



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

PROGRAMA DE ESTUDIOS CONJUNTO EN GRADO EN INGENIERÍA INFORMÁTICA Y
GRADO EN ESTADÍSTICA

**Aplicación multiplataforma de visualización de datos
electorales con agregación de encuestas**

Alumno: D. Javier Martín de Benito

Tutora: D.^a Margarita Gonzalo Tasis

Agradecimientos

En primer lugar, agradecer a mi tutora Margarita, que desde el primer momento confió en mí para sacar adelante las ideas y el proyecto en general.

En segundo lugar, a mi familia, que creyó siempre en mí durante estos 5 años de carrera.

Por último, pero no por ello menos importante, a mis amigos, de antes de la carrera o hechos durante estos años, que han sido mi máximo apoyo durante estos 5 años, y sobre todo durante estos últimos meses soportando mis cambios de humor y mi nerviosismo interminable.

A todos ellos, gracias. Este trabajo que tanto me ha costado va por vosotros.

Resumen

En este trabajo se presenta el desarrollo de una aplicación multiplataforma cuya finalidad es el acceso, de manera centralizada y cómoda, a resultados electorales, tanto a nivel autonómico como nacional, así como el análisis de diferentes encuestas electorales que se publiquen.

La motivación principal que lleva al desarrollo del presente Trabajo de Fin de Grado es permitir el acceso a la gran cantidad de información electoral que se está creando en los últimos años.

En el Trabajo de Fin de Grado se presenta la planificación del mismo, la cual está basada en el método UP, el Proceso Unificado, que está formado por diferentes fases: Inicio, Elaboración, Construcción y Transición, las cuales pueden contar con varias iteraciones, según la necesidad de los trabajadores.

En la fase de inicio se ha realizado el análisis del proyecto desde una perspectiva de su gestión: planificación, costes y la gestión de los riesgos.

La posterior fase de elaboración abarca el análisis y el diseño de la aplicación, como la determinación de los casos de uso, el desarrollo de la arquitectura de la aplicación, su ordenamiento, patrones que seguirá y la elaboración de los pertinentes diagramas para su correcto entendimiento.

La fase de construcción conlleva la implementación de la aplicación a base de su código y los frameworks necesarios, como en este caso serán Ionic o Angular. La fase de transición abarca la creación del manual de usuario y de instalación.

Posteriormente, se presenta el proceso seguido para desarrollar una aplicación con ayuda del framework Ionic, que permite la creación de una aplicación web gracias a frameworks tan conocidos como Angular, Vue o React y permite su conversión en una aplicación Android o iOS.

Abstract

This paper presents the development of a multiplatform application whose purpose is the access, in a centralized and comfortable way, of electoral results, in autonomic and national level, as well as the analysis of different published electoral polls.

The main motivation that leads to the development of this Final Degree Project is to allow access to the large amount of electoral information that is being created in recent years.

In this Final Degree Project, its planning is presented, which is based on the UP method, the Unified Process, which is made up of different phases: Start, Elaboration, Construction and Transition, which can have several iterations, according to the needs of the workers.

In the initial phase, the analysis of the project has been carried out from a management perspective: planning, costs and risk management.

The subsequent development phase covers the analysis and design of the application, such as the determination of use cases, the development of the application architecture, its ordering, patterns that it will follow and the development of the relevant diagrams for its implementation for its correct understanding.

The construction phase entails the implementation of the application based on its code and the necessary frameworks, as in this case they will be Ionic or Angular. The transition phase covers the creation of the user and installation manual.

Subsequently, the process followed in which an application has been developed with the help of the Ionic framework is presented, which allows the creation of a web application thanks to well-known frameworks such as Angular, Vue or React and allows its conversion into an Android or iOS application.

Índice general

Agradecimientos	III
Resumen	V
Abstract	VII
Lista de figuras	XIII
Lista de tablas	XVII
1. Introducción y análisis de otras aplicaciones	1
1.1. Introducción	1
1.1.1. Contexto	1
1.1.2. Motivación	1
1.1.3. Objetivos	1
1.2. Análisis de otras aplicaciones	2
1.2.1. Europe Elects	2
1.2.2. Electomanía	3
1.2.3. Electocracia	3
1.2.4. Electograph	4
1.3. Estructura de la memoria	5
2. Planificación del proyecto	7
2.1. Introducción	7
2.2. Glosario	7
2.3. Artefactos del proyecto	8
2.4. Restricciones del proyecto	8
2.5. Planificación del proyecto	8
2.5.1. Fase de inicio	8
2.5.2. Fase de elaboración	8
2.5.3. Fase de construcción	8
2.5.4. Fase de transición	8
2.6. Calendario del proyecto	9
2.6.1. Fase de inicio	9

2.6.2.	Fase de elaboración	10
2.6.3.	Fase de construcción	11
2.6.4.	Fase de transición	12
2.6.5.	Entregables del proyecto	12
2.7.	Análisis de riesgos	13
2.8.	Estimación de costes	16
2.8.1.	Costes hardware y software	16
2.8.2.	Conexión a internet y tarifa móvil	16
2.9.	Seguimiento del proyecto	16
3.	Análisis	19
3.1.	Análisis de requisitos	19
3.1.1.	Requisitos funcionales	19
3.1.2.	Requisitos de información	20
3.1.3.	Requisitos no funcionales	20
3.1.4.	Reglas de negocio	21
3.2.	Casos de uso	21
3.2.1.	Actores principales	21
3.2.2.	Diagrama de casos de uso	21
3.2.3.	Descripción de los casos de uso	22
3.3.	Modelo de dominio	33
3.3.1.	Región	33
3.3.2.	Parlamento	33
3.3.3.	Encuesta	34
3.3.4.	Partido	34
3.3.5.	PartidoRegion	34
3.3.6.	Pacto	34
3.3.7.	Elección	34
3.3.8.	Calendario	35
3.4.	Diagramas de secuencia	35
4.	Diseño	41
4.1.	Decisiones de diseño	41
4.2.	Arquitecturas y patrones	41
4.2.1.	Arquitectura del servidor	42
4.2.2.	Diseño de la API REST	43
4.2.3.	Arquitectura del cliente	44
4.3.	Persistencia de los datos	49
4.3.1.	Servidor	49
4.4.	Prototipo de la interfaz	50
4.5.	Tratamiento de los datos y GDPR	51

5. Implementación	55
5.1. Herramientas de desarrollo	55
5.2. Implementación de la persistencia de datos	55
5.3. Herramientas utilizadas	55
5.3.1. Django y Django REST framework	55
5.3.2. BeautifulSoup	55
5.3.3. Pandas	56
5.3.4. Angular e Ionic	56
5.3.5. D3	56
5.3.6. ngx-datatable	56
5.3.7. Control de versiones	56
5.4. Implementación del servidor	56
5.4.1. Web scraping	57
5.5. Implementación del cliente	58
5.5.1. Diseño adaptativo	58
6. Pruebas	61
6.1. Pruebas sobre el servidor	61
6.2. Pruebas sobre el cliente	62
7. Conclusiones y trabajo futuro	65
7.1. Conclusiones	65
7.2. Trabajo futuro	65
Bibliografía	67
A. Manual de usuario	69
A.1. Inicio	69
A.1.1. Región	69
A.1.2. Listado de encuestas	70
A.2. Calendario	70
A.3. Encuestas favoritas	71
A.4. Encuesta	73
A.4.1. Comparación de encuestas	73
A.5. Partido	74
B. Manual de Instalación	75
B.1. Contenido del repositorio	75
B.2. Instalación	75
B.2.1. Requisitos y ejecución del servidor	75
B.2.2. Requisitos y ejecución del cliente	76

Índice de figuras

1.1. Tweet de Europe Elects con los resultados	3
1.2. Proyección de Europe Elects.	4
1.3. Gráfico de barras del ElectoPanel	5
1.4. Estimación de escaños del ElectoPanel	5
1.5. Web de Electocracia	6
1.6. Web de Electograph	6
2.1. Fases de la metodología RUP	9
2.2. Fase de inicio	10
2.3. Fase de elaboración	10
2.4. Fase de construcción, iteración 1	11
2.5. Fase de construcción, iteración 2	11
2.6. Fase de transición	12
3.1. Diagrama de Casos de Uso	22
3.2. Caso de Uso 1- Acceder a región	23
3.3. Caso de Uso 2- Acceder a una encuesta	24
3.4. Caso de Uso 3- Acceder a partido	25
3.5. Caso de Uso 4- Acceder a calendario de elecciones	26
3.6. Caso de Uso 5- Añadir encuesta favorita	27
3.7. Caso de Uso 6- Acceder a encuesta favorita	28
3.8. Caso de Uso 7- Conformar pacto	29
3.9. Caso de Uso 8- Guardar pacto	30
3.10. Caso de Uso 9- Acceder a pactos guardados	31
3.11. Caso de Uso 10- Comparar encuestas	32
3.12. Modelo de dominio	33
3.13. Secuencia de caso de Uso 1- Acceder a región	35
3.14. Secuencia de caso de Uso 2- Acceder a una encuesta	36
3.15. Secuencia de caso de Uso 3- Acceder a partido	36
3.16. Secuencia de caso de Uso 4- Acceder a calendario de elecciones	36
3.17. Secuencia de caso de Uso 5- Añadir encuesta favorita	37
3.18. Secuencia de caso de Uso 6- Acceder a encuesta favorita	37

3.19. Secuencia de caso de Uso 7- Conformar pacto	37
3.20. Secuencia de caso de Uso 8- Guardar pacto	38
3.21. Secuencia de caso de Uso 9- Acceder a pactos guardados	38
3.22. Secuencia de caso de Uso 10- Comparar encuestas	39
4.1. Lógica del cliente-servidor	42
4.2. Arquitectura del servidor	42
4.3. Respuesta de la API	43
4.4. Arquitectura del cliente	44
4.5. Paquetes de la aplicación	45
4.6. Paquete inicio	45
4.7. Paquete guardados	45
4.8. Paquete calendario	46
4.9. Paquete servicios	46
4.10. Paquete partido	46
4.11. Paquete región	47
4.12. Paquete encuestas	47
4.13. Paquete encuesta	47
4.14. Paquete comparar	48
4.15. Paquete comparación	48
4.16. Modelo relacional de la base de datos en el servidor	49
4.17. Pantalla de inicio de la app	50
4.18. Calendario electoral	50
4.19. Calendario electoral	51
4.20. Región 1, la seleccionada	52
4.21. Encuestas electorales de la región seleccionada	52
4.22. Datos de la encuesta seleccionada	53
4.23. Guardar encuesta actual	53
4.24. Guardar pacto conformado	54
4.25. Datos del partido seleccionado	54
5.1. Envío de respuesta de encuestas	57
5.2. Envío de respuesta de partidos	58
5.3. Diseño en web	59
5.4. Diseño en un Samsung galaxy S8+	59
A.1. Pantalla de inicio	69
A.2. Pantalla de una región	70
A.3. Listado de encuestas electorales	71
A.4. Próximas elecciones	72
A.5. Encuestas favoritas	72

A.6. Ficha de una encuesta	73
A.7. Comparación de 2 encuestas	74
A.8. Partido político	74

Índice de tablas

2.1. Roles en el proyecto	7
2.2. Glosario	7
2.3. Resumen de las fases de la planificación	9
2.4. Fase de inicio	10
2.5. Fase de elaboración	11
2.6. Fase de construcción, iteración 1	11
2.7. Fase de construcción, iteración 2	11
2.8. Fase de transición	12
2.9. Riesgo 0- Falta de experiencia	13
2.10. Riesgo 1- falta de experiencia en la organización	13
2.11. Riesgo 2- baja calidad del código	13
2.12. Riesgo 3- Falta de disponibilidad	14
2.13. Riesgo 4- Pérdida de entregables	14
2.14. Riesgo 5- Falta de medios	14
2.15. Riesgo 6- Cambios en las fuentes de datos	14
2.16. Riesgo 7- Diseño incorrecto	15
2.17. Riesgo 8- Cambio de plazos	15
2.18. Riesgo 9- Nuevas especificaciones del proyecto	15
2.19. Riesgo 10- Falta de experiencia en planificación de riesgos	15
2.20. Riesgo 11- No se han determinado todos los riesgos	16
2.21. Costes hardware	16
2.22. Costes software	16
2.23. Desvío frente a la planificación inicial	17
3.1. Requisitos funcionales	20
3.2. Requisitos de información	20
3.3. Requisitos no funcionales	21
3.4. Reglas de negocio	21

Capítulo 1

Introducción y análisis de otras aplicaciones

1.1. Introducción

1.1.1. Contexto

La realidad política española ha cambiado de manera radical en los últimos 8 años: derrumbe del bipartidismo, nuevos ciclos electorales autonómicos, repeticiones electorales... Nunca se sabe qué podrá cambiar al día siguiente, pues cualquier decisión tomada puede acarrear cambios que no se hayan podido prever anteriormente. Actualmente, la información electoral obtenida es creada a diario, y uno de los resultados más importantes son las encuestas electorales. Hay tantas que es difícil seguir a diario el resultado de todas ellas, tanto a nivel autonómico como nacional, así como sus cambios. De hecho, no hay que llevarse al engaño, llega a ser abrumador en algunos casos en los que ni tan siquiera esos datos están ordenados. Además, debido al auge de nuevos partidos que han provocado la adhesión (por ideología) de ciudadanos al ámbito político, cada vez hay más gente interesada en el tema y, también, en las encuestas electorales.

Este proyecto se realiza para la asignatura Trabajo de Fin de Grado de Ingeniería Informática, mención en computación. Es un proyecto más, y como tal, ha de seguir una metodología existente y que ya haya sido probada. En este documento se recogen todas las fases realizadas, así como los procedimientos y herramientas utilizadas.

1.1.2. Motivación

El motivo de este proyecto, y la aplicación que resulta del mismo, es el de permitir a los usuarios de la misma el poder acceder a los datos de las encuestas públicas presentadas de una manera cómoda así como de una manera comprensible y sencilla.

1.1.3. Objetivos

El objetivo de este proyecto es el de la realización de una aplicación que obtenga los datos de las encuestas en tiempo real, en la medida de lo posible, y los muestre de una manera amigable y descriptiva. Una vez indicados, los objetivos serían:

- Desarrollar una aplicación multiplataforma utilizando Ionic y Angular que sea compatible con los principales navegadores, primordialmente aquellos basados en Chromium (Google Chrome, Microsoft Edge, Opera, Brave...) y Mozilla Firefox.
- Diseño de una interfaz moderna, limpia y atractiva.
- Desarrollo de diferentes gráficos descriptivos de los datos.
- Centralización de los datos y limpieza de los mismos para presentárselo correctamente al usuario.

- Facilitar al usuario un acceso sencillo a encuestas que haya marcado como favoritas.

Objetivos de aprendizaje:

- Comprender la creación de una aplicación desde 0, desde el back-end hasta el front-end.
- Aprender a utilizar frameworks útiles y actualmente muy usados como son Angular e Ionic.
- Profundizar en el conocimiento sobre web scrapping y lenguajes como JavaScript o distribuciones de SQL como PostgreSQL.
- Realizar todas las etapas y crear la documentación de un proyecto software desde el inicio para comprender la profesión, así como el tratamiento de los datos internos.

1.2. Análisis de otras aplicaciones

Las siguientes aplicaciones no son aplicaciones al uso. Son páginas web tras las cuales hay un grupo de personas que se encargan de mantener la información actualizada de manera constante, con los gráficos realizados de manera manual. Específicamente, no hay aplicaciones similares con el mismo tema que existan, pues la aplicación que se realizará tomará los datos de Wikipedia, actualizada diariamente, para poder mostrar los datos de manera gráfica y cómoda para el usuario.

Las siguientes páginas web se conocen como agregadores de encuestas. Cada vez que se publica una encuesta electoral, sea del ámbito que sea, son informadas y proceden con la recopilación de los pertinentes datos y la creación de los gráficos.

1.2.1. Europe Elects

Europe Elects[1] es una página web alemana especializada en la agregación de encuestas a nivel europeo, hermanada con otras ramas conocidas como *Africa Elects*, *America Elects*, *Asia Elects* y *Oceania Elects*, encargadas de lo mismo en sus respectivos continentes. Ha tenido tanta influencia en los últimos años, que los colores que utilizan para clasificar a los partidos según sus familias políticas europeas (colores rojizos para la izquierda, colores azulados para la derecha, amarillo para liberales y verde para verdes) se han ido utilizando como estándares *de facto*, pese a que son colores que generalmente se asocian a las ideologías mencionadas.

Fue fundada en 2014 por Tobias Gerhard Schminke, como una cuenta personal de agregación de encuestas (ver figura 1.1) gestionada por él mismo, pero debido al éxito que tenía, aumentó el equipo y, a fecha de Abril de 2022, está compuesto por 40 voluntarios de todo el mundo.

La página web, y el grupo en general, provee resultados de sondeos nacionales (y estatales o regionales en países como Alemania, España o Italia) a través de sus cuentas en las diversas redes sociales, como Twitter o su canal de Telegram.

En la página web (ver figura 1.2), además, se encuentran proyecciones para el Parlamento Europeo, elecciones que se convocan cada 5 años. Las próximas, a fecha de 2022, serán para 2024. También tiene un calendario electoral en el que figuran todas las elecciones programadas para el año vigente, para toda Europa.

Los gráficos que aparecen en la web están realizados con una herramienta online llamada Flourish [2], que permite la creación de los mismos de una manera sencilla, vistosa y gratuita.

Puntos fuertes:

- Gráficos vistosos con colores diferenciadores para cada partido.
- Web ordenada.
- Interacción con redes sociales.
- Diferenciación por países (en nuestro caso se puede utilizar como símil la región).

Puntos débiles:

- Gráficos realizados a mano.

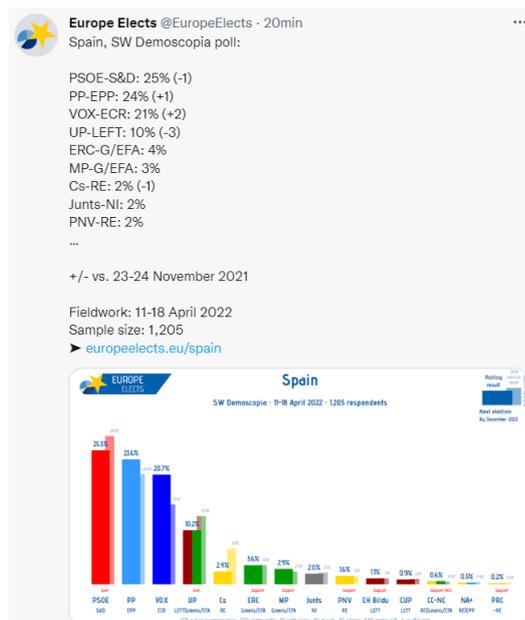


Figura 1.1: Tweet de Europe Elects con los resultados

1.2.2. Electomanía

Electomanía[3] es una empresa que comenzó únicamente como un agregador de encuestas, pero debido a su éxito y la creación de un panel online electoral llamado ElectoPanel y su gran éxito prediciendo los resultados de las elecciones, tanto nacionales como autonómicas, se iniciaron en el negocio demoscópico. Al igual que la anterior, los gráficos utilizados se realizan a través de Flourish Studio de manera manual. Además, aparte de los temas electorales, también realiza estudios demoscópicos generalistas, sobre el comportamiento de los españoles (gustos, predilecciones por la forma de estado, pertenencia a la Unión Europea, etc.). Ver figuras 1.3 y 1.4.

Puntos fuertes:

- Información centralizada.
- Gráficos interactivos y llamativos.
- Foro de usuarios.
- Permite comentarios.

Puntos débiles:

- Web desordenada.
- Web algo obsoleta.
- No se diferencian entre encuestas a nivel nacional, autonómico, municipal o extranjeras.
- Gráficos realizados a mano.

1.2.3. Electocracia

La web de Electocracia[4], a diferencia de las dos anteriores, consta únicamente de una página, en la que se presenta una tabla con la media de la regresión local de las últimas encuestas electorales para cada partido nacional (que por orden de mayor a menor, a fecha de Abril de 2022 y según la misma web son PSOE, PP, VOX, UP, Más País y Cs), así como su proyección media de escaños y una posible tendencia futura, así como el *rating* interno de cada partido (en la que ya constan más criterios personales y de análisis político). A la derecha de la mencionada tabla se encuentra un gráfico de regresión local para cada partido, cada uno con su color representativo. Bajo los

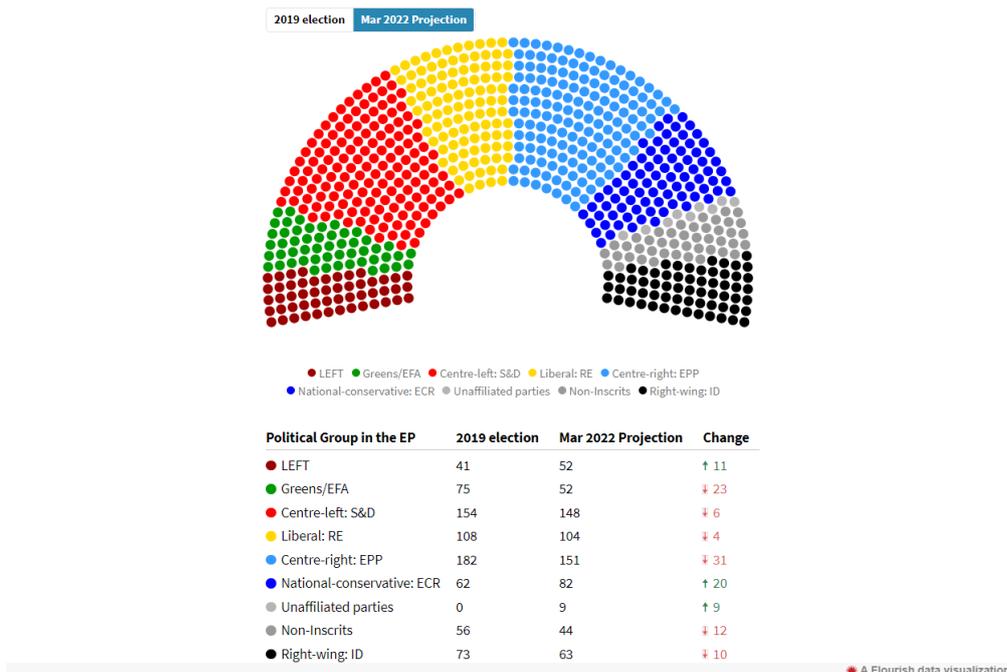


Figura 1.2: Proyección de Europe Elects.

mencionados elementos se encuentran todas las encuestas electorales desde 2017, dando sus resultados así como la empresa encargada de llevarla a cabo y el medio que lo publica. Ver figura 1.5.

Puntos fuertes:

- Web simple. Lo que quieres es lo que tienes.
- Gráfico resumen.
- Gran detalle por partido.
- Tabla con todas las encuestas electorales.

Puntos débiles:

- No se recogen las elecciones autonómicas.
- Datos añadidos a mano.
- Sin gráficos para cada encuesta.

1.2.4. Electograph

Electograph[5], de todas las mencionadas, es aquella que tiene una web mucho más limpia y completa. Tiene desglose para generales y para autonómicas, y también las tiene para europeas y municipales. Ofrece mucha información, así como promedios y tablas con todas las encuestas publicadas actualmente. Ver figura 1.6.

Puntos fuertes:

- Web completa con diseño llamativo.
- Web bien dividida por apartados, muy ordenada.
- Cuadros de resumen con gráficos de medias.
- Información para elecciones generales, autonómicas, municipales y europeas.

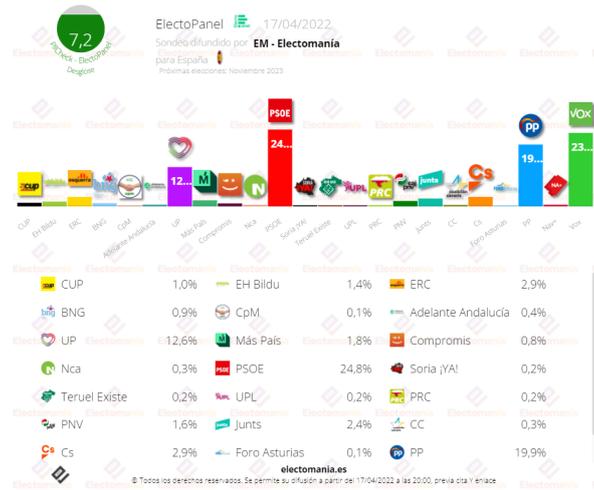


Figura 1.3: Gráfico de barras del ElectoPanel



Figura 1.4: Estimación de escaños del ElectoPanel

- Tabla centralizada con los resultados de todas las encuestas publicadas para la presente legislatura.

Puntos débiles:

- Los datos son añadidos a mano.

1.3. Estructura de la memoria

La presente memoria se estructura en los siguientes capítulos:

- En el segundo capítulo, se presenta el procedimiento de planificación seguido, UP en este caso, así como el análisis de riesgos y las desviaciones sobre la planificación inicial.
- En el tercer capítulo, se presenta el análisis de los requisitos, los casos de uso y los modelos iniciales en la concepción de la aplicación.



Figura 1.5: Web de Electocracia



Figura 1.6: Web de Electograph

- En el cuarto capítulo, se explica el diseño inicial de la aplicación: las intenciones para su creación, el funcionamiento intencionado, etc.
- En el quinto capítulo se comenta la implementación seguida.
- En el sexto capítulo se presentan las pruebas realizadas para el seguimiento del buen funcionamiento de la aplicación.
- En el séptimo y último capítulo se comentan las conclusiones del proyecto, así como el trabajo futuro a realizar en este ámbito.

Capítulo 2

Planificación del proyecto

2.1. Introducción

En este capítulo se muestra el plan de desarrollo del proyecto software de la aplicación **1Poller**.

El proyecto ha sido ideado por el alumno Javier Martín de Benito y aceptado por la profesora Margarita Gonzalo Tasis, y para su desarrollo se empleará la metodología RUP (Rational Unified Process, Proceso Unificado de Rational).

Para ello, hay que adaptar la metodología a la cantidad de personas que participan en el proyecto, asignando roles. En este caso son 2 personas, Margarita Gonzalo, la tutora; y Javier Martín, el alumno encargado del proyecto, al que le serán asignados los siguientes roles, mostrados en la tabla 2.1,

Rol	Responsabilidad
Jefe de proyecto	Es el responsable de la asignación de roles, así como de la planificación del proyecto y de la supervisión del posterior desarrollo. Se compartirá este rol junto a la profesora Margarita Gonzalo.
Analista	Se encarga del análisis de requisitos y de las restricciones del software.
Diseñador	Tras el análisis, es el encargado de realizar el diseño del software.
Programador	Es el responsable de realizar la implementación final de la aplicación.
Encargado de pruebas	Es el responsable del diseño de las pruebas para cumplir los casos de uso.

Tabla 2.1: Roles en el proyecto

2.2. Glosario

En la tabla 2.2 se puede observar diferentes nombres que se utilizarán a lo largo del Trabajo.

Sigla	Significado
MS Project	Microsoft Project
VS Code	Visual Studio Code
HTML5	HyperText Markup Language 5
JS	JavaScript
CSS3	Cascading Style Sheets 3
UP	Unified Process.
TS	TypeScript

Tabla 2.2: Glosario

2.3. Artefactos del proyecto

- Plan de desarrollo de software
- Análisis y diseño
- Implementación de la aplicación
- Test y pruebas

2.4. Restricciones del proyecto

- Momento máximo de entrega: el límite de entrega se establece en 15 de Julio, pues es el límite de petición de defensa.

2.5. Planificación del proyecto

La planificación del proyecto se realizará en base a la metodología RUP (Proceso Unificado de Rational) como ya se ha mencionado antes. Esta metodología impone la división del proyecto en fases y estas se subdividen en iteraciones[6]. En la figura 2.1 se puede observar una representación de las fases de la planificación así como de las actividades a realizar.

2.5.1. Fase de inicio

En la fase de inicio no hay entregables software, pues este tipo de entregables han de presentarse durante la implementación. En esta fase, las metas primordiales son la definición del alcance del proyecto, los casos de uso que se puedan interceptar inicialmente, los requisitos de la aplicación, la estimación del coste, la definición inicial de la gestión de riesgos y la realización de una planificación ideal del proyecto que, posteriormente, podría ser editada si alguno de los riesgos presentados anteriormente llegan a materializarse.

2.5.2. Fase de elaboración

Mientras dure esta fase, se realizarán los pertinentes análisis del sistema que se quiere construir. Además, se realizarán los pertinentes análisis de dominio, creación de diagramas y bocetos iniciales de la aplicación, de manera que para la siguiente fase se pueda iniciar la construcción del software final. Esta fase tiene como fin el crear la documentación teórica de la aplicación, por lo que aún no tendría entregables software.

2.5.3. Fase de construcción

La fase de construcción es la más larga de todo el proyecto, pues es la fase en la que la aplicación se implementa y se prueba, con todo lo que conlleva: programación de la parte algorítmica, el diseño que llevará la aplicación y las sucesivas pruebas que se tienen que realizar para comprobar que no tiene errores y que, además, cumple con los casos de uso que han sido descritos con anterioridad.

2.5.4. Fase de transición

Esta última fase es aquella en la que la aplicación ya se ha terminado pero queda constatar que cumple con todas las necesidades del usuario final, así como el alcance que se determinó en su momento.

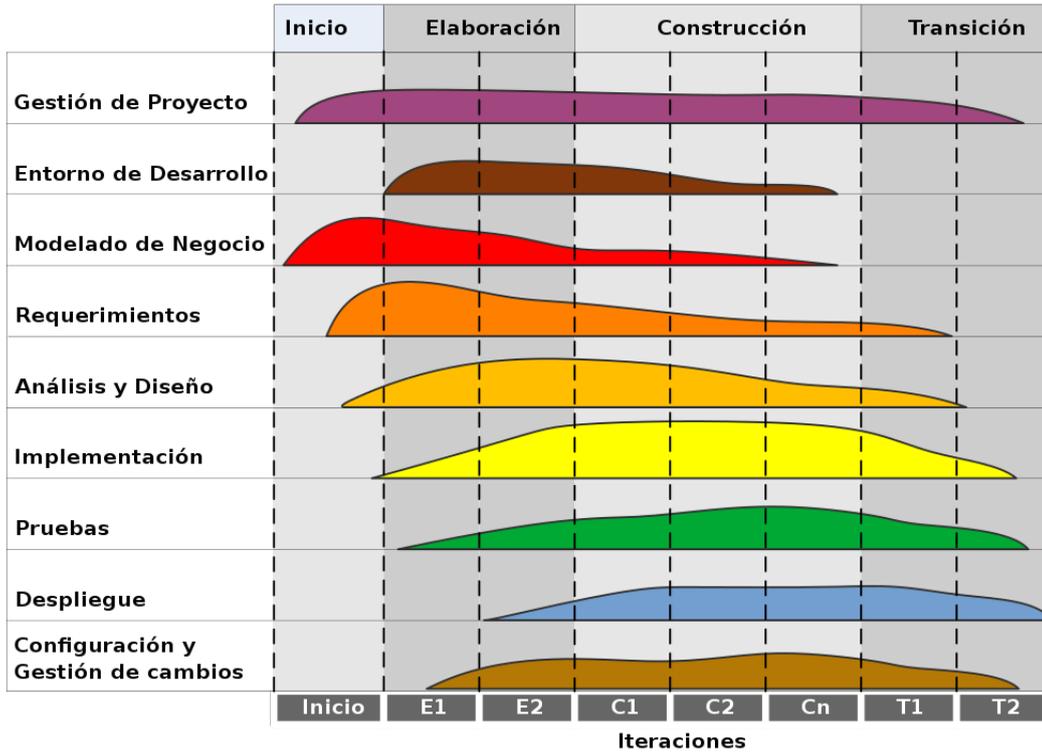


Figura 2.1: Fases de la metodología RUP

2.6. Calendario del proyecto

Como se ha indicado en el apartado anterior, las fases se han decidido en base a la metodología que se está utilizando. El reparto de las horas se muestra en la tabla 2.3.

Fase	Nº Iteraciones	Duración estimada
Inicio	1	28 horas
Elaboración	1	75 horas
Construcción	2	137 horas
Transición	1	60 horas
Total	6	300 horas

Tabla 2.3: Resumen de las fases de la planificación

Se estima que el proyecto dure, en total, unas 300 horas, que es la cantidad de horas determinadas por la asignatura Trabajo de Fin de Grado, ya que es una asignatura con 12 créditos ECTS (cada uno de ellos son 25 horas de trabajo).

2.6.1. Fase de inicio

La fase de inicio tendrá una duración estimada de 28 horas, desde el sábado 2 de abril hasta el martes 19 de abril. Ver figura 2.2 y tabla 2.4.

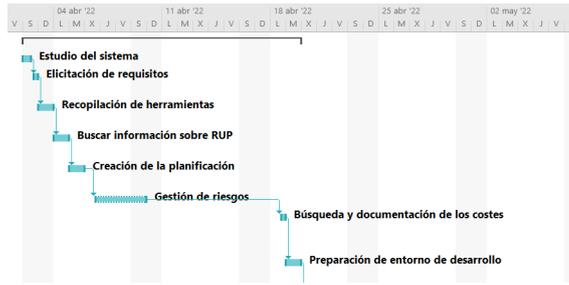


Figura 2.2: Fase de inicio

Tarea	Duración	Comienzo	Fin
Estudio del sistema	4 horas	02/04/22	02/04/22
Elicitación de requisitos	3 horas	02/04/22	02/04/22
Recopilación de herramientas	3 horas	02/04/22	03/04/22
Buscar información sobre RUP	4 horas	03/04/22	04/04/22
Creación de la planificación	6 horas	04/04/22	05/04/22
Gestión de riesgos	6 horas	06/04/22	09/04/22
Búsqueda y documentación de costes	2 horas	18/04/22	18/04/22
Preparación del entorno de desarrollo	4 horas	18/04/22	19/04/22

Tabla 2.4: Fase de inicio

2.6.2. Fase de elaboración

La fase de elaboración tendrá una duración estimada de 75 horas divididas en 2 iteraciones. Dependiendo de la retroalimentación que se reciba, las investigaciones podrán ser enfocadas en un aspecto o en otro. Se espera que se comience con la fase el 18 de abril de 2022 y se termine el 8 de mayo de 2022. Ver figura 2.3 y tabla 2.5

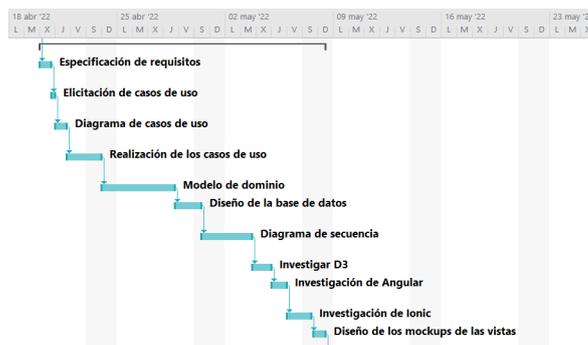


Figura 2.3: Fase de elaboración

Tarea	Duración	Comienzo	Fin
Especificación de requisitos	3 horas	19/04/22	20/04/22
Elicitación de casos de uso	2 horas	20/04/22	20/04/22
Diagrama de casos de uso	2 horas	20/04/22	21/04/22
Realización de los casos de uso	6 horas	21/04/22	23/04/22
Modelo de dominio	18 horas	23/04/22	28/04/22
Diseño de la Base de Datos	4 horas	28/04/22	30/04/22
Diagrama de secuencia	20 horas	30/04/22	03/05/22
Investigar D3	5 horas	03/05/22	04/05/22
Investigación de Angular	5 horas	04/05/22	05/05/22
Investigación de Ionic	5 horas	05/05/22	07/05/22
Diseño de los mockups de las vistas	6 horas	07/05/22	08/05/22
Fin de elaboración	0 horas	08/05/22	08/05/22

Tabla 2.5: Fase de elaboración

2.6.3. Fase de construcción

Como se ha mencionado anteriormente, esta fase es la de mayor duración, ya que engloba el desarrollo y testeo de la aplicación. Se estima una duración de 157 horas divididas en 2 iteraciones, la segunda teniendo retroalimentación de la primera para poder desarrollar nuevas pruebas a las características editadas o añadidas. Ver figura 2.4 y tabla 2.6 para el desarrollo de la iteración número 1 y la figura 2.5 y la tabla 2.7 para el desarrollo de la iteración número 2.

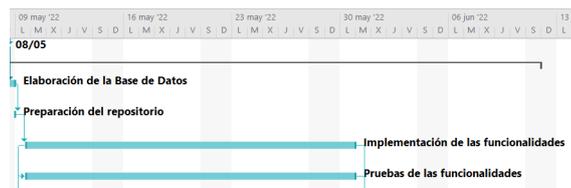


Figura 2.4: Fase de construcción, iteración 1

Tarea	Duración	Comienzo	Fin
Elaboración de la Base de Datos	5 horas	08/05/22	08/05/22
Preparación del repositorio	1 horas	08/05/22	08/05/22
Implementación de las funcionalidades	100 horas	09/05/22	30/05/22
Pruebas de las funcionalidades	100 horas	09/05/22	30/05/22
Documentación del programa	130 horas	09/05/22	06/06/22

Tabla 2.6: Fase de construcción, iteración 1



Figura 2.5: Fase de construcción, iteración 2

Tarea	Duración	Comienzo	Fin
Corrección de problemas	30 horas	31/05/22	06/06/22
Implementación de nuevas pruebas	30 horas	31/05/22	06/06/22
Documentación del programa	130 horas	09/05/22	06/06/22
Creación del icono	1 horas	06/06/22	06/06/22

Tabla 2.7: Fase de construcción, iteración 2

Como se observa en las tablas 2.6 y 2.7, la tarea **Documentación del programa** es compartida por ambas iteraciones. Esto se debe a que es una tarea que se va realizando a lo largo de la creación de la aplicación y no al final ni al comienzo.

2.6.4. Fase de transición

La fase de transición es bastante más corta que las anteriores, pues engloba la corrección de la presente memoria y el despliegue definitivo de la aplicación de manera correcta. El desglose de la fase viene en la figura 2.6 y la tabla 2.8.



Figura 2.6: Fase de transición

Tarea	Duración	Comienzo	Fin
Revisión de la documentación	10 horas	06/06/22	09/06/22
Creación del manual de usuario	20 horas	09/06/22	13/06/22
Revisión de la memoria	30 horas	13/06/22	19/06/22

Tabla 2.8: Fase de transición

2.6.5. Entregables del proyecto

Durante las diferentes fases, se hará entrega a la tutora de diferentes entregables que constituyan maneras de conocimiento para saber cómo va encaminado el proyecto.

- Fase de inicio
 - Plan de gestión software
 - Versión inicial de la gestión de riesgos
- Fase de elaboración
 - Diagramas de caso de uso, dominio, secuencia
 - Versión inicial de los casos de uso
 - Arquitectura del sistema
 - Diseño inicial de la base de datos
 - Prototipos de diseño, mockups
- Fase de construcción
 - Versión inicial del manual de usuario
 - Versión beta de la aplicación
 - Resultados de las pruebas y cambios
- Fase de transición
 - Versión final de la aplicación
 - Versión final del manual de usuario
 - Versión final del manual de instalación
 - Memoria

2.7. Análisis de riesgos

En el presente apartado se definen los riesgos interceptados que puedan obstaculizar el normal desarrollo del proyecto, así como la probabilidad de que estos se presenten, el impacto que puedan ocasionar, la estrategia a seguir si ocurren y los planes que se puedan realizar en caso de ocurrencia.

La descripción de los riesgos interceptados se pueden observar desde la tabla 2.9 hasta la tabla 2.20.

R0	Falta de experiencia
Descripción	El alumno ha utilizado JavaScript pero no Angular ni Ionic
Consecuencia	Ralentización del proyecto
Probabilidad	Alta
Impacto	Bajo
Estrategia	Mitigar el riesgo
Plan de acción	Búsqueda de documentación
Plan de contingencia	Revisar conocimientos

Tabla 2.9: Riesgo 0- Falta de experiencia

R1	Falta de experiencia en la organización
Descripción	El alumno realiza las tareas en orden erróneo
Consecuencia	Replanteamiento de la planificación, ralentización
Probabilidad	Baja
Impacto	Medio
Estrategia	Evitación del riesgo
Plan de acción	Realización de otras tareas bien organizadas
Plan de contingencia	Ordenación de las tareas de manera que realmente dependan de las correctas

Tabla 2.10: Riesgo 1- falta de experiencia en la organización

R2	Baja calidad del código
Descripción	La aplicación está mal codificada, no funciona o produce fallas
Consecuencia	Ralentización de la implementación
Probabilidad	Baja
Impacto	Alto
Estrategia	Comunicar el riesgo
Plan de acción	Corrección de las fallas
Plan de contingencia	Revisión continua de la implementación del código y sus dependencias

Tabla 2.11: Riesgo 2- baja calidad del código

R3	Falta de disponibilidad
Descripción	El alumno se encuentra ausente por realizar el otro TFG o la asignatura pendiente
Consecuencia	Ralentización del proyecto
Probabilidad	Alta
Impacto	Alto
Estrategia	Comunicar el riesgo
Plan de acción	Planificación más realista
Plan de contingencia	Dedicación absoluta al proyecto en los momentos posibles

Tabla 2.12: Riesgo 3- Falta de disponibilidad

R4	Pérdida de entregables
Descripción	Pérdida de información o de los datos
Consecuencia	Volver a realizar el entregable perdido
Probabilidad	Muy baja
Impacto	Crítico
Estrategia	Evitación del riesgo
Plan de acción	No aplica
Plan de contingencia	Utilizar un servidor de control de versiones en la nube gitlab.inf.uva.es

Tabla 2.13: Riesgo 4- Pérdida de entregables

R5	Falta de medios
Descripción	Los medios utilizados para la creación del proyecto comienzan a fallar o desaparecen, impidiendo la continuación del proyecto
Consecuencia	La planificación ha de aplazarse e, incluso, el proyecto puede llegar a pararse definitivamente
Probabilidad	Muy baja
Impacto	Crítico
Estrategia	Reducción del riesgo
Plan de acción	No aplica
Plan de contingencia	Búsqueda de nuevas herramientas

Tabla 2.14: Riesgo 5- Falta de medios

R6	Cambios en las fuentes de datos
Descripción	Las fuentes de datos sufren cambios en su estructura, provocando fallos en su extracción.
Consecuencia	Algunos datos no pueden actualizarse o no extraerse
Probabilidad	Alta
Impacto	Medio
Estrategia	Mitigar el riesgo
Plan de acción	No aplica
Plan de contingencia	Cambio en la extracción

Tabla 2.15: Riesgo 6- Cambios en las fuentes de datos

R7	Diseño incorrecto
Descripción	La aplicación se diseña de manera incorrecta o no coherente con la realidad
Consecuencia	El proyecto ha de replantearse
Probabilidad	Baja
Impacto	Alto
Estrategia	Evitación del riesgo
Plan de acción	Reformulación del diseño de la aplicación
Plan de contingencia	Estudios continuados sobre las nuevas implementaciones

Tabla 2.16: Riesgo 7- Diseño incorrecto

R8	Cambio de plazos
Descripción	Los plazos de los entregables, así como el plazo de finalización del proyecto cambian
Consecuencia	La planificación queda obsoleta (parcial o totalmente)
Probabilidad	Media
Impacto	Bajo
Estrategia	Aceptación del riesgo
Plan de acción	No aplica
Plan de contingencia	Actualización del plan de proyecto

Tabla 2.17: Riesgo 8- Cambio de plazos

R9	Nuevas especificaciones del proyecto
Descripción	Se descubren nuevas especificaciones del proyecto: requisitos, casos de uso, etc.
Consecuencia	Nuevas características han de ser implementadas
Probabilidad	Media
Impacto	Medio
Estrategia	Aceptación del riesgo
Plan de acción	Previsión de todas las especificaciones al comienzo del proyecto
Plan de contingencia	Codificación y adaptación de las nuevas especificaciones

Tabla 2.18: Riesgo 9- Nuevas especificaciones del proyecto

R10	Falta de experiencia en planificación de riesgos
Descripción	No se han planificado correctamente las acciones o respuestas contra los riesgos
Consecuencia	Consecuencias no planteadas ante los riesgos sufridos
Probabilidad	Media
Impacto	Bajo/Medio/Alto/Crítico
Estrategia	Comunicación, evitación, reducción
Plan de acción	Planificación con anterioridad de todos los riesgos posibles
Plan de contingencia	Análisis del riesgo y realizar un plan contra ello

Tabla 2.19: Riesgo 10- Falta de experiencia en planificación de riesgos

R11	No se han determinado todos los riesgos
Descripción	No se han planteado todos los riesgos posibles
Consecuencia	Aparición de riesgos desconocidos
Probabilidad	Media
Impacto	Bajo/Medio/Alto/Crítico
Estrategia	Evitación del riesgo
Plan de acción	Prever todos los riesgos al comienzo del proyecto
Plan de contingencia	Análisis del riesgo y realizar un plan para el nuevo riesgo

Tabla 2.20: Riesgo 11- No se han determinado todos los riesgos

2.8. Estimación de costes

La estimación de costes debería realizarse en base a lo aprendido en anteriores proyectos presentados. Sin embargo, en el Grado los costes no se tienen en cuenta y en proyectos externos no se suelen publicar, pues forma parte de estrategias empresariales. Aún así, como se ha estudiado en la asignatura Planificación y Diseño de Sistemas Computacionales, se pueden estimar de diversas maneras.

2.8.1. Costes hardware y software

Para la estimación de los costes de software únicamente se ha utilizado un ordenador portátil. Teniendo en cuenta su antigüedad y el uso que se le dará, resulta un coste de 60€, mostrado en la tabla 2.21.

Dispositivo	Horas utilizadas	Horas en proyecto
Ordenador portátil	13.140 (4 años y medio, 8h/día)	300 horas

Dispositivo	Coste de compra	Coste
Portátil	1.031€	60€

Tabla 2.21: Costes hardware

Para el software, aunque se prefiere software gratuito, se han utilizado herramientas de pago, aunque realmente, al tratarse de herramientas cuyas licencias las provee la Universidad para el alumnado sin coste alguno, realmente no son costes que cuenten a lo largo del proyecto. Ver tabla 2.22.

Software	Coste
Microsoft Project Professional 2019	1659€
Windows 10 home	15€ (licencia OEM)
Ubuntu	0€
Visual Studio Code	0€
Astah Professional	30€/año
Balsamiq Mockups (versión de prueba)	0€

Tabla 2.22: Costes software

2.8.2. Conexión a internet y tarifa móvil

La conexión a internet y tarifa móvil se obtienen del paquete Vodafone de línea ilimitada + fibra óptica de 300 mb por un total de 49€ mensuales. En total, por 3 meses, son 157€.

2.9. Seguimiento del proyecto

En la tabla 2.23 se puede observar los desvíos de planificación entre las fechas de comienzo y fin planificadas y las reales.

Etapa	Fecha inicio estimada	Fecha fin estimada	Fecha inicio real	Fecha fin real	Desvío
Inicio	02/04/2022	19/04/2022	02/04/2022	01/05/2022	12 días
Elaboración	19/04/2022	08/05/2022	01/05/2022	23/05/2022	15 días
Construcción	08/05/2022	31/05/2022	23/05/2022	13/06/2022	13 días
Construcción	31/05/2022	06/06/2022	13/06/2022	29/06/2022	23 días
Transición	06/06/2022	19/06/2022	29/06/2022	05/07/2022	17 días

Tabla 2.23: Desvío frente a la planificación inicial

Como se puede observar, ha habido grandes desvíos en la planificación inicial del proyecto. Esto es debido, primordialmente, a la aparición continuada de los riesgos R0, falta de experiencia y R3, falta de disponibilidad.

- Para la creación del servidor y del cliente se han tenido que aprender, desde cero, varios frameworks que no se han utilizado a lo largo del Grado. Django para la creación del servidor y tanto Ionic como Angular para la creación del cliente. Estos dos últimos, de una complejidad mucho mayor, han ocupado varios días en la fase de construcción, aunque la formación en Django también ha ocupado unos cuantos días.
- Debido a la necesidad de compaginar el presente Trabajo de Fin de Grado con otro más y una asignatura pendiente que requerían, en algunos momentos del proceso, más tiempo, se ha dejado en un segundo plano el desarrollo de este mismo hasta acabar los otros dos. La asignatura se terminó con la realización de su examen a principios de Junio y el otro Trabajo de Fin de Grado, a finales de Junio, por lo que a partir de entonces, hubo una dedicación completa para este trabajo.

Capítulo 3

Análisis

En el presente capítulo se explica un exhaustivo análisis del funcionamiento deseado de la aplicación. Se realiza un examen de los requisitos deseados, así como un estudio de los casos de uso en el que los actores encontrados se puedan hallar.

3.1. Análisis de requisitos

El análisis de requisitos se realiza acorde a lo necesitado por el usuario. Debido a que el proyecto ha sido ideado por el alumno, es el mismo quien entenderá los requisitos que sean necesarios.

3.1.1. Requisitos funcionales

En la tabla 3.1 se observan el desglose y la descripción de todos los requisitos funcionales del proyecto.

ID	Requisito	Descripción
RF01	Visualización gráfica	El sistema mostrará los resultados de una encuesta de forma gráfica: parlamento y barras.
RF02	Tabla encuestas	El sistema mostrará un resumen de todas las encuestas electorales para unas mismas elecciones de un mismo lugar.
RF03	Resumen numérico encuesta	El sistema mostrará una tabla para cada encuesta en la que se muestren los datos técnicos.
RF04	Actualización	El sistema actualizará los datos de una CCAA cuando el usuario acceda a los datos.
RF05	Resultados electorales	El sistema proveerá un apartado de la última elección.
RF06	Gráfico de pactos	El sistema proveerá un gráfico al estilo "Pactómetro" que permita elegir diferentes partidos y observar si se llega a una mayoría absoluta.
RF07	Tendencias	El sistema mostrará los cambios respecto a la anterior encuesta de una misma casa.
RF08	Favoritos	El sistema permitirá el poder marcar una encuesta como favorita.
RF09	Lista de favoritos	Existirá un apartado de comunidades autónomas favoritas y encuestas marcadas.
RF10	Información de partidos	El sistema proveerá información sobre los partidos políticos que muestren cuántos escaños tienen en cada asamblea.
RF11	Gráficos animados	El sistema proveerá un sistema de gráficos animados.
RF12	Calendario electoral	El sistema proveerá un calendario electoral para las próximas elecciones.
RF13	Guardar pactos	El sistema permitirá guardar los pactos calculados.
RF14	Info región	El sistema proveerá la información de cada región, presidente, partidos, etc.

ID	Requisito	Descripción
RF15	Comparar encuestas	El sistema permitirá comparar dos encuestas cualesquiera de una misma comunidad y empresa encuestadora
RF16	Atrás	El sistema proveerá un botón para volver atrás en cualquier momento

Tabla 3.1: Requisitos funcionales

3.1.2. Requisitos de información

En la tabla 3.2 se observan el desglose y la descripción de los requisitos de información del proyecto.

ID	Requisito	Descripción
RI01	Información elecciones	El sistema almacenará la información de las elecciones, en concreto: partido y escaños obtenidos.
RI02	Información partido	El sistema almacenará la información de los partidos, en especial: nombre, posición, color, etc.
RI03	Información partido extendida	El sistema almacenará la información de los partidos, añadiendo a lo anterior una breve descripción.
RI04	Información región	El sistema almacenará información sobre la región o país si es a nivel general: nombre oficial, cámara legislativa, jefe del ejecutivo actual, composición de la cámara, últimas elecciones y próximas elecciones. También se almacenará la bandera.
RI05	Información pacto	El sistema almacenará el pacto realizado: dónde, encuesta, partidos seleccionados
RI06	Información cámara legislativa	Para cada cámara legislativa, se almacenarán los escaños que la componen y su composición.
RI07	Información favoritos	El sistema almacenará las encuestas favoritas: dónde, cuál.

Tabla 3.2: Requisitos de información

3.1.3. Requisitos no funcionales

Los requisitos no funcionales son aquellos aspectos técnicos que el sistema ha de cumplir a la hora de la presentación frente al usuario, usabilidad, rapidez, diseño o acceso, entre otros.

Respecto a los de este proyecto, se puede observar en la tabla 3.3.

ID	Descripción	Importancia
RNF01	El sistema proveerá un diseño sencillo de aprender	Deseable
RNF02	El sistema será eficiente a la hora de su uso	Deseable
RNF03	El sistema ayudará a la retención del aprendizaje sobre el tiempo	Deseable
RNF04	El sistema deberá ayudar al usuario a no cometer errores	Deseable
RNF05	Satisfacción del usuario con el sistema	Deseable
RNF06	El sistema almacenará los datos en una base de datos SQL	Crítica
RNF07	El sistema proveerá una interfaz sencilla de utilizar	Deseable
RNF08	El sistema proveerá una gama de colores de interfaz agradable	Deseable
RNF09	El sistema tendrá la interfaz dividida en apartados diferenciados	Crítica
RNF10	El sistema será multiplataforma	Crítica
RNF11	El sistema utilizará los colores característicos de cada partido	Deseable
RNF12	El sistema utilizará el formato UTF-08	Crítica
RNF13	El sistema será accesible desde navegadores Chrome y Firefox	Deseable

Tabla 3.3: Requisitos no funcionales

3.1.4. Reglas de negocio

En la tabla 3.4 se presenta la regla de negocio definida en el proyecto.

ID	Descripción	Importancia
RDN01	Extracción de información	La extracción de la información se realizará por Web Scrapping contra las páginas de Wikipedia en inglés, al menos para las encuestas

Tabla 3.4: Reglas de negocio

3.2. Casos de uso

En esta sección se explican los casos de uso que se han encontrado, así como el diagrama principal y sus actores.

3.2.1. Actores principales

Únicamente se ha hallado un actor principal, el usuario de la aplicación:

- **Usuario app:** es la persona que interactúa con el sistema para poder obtener la información que desee.

3.2.2. Diagrama de casos de uso

En la figura 3.1 se puede observar el Diagrama de Casos de uso, con los caso de uso descritos y el actor.

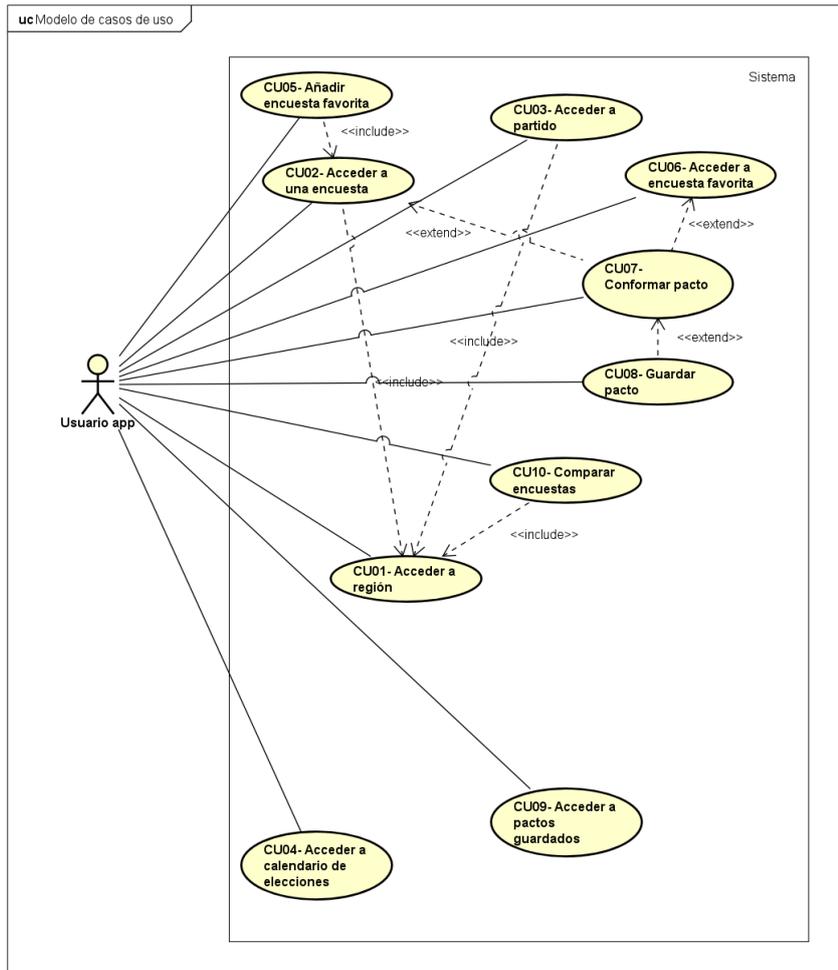


Figura 3.1: Diagrama de Casos de Uso

3.2.3. Descripción de los casos de uso

Los casos de uso se describen desde la figura 3.2 hasta la figura 3.11. Las tablas se han obtenido por medio de Astah.

ITEM	VALUE
UseCase	CU01- Acceder a región
Summary	El actor usuario desea entrar en la ficha de información de una región en específico
Actor	Usuario app
Precondition	
Postcondition	
Base Sequence	<ol style="list-style-type: none">1. El usuario selecciona el listado de regiones2. El sistema muestra el listado de regiones3. El usuario elige una región en concreto4. El sistema obtiene la información de la región seleccionada5. El sistema calcula la información complementaria6. El sistema muestra la ficha de información de la región seleccionada7. El caso de uso finaliza
Branch Sequence	3a. El usuario selecciona ir atrás y el caso de uso queda sin efecto.
Exception Sequence	
Sub UseCase	
Note	

Figura 3.2: Caso de Uso 1- Acceder a región

ITEM	VALUE
UseCase	CU02- Acceder a una encuesta
Summary	El usuario desea visualizar los resultados de una encuesta electoral en específico.
Actor	Usuario app
Precondition	El usuario ha entrado en la ficha de una región
Postcondition	
Base Sequence	<ol style="list-style-type: none"> 1. Se realiza el caso de uso Acceder a región. 2. El usuario selecciona mostrar encuestas electorales 3. El sistema obtiene las encuestas electorales de la región seleccionada 4. El usuario selecciona una encuesta electoral 5. El sistema obtiene los datos de la encuesta 6. El sistema muestra los resultados obtenidos 7. El caso de uso finaliza
Branch Sequence	2a. 4a. El usuario selecciona volver para atrás y el caso de uso queda sin efecto.
Exception Sequence	
Sub UseCase	CU01- Acceder a región
Note	

Figura 3.3: Caso de Uso 2- Acceder a una encuesta

ITEM	VALUE
UseCase	CU03- Acceder a partido
Summary	El usuario desea obtener información de un partido en concreto
Actor	Usuario app
Precondition	El usuario ha entrado en una región
Postcondition	
Base Sequence	<ol style="list-style-type: none">1. El usuario selecciona un partido de una ficha de encuesta2. El sistema obtiene los datos del partido3. El sistema muestra los datos obtenidos4. El caso de uso finaliza
Branch Sequence	
Exception Sequence	
Sub UseCase	CU01- Acceder a región, CU02- Acceder a una encuesta, CU06- Acceder a encuesta favorita
Note	

Figura 3.4: Caso de Uso 3- Acceder a partido

ITEM	VALUE
UseCase	CU04- Acceder a calendario de elecciones
Summary	El usuario desea obtener información acerca de las elecciones más próximas
Actor	Usuario app
Precondition	
Postcondition	
Base Sequence	<ol style="list-style-type: none"> 1. El usuario selecciona el calendario de elecciones 2. El sistema obtiene las fechas de todas las elecciones siguientes de las regiones. 3. El sistema muestra un calendario con la fecha de las elecciones. 4. El caso de uso finaliza
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Figura 3.5: Caso de Uso 4- Acceder a calendario de elecciones

ITEM	VALUE
UseCase	CU05- Añadir encuesta favorita
Summary	El usuario desea marcar una encuesta como favorita
Actor	Usuario app
Precondition	El usuario tiene la aplicación iniciada
Postcondition	
Base Sequence	<ol style="list-style-type: none">1. Se realiza el caso de uso Acceder a una encuesta2. El usuario marca la encuesta como favorita3. El sistema almacena la encuesta como favorita4. El caso de uso finaliza
Branch Sequence	
Exception Sequence	
Sub UseCase	CU02- Acceder a una encuesta
Note	

Figura 3.6: Caso de Uso 5- Añadir encuesta favorita

ITEM	VALUE
UseCase	CU06- Acceder a encuesta favorita
Summary	El usuario desea acceder a una encuesta marcada como favorita
Actor	Usuario app
Precondition	El usuario debe haber marcado una encuesta como favorita
Postcondition	
Base Sequence	<ol style="list-style-type: none"> 1. El usuario selecciona favoritos 2. El sistema muestra los elementos favoritos 3. El usuario selecciona una encuesta favorita 4. El sistema obtiene los datos de la encuesta 5. El sistema muestra los datos de la encuesta 6. El caso de uso finaliza
Branch Sequence	3a. El usuario selecciona ir atrás y el caso de uso queda sin efecto
Exception Sequence	
Sub UseCase	
Note	

Figura 3.7: Caso de Uso 6- Acceder a encuesta favorita

ITEM	VALUE
UseCase	CU07- Conformar pacto
Summary	El usuario desea conformar un pacto entre partidos
Actor	Usuario app
Precondition	El usuario se encuentra dentro de la ficha de una encuesta
Postcondition	
Base Sequence	<ol style="list-style-type: none"> 1. El usuario selecciona "conformar pacto" 2. El sistema muestra la vista de conformar pactos 3. El usuario selecciona tantos partidos como desee 4. El sistema computa los datos 5. El sistema muestra el pacto conformado 6. El caso de uso finaliza
Branch Sequence	<ol style="list-style-type: none"> 1a. 3a. El usuario elige ir hacia atrás y el caso de uso queda sin efecto 6a. El usuario puede seleccionar guardar pacto y entonces se ejecuta el caso de uso Guardar Pacto
Exception Sequence	
Sub UseCase	CU02- Acceder a una encuesta, CU06- Acceder a encuesta favorita
Note	

Figura 3.8: Caso de Uso 7- Conformar pacto

ITEM	VALUE
UseCase	CU08- Guardar pacto
Summary	El usuario desea guardar un pacto formado por él mismo
Actor	Usuario app
Precondition	El usuario ha de haber conformado un pacto electoral
Postcondition	
Base Sequence	<ol style="list-style-type: none"> 1. El usuario selecciona "guardar pacto" 2. El sistema guarda el pacto conformado 3. El caso de uso finaliza
Branch Sequence	
Exception Sequence	
Sub UseCase	
Note	

Figura 3.9: Caso de Uso 8- Guardar pacto

ITEM	VALUE
UseCase	CU09- Acceder a pactos guardados
Summary	El usuario desea acceder a un pacto realizado anteriormente que haya sido guardado
Actor	Usuario app
Precondition	Debe haber sido realizado algún pacto y haber sido guardado
Postcondition	
Base Sequence	<ol style="list-style-type: none">1. El usuario selecciona Favoritos2. El sistema muestra los elementos favoritos3. El usuario selecciona un pacto guardado4. El sistema obtiene los datos del pacto guardado5. El sistema muestra el pacto6. El caso de uso finaliza
Branch Sequence	3a. El usuario selecciona ir atrás y el caso de uso queda sin efecto
Exception Sequence	
Sub UseCase	
Note	

Figura 3.10: Caso de Uso 9- Acceder a pactos guardados

ITEM	VALUE
UseCase	CU10- Comparar encuestas
Summary	El usuario desea comparar 2 encuestas
Actor	Usuario app
Precondition	El usuario se encuentra en la ficha de la región de las encuestas que quiere comparar
Postcondition	
Base Sequence	<ol style="list-style-type: none"> 1. El usuario selecciona ir a las encuestas de la región 2. El sistema obtiene las encuestas de la región seleccionada 3. El usuario selecciona 2 encuestas cualesquiera 4. El sistema obtiene los datos de las encuestas seleccionadas 5. El sistema compara los datos obtenidos 6. El sistema muestra los datos 7. El caso de uso finaliza
Branch Sequence	1a. 3a. El usuario selecciona ir para atrás y el caso de uso queda sin efecto
Exception Sequence	
Sub UseCase	CU01- Acceder a región
Note	

Figura 3.11: Caso de Uso 10- Comparar encuestas

3.3. Modelo de dominio

El modelo de dominio que se presenta en las siguientes líneas representa las clases conceptuales con las que se trabajará posteriormente. Estas clases son conceptos que se utilizarán posteriormente como base del proyecto. Ver figura 3.12.

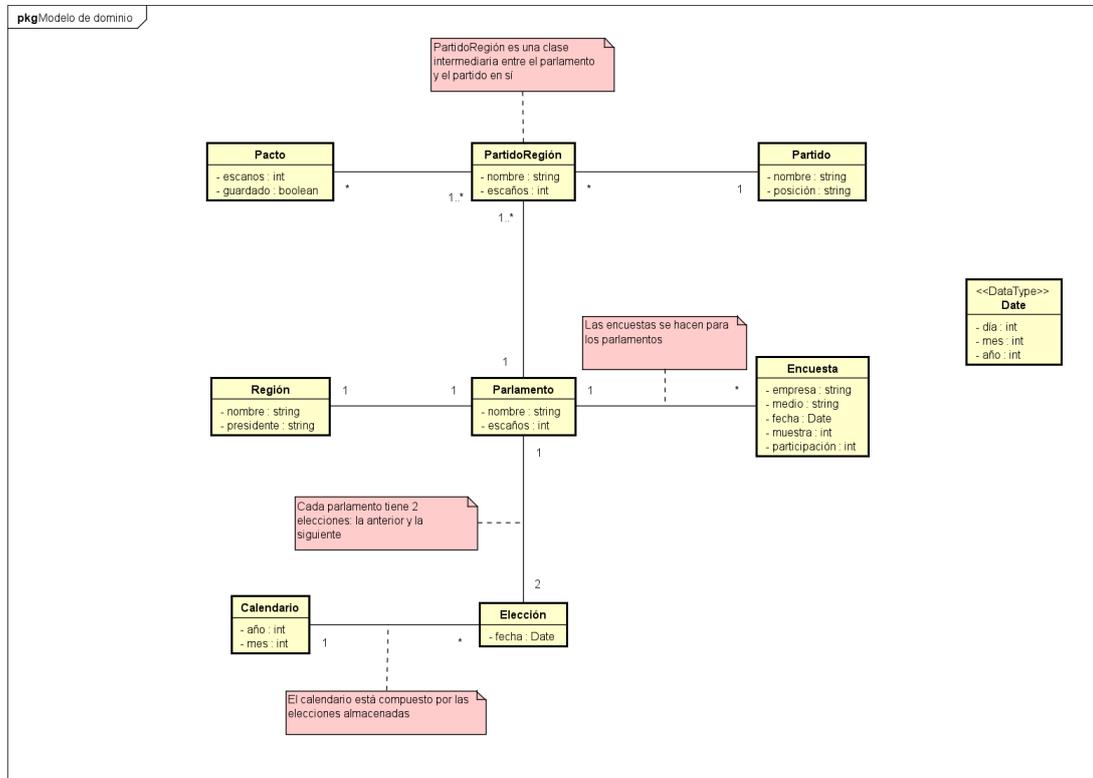


Figura 3.12: Modelo de dominio

3.3.1. Región

- **Descripción:** clase que modela una región que tiene un parlamento.
- **Responsabilidad:** representa a la Comunidad o a España y hace las veces de nexo de su información.
- **Atributos:**
 - nombre: nombre de la región.
 - presidente: presidente de la región.

3.3.2. Parlamento

- **Descripción:** clase que modela el parlamento de una región.
- **Responsabilidad:** representa toda la información del parlamento que hay en cada región, así como de referencia a todas las regiones. Está compuesto por partidos.
- **Atributos:**
 - nombre: nombre del parlamento (Congreso de los Diputados, Asamblea, Parlamento, Cortes).
 - escaños: número de escaños que hay en el parlamento.

3.3.3. Encuesta

- **Descripción:** clase que modela una encuesta electoral.
- **Responsabilidad:** asociada a un Parlamento, informa de los resultados para cada Partido que se presenta a una Elección.
- **Atributos:**
 - casa: empresa demoscópica que realiza la encuesta.
 - medio: puede no tenerlo, medio de comunicación que ha encargado la encuesta.
 - fecha: fecha de la obtención de las respuestas.
 - muestra.

3.3.4. Partido

- **Descripción:** clase que modela un partido político.
- **Responsabilidad:** sirve de almacén de la información general de un partido político.
- **Atributos:**
 - nombre: nombre del partido político.
 - posición: posición en el espectro político.

3.3.5. PartidoRegion

- **Descripción:** modela el partido político de una región.
- **Responsabilidad:** representa al partido político que se encuentra en un Parlamento en específico.
- **Atributos:**
 - nombre: nombre del partido político.
 - escaños: escaños que tiene en un Parlamento.

3.3.6. Pacto

- **Descripción:** clase que modela un pacto realizado.
- **Responsabilidad:** sirve de representación de los pactos que se puedan formar a base de PartidosRegion.
- **Atributos:**
 - escaños: número total de escaños que tienen los PartidoRegion añadidos al pacto, en total.

3.3.7. Elección

- **Descripción:** clase que modela una elección a un Parlamento.
- **Responsabilidad:** es el depósito de la información que hay respecto a una elección en particular.
- **Atributos:**
 - fecha: fecha en la que está programada una elección.

3.3.8. Calendario

- **Descripción:** clase que modela el calendario electoral.
- **Responsabilidad:** hace las veces de conjunto de todas las elecciones programadas para un futuro.
- **Atributos:**
 - año: año en el que ocurrirá la elección, representado de manera diferenciada.
 - mes: mes en el que ocurrirá la elección, representado de manera diferenciada.

3.4. Diagramas de secuencia

En este apartado se enseñan los diagramas de secuencia, que muestran el funcionamiento conceptual del sistema en los diferentes casos de uso. Ver figuras desde 3.13 hasta 3.22.

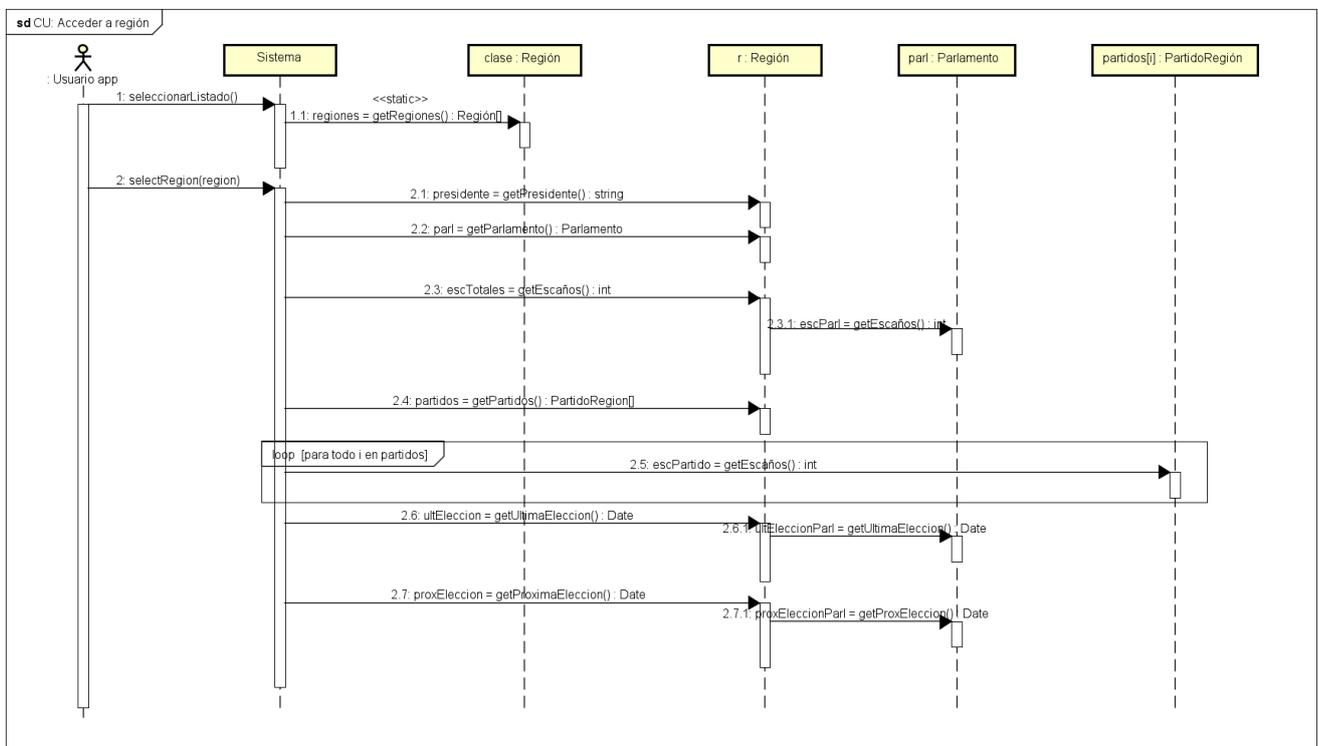


Figura 3.13: Secuencia de caso de Uso 1- Acceder a región

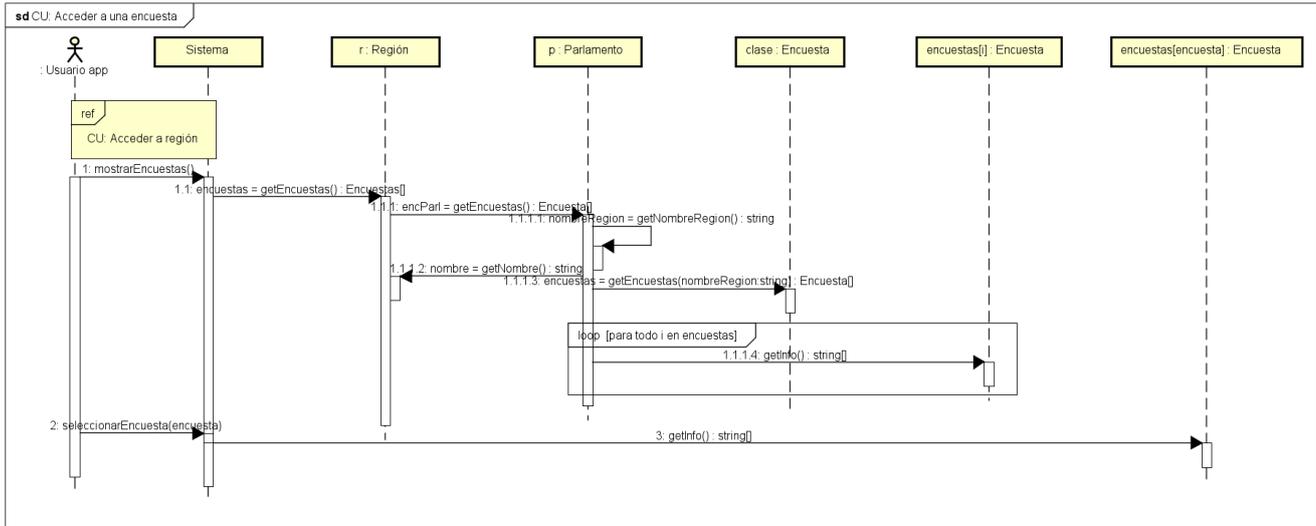


Figura 3.14: Secuencia de caso de Uso 2- Acceder a una encuesta

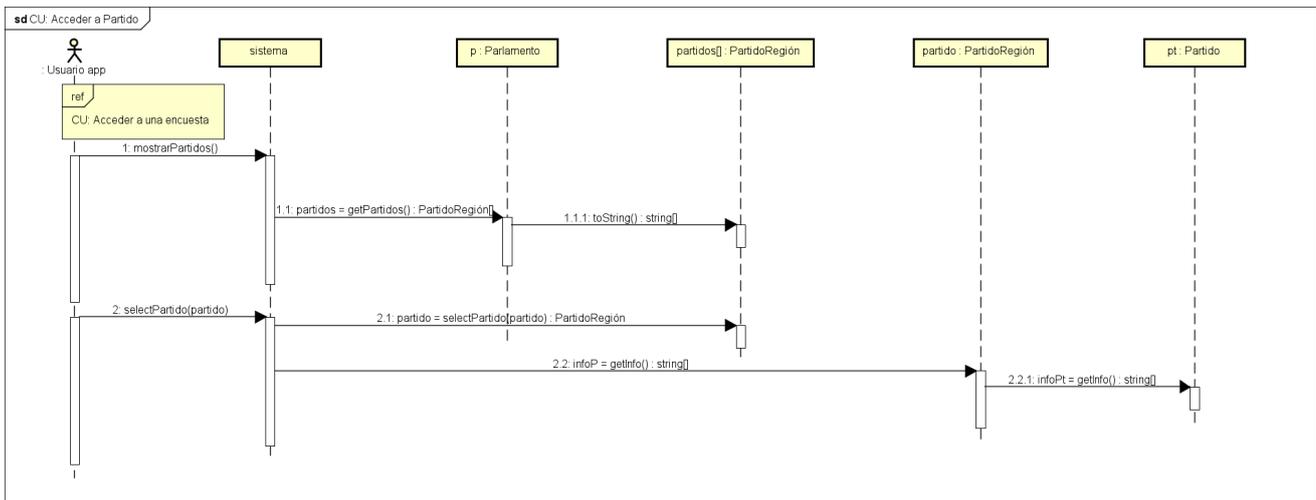


Figura 3.15: Secuencia de caso de Uso 3- Acceder a partido

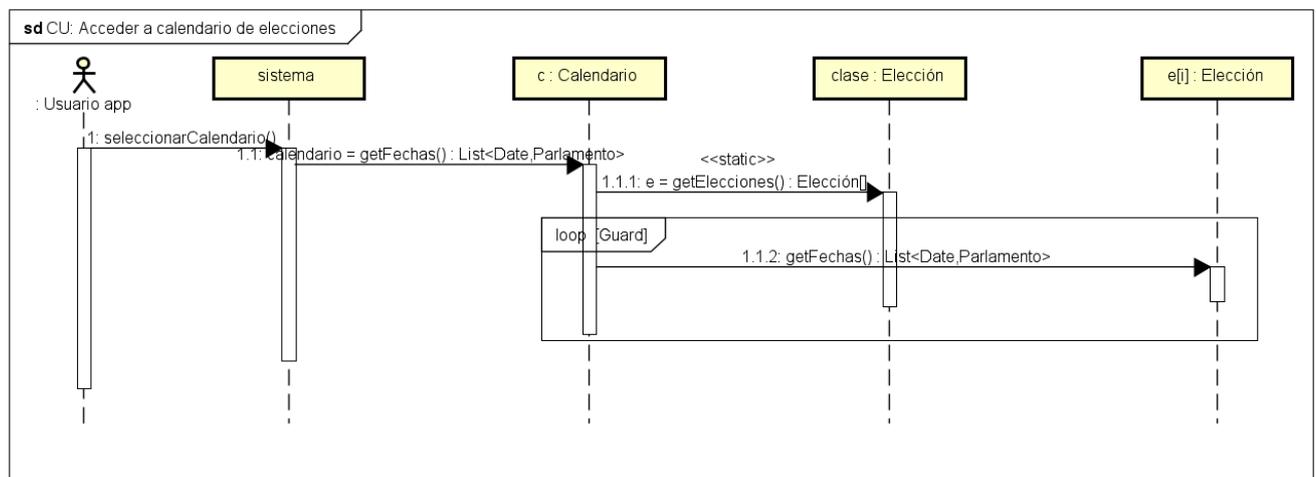


Figura 3.16: Secuencia de caso de Uso 4- Acceder a calendario de elecciones

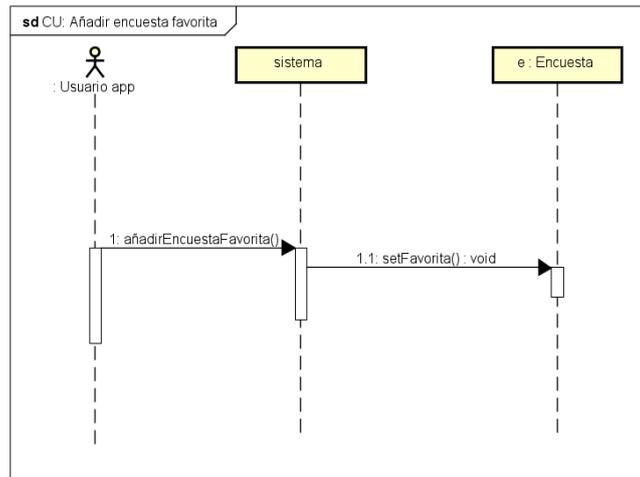


Figura 3.17: Secuencia de caso de Uso 5- Añadir encuesta favorita

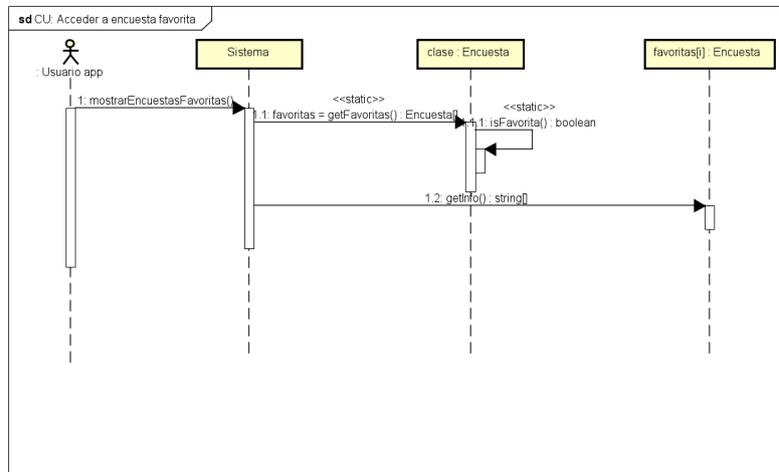


Figura 3.18: Secuencia de caso de Uso 6- Acceder a encuesta favorita

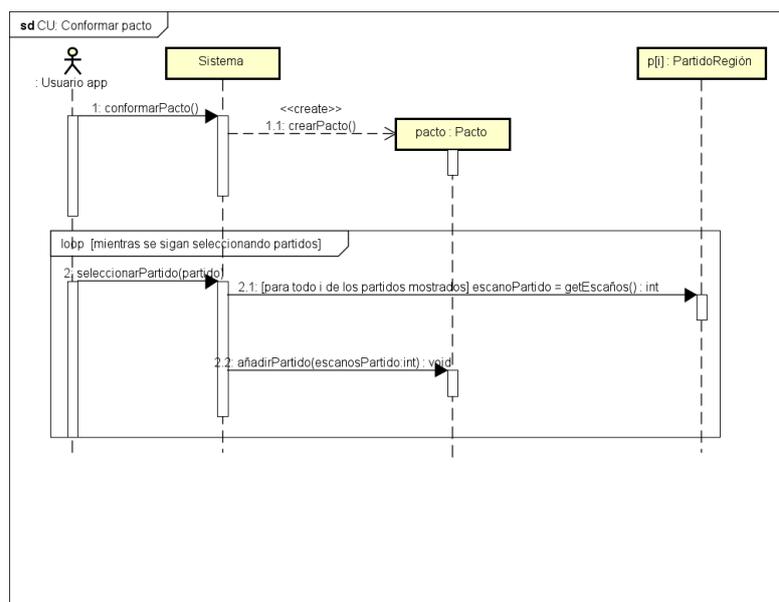


Figura 3.19: Secuencia de caso de Uso 7- Conformar pacto

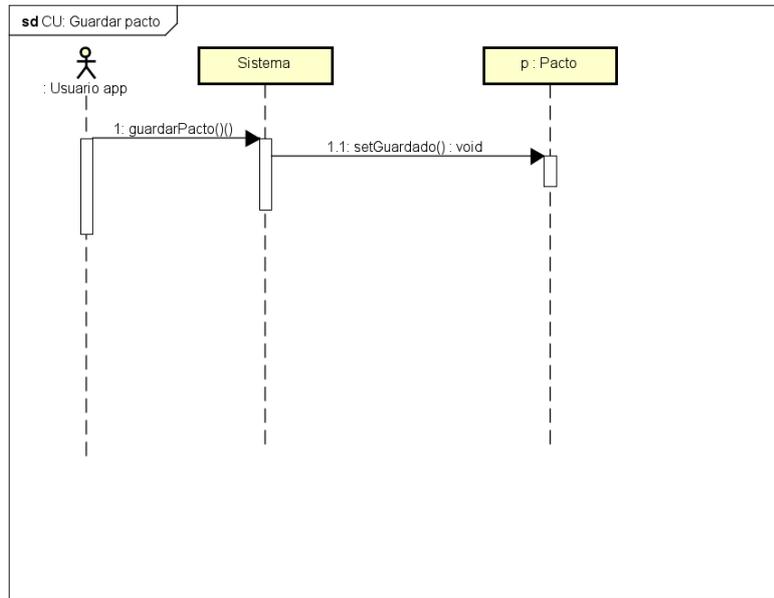


Figura 3.20: Secuencia de caso de Uso 8- Guardar pacto

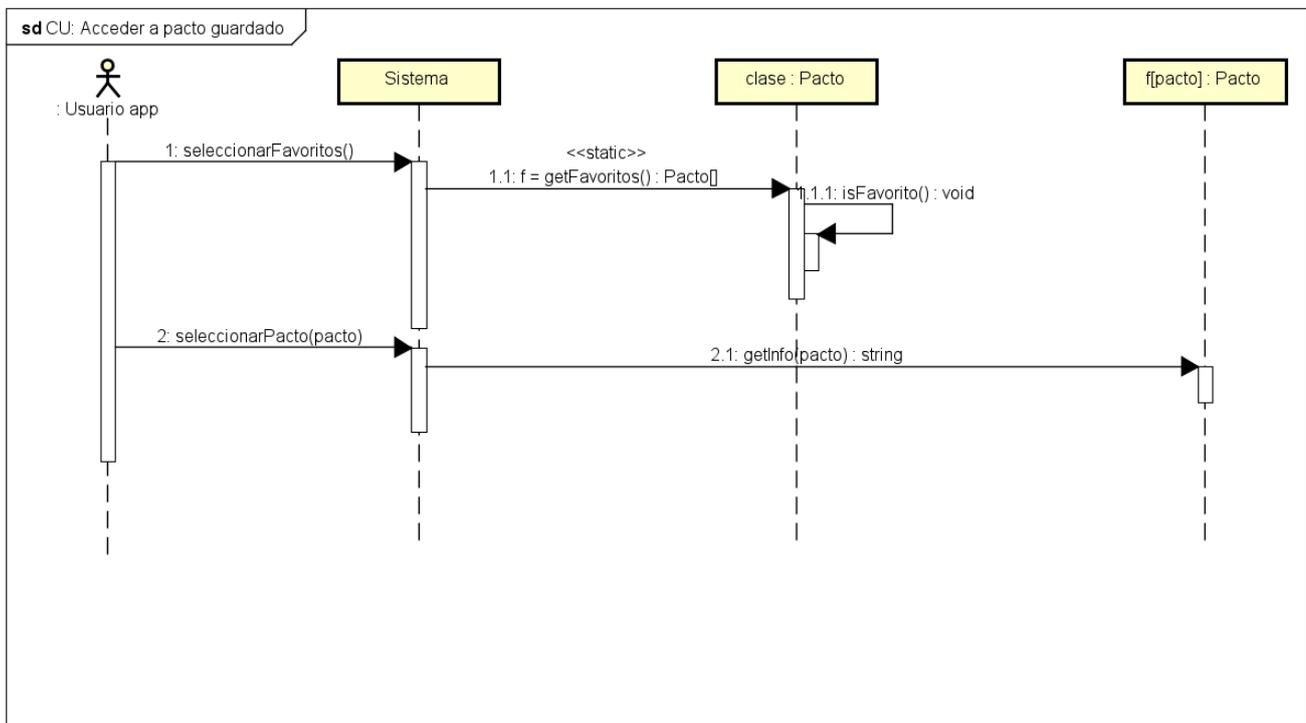


Figura 3.21: Secuencia de caso de Uso 9- Acceder a pactos guardados

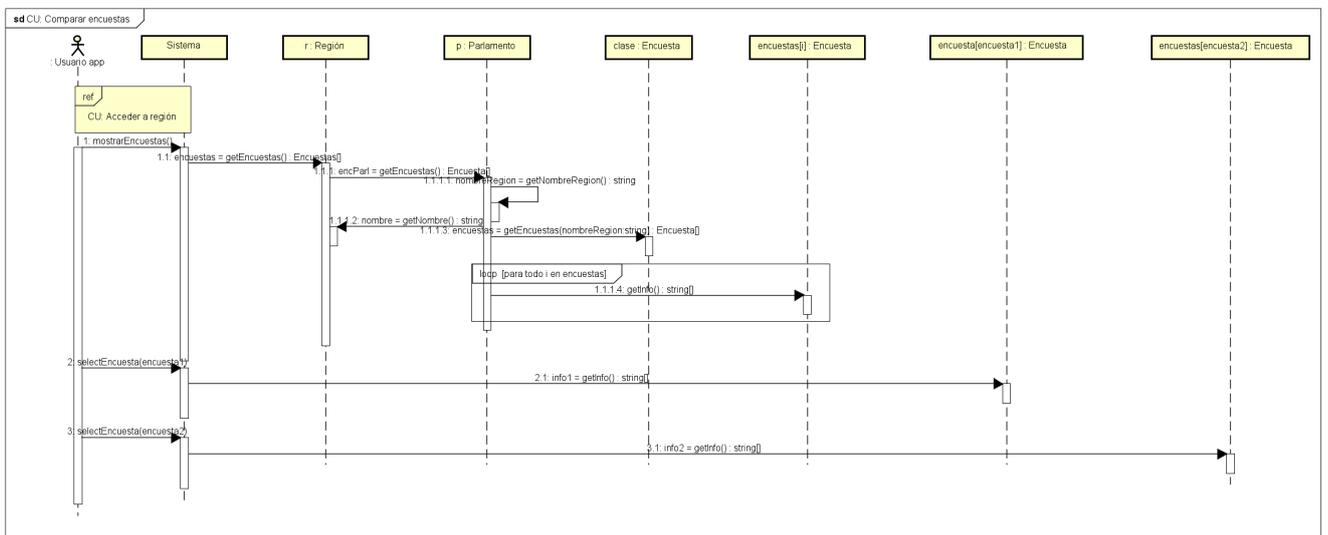


Figura 3.22: Secuencia de caso de Uso 10- Comparar encuestas

Capítulo 4

Diseño

Como paso posterior al Análisis y antes de iniciar con la implementación de la aplicación, hay que describir las decisiones de diseño de la misma. En los siguientes apartados se presentan las decisiones de diseño tomadas, así como la arquitectura utilizada y la organización interna del diseño de la aplicación.

4.1. Decisiones de diseño

En el apartado actual se presentan las decisiones que se han tomado a la hora de la utilización de diversos patrones o herramientas.

- Requisitos de la aplicación: cualquier navegador de versión actualizada y terminal Android con API 30 (Android 11 R) o superior.
- Se utilizará una arquitectura cliente-servidor.
- En la parte del cliente, la aplicación se encargará tanto de la parte gráfica como del almacenamiento de los datos del usuario.
- En la parte del servidor, la aplicación se encargará de la extracción de los datos y de su correcto envío al cliente.
- Se utilizará Python a la hora de crear el script de web scrapping.
- Se utilizará JavaScript a la hora de la definición de la interfaz de la aplicación.
- Para el servidor se utilizará el framework Django. Además, contiene un *plug-in* denominado Django REST framework que permitirá la creación de una API para que el cliente se pueda comunicar.
- Para el cliente se utilizarán los frameworks Angular e Ionic, pues proveen una manera completa de creación de aplicaciones web, por la parte de Angular y su adaptación a dispositivos móviles por parte de Ionic.
- Para el almacenamiento persistente de los datos se utilizará SQLite.
- Para la presentación de los gráficos, se utilizará la librería D3.js, que permite la creación de gráficos interactivos desde cero y provee una edición a gran nivel de detalle.

4.2. Arquitecturas y patrones

Como ya se ha mencionado, la aplicación se dividirá en cliente y en servidor, mostrándose en la figura 4.1 los componentes que tienen cada uno, así como la interacción que tienen entre ambas partes, a través de una API REST.

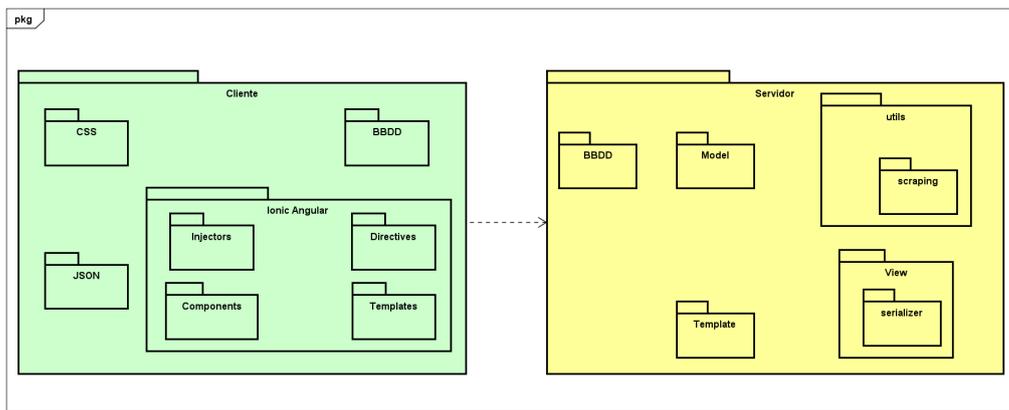


Figura 4.1: Lógica del cliente-servidor

4.2.1. Arquitectura del servidor

El servidor de la aplicación utilizará Django REST Framework, que permite la creación de APIs REST de manera sencilla, utilizando el lenguaje Python. Django utiliza un patrón parecido a MVC (Model-View-Controller) conocido como MVT (Model-View-Template). Esto se debe a que el framework Django está enfocada a las bases de datos.

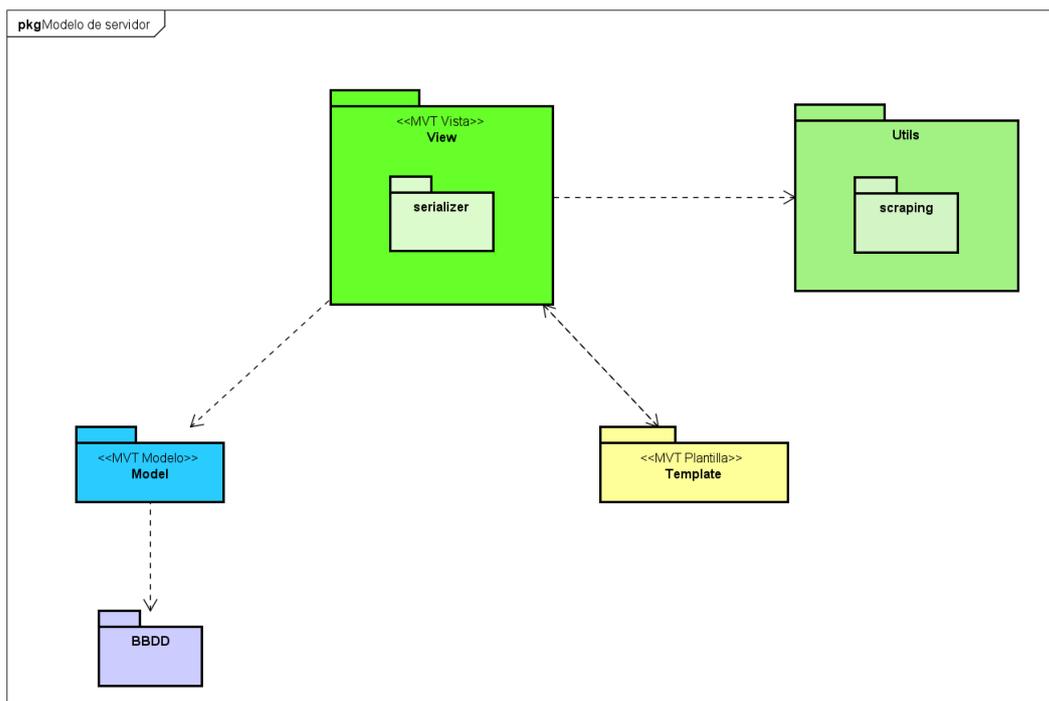


Figura 4.2: Arquitectura del servidor

En la figura 4.2 se observa que el patrón utiliza el servidor MVT, el cual posee 3 componentes:

- El **Modelo** se encarga de las operaciones CRUD (Create, Read, Update, Delete) de la base de datos. Se define una serie de clases que posteriormente actúan como tablas, y contienen sus atributos. Esta aproximación permite la creación de una base de datos relacional (ya sea en SQLite, MySQL o PostgreSQL, por ejemplo) utilizando el lenguaje Python.
- La **Vista** contiene toda la carga lógica de la aplicación. Se encarga de la comunicación con el Modelo, así como de las operaciones a realizar. También se comunica con las Plantillas (Plantillas). Contiene, en este caso, un *serializer* que convierte las consultas al modelo en un JSON.

- Las **Plantillas** son la parte visual de la aplicación. Se comunican con la Vista. En este caso, la Plantilla es muy sencilla y viene provista por *Django REST Framework*, para devolver un objeto **Response** y se pueda visualizar de manera estructurada.

Además, un componente primordial del servidor se encuentra fuera de estos 3 mencionados anteriormente, se trata del **script de web scraping**. El mismo se encuentra en el paquete de **utils** en el proyecto y se encarga de proveer obtener los datos.

4.2.2. Diseño de la API REST

A la hora de ir a un restaurante a comer, los clientes no se dirigen directamente a la cocina a pedir la comida, sino que se lo piden a un camarero. Con este símil, la cocina es el servidor y el camarero es la API.

Las APIs (Interfaz de Programación de Aplicaciones) hacen de intermediario entre el cliente y el servidor. En el caso de la aplicación 1poller, será una API REST, que sigue los principios de REST [9].

- Acceso HTTP: los accesos a la API se realizan mediante HTTP y se obtienen los códigos de estado para enriquecer el uso de la interfaz.
- Uso de URLs: los accesos a los recursos se realizan mediante el uso de URLs.
- Protocolo cliente/servidor sin estado: la API no necesitará más información por parte del cliente para ser accedida.
- Operaciones definidas: las operaciones están limitadas para los datos. En el caso de la presente aplicación, únicamente se utilizará GET.



```
Django REST framework
GET /polls/AS

HTTP 200 OK
Allow: OPTIONS, GET
Content-Type: application/json
Vary: Accept

[
  {
    "clave": "AS_electoPanel_15 Oct 2021",
    "region": "AS",
    "encuestadora": "ElectoPanel",
    "medio": "Electomania",
    "fecha_inicio": "2021-10-15",
    "fecha_fin": "2021-11-29",
    "muestra": 1093,
    "part_est": null
  },
  {
    "clave": "AS_electoPanel_15 Apr 2021",
    "region": "AS",
    "encuestadora": "ElectoPanel",
    "medio": "Electomania",
    "fecha_inicio": "2021-04-15",
    "fecha_fin": null,
    "muestra": 800,
    "part_est": null
  },
  {
    "clave": "AS_SyM Consulting_27 Dec 2020",
    "region": "AS",
    "encuestadora": "SyM Consulting",
    "medio": null,
    "fecha_inicio": "2020-12-27",
    "fecha_fin": "2020-12-31",
    "muestra": 1276,
    "part_est": 60.6
  },
  {
    "clave": "AS_electoPanel_1 Apr 2020",
    "region": "AS",
    "encuestadora": "ElectoPanel",
    "medio": "Electomania",
    "fecha_inicio": "2020-04-01",
    "fecha_fin": "2020-05-15",
    "muestra": null,
    "part_est": null
  },
  {
    "clave": "AS_SyM Consulting_4 May 2020",
    "region": "AS",
    "encuestadora": "SyM Consulting",
    "medio": null,
    "fecha_inicio": "2020-05-04",
    "fecha_fin": "2020-05-06",
    "muestra": 1194,
    "part_est": 62.2
  }
]
```

Figura 4.3: Respuesta de la API

La respuesta de la API será siempre un JSON, con la estructura mostrada en la figura 4.3.

4.2.3. Arquitectura del cliente

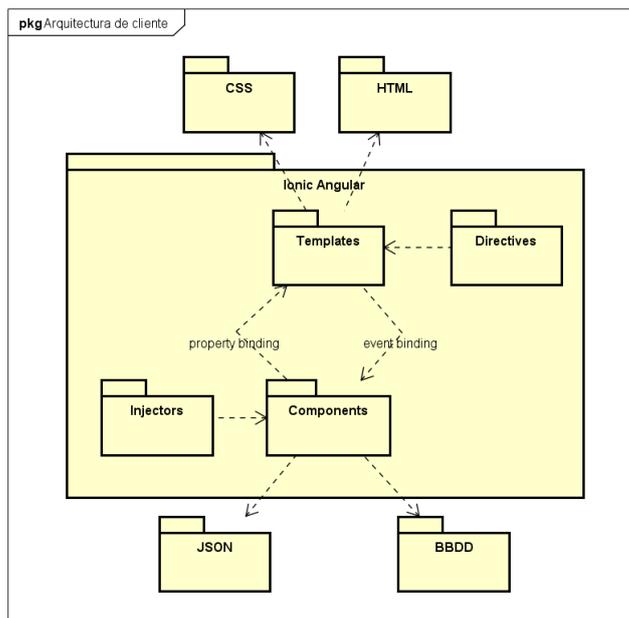


Figura 4.4: Arquitectura del cliente

En la figura 4.4 se muestra la arquitectura del Cliente. Se compone de la arquitectura de Angular, que sigue un patrón MV-VM o MVW (Modelo Vista- Vista Modelo o Model-View-Whatever) [10][11]. No es un patrón Modelo-Vista-Controlador clásico debido a los *bindings* que hay entre los Components y los Templates (algo parecido a los Modelos y Vistas). Los bindings se encargan de reaccionar ante eventos y de enviar la información a los Templates. Esta indefinición ha llevado, también, a ser llamado MVW (Modelo-Vista-Lo que sea).

Externamente se completará con los accesos a los ficheros de información JSON y a las sencillas bases de datos, así como completar los Templates con CSS y HTML.

Estructura de la aplicación

La aplicación se estructura en los paquetes Components, que contiene todas las páginas por las que se navegará y Services, que servirá de acceso a la ligera base de datos de las encuestas guardadas.

Cabe resaltar que, en caso de Ionic Angular, cada componente conforma una página entera. Además, para la eficiencia del acceso a los datos encontrados en la lógica de cada una, se declaran varias variables globales para su acceso desde cada una de las Templates HTML.

De una forma global, la aplicación se estructura como se muestra en la figura 4.5.

Clases y paquetes detallados

Una vez presentada la visión general, se explican detalladamente los contenidos de cada clase y su paquete. Son componentes sencillos, así como un servicio que provee el acceso a la base de datos ligera del programa. La descripción de cada paquete y clase se muestran desde la figura 4.6 hasta la figura 4.15.

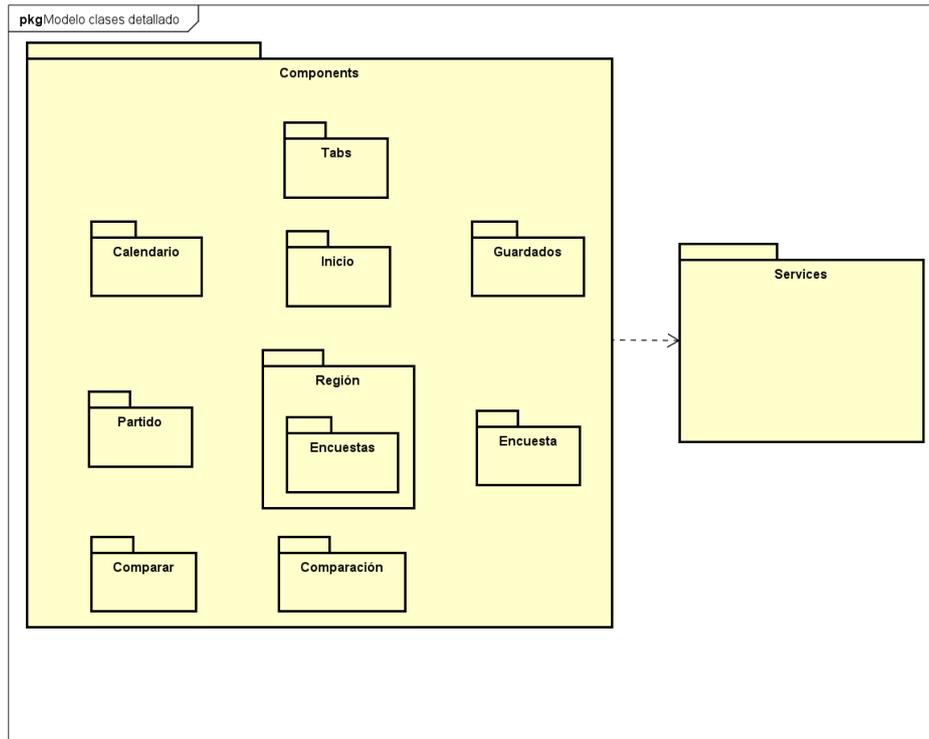


Figura 4.5: Paquetes de la aplicación

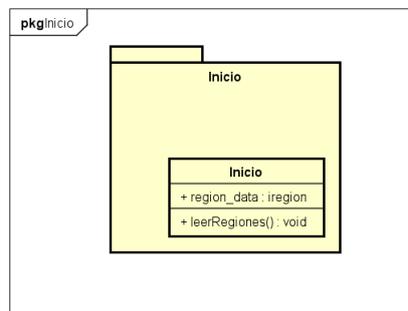


Figura 4.6: Paquete inicio

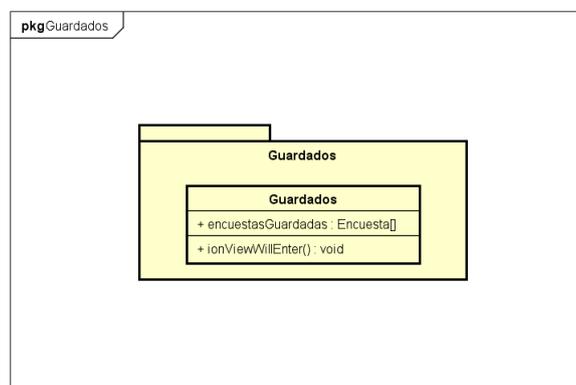


Figura 4.7: Paquete guardados

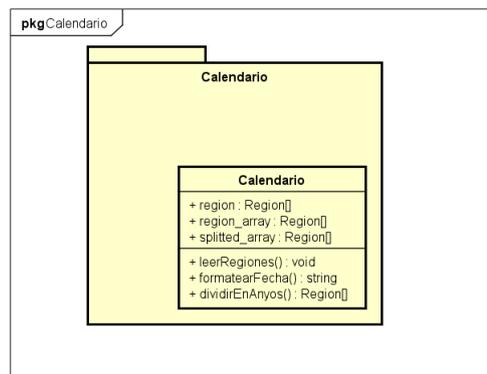


Figura 4.8: Paquete calendario

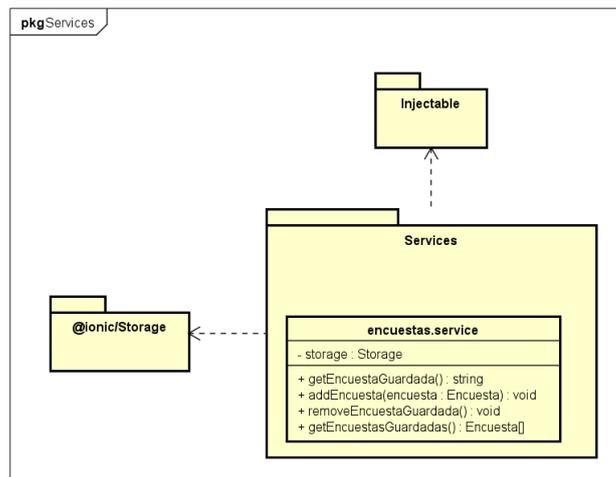


Figura 4.9: Paquete servicios

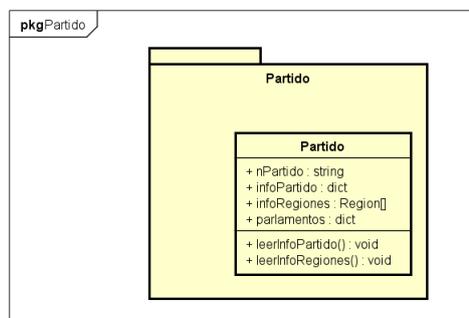


Figura 4.10: Paquete partido

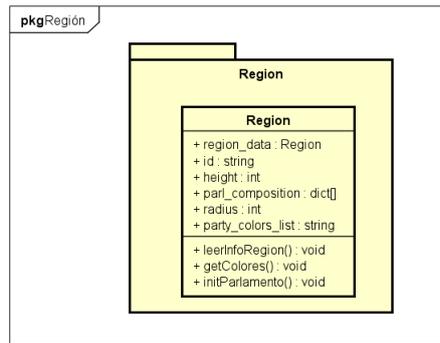


Figura 4.11: Paquete región

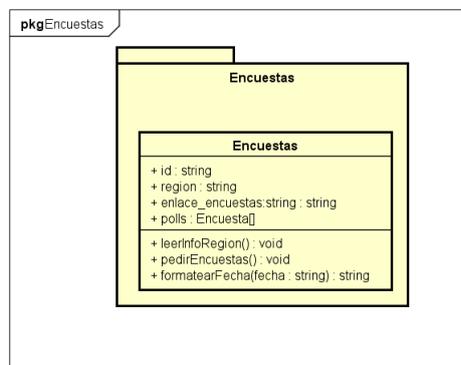


Figura 4.12: Paquete encuestas

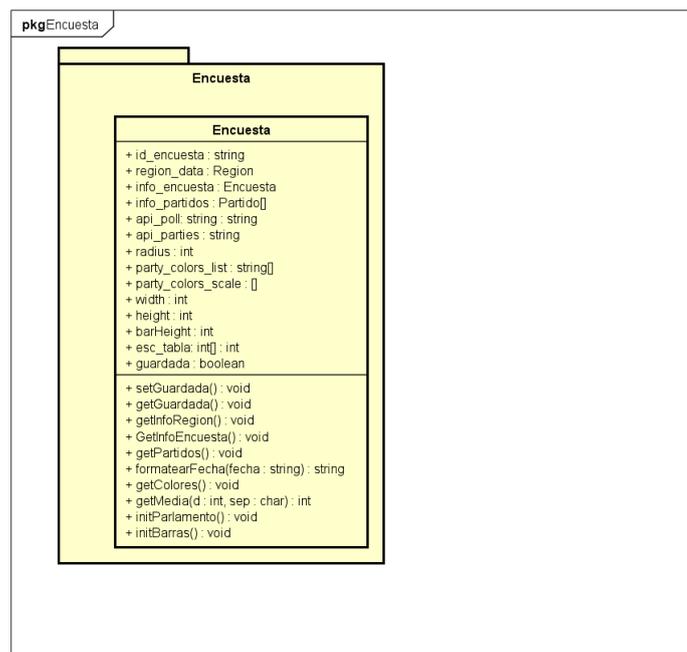


Figura 4.13: Paquete encuesta

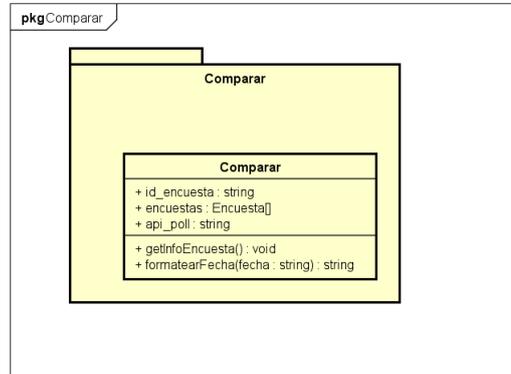


Figura 4.14: Paquete comparar

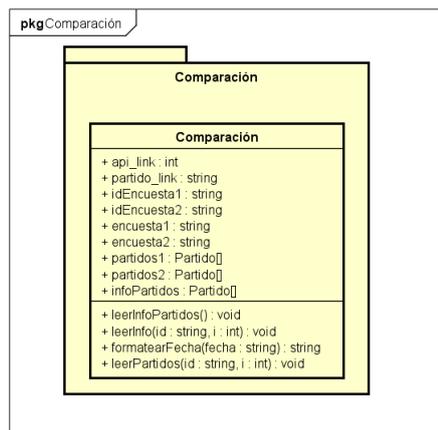


Figura 4.15: Paquete comparación

4.3. Persistencia de los datos

La arquitectura cliente-servidor se divide en:

- El cliente, el demandante de recursos, que poseerá una base de datos compleja con todos los datos que hayan de almacenarse de manera persistente para el usuario.
- El servidor, el proveedor de recursos, que almacenará los datos obtenidos por web scraping.

4.3.1. Servidor

El servidor, al encargarse de la obtención de los datos desde las fuentes de datos, ha de almacenarlos acorde a sus necesidades también. Debido a su implementación en Django y su enfoque en bases de datos, se utilizará una base de datos relacional. En algunos momentos la consulta en las fuentes puede fallar, y por lo tanto es preciso mantener la información de manera persistente en el servidor. El modelo relacional sería el siguiente:

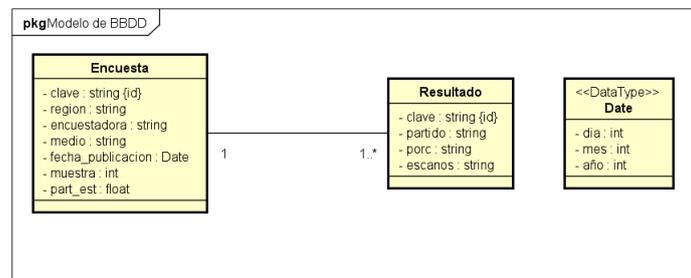


Figura 4.16: Modelo relacional de la base de datos en el servidor

En la figura 4.16 se puede observar el diagrama ER de la base de datos en el servidor.

La tabla Encuesta estará formada por:

- **clave**: la clave primaria, será una concatenación de la encuestadora, la región, el medio y la fecha. Esto es debido a que Django no trabaja con claves primarias compuestas, por lo que se requiere realizarlo así.
- **region**: la región a la que se refiere (comunidad o España).
- **encuestadora**: la encuestadora que realiza la encuesta.
- **medio**: el medio al que va dirigida la encuesta. Podría no existir.
- **fecha_publicacion**: fecha de la publicación de la encuesta.
- **muestra**: la muestra que se ha utilizado para la encuesta. Podría no existir.
- **part_est**: participación estimada en la elecciones según la encuesta. Podría no existir.

La tabla Resultado estaría compuesta por los siguientes campos:

- **encuesta** (no aparece): es la clave foránea y se refiere a la clave primaria de la encuesta.
- **clave**: es la clave primaria y es la concatenación de encuesta y el partido. Se crea debido al mismo "problema" que se tenía anteriormente.
- **partido**: partido al que se refiere el resultado.
- **porc**: el porcentaje obtenido por el partido. Es string porque puede haber una horquilla en la estimación, separada por un '-'.
- **escanos**: la cantidad de escaños que han obtenido. String porque puede ocurrir que haya una horquilla como estimación, separada por una '/'.

4.4. Prototipo de la interfaz

En el presente apartado se muestra un prototipo de la interfaz creado con la herramienta *Balsamiq mockups*.

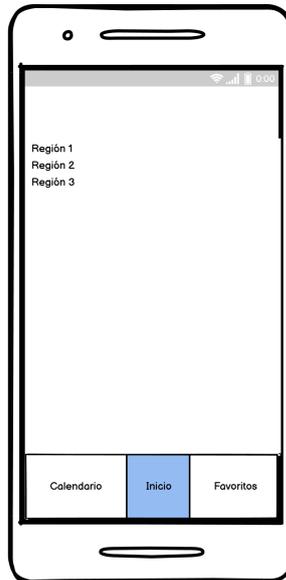


Figura 4.17: Pantalla de inicio de la app

En la figura 4.17 se muestra un prototipo muy básico de la aplicación. Se dispone de un listado de todas las regiones, así como una barra de navegación inferior, entre las que se encuentran el calendario, la pantalla de inicio actual y los favoritos.

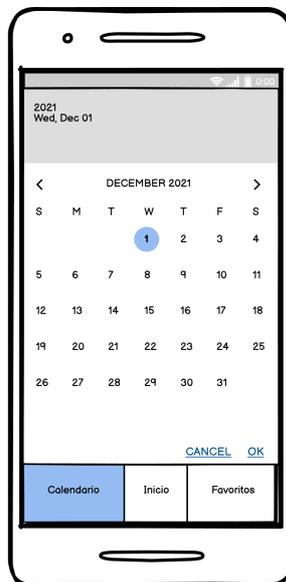


Figura 4.18: Calendario electoral

En la figura 4.18 se ha navegado hasta el Calendario electoral, que si bien se deja mostrado como si fuera un calendario, será mejor un listado en el que se muestren las siguientes elecciones.

En el prototipo visualizado en la figura 4.19 se encuentran tanto las encuestas favoritas como los pactos guardados.

En la ficha de la región seleccionada desde la pantalla de inicio, mostrada en la figura 4.20, se mostrará la composición actual del parlamento de la región seleccionada, así como información de la misma. Además, se mostrará una imagen del actual líder del ejecutivo. En esta pantalla se podrá navegar hasta el listado de encuestas.



Figura 4.19: Calendario electoral

En el listado de encuestas, mostrado en 4.21, se encontrarán todas las encuestas publicadas para la región seleccionada. Se pueden seleccionar del listado.

En la ficha de información de la encuesta seleccionada, que se puede observar en 4.22, se mostrará un gráfico de hemiciclo representando los escaños estimados para cada partido, así como un diagrama de barras que mostrará el porcentaje de los mismos, también estimado. Una tabla completará la información para tener esos mismos datos de manera numérica, así como la variación respecto a la anterior encuesta de la misma casa, si existe. De la tabla se podrá seleccionar los partidos de los que formarán un pacto, así como guardarlos. Se añadirá un icono de guardar para poder añadir a favoritos la encuesta.

Tanto la figura 4.23 como 4.24 muestran las confirmaciones de guardado de las encuestas y los pactos, respectivamente.

En una encuesta, se puede seleccionar un partido. Esta acción mostrará la ficha de información del mismo, mostrada en 4.25, con su posición ideológica (sacada de Wikipedia, nunca opinión propia) como la representación regional y nacional.

4.5. Tratamiento de los datos y GDPR

Según el Reglamento General de Protección de Datos (por sus siglas en inglés, GDPR), los datos personales deben seguir un estricto tratamiento, así como un almacenamiento seguro y el permiso del usuario para que puedan ser almacenados[12].

Sin embargo, en la aplicación 1poller, no es necesario, ya que no se almacenan datos personales más allá de los nombres de cada presidente, los cuales son públicos.

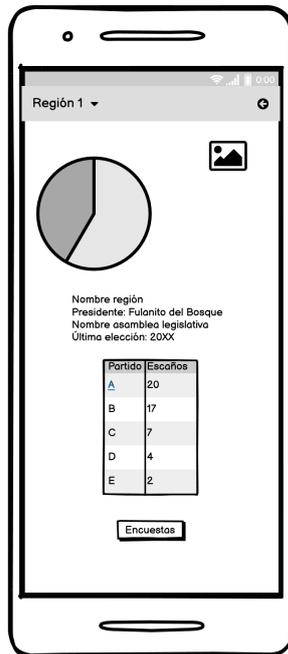


Figura 4.20: Región 1, la seleccionada

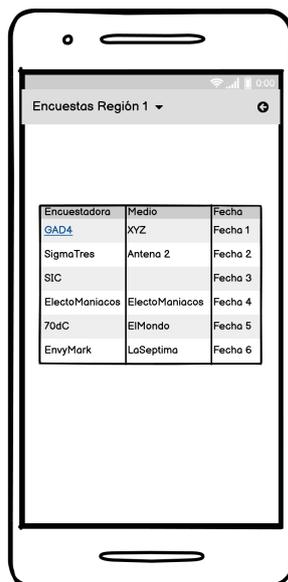


Figura 4.21: Encuestas electorales de la región seleccionada

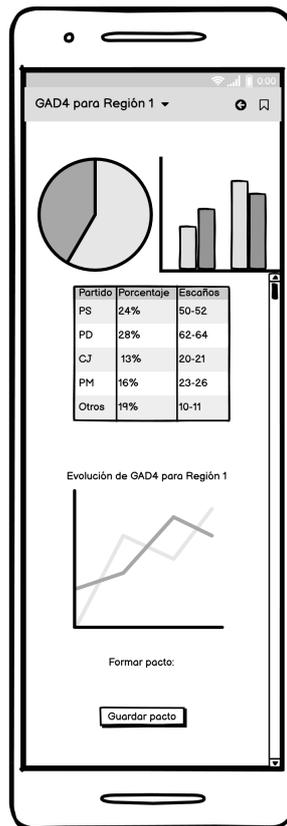


Figura 4.22: Datos de la encuesta seleccionada

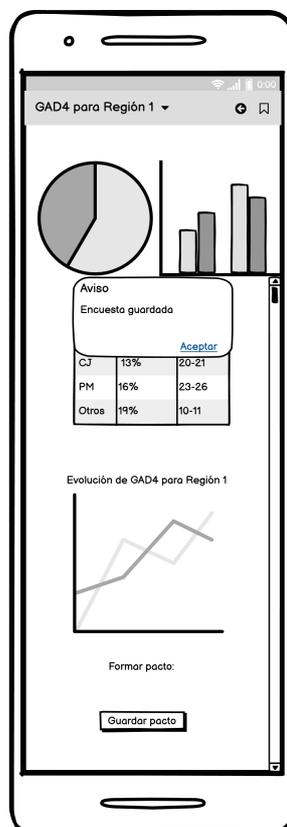


Figura 4.23: Guardar encuesta actual

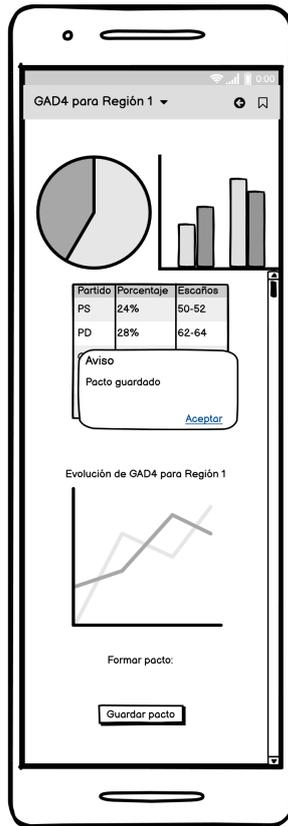


Figura 4.24: Guardar pacto conformado

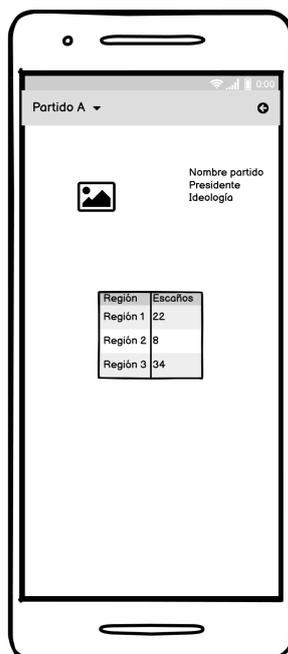


Figura 4.25: Datos del partido seleccionado

Capítulo 5

Implementación

En el presente capítulo se explica la metodología implementada en la aplicación y las herramientas utilizadas para el desarrollo de la misma.

5.1. Herramientas de desarrollo

Para el desarrollo de la aplicación se ha utilizado Visual Studio Code, un editor de código ampliamente conocido. Este entorno ha sido utilizado tanto en Windows 10 como en Ubuntu 21.10.

5.2. Implementación de la persistencia de datos

La implementación de la persistencia de datos se ha realizado, para la parte del servidor, con SQLite. SQLite es una implementación ligera de SQL-92 que permite la manipulación de bases de datos de manera sencilla sin instalar distribuciones SQL completas como PostgreSQL o MySQL[13]. Es el gestor de bases de datos por defecto en Django, framework con el que se implementa el servidor.

Por otra parte, para el cliente, la persistencia de datos se mantiene con ficheros JSON, son simples y se pueden comprender y editar fácilmente. No hay relaciones entre los componentes, por lo que no se necesita una base de datos relacional completa.

5.3. Herramientas utilizadas

Para la implementación de las funcionalidades de la aplicación se han utilizado diversas herramientas, entre frameworks y librerías.

5.3.1. Django y Django REST framework

Django es un framework que permite el desarrollo de aplicaciones web utilizando el lenguaje de programación Python, de propósito general[14]. Django está enfocado a la persistencia de datos y los formularios, pero es suficiente para poder crear un servidor y una implementación de una API REST con ayuda de un plug-in llamado Django REST framework, que permite su desarrollo de una manera sencilla y cómoda, sin conocimiento de otros lenguajes más complicados como PHP[15].

5.3.2. BeautifulSoup

Para la obtención de los datos de las encuestas, se hace *web scraping*. Existen formas manuales para ello, pero en este caso no sirve porque ha de estar automatizado. Para ello, se ha utilizado una librería de Python llamada BeautifulSoup[20], que permite la extracción de datos de documentos HTML o XML.

5.3.3. Pandas

Una vez extraídos los datos, hay que tratarlos para que las cabeceras tengan los nombres de los partidos que se conocen, pues al extraerlos no se extraen sus nombres. Para ello, se utilizan una librería muy utilizada para el tratamiento de datos en Python, conocida como Pandas[21], que lo permite realizar de una manera muy sencilla.

5.3.4. Angular e Ionic

Para la creación del cliente, se han utilizado Angular e Ionic.

Angular es un *framework* Javascript (lenguaje utilizado para creación de aplicaciones web, principalmente) que simplifica la tarea de crear aplicaciones web *single-page*[16]. Se ha utilizado como parte de la aplicación desarrollada en Ionic[17], *framework* que ayuda a la creación de aplicaciones multiplataforma utilizando un único código, en lenguaje web. Además, tanto Ionic como Angular proveen implementaciones de interfaces de usuario y elementos gráficos, lo cual simplifica al desarrollador la creación de las mismas. Ionic, además de poder utilizarse con Angular, puede utilizarse con Vue o React.

5.3.5. D3

D3.js es una librería JavaScript que permite la creación de visualizaciones y su manipulación hasta el más mínimo detalle, utilizando JavaScript, HTML y CSS. Su nombre es un acrónimo de Data-Driven Documents[18].

5.3.6. ngx-datatable

ngx-datatable es un componente Angular que permite la creación de tablas para su visualización sin dependencias externas.

5.3.7. Control de versiones

Para el control de versiones se ha utilizado la herramienta Git [22], así como la herramienta GitLab proporcionada por la escuela para poder almacenar las versiones de forma online.

Git permite el control de versiones de una manera sencilla y cómoda, permitiendo la creación de diferentes ramas de desarrollo para cada una de las funcionalidades del software.

5.4. Implementación del servidor

El servidor, en la aplicación 1poller, se encarga de obtener los datos de las encuestas de Wikipedia.

Para su implementación se han utilizado los ya mencionados Django y Django REST framework que permiten la creación de una aplicación web enfocada en las bases de datos y una API REST, respectivamente.

El servidor, al recibir una llamada para obtener las encuestas las actualiza en su base de datos, siempre comprobando que no habían sido ya añadidas (por medio de un identificador único en la base de datos) para mejorar la eficiencia del mismo y no repetir operaciones. También actualiza los partidos añadidos para que, una vez seleccionada la encuesta en el cliente, se envíe la solicitud por encuesta al servidor y responda con los datos necesarios.

Las respuestas del servidor son serializadas en formato JSON.

En la figura 5.1 se pueden observar unas encuestas obtenidas para el Principado de Asturias en formato JSON. Se puede observar que el formato contiene la clave que identifica a la encuesta, el nombre ISO de la región a la que pertenece, la fecha de publicación, la muestra (si existe) y la participación estimada (si existe).

En la figura 5.2 se observa, para la encuesta de “La Nueva España”, algunos partidos almacenados y serializados en JSON. Cada uno tiene la clave de la encuesta de la que depende, una clave individual para cada partido, el nombre del partido, el porcentaje en la encuesta (si existe) y el número de escaños estimado (si existe).

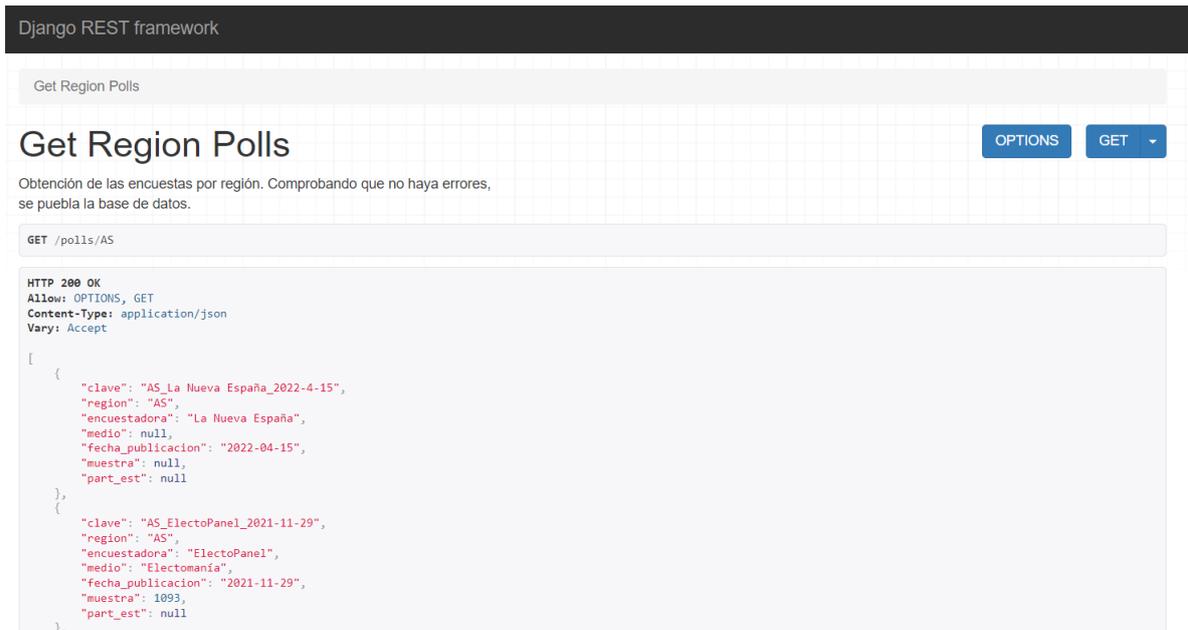


Figura 5.1: Envío de respuesta de encuestas

Además, para obtener una encuesta en concreto, se filtra la base de datos con la información de la encuesta, con el mismo formato que en la figura 5.1.

Para la obtención de todas las encuestas de una misma encuestadora, se envía el identificador de la misma y se obtiene la región y la encuestadora. Una vez obtenidos esos campos, se filtra la base de datos de tal manera que se envíe como respuesta las encuestas que no sea la misma enviada como petición pero coincida en región y en encuestadora.

5.4.1. Web scraping

La técnica de *web scraping* es una metodología para la automatización de la obtención de información en la web. En este caso, se utiliza Python con BeautifulSoup 4. El script Python accede a páginas web de las próximas elecciones de cada región y extrae la información necesaria.

El proceso que se sigue es el siguiente:

- La herramienta BeautifulSoup intenta acceder a la primera tabla de estilo “wikitable collapsible” (se da el caso que, por patrón, la primera es la de encuestas electorales). En caso de no poder acceder a ninguna o a no encajar la cabecera de la tabla con ninguna, intenta con el estilo de tabla “wikitable collapsible collapsed”. En caso de fallar de nuevo, es que no existe y se envían los datos ya almacenados en la base de datos. En caso de obtener correctamente las encuestas, hay que realizar una limpieza: obligar a que el separador de miles sea un punto para poder transformar los dígitos a *string* sin problemas, separar el porcentaje de votos estimado y los escaños por medio de un espacio (pues viene junto) por medio de expresiones regulares, etc.
- Una vez obtenidos los datos, hay que tratar cada campo para poder ser almacenado en la base de datos:
 - Eliminación de campos que no se utilizan.
 - Separación de encuestadora y medio.
 - Obtención de la fecha de publicación y darle el formato adecuado (YYYY-MM-DD).
 - Obtención de la muestra y de la participación estimada.
 - Separación y almacenamiento de porcentaje de voto y escaños estimados por partidos.
 - Creación de las claves primarias para la encuesta (concatenación de la región, el nombre de la encuestadora y la fecha de publicación) y para cada partido de una misma encuesta (concatenación de la clave primaria de la encuesta y el nombre del partido).

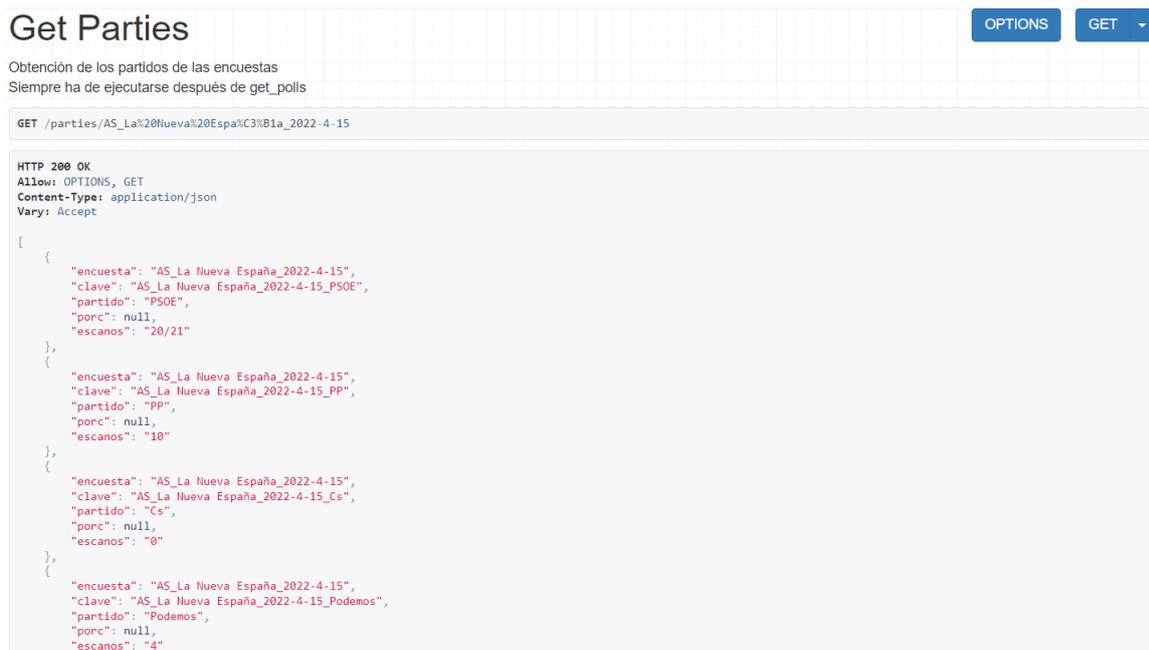


Figura 5.2: Envío de respuesta de partidos

- Almacenamiento en la base de datos de cada encuesta y sus partidos. Para reducir el tiempo de espera significativamente, se omiten aquellas encuestas ya almacenadas (comprobando que ya existan).

5.5. Implementación del cliente

Para la interacción del usuario con la aplicación se ha utilizado Ionic, que utilizará Angular para la lógica del cliente y será quien se encargue de las peticiones al servidor.

Para los gráficos se ha utilizado D3.js y para mostrar la información, ngx-datatable.

5.5.1. Diseño adaptativo

La interfaz de usuario implementada se adapta al tamaño de las pantallas en las partes centrales de la aplicación, como son las fichas de información tanto de partidos, como de encuestas y regiones. En la figura 5.3 se observa el diseño que una de estas pantallas tendrían en web y en la figura 5.4, en un móvil Samsung Galaxy S8+.

← Castilla y León

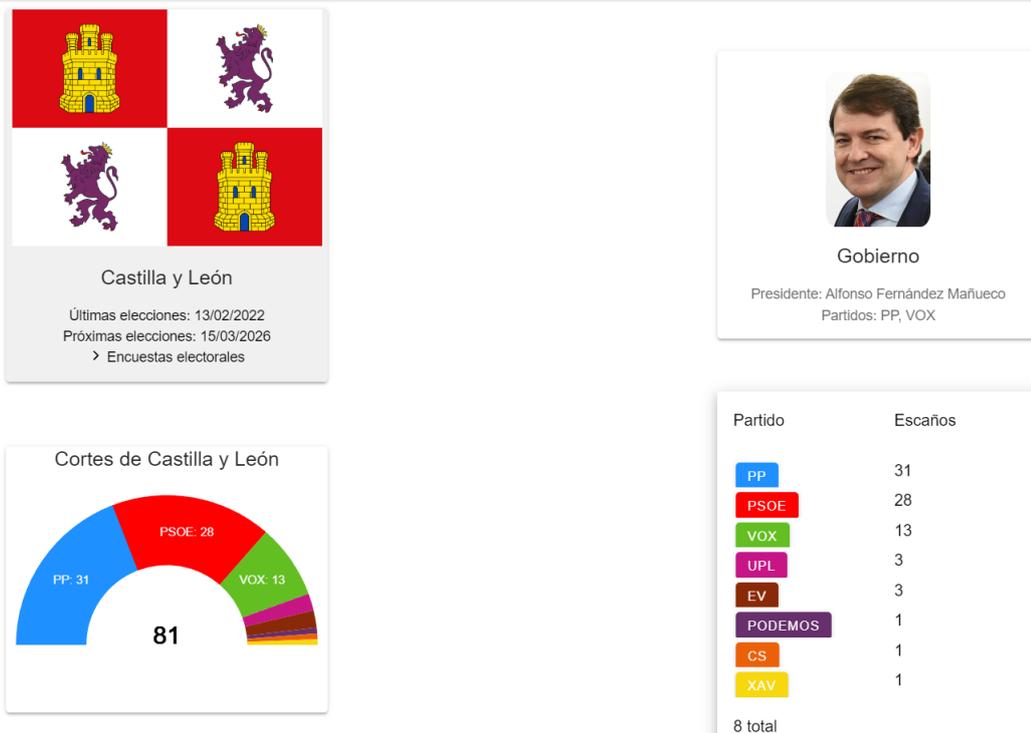


Figura 5.3: Diseño en web

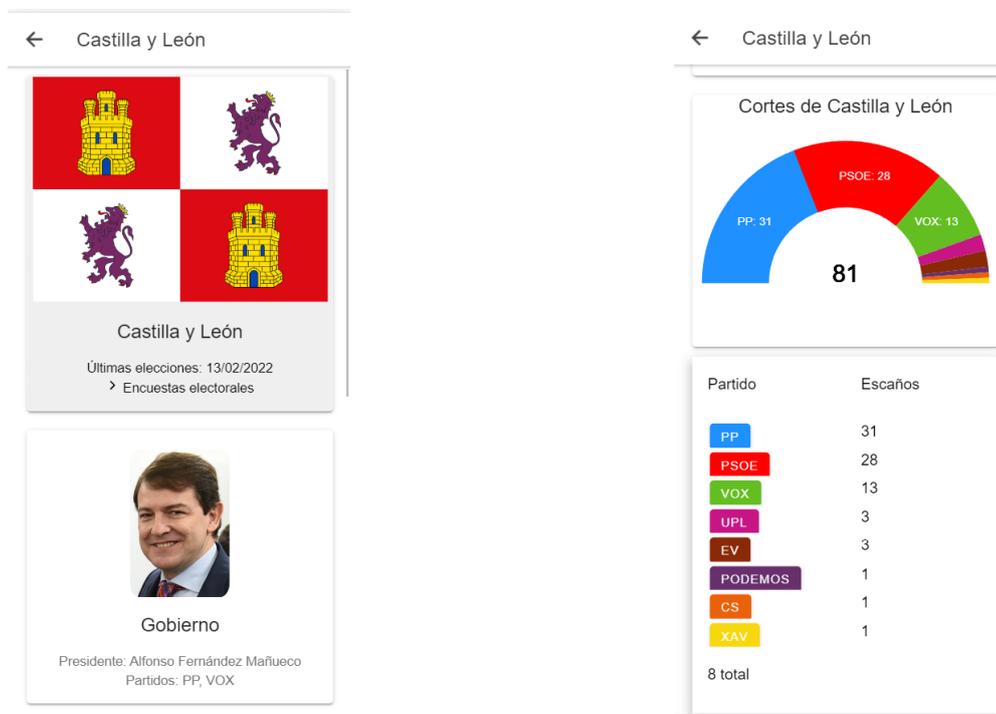


Figura 5.4: Diseño en un Samsung galaxy S8+

Capítulo 6

Pruebas

En el presente capítulo se presentan las pruebas realizadas para la comprobación del correcto funcionamiento del sistema. Si bien se han ido realizando a lo largo de la implementación, en este capítulo se muestran aquellas que tienen especial relevancia en el funcionamiento, así como su resultado final.

Se dividen en las pruebas realizadas sobre el servidor y las pruebas realizadas sobre el cliente.

6.1. Pruebas sobre el servidor

No se han encontrado problemas en el funcionamiento del servidor.

S1	El servidor obtiene las encuestas
Descripción	El servidor accede a la página de las encuestas y obtiene los datos
Resultado esperado	Las encuestas son almacenadas en el formato correcto
Resultado obtenido	Resultado esperado

S2	El servidor no puede obtener las encuestas
Descripción	El servidor detiene la extracción de datos y envía los datos almacenados
Resultado esperado	Envío de las encuestas ya almacenadas
Resultado obtenido	Resultado esperado

S3	Petición de los datos de una encuesta al servidor
Descripción	El servidor recibe una petición de datos de una encuesta específica
Resultado esperado	Envío de la información de la encuesta enviada
Resultado obtenido	Resultado esperado

S4	Petición de los resultados de una encuesta
Descripción	El servidor recibe una petición de los resultados de una encuesta
Resultado esperado	Envío de los resultados de la encuesta
Resultado obtenido	Resultado esperado

S5	Petición de las encuestas de una misma encuestadora
Descripción	El servidor recibe una petición de las encuestas de una encuestadora y región
Resultado esperado	Envío de la información de todas las encuestas de una encuestadora y región, ordenadas por fecha
Resultado obtenido	Resultado esperado

S6	El servidor acepta peticiones desde dominios diferentes
Descripción	El servidor permite a cualquier dominio acceder a sus datos
Resultado esperado	El servidor permite al cliente acceder a su información, es decir, las cabeceras CORS funcionan correctamente
Resultado obtenido	Resultado esperado

6.2. Pruebas sobre el cliente

No se han encontrado problemas en el funcionamiento del cliente.

C1	Inicio de la aplicación
Descripción	Se inicia la aplicación sin errores
Resultado esperado	El cliente es capaz de iniciar la aplicación sin problemas, obteniendo el listado de regiones, de fechas de elecciones y guardados sin problemas
Resultado obtenido	Resultado esperado

C2	Acceso a una región
Descripción	El usuario accede a una aplicación seleccionada
Resultado esperado	La aplicación muestra todos los datos especificados, crea todos los gráficos y muestra toda la información
Resultado obtenido	Resultado esperado

C3	Acceso a una encuesta
Descripción	El usuario accede a una encuesta
Resultado esperado	La aplicación obtiene de manera satisfactoria todas las encuestas de la región y, una vez el usuario ha seleccionado una, obtiene su información y la muestra de manera correcta
Resultado obtenido	Resultado esperado

C4	Comparación de dos encuestas
Descripción	El usuario quiere comparar 2 encuestas cualesquiera
Resultado esperado	La aplicación obtiene de manera satisfactoria todas las encuestas de la encuestadora en la región y, una vez seleccionada una, muestra la comparación de manera exitosa
Resultado obtenido	Resultado esperado

C5	Guardar una encuesta como favorita
Descripción	El usuario desea guardar una encuesta como favorita
Resultado esperado	La aplicación guarda, en su pequeña base de datos, la información básica de la encuesta
Resultado obtenido	Resultado esperado

C6	Acceder a una encuesta favorita
Descripción	El usuario desea acceder a una encuesta favorita
Resultado esperado	La aplicación muestra el listado de encuestas favoritas ordenadas por fecha de guardado y permite acceder a su información
Resultado obtenido	Resultado esperado

C7	Acceder al calendario de elecciones
Descripción	El usuario desea acceder al calendario de próximas elecciones La aplicación muestra el listado de regiones y sus próximas elecciones, divididas
Resultado esperado	por años y ordenadas por fecha
Resultado obtenido	Resultado esperado

C8	Acceder a un partido desde ficha de región
Descripción	El usuario desea acceder a la ficha de un partido desde una región
Resultado esperado	La aplicación muestra la página de información del partido seleccionado
Resultado obtenido	Resultado esperado

C9	Acceder a un partido desde ficha de encuesta
Descripción	El usuario desea acceder a la ficha de un partido desde una encuesta
Resultado esperado	La aplicación muestra la página de información del partido seleccionado
Resultado obtenido	Resultado esperado

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

En el presente apartado se detallan los objetivos conseguidos a lo largo del Trabajo de Fin de Grado, así como las competencias adquiridas o reforzadas.

- Se ha desarrollado una aplicación cliente-servidor, utilizando diferentes tecnologías.
- Se han reforzado los conocimientos del lenguaje de programación Python.
- Se ha aprendido a utilizar el framework Django, así como Django REST framework para la creación de una API. Son tecnologías muy utilizadas en la actualidad.
- Se han ampliado los conocimientos sobre programación en JavaScript, así como de TypeScript.
- Se ha aprendido a utilizar los framework Ionic y Angular para el desarrollo de aplicaciones multiplataforma.
- Se ha desarrollado una aplicación multiplataforma que permite el acceso a los datos y visualizaciones de diferentes encuestas electorales para cada comunidad autónoma y el país entero.
- Se han utilizado las librerías D3.js, para la creación de visualizaciones y ngx-datatable para la creación de tablas interactivas con datos electorales.
- Se ha diseñado una interfaz sencilla de utilizar y atractiva para el usuario.
- Se ha implementado un sistema que permite guardar encuestas electorales como favoritas.
- Se han implementado páginas para mostrar todos los datos informativos de las comunidades autónomas y de España.
- Se han implementado las fichas informativas con los datos de cada partido político, así como su representación en las diferentes asambleas legislativas.
- Se ha implementado un método de scraping eficiente en el servidor.
- Se ha implementado una API REST para la obtención de la información electoral con éxito.
- Se ha desarrollado un proyecto software completo sobre un tema propuesto por el alumno D. Javier Martín de Benito.

7.2. Trabajo futuro

Si bien la mayor parte de los objetivos iniciales se han cumplido, hay algunos que no han podido cumplirse por diversas razones.

También se han dado casos en los que han aparecido nuevos objetivos aunque no se hayan podido cumplir.

- Detección automática de los cambios de cabeceras en las tablas de donde se obtienen los datos de las encuestas electorales. Las cabeceras cambian según la adhesión de nuevos partidos a los diversos procesos electorales y actualmente la herramienta de scraping no lo detecta y se han de añadir manualmente.
- Obtención automática de la información de los diversos partidos. Actualmente, esta información se añade manualmente a base de ficheros de configuración e información.
- Obtención de la información de cada región en una base de datos en el servidor, lo que permitiría que el cliente no fuera modificado pero sí la información del servidor.
- Creación de un método de formación de pactos. Era un objetivo inicial, pero no era una característica obligatoria. No se ha encontrado una manera satisfactoria para su implementación, por lo que se ha tomado la decisión de no ser incluido en la versión final.
- Ampliación de la información de encuestas electorales a consistorios de ciudades grandes, como Madrid o Barcelona.
- Traducción de la aplicación a otros idiomas cooficiales de España, aparte del castellano.

Bibliografía

- [1] *Europe Elects*,
EUROPEELECTS.EU.
Recuperado a 21 de Abril de 2022,
de <https://europeelects.eu/team/>
- [2] *Flourish Studio*,
FLOURISH.STUDIO/.
Recuperado a 21 de Abril de 2022,
de <https://flourish.studio/>
- [3] *EM- ElectoManía*,
EM-ANALYTICS.COM/.
Recuperado a 21 de Abril de 2022,
de <https://www.em-analytics.com/>
- [4] *Electocracia*,
ELECTOCRACIA.COM/.
Recuperado a 21 de Abril de 2022,
de <https://electocracia.com/>
- [5] *Electograph*,
ELECTOGRAPH.COM/.
Recuperado a 21 de Abril de 2022,
de <https://www.electograph.com/>
- [6] *Rational Unified Process*,
[HTTPS://WWW.ELECTOGRAPH.COM/](https://www.electograph.com/).
Recuperado a 21 de Abril de 2022,
de https://en.wikipedia.org/wiki/Rational_Unified_Process
- [7] *Microsoft Project 2021*,
MICROSOFT.COM.
Recuperado a 27 de Abril de 2022,
de <https://www.microsoft.com/es-es/microsoft-365/project/compare-microsoft-project-management-software>
- [8] *Sánchez W. (2011) La usabilidad en Ingeniería de Software: definición y características*,
CORE.AC.UK.
Recuperado a 27 de Abril de 2022,
de <https://core.ac.uk/download/pdf/47264961.pdf>
- [9] *¿Qué es una API de REST?*,
REDHAT.COM.
Recuperado a 9 de Junio de 2022,
de <https://www.redhat.com/es/topics/api/what-is-a-rest-api>
- [10] *Bhabika Garg, Angular Architecture*,
MEDIUM.COM.
Recuperado a 9 de Junio de 2022,
de <https://medium.com/@bhavikagarg8/angular-architecture-overview-1e7cc7483a0#:~:text=Angular>

- [11] *Nacho Blanco, ¿qué patrón usa Angular?*,
OPENWEBINARS.NET.
Recuperado a 9 de Junio de 2022,
de <https://openwebinars.net/blog/que-patron-usa-angular-mvc-o-mvvm/>
- [12] *Protección de datos*,
EUROPA.EU.
Recuperado a 16 de Junio de 2022,
de https://ec.europa.eu/info/law/law-topic/data-protection_es
- [13] *SQLite*,
SQLITE.ORG.
Recuperado a 25 de Junio de 2022,
de <https://www.sqlite.org/about.html>
- [14] *Django*,
DJANGOPROJECT.COM.
Recuperado a 25 de Junio de 2022,
de <https://www.djangoproject.com/>
- [15] *Django REST framework*,
DJANGO-REST-FRAMEWORK.ORG.
Recuperado a 25 de Junio de 2022,
de <https://www.django-rest-framework.org/>
- [16] *Angular*,
ANGULAR.IO.
Recuperado a 25 de Junio de 2022,
de <https://angular.io/>
- [17] *Ionic*,
IONICFRAMEWORK.COM.
Recuperado a 25 de Junio de 2022,
de <https://ionicframework.com/>
- [18] *D3.js- Data-Driven Documents*,
D3JS.ORG.
Recuperado a 25 de Junio de 2022,
de <https://d3js.org/>
- [19] *ngx-datatable documentation*,
Recuperado a 25 de Junio de 2022,
de <https://swimlane.gitbook.io/ngx-datatable/>
- [20] *Beautiful Soup 4*,
Recuperado a 25 de Junio de 2022,
de <https://beautiful-soup-4.readthedocs.io/en/latest/>
- [21] *Pandas*,
Recuperado a 25 de Junio de 2022,
de <https://pandas.pydata.org/>
- [22] *Git*,
Recuperado a 5 de Julio de 2022,
de <https://git-scm.com/>

Apéndice A

Manual de usuario

En el presente apéndice se muestra al usuario el funcionamiento de la aplicación 1poller.

A.1. Inicio

La aplicación inicia en la sección del listado de regiones, la principal de la aplicación, mostrándose y una barra de pestañas inferior: **Calendario, Inicio y Guardados**.

En esta sección se muestra un listado de las Comunidades Autónomas y de España: se muestran sus nombres y su bandera. Los botones invitan, con una pequeña flecha, a ser pulsados, acción que mostraría la información de la región seleccionada (ver figura A.1).

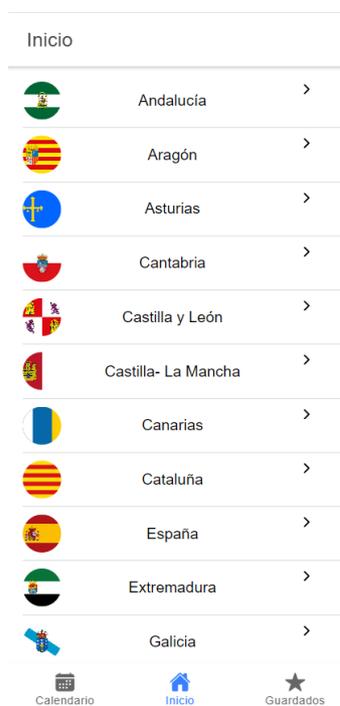


Figura A.1: Pantalla de inicio

A.1.1. Región

Una vez seleccionada una región del listado inicial, se carga la ficha de información de la región en cuestión. En la misma, se puede encontrar:

- La bandera de la región.
- El nombre oficial de la región.
- La fecha de las últimas elecciones celebradas.
- La fecha de las próximas elecciones a celebrar. Normalmente son fechas tentativas, pues son la fecha límite en la que se pueden convocar, a no ser que ya se hayan convocado para una fecha en específico.
- Un botón para acceder al listado de encuestas publicadas para la región seleccionada.
- El nombre del presidente del ejecutivo, así como de los partidos que conforman su gabinete.
- Una representación gráfica en forma de hemiciclo que representa los resultados de las últimas elecciones. Los colores de los partidos son los representativos.
- Una tabla en la que se muestran los escaños de cada uno de los partidos.

La página se puede observar en la figura A.2.



Figura A.2: Pantalla de una región

A.1.2. Listado de encuestas

En la pantalla del listado, se presentan todas las encuestas publicadas para la región seleccionada. Se pueden seleccionar y acceder a su información. Es posible que haya veces en las que tarde en cargar si el servidor tiene que extraer la información. Ver figura A.3.

Las encuestas se encuentran ordenadas por fecha estimada de publicación.

A.2. Calendario

La página Calendario muestra las siguientes elecciones de cada región, junto a su nombre y bandera, separadas por años y ordenadas de más próxima a más lejana. Es únicamente un listado, por lo que no se puede accionar ningún botón. Ver figura A.4. Cabe mencionar que son las fechas límite en las que se pueden celebrar las próximas elecciones, pues se pueden convocar anticipadamente en algunos casos.

← Encuestas de España	
Encuesta de Sigma Dos Publicada el 26-06-2022	>
Encuesta de Sociométrica Publicada el 24-06-2022	>
Encuesta de InvyMark Publicada el 24-06-2022	>
Encuesta de ElectoPanel Publicada el 24-06-2022	>
Encuesta de KeyData Publicada el 23-06-2022	>
Encuesta de GESOP Publicada el 22-06-2022	>
Encuesta de Data10 Publicada el 20-06-2022	>
Encuesta de ElectoPanel Publicada el 10-06-2022	>
Encuesta de Simple Lógica Publicada el 09-06-2022	>
Encuesta de CIS Publicada el 09-06-2022	>
Encuesta de InvyMark	>

Figura A.3: Listado de encuestas electorales

A.3. Encuestas favoritas

La página de Favoritas, accesible desde la barra de pestañas, contiene todas las encuestas que han sido marcadas como favoritas. Se muestran en un listado que permite el acceso a las mismas con un toque. Esta página se puede ver en la figura A.5.

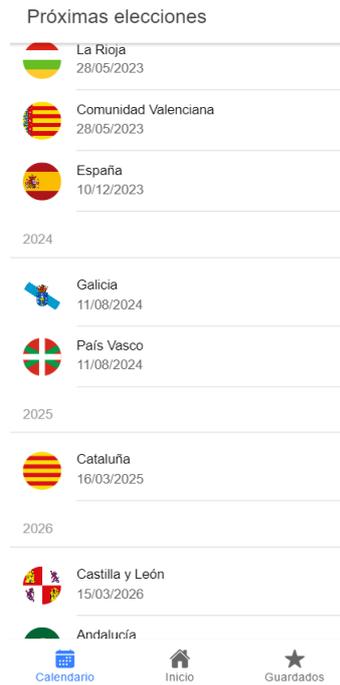


Figura A.4: Próximas elecciones



Figura A.5: Encuestas favoritas

A.4. Encuesta

Para acceder a una encuesta se puede acceder tanto desde el listado de encuestas de una región como desde el listado de encuestas favoritas y se puede encontrar, en su ficha informativa, la siguiente información:

- Un marcador para poder añadir la encuesta como favorita.
- Una proyección de escaños, tanto en forma de tabla como gráfica. Hay casos en los que las encuestas publican una horquilla de los mismos, por lo que para su representación se calcula la media entre ambos extremos.
- Una estimación del voto, tanto en forma de tabla como gráfica, con las alturas ordenadas. En el caso del diagrama de barras, no se dibujan aquellos partidos con una estimación de voto menor a un 1%, para no sobrecargar el gráfico de información.
- La información técnica de la encuesta:
 - Encuestadora.
 - Medio que la publica (si existe).
 - Muestra utilizada.
 - Participación estimada.
- Un botón para poder realizar comparaciones con encuestas de la misma empresa encuestadora y para la misma región.

La pantalla se puede observar en la figura A.6.

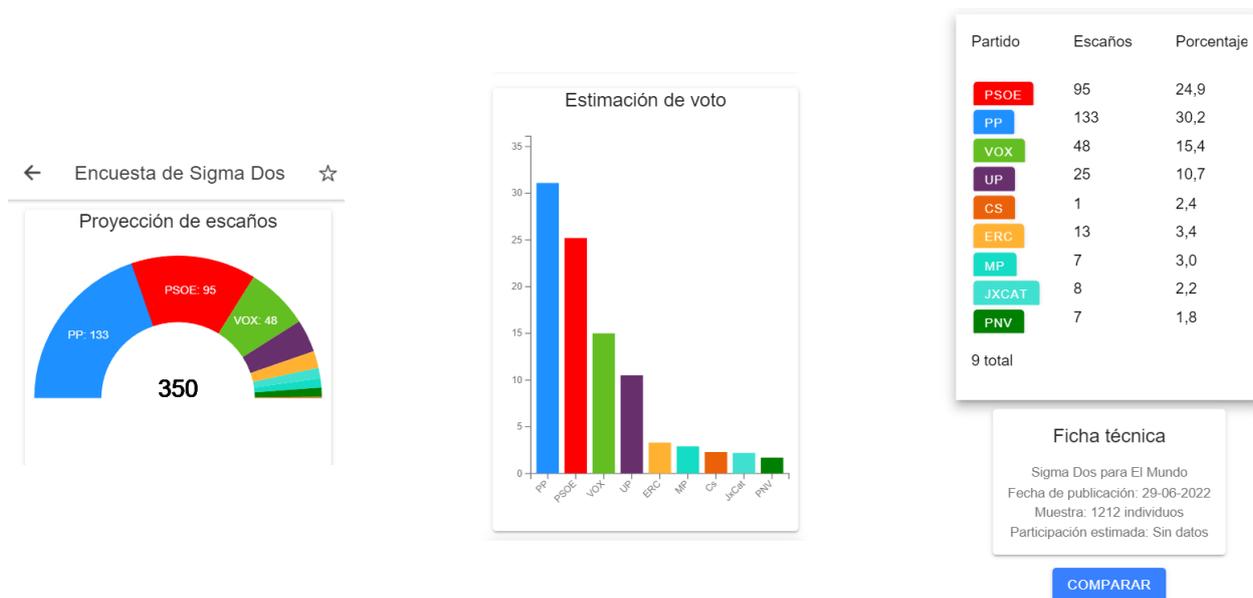


Figura A.6: Ficha de una encuesta

A.4.1. Comparación de encuestas

En la ficha de una encuesta, se puede realizar una comparación entre encuestas de la misma empresa encuestadora. Esta decisión es así porque las tendencias más interesantes se dan entre encuestas de una misma encuestadora. La comparación consta de varias fases:

- Se muestra un listado de encuestas, contenido en el listado de todas las encuestas de la región pero filtrados.
- El usuario debe elegir una encuesta del listado.

- Se muestran 2 tablas: la primera es de la encuesta inicial y la segunda, que se encuentra en el final de la página, la seleccionada para comparar. En ambas tablas se muestran los resultados en escaños y en porcentaje.

Utilizando la encuesta de la figura A.6, la comparación con más de su empresa se muestra en A.7.



Figura A.7: Comparación de 2 encuestas

A.5. Partido

A la ficha de los partidos se puede acceder pulsando los botones que se pueden encontrar en las tablas de las regiones y de las encuestas.

En la ficha, se encuentra el logotipo del partido elegido, su nombre, líder e ideología (si se tienen datos). En una tarjeta informativa diferenciada se encuentra la representación que dicho partido tiene en cada asamblea legislativa. Esta pantalla se puede ver en A.8.



Figura A.8: Partido político

Apéndice B

Manual de Instalación

B.1. Contenido del repositorio

El repositorio se encuentra en la cuenta personal del alumno en el GitLab de la Escuela¹. En el mismo, se puede encontrar:

- ionic/cliente: la carpeta donde se encuentra el código fuente del cliente de la aplicación. En esta hay diferentes directorios pero los que interesan son los siguientes:
 - src: donde se encuentra el código fuente. En esta se encuentra el directorio app, compuesto por las carpetas de cada página.
 - assets: en este directorio se encuentran los archivos JSON de información.
- server: directorio en el que se encuentra el proyecto Django. Se divide en los siguientes directorios:
 - scraping: aplicación de scraping, contiene todos los ficheros necesarios para la ejecución.
 - server: directorio en el que se encuentran las configuraciones internas de Django.
- doc: en este directorio se encuentran la presente memoria, la planificación ideal, el *mockup* realizado con Balsamiq y los diagramas hechos en Astah.

Cabe mencionar que es posible que, desde la subida del Trabajo de Fin de Grado hasta su presentación, los ficheros se hayan actualizado si algunos partidos se añaden a la escena política actual.

B.2. Instalación

En la sección presente se detalla el procedimiento de instalación y ejecución de la aplicación, así como los requisitos de instalación.

Para la descarga, basta con acceder al repositorio y descargar en zip todos los directorios.

B.2.1. Requisitos y ejecución del servidor

El servidor se ha desarrollado utilizando Python 3.10².

Los siguientes paquetes se han utilizado para el desarrollo de la aplicación y han de ser instalados utilizando `pip install *paquete*`.

¹Repositorio de la aplicación: <https://gitlab.inf.uva.es/martind/1poller>
Se encontrará disponible únicamente para el tribunal

²Página de descarga de Python: <https://www.python.org/downloads/>

- BeautifulSoup 4.11.1.
- Django 4.0
- django-cors-headers 3.13
- djangorestframework 3.13
- Pandas 1.4

Aún así, en el directorio `server`, se puede encontrar el fichero `requirements.txt`, en el que se encuentra una congelación de las versiones instaladas en el sistema operativo del alumno (Windows 10).

Para la ejecución del servidor, serviría con acceder al directorio raíz del servidor y ejecutar en consola `python manage.py runserver`. De estar todas las dependencias correctamente instaladas, el servidor se iniciaría sin problemas ni errores.

B.2.2. Requisitos y ejecución del cliente

El cliente se ha desarrollado bajo la última versión LTS de Node.js, la 16³. Una vez instalada la versión, junto a npm (node package manager):

1. Se siguen los pasos de instalación de Ionic v6⁴.
2. Ionic previsiblemente habrá instalado Angular, pero si no es así, se siguen los pasos de instalación de Angular⁵.
3. Se instala D3.js con `npm install d3`, ngx-datatable con `npm install @swimlane/ngx-datatable` e Ionic Storage con `npm install @ionic/storage-angular`. Todos los paquetes utilizados se encuentran en `package.json`, que se puede encontrar en el directorio raíz `client`.
4. Una vez instalados los paquetes necesarios, se ejecuta en terminal `ionic serve` en la raíz del cliente, el directorio `client`. Si la instalación ha ido correctamente, no debería causar ningún problema.

Cabe mencionar que los componentes y páginas se conectan al alojamiento del servidor online en Heroku⁶, aunque editando los ficheros `typescript` con los enlaces al servidor, se puede sustituir por la URL local del servidor que haya ejecutado Django.

Cabe mencionar que se puede ejecutar en Android, siguiendo los pasos descritos en la página web oficial de Ionic⁷. Es necesario instalar la última versión del SDK Android.

³Instalación de Node.js: <https://nodejs.org/es/>

⁴Instalación de Ionic CLI v6: <https://ionicframework.com/docs/intro/cli>

⁵Instalación de Angular <https://angular.io/guide/setup-local>

⁶URL del servidor desplegado <https://servidor-1poller.herokuapp.com/>

⁷Android App Development Guide <https://ionicframework.com/docs/developing/android>