aunque para este punto se han considerado solo dos de ellas: WhatsApp [6] y Telegram [7]. En un artículo de la web Xataka Android [21], se lleva a cabo una explicación punto por punto de las diferencias entre ambas aplicaciones finalizando con una tabla que mostramos en la Figura 2.6

| | WHATSAPP | TELEGRAM |
|------------------------|----------|----------|
| USUARIOS | 1 | 0 |
| PRIVACIDAD Y SEGURIDAD | 0 | 1 |
| CHAT DE TEXTO | 1 | 1 |
| COMPARTIR | 0 | 1 |
| LLAMADAS | 1 | 0 |
| MULTIPLATAFORMA | 0 | 1 |
| PERSONALIZACIÓN | 0 | 1 |
| TOTAL | 3 | 5 |

Figura 2.6: Tabla comparativa WhatsApp vs Telegram. Fuente: Xataka [21]

Existen varias razones por las que se ha optado por hacer uso de **Telegram**, entre ellas podemos destacar las siguientes:

1. **Contenido disponible a través de Bots:** en el caso de **Telegram**, la lista de contenidos se ve multiplicada gracias a estos. Es cierto que **WhatsApp** también puede permitir la implementación de Bots, pero aún no es algo que se pueda realizar de manera generalizada, mientras que en **Telegram** cualquier usuario puede crear sus propios Bots.
2. **Cuentas asociadas a un nº de teléfono:** **WhatsApp** tiene sus cuentas de usuario enlazadas a los números de teléfono de los usuarios, lo que implica que no se puede chatear con alguien sin tener su nº de teléfono. Sin embargo, **Telegram** facilita la comunicación entre usuarios asignándole nombres de usuarios a los mismos. Esto implica un importante punto a favor de **Telegram** en el ámbito de la privacidad, ya que en **WhatsApp** mientras formemos parte de un grupo todos los usuarios que se encuentre en este pueden ver nuestro nº de teléfono, **Telegram** mostrará en su lugar el nombre de usuario, impidiendo al resto conocer nuestro nº de teléfono.
3. **Versión Web:** En este punto **Telegram** es el claro ganador. Aunque las dos aplicaciones cuentan con una versión Web, **WhatsApp** presenta un gran defecto que no se puede dejar pasar, y más a la hora de desarrollar una aplicación como esta: la estricta necesidad de tener el dispositivo móvil activo y conectado a Internet. En este aspecto, **Telegram** presenta una independencia total entre la versión de dispositivos y Web.
4. **Descargas de contenido:** Sabiendo que queremos mostrar diferentes imágenes para favorecer el entendimiento de los datos a los usuarios, el evitar llenar la memoria de sus dispositivos es algo muy importante. **Telegram** almacena estos contenidos en la nube, por lo que no será necesario descargarlos para poder visionarlos, mientras que en **WhatsApp** por el contrario si queremos visualizar este contenido, tendríamos que permitir su almacenamiento en la memoria del dispositivo.

2.2.2. Python

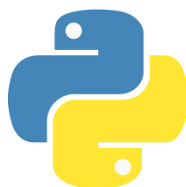


Figura 2.7: Logotipo de Python.

Como lenguaje de programación se ha optado por elegir **Python**. En primer lugar se debe a que es un lenguaje sobre el que he trabajado en varias ocasiones

y es uno de los lenguajes más fáciles de manejar. Otra de las razones por las que he decidido trabajar en **Python** se debe a que a la hora de analizar datos y poder representarlos de manera gráfica, acciones que he podido probar en diversos lenguajes de programación y que **Python** permite realizar de manera más sencilla. Y no solo por ello, si no también por su popularidad. Según un artículo de Stackscale [25], **Python** se ha convertido en el más utilizado, llegando a superar a **Java**, en los rankings de GitHub en 2019.

No solo por ello, se ha tenido en cuenta las características del proyecto y con que se va a trabajar. No solo es importante la interfaz del Bot, también lo es el contexto en el que trabajará, siendo en este caso el análisis de datos. Teniendo esto, aparece un nuevo lenguaje de uso común para el análisis de datos, **R**. Pero, ¿por que escogeremos a **Python** como el lenguaje principal de nuestro proyecto?

Según Paula Rochina en su artículo [22] estas son las principales causas por las que debemos decantarnos por **Python**:

1. **Python** es un lenguaje dinámico, lo que quiere implica que podremos cambiar el tipo de una variable o agregar nuevas propiedades o métodos a un objeto mientras el programa está en ejecución.
2. La curva de aprendizaje de **Python** no es muy exigente, es un lenguaje sencillo de aprender tanto para nuevos programadores como para programadores que quieren incrementar las habilidades que tienen del mismo.
3. Dado que el análisis de datos que se lleva a cabo en el proyecto no es independiente al mismo, si no que ha de estar integrado en nuestro Bot, **Python** es la mejor opción frente a **R**

Para una mayor seguridad a la hora de elegir este lenguaje mencionaremos a artículo de Planeta CHATBOT, escrito por Marvin G. Soto [24] donde, como resumen, **Python** es la mejor alternativa para un chatbot al admitir todo tipo de funcionalidades y garantiza un acceso rápido y fácil a la información y los servicios de la aplicación como usuarios.

Una vez decidido sobre que lenguaje se va a desarrollar el proyecto explicaremos las diferentes librerías o paquetes que van a ser necesarios para poder desarrollar el Bot: python-telegram-bot, pandas y Matplotlib.

2.2.2.1. python-telegram-bot



Figura 2.8: Logotipo de python-telegram-bot.

Se ha decidido escoger esta librería ya que proporciona al usuario una interfaz Python pura para el **Telegram Bot API**. Además de que es una librería que se actualiza con regularidad, ésta es compatible con las versiones más actuales de **Python**, siendo éstas las superiores a su versión 3.6. Además la biblioteca cuenta con algunas funciones para facilitar el desarrollo de Bots, haciéndolo mas fácil y sencillo.

Para mas información puede consultarse su repositorio en GitHub [\[20\]](#)

2.2.2.2. pandas



Figura 2.9: Logotipo de pandas.

la librería **pandas**, se considera una extensión de NumPy para poder manipular y analizar datos en **Python**. Se ha decidido utilizarla ya que previamente he trabajado con ella para aplicarla en ámbitos de Machine Learning, además en éste caso porque nos permite leer de manera fácil archivos de formato CSV (explicaremos más adelante porque hacemos uso de éstos archivos). También nos permite de una manera sencilla poder trabajar con tablas, acceder a sus índices, reordenarlas, modificarlas y combinarlas.

Para más información puede consultarse desde su web [\[19\]](#)

2.2.2.3. matplotlib



Figura 2.10: Logotipo de Matplotlib.

Matplotlib es una librería de **Python** dedicada a la generación de gráficos por medio de datos. Ésta combina a la perfección con **pandas**. Durante principios

de 2020 he tenido la oportunidad de trabajar con diferentes bibliotecas gráficas para **Python**, como Sklearn. Al trabajar con ambas al mismo tiempo me llegué a sentir mas cómodo con **Matplotlib**, ya que me parecía más sencilla de utilizar. Por ello se ha optado por su uso para este proyecto.

Para más información puede consultarse desde su web [15]

2.2.3. Git y GitHub



Figura 2.11: Logotipo de Git.

Git es un software de control de versiones. Está pensado para la eficiencia y la confiabilidad del mantenimiento entre versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. **Git** nace de la necesidad de llevar un control de los cambios de los archivos y sobre el trabajo que diferentes personas puedan realizar sobre archivos que se encuentran compartidos en un proyecto.



Figura 2.12: Logotipo de GitHub.

GitHub es una plataforma que nos permite alojar proyectos haciendo uso del **Git**. Se ha decidido utilizar esta plataforma para alojar nuestro código ya que se pretende que sea un proyecto de software libre haciendo uso de una licencia **AGPL** [12].

2.2.4. Heroku



Figura 2.13: Logotipo de Heroku.

Partiendo del punto en el que al crear un Bot para **Telegram** se obtienen unos **TOKENS** que son necesarios y están asociados al Bot en la aplicación y al ser un **TOKEN** que otorga manejo total sobre nuestro Bot no debe ser mostrado en ningún momento. Desde **Heroku** podremos proteger este **TOKEN**, permitiendo guardarlo con una variable de configuración, donde solo el dueño de la aplicación puede tener acceso a ellos y no son visibles en el código. **Heroku** es un PaaS (Platform as a Service) que nos permitirá poder desplegar nuestro Bot en la nube de manera que siempre esté en funcionamiento y podamos hacer uso de él en todo momento.

Para más información puede consultarse desde su web [\[13\]](#)

A lo largo de este capítulo se ha podido observar y analizar los puntos importantes del arte de nuestro proyecto, comparando con otras aplicaciones o webs que pueden asimilarse a lo que se quiere llevar a cabo, así como las principales herramientas que se van a utilizar para desarrollar el mismo.

Capítulo 3

Planificación

3.1. Metodología utilizada

Para poder planificar este proyecto se ha optado por hacer uso de una de las metodologías ágiles mas utilizadas hoy en día, la metodología Scrum. Scrum es una metodología ágil para el desarrollo de software la gestión de proyectos. Antes de centrarnos en definir en que consiste aplicar una metodología Scrum es muy importante conocer que es una metodología ágil.

3.1.1. Metodología Ágil

La metodología ágil, más que una forma para llevar a cabo proyectos que requieren de rapidez y flexibilidad podría llegar a concebirse como una filosofía que presenta una forma distinta de trabajar y organizarse, adaptándose a las condiciones cambiantes que pueden surgir, aprovechando los cambios para obtener ventajas. Por medio de esta lo que se pretende es dividir un proyecto en pequeñas partes, de manera que este se realice de forma incremental o por fases, permitiendo así que dichas partes se pueden completar y entregarse lo mas rápido posible.

Es importante tener en cuenta al definir la metodología ágil el propio manifiesto ágil [8], que contiene los valores sobre los que se asientan dichos métodos y fueron definidos en 2001. Estos valores son los siguientes:

- Individuos e interacciones sobre procesos y herramientas
- Software funcionando sobre documentación extensiva
- Colaboración con el cliente sobre negociación contractual

- Respuesta ante el cambio sobre seguir un plan

Llevar a cabo una implementación ágil consta de 2 fases: análisis y realización de Sprint como se muestra en la Figura 3.1.



Figura 3.1: Ciclo de entrega en proyectos ágiles.

A continuación veremos en que consisten ambas fases:

3.1.1.1. Fase 1: Análisis

Esta fase consiste en 4 pasos:

1. Preparación del proyecto durante el cual se definen los diferentes elementos, tales como labores, responsabilidades, estándares para la documentación y requisitos del hardware.
2. Proceso de visualización en el que se identifican cuidadosamente todos los procesos operativos y procesos que dependen de condiciones específicas, como seguridad, autorizaciones e interfaces. Estos resultados serán la base para la construcción de unos cimientos sólidos para el proyecto entero.
3. Funcionamiento de referencia del sistema, el cual se basará en el software de ERP.

4. Fase de evaluación. En esta fase, se determina la prioridad de los requisitos adicionales y las funcionalidades, en orden del valor del negocio. Después de esto, el equipo de implementación estima el esfuerzo que será necesario para realizar esto y determina la planificación de los Sprints para que se suministren los componentes del sistema.

3.1.1.2. Fase 2: Realización del Sprint

Esta fase consiste en 5 pasos:

1. Reuniones de planificación al comienzo de cada Sprint. Se define el objetivo del Sprint entre el propietario y el equipo de implementación.
2. Realización de los requisitos, de pruebas y documentación.
3. Reuniones diarias del estado del proyecto. En esta fase se registra el estado del proyecto y se discuten los diversos obstáculos que el equipo ha podido encontrar.
4. Sesión de prueba del Sprint. Durante esta fase los usuarios y el equipo IT determinan si los procesos cumplen los requisitos.
5. Se llevará a cabo una revisión del Sprint para comprobar que se puede mejorar en los futuros Sprints.

3.1.2. Metodología Scrum

Como hemos dicho al inicio de este capítulo, Scrum es una metodología ágil para el desarrollo de software o la gestión de proyectos, más específicamente, es un marco de trabajo a través del cual las personas pueden abordar problemas complejos adaptativos, mientras que se entregan productos de forma eficiente y creativa con el máximo valor [9].

Scrum está compuesto por diferentes procesos que se utilizan para la gestión del trabajo de producto complejos desde inicios de los años 90. Como tal, Scrum no es un proceso, una técnica o un método definitivo, es un marco de trabajo donde se puede emplear un conjunto de diversos procesos o técnicas. Scrum muestra la eficacia relativa de las técnicas de gestión de producto y de trabajo de modo que podamos mejorar de forma continua el producto, equipo y entorno de trabajo. El marco de trabajo de Scrum está compuesto por los Equipos Scrum, sus Roles, Eventos, Artefactos y Reglas asociadas. Cada uno de ellos sirve a un propósito específico y de la misma forma es esencial para el éxito de Scrum y para su uso.

En un principio, Scrum fue desarrollado para gestionar y desarrollar productos. Con el tiempo, y más a partir de 1990, se ha llegado a utilizar en diferentes ámbitos: desarrollo de software, hardware, redes funcionales, vehículos autónomos, escuelas, gobiernos, marketing y en la totalidad de todo lo que hacemos uso en nuestra vida, tanto como individuos y sociedad.

La esencia de Scrum reside en trabajar en pequeños equipos, siendo estos individualmente flexibles y adaptativos. Estas características siguen manteniéndose tanto en equipos individuales, varios, muchos u redes que equipos encargados de desarrollar, lanzar, operar y mantener el trabajo y el producto. Entre los diferentes equipos colaboran e ínter-operan a través del desarrollo de sofisticadas arquitecturas y objetivos en entornos de desarrollo.

3.1.2.1. Características de Scrum

Entre todas las metodologías ágiles, Scrum se basa en la teoría de control empírica de los procesos. Esto significa que se utiliza el progreso real de un proceso para poder planificar y concretar los lanzamientos. Scrum trabaja de forma que los proyectos están divididos en Sprints, los cuales tienen una duración de entre dos a cuatro semanas. Cuando finaliza un Sprint, se realiza una reunión entre los miembros del equipo y el cliente con el fin de mostrar al cliente el proyecto y que este pueda ser evaluado de cara a poder planificar las tareas que se deberán de utilizar en el siguiente Sprint como se ve en la Figura 3.2. Con esto lo que se pretende es que la dirección del proyecto pueda ajustarse o reorientarse una vez haya finalizado y durante el mismo mantener un control de los riesgos.

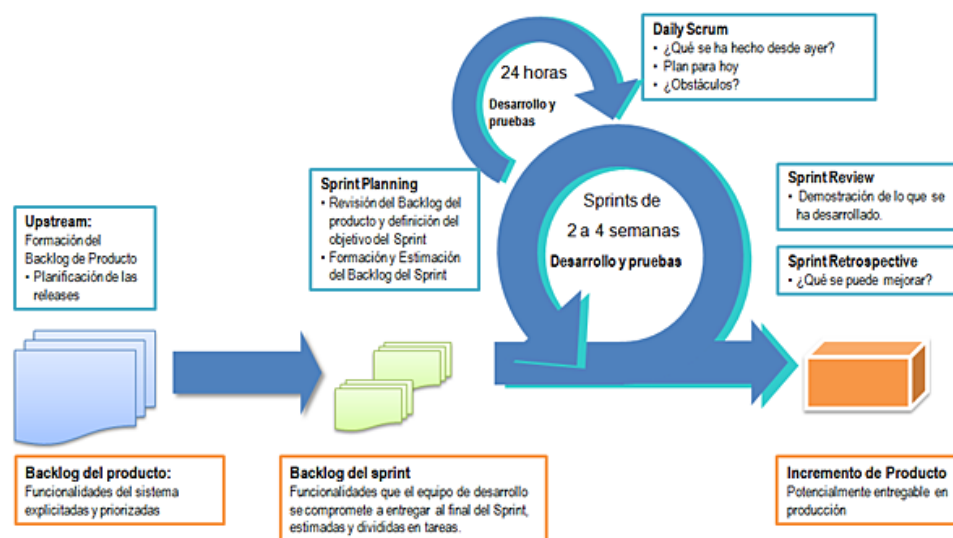


Figura 3.2: Ciclo de vida de Scrum.

Existen tres pilares sobre los que se fundamentará el control empírico de Scrum:

1. **Transparencia:** Los aspectos significativos han de ser visibles para todos aquellos responsables del resultado.
2. **Inspección:** Se han de inspeccionar frecuentemente los artefactos y el progreso hacia el objetivo, para detectar variaciones.
3. **Adaptación:** Capacidad para adaptarse en el caso de que el proceso desvíe de los límites aceptables.

3.1.2.2. Roles de Scrum

Un equipo de **Scrum** suele estar formado por unos pocos miembros, entre 3 y 9, sin incluir al Scrum Manager y el Product Owner. Cada uno de ellos tiene por tanto diferentes roles y responsabilidades dentro del proyecto. Procederemos a continuación a explicar las diferentes responsabilidades de cada uno de ellos

- **Product Owner:** Es el rol principal del proyecto. Suele ser el encargado de representar al cliente o ser directamente el propio cliente. ha de preservar los intereses del cliente priorizando las diferentes tareas para alcanzar los objetivos propuestos y establecer los diferentes requisitos del proyecto. De los tres roles, es el que mayor responsabilidades tiene, por lo que si algo no funciona o sale mal la responsabilidad recae sobre el. Para que el Product Owner pueda hacer bien su trabajo, todos han de respetar sus decisiones. A su vez el Product Owner debería evitar la supervisión de cada detalle del proyecto, pero sin embargo ha de estar disponible en caso de que el equipo de desarrollo tenga alguna pregunta o duda.
- **Scrum Master:** Es el rol encargado de actuar como enlace entre el Product Owner y el equipo de desarrollo. Como tal el Scrum Master ha de encargarse de resolver los diferentes conflictos que puedan obstaculizar el ritmo del proyecto. Este a su vez ha de incentivar u motivar al equipo de desarrollo para poder hacer visibles los logros del mismo ante el Product Owner. De cara al Product Owner, el Scrum Master le ofrecerá el apoyo necesario para poder maximizar los resultados.
- **Equipo de desarrollo:** Es el rol que se le da al conjunto del equipo que trabaja en el equipo, el cual será el encargado de terminar el trabajo. Estos equipos han de ser auto-organizados, multifuncionales, aunque cada miembro del equipo puede tener habilidades especializadas en un área, aunque la responsabilidad recae sobre el equipo y no sobre el individuo. A su vez, no se reconocen títulos dentro del equipo, independientemente

del trabajo que realicen sus miembros así como tampoco se reconocen los sub-equipo, independientemente de los dominios sobre los que se trabajen

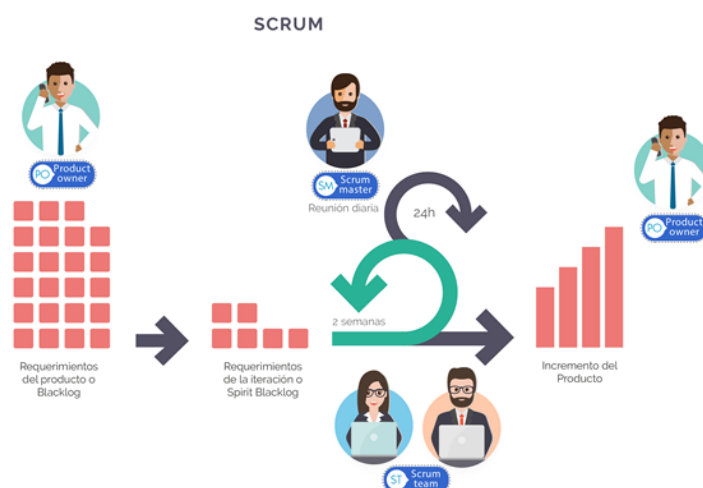


Figura 3.3: Estructura de la participación de los roles en el Scrum.

3.2. Historias de usuario y tareas

En este punto se tiene que ser capaz de plantear los objetivos que han de ser alcanzados a lo largo del desarrollo del producto final. Para hacer esto se hará uso de una serie de **Historias de Usuarios**, un elemento básico a la hora de aplicar metodologías ágiles en un proyecto y especialmente para poder aplicar **Scrum**. Las **Historias de Usuario** representarán de manera breve las características demandadas por el cliente las cuales deberán formar parte de la funcionalidad del producto, satisfaciendo sus exigencias.

El proceso por el cual se realiza la extracción de información relacionada con la funcionalidad del proyecto se debe llevar a cabo entre los miembros del equipo y el propio cliente. Al aplicar **Scrum**, este proceso no solo se realizará en la fase inicial del proyecto, si no que se realizan en cada Sprint del proyecto, de manera que se pueda obtener el resultado esperado en un corto espacio de tiempo y permitiendo amoldar el proyecto a lo requerido por el cliente de la forma mas eficiente posible.

Las **Historias de Usuario**, en términos generales, siempre han de extraerse durante las reuniones con el cliente y es deseable que sean escritas por el mismo y en un lenguaje claro, sin entrar en detalles. Estas han de aportarnos la funcio-

nalidad requerida por el proyecto, entregando de esta forma un valor particular al cliente.

Estas han de desglosarse en tres apartados:

- **Como:** representa el rol que va a utilizar el proyecto
- **Quiere:** representa la acción o evento que quiere que ocurra
- **Para:** representa la funcionalidad que se quiere cubrir.

A su vez también puede usarse la estructura presentada por la web Scrum Manager [2]:

- Nombre breve y descriptivo.
- Descripción de la funcionalidad en forma de diálogo o monólogo del usuario describiendo la funcionalidad que desea realizar.
- Criterio de validación y verificación que determinará para considerar terminado y aceptable por el cliente el desarrollo de la funcionalidad descrita.

Como se ha dicho antes, las **Historias de Usuario** ayudan a modelar el producto según las necesidades del cliente y mediante una reunión entre el equipo y este. El cliente aportará la idea que tiene, las necesidades que pretende cubrir y las funcionalidades que en cada momento el estima oportunas para el proyecto. A su vez, el equipo encargado del desarrollo también podrán aportar su punto de vista en ciertos puntos con la finalidad de poder enriquecer el proyecto. Para finalizar el Product Owner, que actúa como la voz del cliente dentro del equipo, será el encargado de redactar las **Historias de Usuario** y de extraer las diferentes tareas resultantes de las mismas, identificándolas según el coste su coste y prioridad. Con esto lo que se consigue es definir el Product Backlog, base a la hora de aplicar Scrum a un proyecto.

Es importante recalcar que en el Product Backlog se indicaran los diferentes Sprints del proyecto y las tareas asociadas a los mismos. Esto no quiere decir que se mantenga durante todo el proyecto, pues la definición obtenida al inicio del mismo puede variar dependiendo de las necesidades del cliente. Dado que el proyecto estará dividido en una serie de Sprints, durante los mismos debería realizarse una reunión entre los diferentes miembros del equipo y el cliente donde se podrán hacer adaptaciones que se consideren convenientes permitiendo cambiar o replantear los objetivos del proyecto con el fin de maximizar su utilidad.

Las **Historias de Usuario** serán definidas con la siguiente estructura:

| Historia de Usuario | |
|---------------------|------|
| ID | HUXX |
| Nombre | |
| Prioridad | |
| Riesgo | |
| Descripción | |
| Validación | |

Tabla 3.1: Modelo Historia de Usuario.

Cada campo de las **Historias de Usuario** representa lo siguiente:

- **ID:** Identificar único de la Historia de Usuario.
- **Nombre:** Nombre asignado a la Historia de Usuario
- **Prioridad:** Importancia a la hora de llevar a cabo en el desarrollo, pudiendo ser alta, media o baja
- **Riesgo:** Importancia en relación al conjunto del proyecto, indicando así en caso de fallo el daño provocado, pudiendo ser alto, medio o bajo. .
- **Descripción:** Explicación de la Historia de Usuario, dejando clara la idea de la misma
- **Validación:** Condiciones que se han de cumplir para dar la historia por finalizada.

3.2.1. Historias de Usuario

Las **Historias de Usuario** que se han creado para este proyecto son las siguientes:

| Historia de Usuario | |
|---------------------|--|
| ID | HU01 |
| Nombre | Apariencia |
| Prioridad | Media |
| Riesgo | Baja |
| Descripción | Como usuario quiero que el bot tenga un diseño simple, sencillo e intuitivo. |
| Validación | <ul style="list-style-type: none"> ■ Quiero acceder a la información pulsando un botón. ■ Quiero ver todos los datos de manera clara. ■ Todos los botones han de tener un funcionamiento claro. |

Tabla 3.2: Historia de Usuario - Apariencia.

| Historia de Usuario | |
|---------------------|---|
| ID | HU02 |
| Nombre | Funcionamiento |
| Prioridad | Media |
| Riesgo | Baja |
| Descripción | Como usuario quiero poder hacer uso del Bot en todo momento, principalmente en un dispositivo móvil, para poder consultar información en cualquier lugar. |
| Validación | <ul style="list-style-type: none"> ■ Quiero que el Bot funcione sobre todo en móviles. ■ Quiero que tenga un acceso fácil. |

Tabla 3.3: Historia de Usuario - Funcionamiento.

| Historia de Usuario | |
|---------------------|---|
| ID | HU03 |
| Nombre | Menú Principal |
| Prioridad | Alta |
| Riesgo | Baja |
| Descripción | Como usuario quiero tener un menú principal que se muestre en todo momento y me permita acceder a las diferentes utilidades del Bot. |
| Validación | <ul style="list-style-type: none"> ■ Quiero que el menú se muestre siempre por defecto salvo que el usuario decida ocultarlo. ■ Quiero poder acceder al menú principal del Bot. ■ Quiero poder acceder a la ayuda del Bot. ■ Quiero poder acceder a la información de desarrollo del Bot. |

Tabla 3.4: Historia de Usuario - Menú Principal.

| Historia de Usuario | |
|---------------------|---|
| ID | HU04 |
| Nombre | Seleccionar Comunidad Autónoma |
| Prioridad | Alta |
| Riesgo | Baja |
| Descripción | Como usuario quiero poder seccionar la provincia de la que quiero consultar los datos. |
| Validación | <ul style="list-style-type: none"> ■ Quiero poder seleccionar entre todas las CCAA del territorio español. ■ Quiero poder seleccionar toda España. ■ Las CCAA deben verse todas a la vez en pantalla. ■ Las CCAA han de mostrarse de manera clara y ordenada. |

Tabla 3.5: Historia de Usuario - Selección Comunidad Autónoma.

| | |
|---------------------|--|
| Historia de Usuario | |
| ID | HU05 |
| Nombre | Seleccionar datos |
| Prioridad | Alta |
| Riesgo | Media |
| Descripción | Como usuario quiero poder seccionar los datos que quiero consultar. |
| Validación | <ul style="list-style-type: none"> ■ Los datos han de mostrarse en diferentes apartados. ■ Pueden consultarse los datos mas actuales. ■ Pueden consultase los datos desde el inicio de la pandemia. ■ En algunos datos deben mostrarse gráficas (mostrar evolución de manera clara). ■ Poder consultar la acumulación de casos. ■ Poder ver el nº de fallecidos. ■ Poder ver el nº de hospitalizados. ■ Poder ver una lista de las provincias con más casos (solo al consultar España) ■ Poder ver las incidencias actuales por cada 100k habitantes por provincias (solo al consultar España) ■ Poder ver las incidencias desde el inicio de la pandemia por cada 100k habitantes por provincias (solo al consultar España) ■ Poder ver como afecta el virus dependiendo de la edad (solo al consultar España) |

Tabla 3.6: Historia de Usuario - Selección datos.

| Historia de Usuario | |
|---------------------|---|
| ID | HU06 |
| Nombre | Seleccionar ayuda |
| Prioridad | Alta |
| Riesgo | Baja |
| Descripción | Como usuario quiero poder acceder a una opción en todo momento por si tengo alguna duda. |
| Validación | <ul style="list-style-type: none"> ■ Esta opción ha de mostrarse en todo momento salvo que el usuario la oculte. ■ Esta opción ha de estar en el menú principal. ■ Quiero poder ver que es cada uno de los apartados del proyecto. |

Tabla 3.7: Historia de Usuario - Selección ayuda.

| Historia de Usuario | |
|---------------------|---|
| ID | HU07 |
| Nombre | Seleccionar información |
| Prioridad | Alta |
| Riesgo | Baja |
| Descripción | Como desarrollador quiero que el usuario pueda consultar datos sobre el proyecto. |
| Validación | <ul style="list-style-type: none"> ■ Esta opción ha de mostrarse en todo momento salvo que el usuario la oculte. ■ Esta opción ha de estar en el menú principal. ■ Quiero mostrar al usuario la información principal del desarrollo del proyecto. |

Tabla 3.8: Historia de Usuario - Selección información.

| Historia de Usuario | |
|---------------------|---|
| ID | HU08 |
| Nombre | Despliegue |
| Prioridad | Alta |
| Riesgo | Baja |
| Descripción | Como desarrollador quiero que el Bot funcione en todo momento. |
| Validación | <ul style="list-style-type: none"> ■ El Bot y sus datos han de estar en un contenedor. ■ El Bot ha de desplegarse como un PaaS. |

Tabla 3.9: Historia de Usuario - Despliegue.

| Historia de Usuario | |
|---------------------|--|
| ID | HU09 |
| Nombre | Modularización |
| Prioridad | Alta |
| Riesgo | Baja |
| Descripción | Como programador, quiero modularizar algunas partes de forma que sean fáciles de manejar y testear. |
| Validación | <ul style="list-style-type: none"> ■ Quiero de los diferentes módulos puedan pasar los test. ■ Quiero que el código esté dividido por funcionalidad. |

Tabla 3.10: Historia de Usuario - Modularización.

3.2.2. Tareas

A su vez, las **Historias de Usuario** estarán divididos en una serie de **Tareas**. Las **Tareas** serán definidas con la siguiente estructura:

| | |
|----------------------------|-----|
| Tarea | TXX |
| Historia de Usuario | |
| Estado | |
| Descripción | |

Tabla 3.11: Modelo Tarea.

Cada campo de las **Historias de Usuario** representa lo siguiente:

- **Tarea:** Identificar único de la Tarea.
- **Historia de Usuario:** Historia de Usuario a la que está asociada la Tarea
- **Estado:** Fase en la que se encuentra la Tarea dentro del proyecto, pudiendo ser No iniciada, En proceso o Completada
- **Descripción:** Explicación de la Historia de Usuario, dejando clara la idea de la misma

Las **Tareas** que se han creado para este proyecto son las siguientes:

| | |
|----------------------------|--|
| Tarea | T01 |
| Historia de Usuario | HU02 |
| Estado | Completada |
| Descripción | Seleccionar la plataforma donde vamos a desarrollar el Bot |

Tabla 3.12: Tarea 01.

| | |
|----------------------------|---|
| Tarea | T02 |
| Historia de Usuario | HU01 |
| Estado | Completada |
| Descripción | Crear la disposición y diseño de botones del Menú principal |

Tabla 3.13: Tarea 02.

| | |
|----------------------------|---|
| Tarea | T03 |
| Historia de Usuario | HU01 |
| Estado | Completada |
| Descripción | Crear la disposición y diseño de botones del Menú de selección de comunidades |

Tabla 3.14: Tarea 03.

| | |
|----------------------------|--|
| Tarea | T04 |
| Historia de Usuario | HU01 |
| Estado | Completada |
| Descripción | Crear la disposición y diseño de botones del Menú de información a consultar de cada comunidad en sus diferentes estados |

Tabla 3.15: Tarea 04.

| | |
|----------------------------|--|
| Tarea | T05 |
| Historia de Usuario | HU01 |
| Estado | Completada |
| Descripción | Crear la disposición y diseño de botones del Menú de información a consultar de España en sus diferentes estados |

Tabla 3.16: Tarea 05.

| | |
|----------------------------|---|
| Tarea | T06 |
| Historia de Usuario | HU03 |
| Estado | Completada |
| Descripción | Crear el Menú principal y configurarlo para que sea visible en todo momento salvo que el usuario decida ocultarlo |

Tabla 3.17: Tarea 06.

| | |
|----------------------------|--|
| Tarea | T07 |
| Historia de Usuario | HU03 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad del botón Menú para acceder al listado de comunidades |

Tabla 3.18: Tarea 07.

| | |
|----------------------------|--|
| Tarea | T08 |
| Historia de Usuario | HU03 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad del botón Ayuda |

Tabla 3.19: Tarea 08.

| | |
|----------------------------|--|
| Tarea | T09 |
| Historia de Usuario | HU03 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad del botón Información |

Tabla 3.20: Tarea 09.

| | |
|----------------------------|---------------------------------------|
| Tarea | T10 |
| Historia de Usuario | HU04 |
| Estado | Completada |
| Descripción | Crear el Menú de comunidades y España |

Tabla 3.21: Tarea 10.

| | |
|----------------------------|---|
| Tarea | T11 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Andalucía |

Tabla 3.22: Tarea 11.

| | |
|----------------------------|--|
| Tarea | T12 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Aragón |

Tabla 3.23: Tarea 12.

| | |
|----------------------------|--|
| Tarea | T13 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Asturias |

Tabla 3.24: Tarea 13.

| | |
|----------------------------|---|
| Tarea | T14 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para C. Valenciana |

Tabla 3.25: Tarea 14.

| | |
|----------------------------|--|
| Tarea | T15 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Canarias |

Tabla 3.26: Tarea 15.

| | |
|----------------------------|---|
| Tarea | T16 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Cantabria |

Tabla 3.27: Tarea 16.

| | |
|----------------------------|--|
| Tarea | T17 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Castilla La Mancha |

Tabla 3.28: Tarea 17.

| | |
|----------------------------|---|
| Tarea | T18 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Castilla y León |

Tabla 3.29: Tarea 18.

| | |
|----------------------------|--|
| Tarea | T19 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Cataluña |

Tabla 3.30: Tarea 19.

| | |
|----------------------------|---|
| Tarea | T20 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Ceuta |

Tabla 3.31: Tarea 20.

| | |
|----------------------------|---|
| Tarea | T21 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Extremadura |

Tabla 3.32: Tarea 21.

| | |
|----------------------------|---|
| Tarea | T22 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Galicia |

Tabla 3.33: Tarea 22.

| | |
|----------------------------|--|
| Tarea | T23 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Baleares |

Tabla 3.34: Tarea 23.

| | |
|----------------------------|--|
| Tarea | T24 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para La Rioja |

Tabla 3.35: Tarea 24.

| | |
|----------------------------|--|
| Tarea | T25 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Madrid |

Tabla 3.36: Tarea 25.

| | |
|----------------------------|---|
| Tarea | T26 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Melilla |

Tabla 3.37: Tarea 26.

| | |
|----------------------------|--|
| Tarea | T27 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Murcia |

Tabla 3.38: Tarea 27.

| | |
|----------------------------|---|
| Tarea | T28 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para Navarra |

Tabla 3.39: Tarea 28.

| | |
|----------------------------|--|
| Tarea | T29 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para País Vasco |

Tabla 3.40: Tarea 29.

| | |
|----------------------------|--|
| Tarea | T30 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar la funcionalidad de los botones de consulta de datos para España |

Tabla 3.41: Tarea 30.

| | |
|----------------------------|--|
| Tarea | T31 |
| Historia de Usuario | HU05 |
| Estado | Completada |
| Descripción | Implementar las diferentes gráficas para cada consulta disponible para mostrar los datos de manera gráfica |

Tabla 3.42: Tarea 31.

| | |
|----------------------------|---|
| Tarea | T32 |
| Historia de Usuario | HU06 |
| Estado | Completada |
| Descripción | Redactar la información que ha de aparecer al seleccionar la opción Ayuda |

Tabla 3.43: Tarea 32.

| | |
|----------------------------|---|
| Tarea | T33 |
| Historia de Usuario | HU07 |
| Estado | Completada |
| Descripción | Redactar la información que ha de aparecer al seleccionar la opción Información |

Tabla 3.44: Tarea 33.

| | |
|----------------------------|---|
| Tarea | T34 |
| Historia de Usuario | HU08 |
| Estado | Completada |
| Descripción | Seleccionar la plataforma y las configuración donde se va a Desplegar el proyecto |

Tabla 3.45: Tarea 34.

| | |
|----------------------------|--|
| Tarea | T35 |
| Historia de Usuario | HU08 |
| Estado | En proceso |
| Descripción | Implementar el despliegue de manera que se realice de forma automática |

Tabla 3.46: Tarea 35.

| | |
|----------------------------|--|
| Tarea | T36 |
| Historia de Usuario | HU09 |
| Estado | Completada |
| Descripción | Modularizar las funciones que crean las gráficas asociadas a los datos |

Tabla 3.47: Tarea 36.

| | |
|----------------------------|--|
| Tarea | T37 |
| Historia de Usuario | HU09 |
| Estado | Completada |
| Descripción | Modularizar las funciones que muestran los datos |

Tabla 3.48: Tarea 37.

Nota: Como se describió en el Capítulo 2, se ha utilizado **GitHub** para llevar el control del proyecto, por lo que el control de este apartado se ha llevado a cabo mediante el siguiente proyecto dentro del repositorio, donde se ha intentado representar las diversas Historias de Usuario y Tareas aquí mostradas. Podéis acceder a él desde aquí [\[4\]](#)

3.3. Temporización

Como hemos dicho antes, al aplicar la metodología **Scrum** el proyecto se dividirá en diferentes Sprints. Al final de cada uno de ellos se deberá tener un resultado completo y funcional o un incremento de las funcionalidades del producto que pueda ser entregable de manera que al ser solicitado por el **Product Owner** este pueda, por medio de un esfuerzo mínimo, ser utilizado.

Como se mencionó en la subsección 3.2, el uso de GitHub permitirá llevar un control de los diferentes Sprints del proyecto. En este punto procederemos a desglosar los diferentes Sprints en los que se ha dividido el proyecto.

3.3.1. Sprint 0

El **Sprint 0** se denomina así porque es el que representa a la fase inicial de todo proyecto. La duración de este Sprint suele ser de aproximadamente 1 semana. Este Sprint servirá para preparar y tomar las decisiones convenientes sobre el proyecto: el equipo, la tecnología que se va a usar, dejar clara la metodología que se va a aplicar (en este caso **Scrum**) y organizarse de la mejor manera posible para poder evitar fallos a futuro en el proyecto.

En este punto al estar desarrollando un proyecto de final de grado, no existen los diferentes roles que se aplican en la metodología **Scrum**, por ello, el alumno ha sido el encargado de desempeñar todos los papeles, imaginando un cliente ficticio y actuando como equipo y Scrum Master al mismo tiempo, haciendo así que se cumplan las pautas.

Como hemos dicho, el **Sprint 0** será la fase de análisis definida en la subsección 3.1.1.1, donde nos encargaremos de la toma de decisiones relativas al proyecto: como se va a desarrollar, en que lenguaje, en que plataforma, etc.

Para poder ver los datos del Sprint debemos definir la capacidad de trabajo por iteración. Por ello se define la siguiente tabla:

| | |
|----------------------------|------------------------------|
| Duración del Sprint | 1 semana (5 días laborables) |
| Trabajo diario | 4 horas |
| Horas del Sprint | 20 horas |

Tabla 3.49: Capacidad de trabajo Sprint 0.

En este caso las horas utilizadas son escasas, esto como hemos comentado antes se debe a la naturaleza del trabajo, ya que se trata de un proyecto realizado por una única persona. Debido a esto, la estimación del desarrollo será enfocado a dicha persona, aunque se simulará de forma sencilla los diferentes aspecto que puedan darle un aspecto mas real al proyecto.

3.3.2. Sprint 1

Lo primero que tendremos que hacer en este primer Sprint es definir el **Product Backlog** y usarlo para poder estimar el tiempo de desarrollo del proyecto en su conjunto. Para este caso representaremos a los 3 tipos de roles que existen dentro de Scrum los cuales representaremos como PO (Product Owner), SM (Scrum Master) y TD (Equipo de desarrollo). El **Product Backlog** puede verse en la siguiente tabla ??.

| Tareas | Historia de Usuario | PO | SM | TD | Tiempo estimado | Prioridad |
|--------|---------------------|-----|-----|-----|-----------------|-----------|
| T01 | HU02 | 2 | 2 | 2 | 2 | Alta |
| T02 | HU01 | 2 | 2.5 | 3 | 2.5 | Alta |
| T03 | HU01 | 2 | 2.5 | 3 | 2.5 | Alta |
| T04 | HU01 | 2 | 2.5 | 3 | 2.5 | Alta |
| T05 | HU01 | 2 | 2.5 | 3 | 2.5 | Alta |
| T06 | HU03 | 1 | 3 | 3.5 | 2.5 | Alta |
| T07 | HU03 | 1 | 1 | 1 | 1 | Alta |
| T08 | HU03 | 1 | 1 | 1 | 1 | Alta |
| T09 | HU03 | 1 | 1 | 1 | 1 | Alta |
| T10 | HU04 | 1 | 3.5 | 4.5 | 3 | Alta |
| T11 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T12 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T13 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T14 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T15 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T16 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T17 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T18 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T19 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T20 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T21 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T22 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T23 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T24 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T25 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T26 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T27 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T28 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T29 | HU05 | 1 | 2.5 | 3.5 | 2.33 | Alta |
| T30 | HU05 | 1 | 2.5 | 3.5 | 4 | Alta |
| T31 | HU05 | 10 | 20 | 36 | 22 | Alta |
| T32 | HU06 | 0.5 | 0.5 | 0.5 | 0.5 | Baja |
| T33 | HU07 | 0.5 | 0.5 | 0.5 | 0.5 | Baja |
| T34 | HU08 | 2 | 3 | 4 | 3 | Alta |
| T35 | HU08 | 1 | 3 | 5 | 3 | Alta |
| T36 | HU09 | 2 | 9 | 13 | 8 | Alta |
| T37 | HU09 | 2 | 9 | 13 | 8 | Alta |

Tabla 3.51: Product Backlog.

Como se ve en la tabla, se han recogido todas las tareas que forman el proyecto, estas después serán administradas en la manera que el equipo (en este caso el alumno) crea conveniente.

Para este Sprint se han decidido que se realizaran las tareas T01-T10, T32 y T33. Éstas son las principales que debemos desarrollar a la hora de iniciar el proyecto, las cuales consisten en el diseño de la interfaz y el menú inicial del Bot, aplicando las funcionalidades a los diferentes botones del Menú Principal del proyecto.

Para poder ver los datos del Sprint definiremos la capacidad de trabajo en la siguiente tabla:

| | |
|----------------------------|------------------------------|
| Duración del Sprint | 1 semana (5 días laborables) |
| Trabajo diario | 5 horas |
| Horas del Sprint | 25 horas |

Tabla 3.52: Capacidad de trabajo Sprint 1.

Una vez definida la capacidad de trabajo que debería tener este Sprint procederemos a mostrar una tabla con el resultado del trabajo comparando el tiempo estimado con el tiempo real que se ha utilizado para llevar a cabo este primer Sprint.

| Tareas | Tiempo estimado | Tiempo real |
|--------------|-----------------|-------------|
| T01 | 2 | 3 |
| T02 | 2.5 | 2 |
| T03 | 2.5 | 3 |
| T04 | 2.5 | 2 |
| T05 | 2.5 | 3 |
| T06 | 2.5 | 3 |
| T07 | 1 | 2 |
| T08 | 1 | 0.5 |
| T09 | 1 | 0.5 |
| T10 | 3 | 3 |
| T32 | 0.5 | 0.5 |
| T33 | 0.5 | 0.5 |
| Total | 21.5 | 23 |

Tabla 3.53: Tiempo invertido en el Sprint 1.

3.3.3. Sprint 2

Para este Sprint se han decidido que se realizaran las tareas T11-T30. Estas son las asociadas a la implementación de los diferentes menús de consulta para todas las Comunidades Autónomas del país y para el conjunto de España. Además de la implementación se han llevado a cabo las respectivas pruebas de funcionamiento para comprobar que todo el proyecto funciona.

Para poder ver los datos del Sprint definiremos la capacidad de trabajo en la siguiente tabla:

| | |
|----------------------------|------------------------------|
| Duración del Sprint | 2 semana (5 días laborables) |
| Trabajo diario | 5 horas |
| Horas del Sprint | 50 horas |

Tabla 3.54: Capacidad de trabajo Sprint 2.

Una vez definida la capacidad de trabajo que debería tener este Sprint procederemos a mostrar una tabla con el resultado del trabajo comparando el tiempo estimado con el tiempo real que se ha utilizado para llevar a cabo este segundo Sprint.

| Tareas | Tiempo estimado | Tiempo real |
|--------------|-----------------|-------------|
| T11 | 2.33 | 2.5 |
| T12 | 2.33 | 2.5 |
| T13 | 2.33 | 2.5 |
| T14 | 2.33 | 2.5 |
| T15 | 2.33 | 2.5 |
| T16 | 2.33 | 2.5 |
| T17 | 2.33 | 2.5 |
| T18 | 2.33 | 2.5 |
| T19 | 2.33 | 2.5 |
| T20 | 2.33 | 2.5 |
| T21 | 2.33 | 2.5 |
| T22 | 2.33 | 2.5 |
| T23 | 2.33 | 2.5 |
| T24 | 2.33 | 2.5 |
| T25 | 2.33 | 2.5 |
| T26 | 2.33 | 2.5 |
| T27 | 2.33 | 2.5 |
| T28 | 2.33 | 2.5 |
| T29 | 2.33 | 2.5 |
| T30 | 4 | 4.5 |
| Total | 48.27 | 52 |

Tabla 3.55: Tiempo invertido en el Sprint 2.

3.3.4. Sprint 3

Para este Sprint se han decidido que se realizará la tarea T31. Ésta es la asociada al diseño e implementación de la generación de las gráficas asociadas a las diferentes opciones de consulta que se definieron en el Sprint anterior. De esta forma no solo se mostrará los datos solicitados, si no que además se añadirá un imagen de manera que aporte una mayor información.

Para poder ver los datos del Sprint definiremos la capacidad de trabajo en la siguiente tabla:

| | |
|----------------------------|------------------------------|
| Duración del Sprint | 1 semana (5 días laborables) |
| Trabajo diario | 5 horas |
| Horas del Sprint | 25 horas |

Tabla 3.56: Capacidad de trabajo Sprint 3.

Una vez definida la capacidad de trabajo que debería tener este Sprint procederemos a mostrar una tabla con el resultado del trabajo comparando el tiempo estimado con el tiempo real que se ha utilizado para llevar a cabo este tercer Sprint.

| Tareas | Tiempo estimado | Tiempo real |
|--------------|-----------------|-------------|
| T33 | 22 | 25.5 |
| Total | 22 | 25.5 |

Tabla 3.57: Tiempo invertido en el Sprint 3.

3.3.5. Sprint 4

Para este Sprint se han decidido que se realizarán las tareas T34-35. Éstas lo que pretenden es mejorar el código mediante la modularización de diferentes funcionalidades, como podrían ser las funciones que se encargan de la generación de las gráficas o las que se encargan del calculo de los datos, así como de llevar a cabo el despliegue automático del Bot en una plataforma.

Para poder ver los datos del Sprint definiremos la capacidad de trabajo en la siguiente tabla:

| | |
|----------------------------|------------------------------|
| Duración del Sprint | 1 semana (5 días laborables) |
| Trabajo diario | 5 horas |
| Horas del Sprint | 25 horas |

Tabla 3.58: Capacidad de trabajo Sprint 4.

Una vez definida la capacidad de trabajo que debería tener este Sprint procederemos a mostrar una tabla con el resultado del trabajo comparando el tiempo estimado con el tiempo real que se ha utilizado para llevar a cabo este tercer Sprint.

| Tareas | Tiempo estimado | Tiempo real |
|--------------|-----------------|-------------|
| T34 | 3 | 3 |
| T35 | 3 | 4 |
| T36 | 8 | 8.5 |
| T37 | 8 | 8.5 |
| Total | 22 | 24 |

Tabla 3.59: Tiempo invertido en el Sprint 4.

3.4. Costes

En este apartado, lo que se pretende es hacer una estimación de los costes que se necesitan para llevar a cabo el proyecto. Dentro de estos podremos encontrarnos dos apartados diferenciados: los costes de desarrollo y los costes de producción.

3.4.1. Costes de desarrollo

En este apartado se calcularán los costes derivados del desarrollo, estos son: el tiempo empleado en el desarrollo, el material utilizado, en este caso el equipo informático asociado, así como una estimación de los costes de amortización del mismo.

3.4.1.1. Costes del personal

Para este apartado se ha hecho una estimación teniendo en cuenta la existencia de los tres roles diferenciales de Scrum: Product Owner, Scrum Master y Equipo de desarrollo.

En primer lugar lo que tendremos que calcular son las horas totales que se han implementado. En este caso el total de horas empleadas para este proyecto se define en la siguiente tabla:

| Sprint | Tiempo invertido |
|---------------|-------------------------|
| Sprint 0 | 20 |
| Sprint 1 | 23 |
| Sprint 2 | 52 |
| Sprint 3 | 25.5 |
| Sprint 4 | 24 |
| Documentación | 50 |
| Total | 194.5 |

Tabla 3.60: Resumen tiempo invertido

Una vez obtenido el total de horas utilizadas para este proyecto las dividiremos según hayan sido empleadas por los diferentes roles.

| Rol | Tiempo dedicado | Coste (empleado/hora) | Coste |
|---------------|-----------------|-----------------------|--------|
| Product Owner | 28 | 15€ | 420€ |
| Scrum Master | 56 | 10€ | 560€ |
| Equipo | 111 | 10€ | 1.110€ |
| | | Total | 2.090€ |

Tabla 3.61: Estimación coste de los empleados.

3.4.1.2. Costes de Hardware

Para el desarrollo del proyecto se ha necesitado el uso de un equipo informático, por lo que su coste y sus costes de amortización son los siguientes.

| Equipo | Coste | % Uso dedicado | Dedicación (meses) | Periodo de depreciación (meses) | Coste imputable |
|---|-----------|----------------|--------------------|---------------------------------|-----------------|
| MSI GL63 8RD-642XES Intel Core i7-8750H/16GB/1TB+512GB SSD/GTX 1050Ti/15.6" | 1.154,95€ | 100 % | 2 | 36 | 64.17€ |

Tabla 3.62: Estimación costes del hardware.

El coste imputable se ha calculado de la siguiente manera:

$$\frac{\text{Coste} * \text{Dedicación}}{\text{Periodo de depreciación} * \text{Usodedicado}}$$

3.4.2. Costes de producción

En este apartado se calcularán los costes derivados de la producción, estos son: el software empleado para la producción del proyecto, así como los costes de los recursos cloud utilizados.

| Producto | Coste |
|-----------------------------|-------------|
| Sistema Operativo | 0€ |
| Pycharm Professional 2020.2 | 199€ |
| Heroku | 0€ |
| Total | 199€ |

Tabla 3.63: Resumen costes producción.

3.4.3. Costes de operaciones

Los costes de operaciones son aquellos gastos que están relacionados con las operaciones de un negocio, o para el funcionamiento del mismo. En resumen, son el coste de los recursos utilizados por una organización.

En este casos los costes de operaciones de nuestro proyecto serian los siguientes:

| Recurso | Coste | Tiempo (meses) | Coste Total |
|----------|--------|----------------|---------------|
| Luz | 80€ | 2 meses | 160€ |
| Internet | 35,95€ | 2 | 71,9€ |
| | | Total | 231.9€ |

Tabla 3.64: Resumen costes producción.

3.4.4. Resumen de costes

Para terminar, tras haber estimado los costes de las diferentes partes del proyecto, procederemos ha calcular el coste total del mismo. Añadiremos, además de los calculados, un apartado de costes indirectos para cubrir los posibles riesgo o gastos imprevistos que puedan surgir a lo largo del desarrollo, los cuales serán un 10 % del total calculado.

| Tipo Coste | Coste |
|------------------|------------------|
| Personal | 2.090€ |
| Equipo | 64,17€ |
| Producción | 199€ |
| Operaciones | 231,9€ |
| Indirectos(10 %) | 235,32€ |
| Total | 2.843.58€ |

Tabla 3.65: Resumen costes del proyecto.

A lo largo de este capítulo se ha podido ver las decisiones que se han tomado a la hora de llevar a cabo el proyecto. Esta decisiones pueden marcar la diferencia entre un buen o mal proyecto. Saber definirlo desde el primer momento y aplicar una metodología de trabajo óptima puede facilitar mucho la tarea, así como mejorar el rendimiento a la hora de llevar a cabo el desarrollo. En cuanto a la estimación de los costes del proyecto es igual de importante que las decisiones previas, un proyecto que no puede asumir los costes para su producción y desarrollo no puede seguir adelante.

Capítulo 4

Análisis

A lo largo de este apartado se presentará todo el análisis previo realizado a la hora de llevar a cabo la implementación del Bot. Entre los diferentes puntos a tratar analizaremos los requisitos que ha de cumplir nuestro Bot, así como el proceso de diseño de la interfaz del mismo

4.1. Requisitos del sistema

En este apartado se definen los requisitos funcionales y no funcionales con los que cuenta el sistema. Antes de proceder a exponer los diferentes requisitos debemos dejar claro que son cada uno de ellos:

- **Requisitos funcionales:** Definen las funcionalidades del sistema, es decir, el comportamiento del mismo. Normalmente estos requisitos están vinculados con las entradas, las salidas de los procesos y los datos almacenados en el sistema.
- **Requisitos no funcionales:** Describen prestaciones, características y limitaciones. Entre estos podemos encontrar propiedades como el rendimiento, la facilidad de uso, disponibilidad, seguridad.

4.1.1. Requisitos funcionales

- El sistema permitirá el acceso de usuarios sin registro previo.
- El sistema permitirá consultar las diferentes Comunidades Autónomas.
- El sistema permitirá consultar los datos de España.

- El sistema permitirá consultar el incremento de casos de Covid.
- El sistema permitirá consultar los casos acumulados.
- El sistema permitirá consultar los datos de fallecidos.
- El sistema permitirá consultar los datos hospitalarios de las Comunidades Autónomas.
- El sistema permitirá consultar los datos de los casos y fallecimientos por edades en España.
- El sistema permitirá consultar el nº de casos por cada 100mil habitantes.
- El sistema permitirá consultar el nº de casos por cada 100mil habitantes en la última semana.
- El sistema permitirá consultar los fallecimientos por cada 100mil habitantes.
- El sistema permitirá consultar los fallecimientos por cada 100mil habitantes en la última semana.
- El sistema permitirá consultar la ayuda del sistema.
- El sistema deberá ofrecer gráficas para apoyar los datos mostrados.
- El sistema deberá ofrecer la fecha en la que se han actualizado los datos por última vez.
- El sistema deberá ofrecer información de ayuda para el usuario.
- El sistema deberá ofrecer información sobre donde se puede consultar el código y quien es su desarrollador.

4.1.2. Requisitos no funcionales

- El sistema estará disponible para cualquier plataforma que soporte **Telegram**
- El sistema ha de poder utilizarse en todo momento.
- El sistema no debe tener largos tiempos de espera.
- El sistema ofrecerá un manual de ayuda de los diferentes opciones de consulta.
- El sistema ofrecerá un listado con información del sistema.
- El sistema ha de cumplir con la política de privacidad de **Telegram**

4.2. Diseño de la Interfaz

En este apartado se mostrará como se ha llevado a cabo el diseño del sistema, tanto el diseño del **sitemap** como del **wireframe**.

4.2.1. Diseño del sitemap

Se conoce como el **sitemap**, en el ámbito de las páginas web, al listado de las diferentes páginas que componen la web [14]. En este caso, al aplicar el **sitemap** al Bot, este estará compuesto por los botones de los que constará nuestro Bot y permiten al usuario a navegar por él. La Figura 4.1 representa la estructura de **Covid-19 Reports**

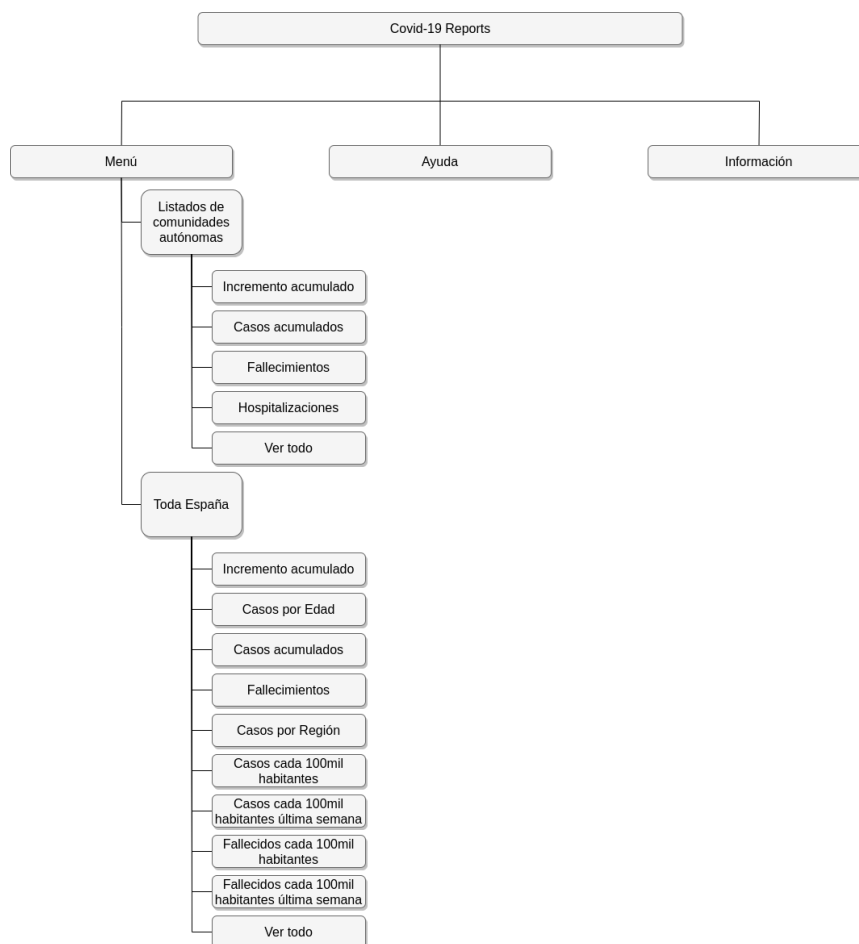


Figura 4.1: Sitemap del Bot.

4.2.2. Diseño del wireframe

El **wireframe** es un boceto donde se representan visualmente de manera sencilla y estemática la estructura de una web. En este caso . Las Figuras 4.2 a 4.13 representan la estructura de **Covid-19 Reports**.

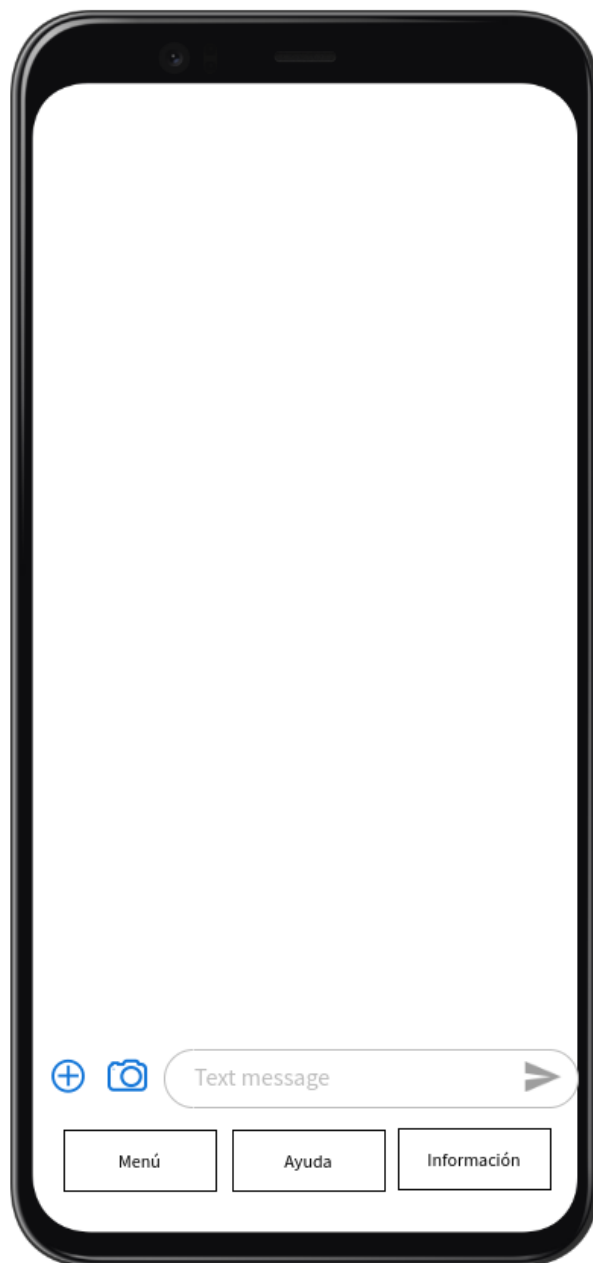


Figura 4.2: Boceto de la pantalla principal.



Figura 4.3: Boceto Menú principal de selección.



Figura 4.4: Boceto Menú Comunidad Autónoma.

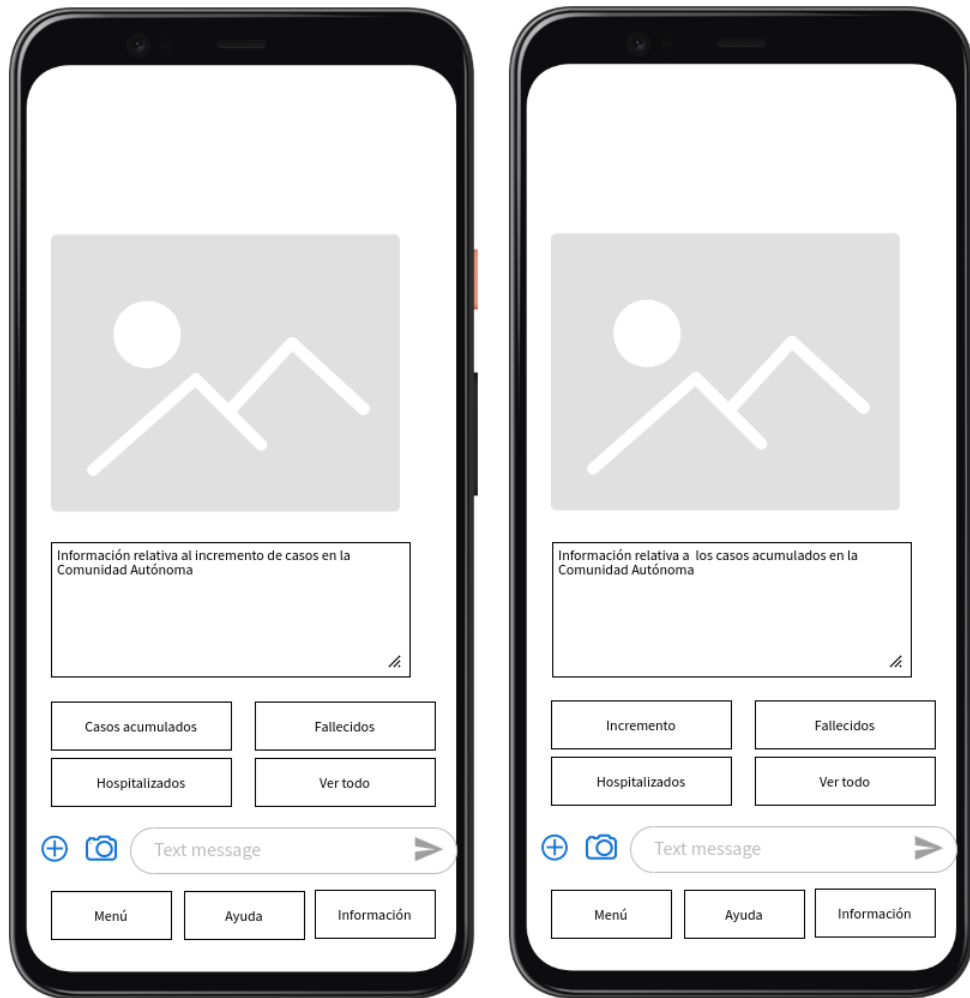


Figura 4.5: Boceto Menús Incremento y Casos acumulados.

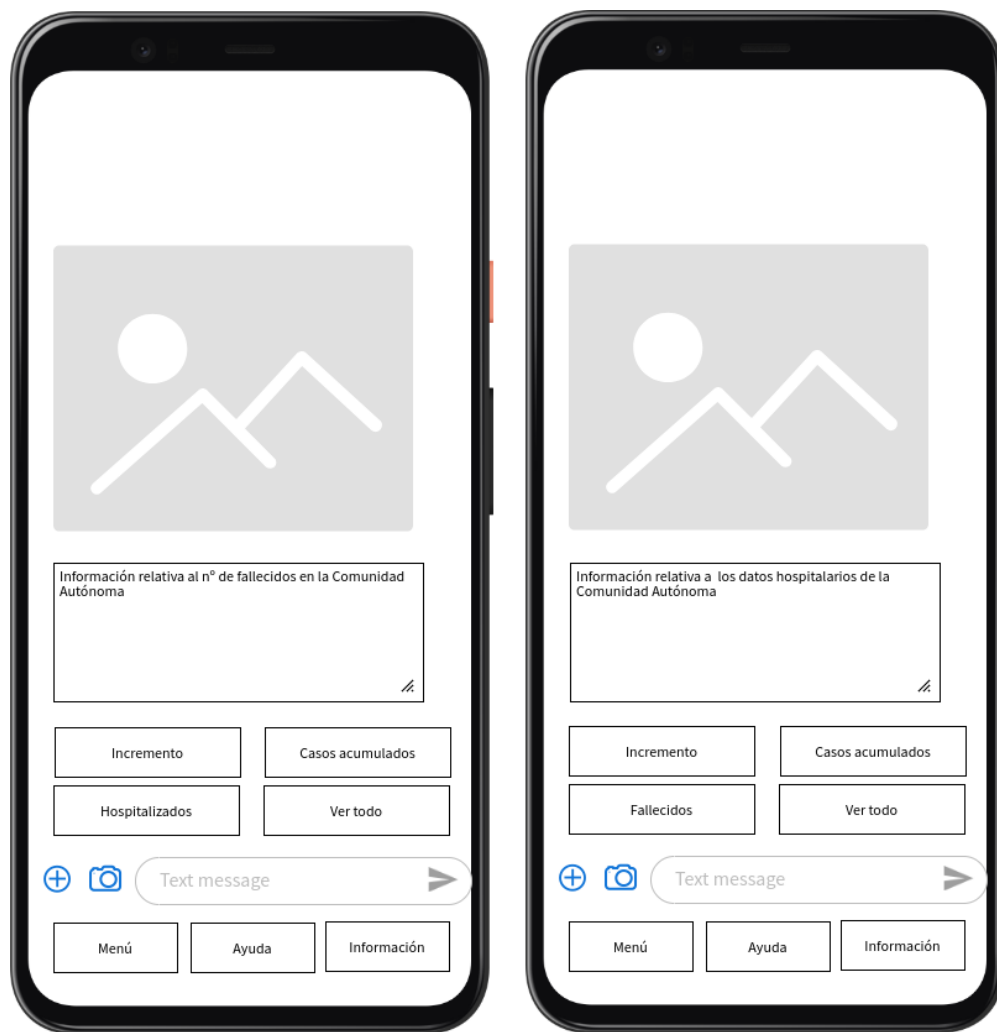


Figura 4.6: Boceto Menús Fallecidos y Hospitalizaciones.



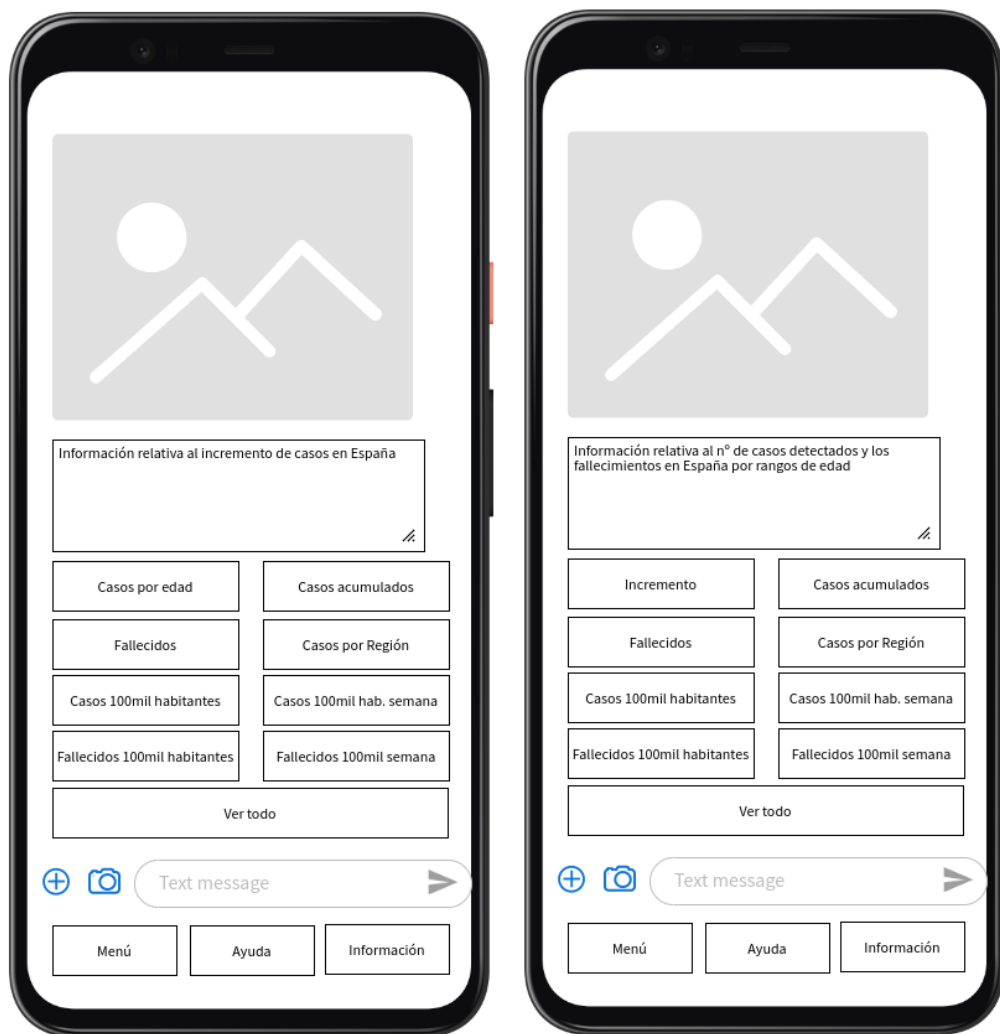


Figura 4.8: Boceto Menús Incremento y Casos por edad en España.

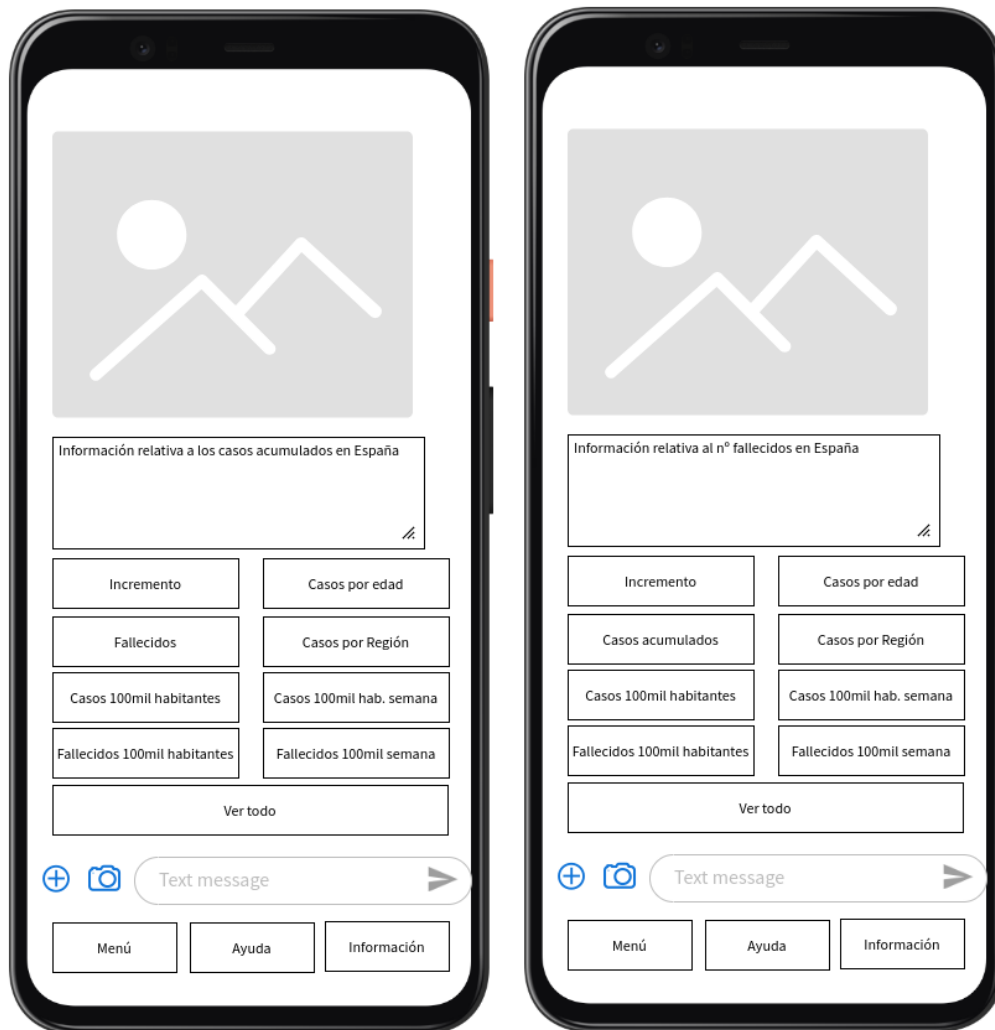


Figura 4.9: Boceto Menús Casos acumulados y Fallecidos en España.

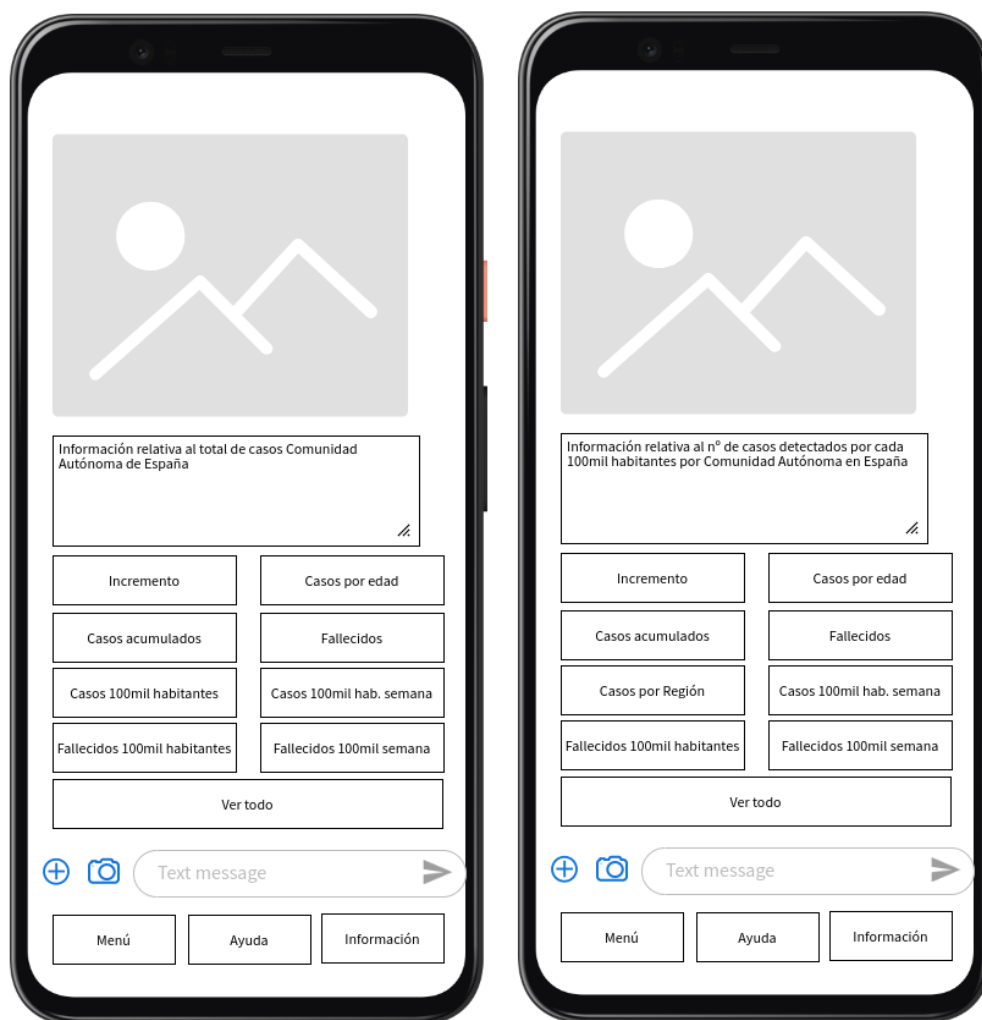


Figura 4.10: Boceto Menús Casos por Región y Casos cada 100mil habitantes en España.

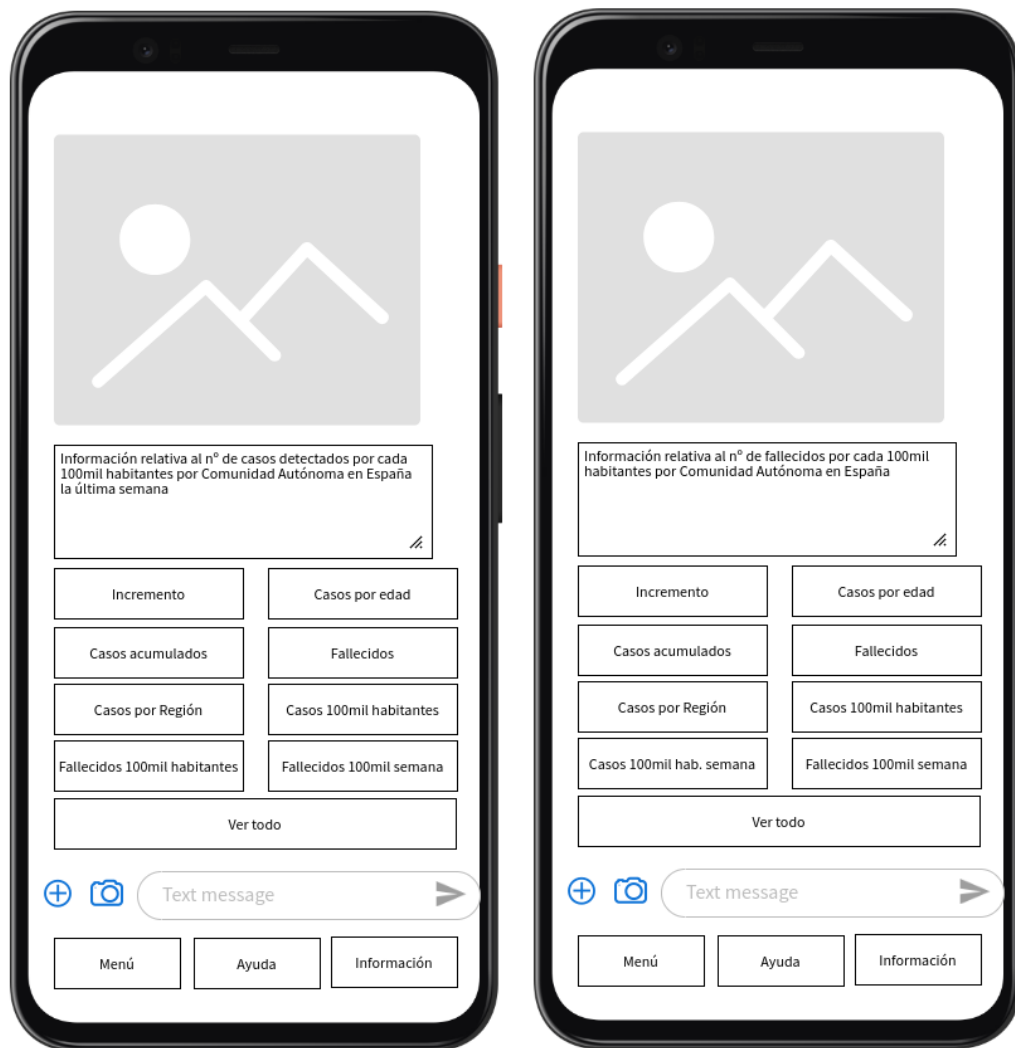


Figura 4.11: Boceto Menús Casos cada 100mil hab. semana y Fallecidos cada 100mil habitantes en España.



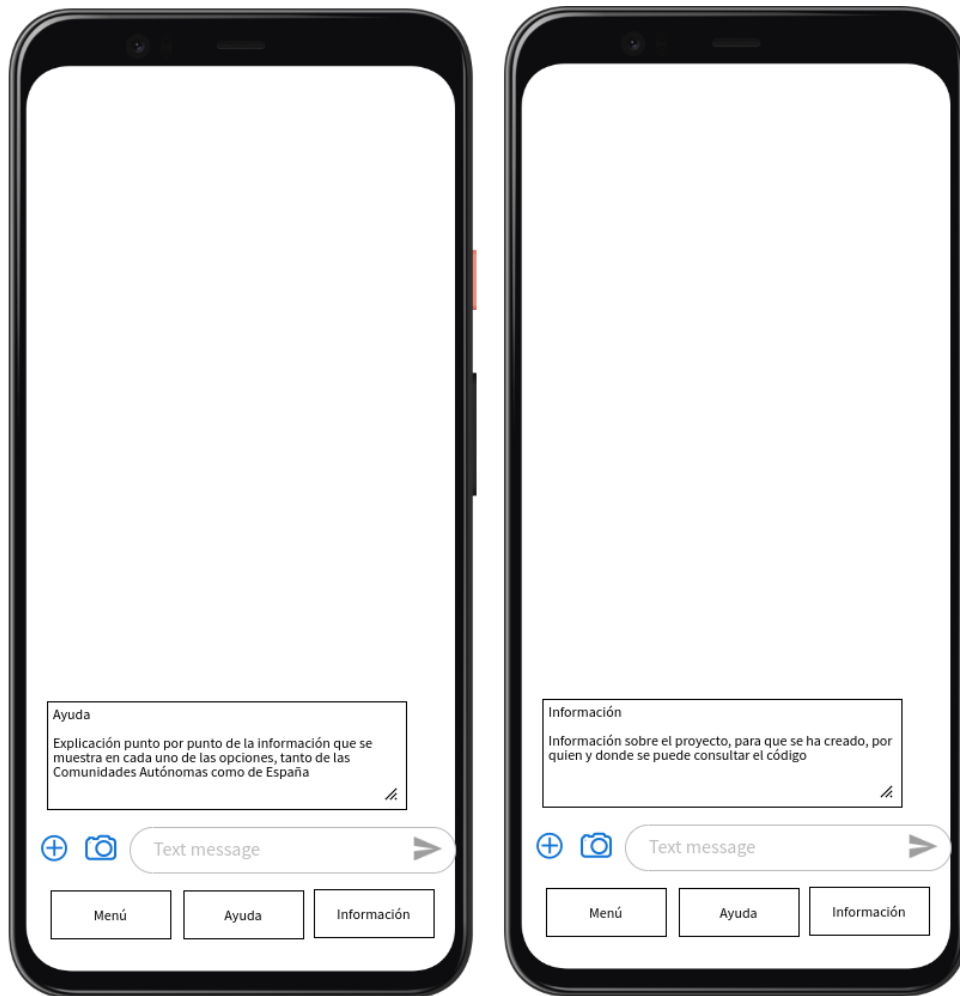


Figura 4.13: Boceto Menús Ayuda e Información.

Como hemos visto, a lo largo de este capítulo se han mostrados los diferentes puntos del análisis del proyecto, los cuales nos serán útiles a la hora de llevar a cabo del mismo. En el próximo capítulo se muestra como se ha llevado a cabo la implementación del proyecto siguiendo las bases que se han desarrollado.

Capítulo 5

Implementación

Como ya se explicó en el Capítulo 3 la implementación del proyecto se ha llevado a cabo en una serie de **Sprints**. Estos se han definido en **GitHub** a modo de *hitos* y cada uno de ellos contendrá un grupo de *issues* que corresponden a las diferentes **Historias de Usuario** y **Tareas** que se han ido incorporando al proyecto a lo largo desarrollo del mismo.

Capítulo 6

Pruebas

Capítulo 7

Conclusiones

Capítulo 8

Trabajos futuros

Bibliografía

- [1] Actualización nº 246. enfermedad por el coronavirus (covid-19). 09.11.2020. https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov/documentos/Actualizacion_246_COVID-19.pdf.
- [2] Historia de usuario. https://www.scrummanager.net/bok/index.php/Historia_de_usuario.
- [3] La moncloa. <https://www.lamoncloa.gob.es/Paginas/index.aspx>.
- [4] Proyecto github. https://github.com/JmZero/TFG_Covid-19_reports/projects/1.
- [5] Situación actual del covid-19 en España. <https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov/situacionActual.htm>.
- [6] Whatsapp. <https://www.whatsapp.com/?lang=es>.
- [7] Whatsapp. <https://telegram.org/>.
- [8] Kent Beck. Manifiesto por el desarrollo Ágil de software. <https://agilemanifesto.org/iso/es/manifesto.html>.
- [9] La Guía de Scrum™. Ken schwaber y jeff sutherland. <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Spanish-European.pdf>.
- [10] Statista Research Department. Usuarios de teléfonos móviles y smartphones a nivel mundial de 2014 a 2019. <https://es.statista.com/estadisticas/723622/usuarios-de-telefonos-moviles-y-smartphones-en-el-mundo/>.

- [11] elPeriódico. Enganchados al móvil: ¿somos una sociedad 'empantallada'? <https://byzness.elperiodico.com/es/innovadores/20200102/empantallados-somos-adictos-nuestro-movil-7790578>.
- [12] Free Software Foundation. GNU General Public License. <https://www.gnu.org/licenses/agpl-3.0.en.html>.
- [13] Heroku. The heroku platform. <https://www.heroku.com/platform>.
- [14] Anastasia Kurmakaeva. ¿qué es un sitemap y cómo crearlo? <https://www.humanlevel.com/diccionario-marketing-online/sitemap>.
- [15] Matplotlib. Matplotlib: Visualization with python. <https://matplotlib.org/>.
- [16] Gabriela Nova. Los beneficios de estar informado. *Medium*, Dec 2016.
- [17] OMS. Preguntas y respuestas sobre la enfermedad por coronavirus (covid-19). <https://www.who.int/es/emergencies/diseases/novel-coronavirus-2019/advice-for-public/q-a-coronaviruses>.
- [18] OMS. Reglamento sanitario internacional (2005). <https://www.who.int/ihr/publications/9789241580496/es/>.
- [19] pandas. About pandas. <https://pandas.pydata.org/about/>.
- [20] python-telegram bot. Repositorio de python-telegram-bot. <https://github.com/python-telegram-bot/python-telegram-bot>.
- [21] Iván Ramírez. Whatsapp vs telegram: ¿cuál es la mejor aplicación de mensajería? <https://www.xatakandroid.com/aplicaciones-android/whatsapp-vs-telegram-cual-mejor-aplicacion-mensajeria-1>.
- [22] Paula Rochina. Python vs r para el análisis de datos. <https://revistadigital.inesem.es/informatica-y-tics/python-r-analisis-datos/>.
- [23] RTVE. La crisis económica y el coronavirus preocupan ahora más a los españoles que el paro. <https://www.rtve.es/noticias/20201015/crisis-economica-coronavirus-preocupan-ahora-mas-espanoles-paro/2045610.shtml>.
- [24] Marvin G. Soto. ¿por qué usar python para programar chatbots? <https://planetachatbot.com/por-qu%C3%A9-usar-python-para-programar-chatbots-3fce4b44df08>.
- [25] Stackscale. Top 10 de lenguajes de programación 2020. <https://www.stackscale.com/es/blog/top-10-lenguajes-programacion-2020/>.