



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

DESARROLLO DE UN SISTEMA WEB DE GESTIÓN PARA LA ADQUISICIÓN DE TALENTO

Alumno: Antonio Manuel Venzalá Campaña

Tutor: Prof. D. José Ignacio Gómez Espínola
Dpto: Informática

Septiembre, 2023



Universidad de Jaén

Escuela Politécnica Superior de Jaén

Departamento de Informática

Don José Ignacio Gómez Espínola, tutor del Trabajo Fin de Grado titulado:
Petro Talent - Desarrollo de un sistema web de gestión para la adquisición de
talento, que presenta Antonio Manuel Venzalá Campaña, autoriza su
presentación para defensa y evaluación en la Escuela Politécnica Superior de
Jaén.

Jaén, Septiembre de 2023

El alumno:

El tutor:

Antonio Manuel Venzalá Campaña

José Ignacio Gómez Espínola

Índice

1. INTRODUCCIÓN	9
1.1. Introducción al proyecto	9
1.2. Motivación	10
1.3. Objetivos	12
2. ESTADO DEL ARTE	13
2.1. Sistemas HRMS (Human Resources Management System)	13
2.1.1. Factorial HR	13
2.1.2. Sage HMRS	14
2.1.3. Conclusiones	14
2.2. Sistemas TMS (Talent Management System)	15
2.2.1. SAP SuccessFactors	15
2.2.2. Oracle HCM Cloud	16
2.2.3. Conclusiones	16
2.3. Sistemas ATS (Applicant Tracking System)	16
2.3.1 Bizneo HR	17
2.3.2 Taleo	18
2.3.3. Conclusiones	18
3. TECNOLOGÍAS UTILIZADAS	20
3.1. Estudio sobre tecnologías actuales	20
3.1.1. Editores de código	20
3.1.1.1. Visual Studio Code	20
3.1.1.2. Sublime Text	21
3.1.1.3. Atom	22
3.1.1.4. Conclusión y elección	22
3.1.2. Control de versiones	23
3.1.2.1. Git	23
3.1.2.2. Subversion	24
3.1.2.3. Mercurial	25
3.1.2.4. Conclusión y elección	25
3.1.3. Gestión de proyectos	27
3.1.3.1. Trello	27
3.1.3.2. Jira	28
3.1.4. Frameworks de desarrollo	29
3.1.4.1. Framework Back-End	31
3.1.4.1.1. Symfony	31
3.1.4.1.2. Laravel	33
3.1.4.1.3. Comparativa	35
3.1.4.2. Framework Front-End	36
3.1.4.2.1. React	36
3.1.4.2.2. Angular	38
3.1.4.2.3. Comparativa	40

4. INGENIERÍA DEL SOFTWARE	42
4.1. Introducción	42
4.2. Definición de requisitos	43
4.2.1. Requisitos funcionales	44
4.2.2. Requisitos no funcionales	46
4.3. Planificación	47
4.3.1. Selección del enfoque de Ingeniería del Software	47
4.3.1.1. Elección de metodología ágil a usar	48
4.3.1.2. Aplicación de la metodología seleccionada	51
4.3.1.2.1. Roles	51
4.3.1.2.2. Artefactos	52
4.3.1.2.3. Reuniones	52
4.3.1.2.4. Divisiones del proyecto en Sprints	54
4.3.1.2.5. Estimación de costes del proyecto	56
4.4. Análisis	60
4.4.1. Diagrama de clases	60
4.4.2. Diagrama de casos de uso	61
4.4.3. Diagrama de secuencia	62
4.4.3.1. Diagrama de secuencia de Inicio de Sesión	62
4.4.3.2. Diagrama de secuencia de candidatos	63
4.4.3.2. Diagrama de secuencia de ofertas	66
4.4.3.3. Diagrama de secuencia del listado de candidatos para una oferta	67
4.4.3.4. Diagrama de secuencia del Talent Pool	68
4.5. Diseño	70
4.5.1. Sketch	70
4.5.2. Wireframe	76
4.5.3. Mockup	80
4.6. Implementación	83
4.6.1. Sprint 2 - Desarrollo de la vista inicio de sesión	85
4.6.2. Sprint 3 - Desarrollo del dashboard y menubars	87
4.6.3. Sprint 4 - Desarrollo de la vista ofertas y candidatos	91
4.6.4. Sprint 5 - Desarrollo del tablero Kanban para administrar candidatos de una oferta	94
4.6.5. Sprint 6 - Desarrollo del tablero Kanban para administrar candidatos de una oferta	97
4.6.6. Sprint 7 - Generar proyecto e implementación de autenticación.	100
4.6.7. Sprint 8 - Implementación de controladores, entidades, modelos 1	103
4.6.8. Sprint 10 - Implementación de controladores, entidades, modelos 2	105
4.6.9. Sprint 11 - Implementación de controladores, entidades, modelos 3	107
4.6.10. Sprint 12 - Últimos Ajustes	108
4.7. Pruebas/Resumen aplicación	110
4.7.1. Inicio de sesión	110
4.7.2. Dashboard	112

4.7.3. Apartado candidatos	114
4.7.4. Vista ofertas - Candidatos por oferta	117
4.7.5. Apartado Talent-Pool y Candidatos de un puesto	124
5. CONCLUSIÓN	128
5.1. Conclusiones	129
5.2. Posibles desarrollos futuros	130
6. BIBLIOGRAFÍA	134

Índice de gráficos/ilustraciones

Ilustración 1. Logotipo de Factorial HR.....	13
Ilustración 2. Logotipo de Sage HMRS.....	14
Ilustración 3. Logotipo de SAP SuccessFactors.....	15
Ilustración 4. Logotipo de Oracle HCM Cloud.....	16
Ilustración 5. Logotipo Bizneo HR.....	17
Ilustración 6. Logotipo Taleo.....	18
Ilustración 7. Logotipo Visual Studio Code.....	20
Ilustración 8. Logotipo Sublime Text.....	21
Ilustración 8. Logotipo Atom.....	22
Ilustración 9. Logotipo Git.....	23
Ilustración 10. Logotipo Subversion.....	24
Ilustración 11. Logotipo Mercurial.....	25
Ilustración 12. Imagen comparativa de herramientas de control de versiones [6].....	26
Ilustración 13. Logotipo Trello.....	27
Ilustración 14. Imagen con los distintos planes de Trello.....	28
Ilustración 15. Logotipo Jira.....	28
Ilustración 16. Imagen con los distintos planes de Jira.....	29
Ilustración 17. Logotipo Symfony.....	31
Ilustración 18. Logotipo laravel.....	33
Ilustración 19. Logotipo React.....	36
Ilustración 20. Logotipo Angular.....	38
Ilustración 21. Comparativa de popularidad de frameworks Front-End.....	40
Ilustración 22. Diagrama de clases.....	60
Ilustración 23. Diagrama de casos de uso.....	61
Ilustración 24. Diagrama de secuencia de Inicio de Sesión.....	62
Ilustración 25. Diagrama de secuencia de Candidatos (Visualizar).....	63
Ilustración 26. Diagrama de secuencia de Candidatos (Añadir).....	64
Ilustración 27. Diagrama de secuencia de Candidatos (Editar).....	65
Ilustración 28. Diagrama de secuencia de Candidatos (Eliminar).....	65
Ilustración 29. Diagrama de secuencia de Ofertas.....	67
Ilustración 30. Diagrama de secuencia de candidatos de una oferta.....	68
Ilustración 31. Diagrama de secuencia de talent pool.....	69
Ilustración 32. Sketch del dashboard/vista principal.....	71
Ilustración 33. Sketch de la vista candidatos.....	72
Ilustración 34. Sketch de la vista ofertas de trabajo.....	73
Ilustración 35. Sketch de la vista etapas del proceso de selección.....	74
Ilustración 36. Sketch de la vista talent pool.....	75
Ilustración 37. Wireframe pantalla de inicio.....	76
Ilustración 38. Wireframe del panel/vista principal.....	77
Ilustración 39. Wireframe de la vista de listado de candidatos.....	77

Ilustración 40. Wireframe de la vista de ofertas.....	78
Ilustración 41. Wireframe de la vista para modificar estado de candidatos en una oferta de empleo.....	78
Ilustración 42. Wireframe de la vista para gestionar el talent pool.....	79
Ilustración 43. Mock Up del panel/vista principal.....	80
Ilustración 44 . Mock Up de la vista de listado de candidatos.....	81
Ilustración 45 . Mock Up de la vista con las ofertas de empleo.....	81
Ilustración 46 . Mock Up de la vista para modificar estado de candidatos en una oferta de empleo.....	82
Ilustración 47 . Mock Up de la vista para gestionar el talent pool.....	82
Ilustración 48 . Cronograma inicial del desarrollo Front-End.....	84
Ilustración 49 . Cronograma inicial del desarrollo Back-End.....	85
Ilustración 50 . Tablero a mitad Sprint 2.....	86
Ilustración 51 . Tablero a final Sprint 2.....	86
Ilustración 52 . Imagen formulario Login.....	87
Ilustración 53 . Tablero a mitad Sprint 3.....	88
Ilustración 54 . Tablero a final Sprint 3.....	89
Ilustración 55 . Imagen Dashboard.....	90
Ilustración 56 . Tablero a mitad Sprint 4.....	91
Ilustración 57 . Tablero a final Sprint 4.....	92
Ilustración 58 . Vistas candidatos y ofertas.....	93
Ilustración 59 . Tablero a mitad Sprint 5.....	94
Ilustración 60 . Tablero a final Sprint 5.....	95
Ilustración 61 . Vista Offers Candidates.....	96
Ilustración 62 . Tablero a mitad Sprint 6.....	97
Ilustración 63 . Tablero a final Sprint 6.....	98
Ilustración 64 . Vista Talent Pool.....	99
Ilustración 65 . Tablero a mitad Sprint 7.....	101
Ilustración 66 . Tablero a final Sprint 7.....	102
Ilustración 67 . Tablero a mitad Sprint 8.....	103
Ilustración 68 . Tablero a final Sprint 8.....	104
Ilustración 69 . Tablero a mitad Sprint 10.....	105
Ilustración 70 . Tablero a final Sprint 10.....	106
Ilustración 71 . Tablero a mitad Sprint 11.....	107
Ilustración 72 . Tablero a final Sprint 11.....	107
Ilustración 73 . Tablero a mitad Sprint 12.....	108
Ilustración 74 . Tablero a final Sprint 12.....	109
Ilustración 75 . Vista inicio de sesión correcto.....	111
Ilustración 76 . Vista inicio de sesión incorrecto.....	111
Ilustración 77 . Vista dashboard.....	112
Ilustración 78 . Ejemplo menú calendar (Agenda).....	113
Ilustración 79 . Vista candidatos.....	114
Ilustración 80 . Modal Añadir Candidatos.....	115

Ilustración 81 . Modal Editar Candidatos.....	115
Ilustración 82 . Eliminar candidatos.....	116
Ilustración 83 . Descargar curriculum vitae.....	116
Ilustración 84 . Vista Ofertas de empleo.....	117
Ilustración 85 . Formularios Ofertas de empleo.....	118
Ilustración 86 . Formularios Ofertas de empleo.....	118
Ilustración 87 . Vista Candidatos por oferta de empleo.....	119
Ilustración 88 . Modificar etapa proceso selección.....	120
Ilustración 89 . Confirmación modificación etapa proceso selección.....	120
Ilustración 90 . Mail enviado a correo asociado.....	121
Ilustración 91. Formulario entrevista.....	121
Ilustración 92. Correo notificando entrevista.....	122
Ilustración 93 . Formulario añadir candidatos a ofertas de empleo.....	122
Ilustración 94 . Vista candidatos de una oferta al finalizar añadir.....	123
Ilustración 95 . Botón eliminar candidato.....	123
Ilustración 96 . Candidato eliminado de la oferta.....	123
Ilustración 97 . Listado de puestos del talent pool.....	124
Ilustración 98 . Añadir puesto al talent pool.....	125
Ilustración 99 . Editar puesto del talent pool.....	125
Ilustración 100. Eliminar puesto del talent pool.....	126
Ilustración 101 . Vista candidatos por puesto del talent pool.....	127
Ilustración 102 . Añadir candidatos a puesto del talent pool.....	127
Ilustración 103 . Eliminar candidato de puesto del talent pool.....	128
Ilustración 104 . Confirmación eliminación candidato puesto del talent pool....	128

Índice de tablas

Tabla 1. Planificación Sprint	56
-------------------------------------	----

1.INTRODUCCIÓN

1.1. Introducción al proyecto

En este trabajo de Fin de Grado se va a realizar el desarrollo de una aplicación web que ayude al personal encargado del proceso de reclutamiento y selección de talento humano de una empresa. En este caso, esta aplicación va dirigida a la empresa Petroprix.

Los objetivos principales de esta aplicación son:

- Generar ofertas de trabajo según las necesidades de la empresa.
- Gestionar a los candidatos entre las distintas etapas del proceso de reclutamiento y selección.
- Almacenar la información de los candidatos potenciales en lo que se denomina un “Talent Pool”.

Esta aplicación está dirigida única y exclusivamente a los reclutadores de una empresa, ya que serán los encargados de realizar la gestión de este proceso.

Los reclutadores podrán acceder a la aplicación mediante un formulario de inicio de sesión. Las credenciales de acceso serán generadas por el administrador, por lo que solo el personal autorizado podrá acceder a la aplicación.

Los reclutadores podrán navegar por las distintas vistas de la aplicación, en donde podrán generar o consultar ofertas de trabajo. Cada una de las ofertas dejará visualizar los distintos candidatos que han solicitado un puesto para la oferta de trabajo.

Los reclutadores tendrán acceso a un apartado denominado Talent Pool, en donde aparecerán aquellos candidatos potenciales que cuentan con la

formación y competencias que necesitan o necesitarán en un futuro. De esta manera, los reclutadores podrán tener un seguimiento más cercano de estos candidatos y tenerlos más controlados para poder realizarles una oferta de trabajo en el momento idóneo para ambas partes.

1.2. Motivación

La adquisición de talento se ha vuelto cada vez más competitiva en el mercado laboral actual. Las empresas necesitan encontrar maneras cada vez más eficientes de atraer candidatos altamente cualificados, analizarlos y tomar decisiones sobre qué candidatos son idóneos para sus necesidades actuales.

La adquisición de talento y el proceso de reclutamiento ha tenido la siguiente evolución con el paso de los años[1]:

- **Antecedentes de la evolución (Reclutamiento 1.0):**

En la década de los 90, debido a las limitaciones que existían, el proceso de adquisición de talento se resumía en entregar el Curriculum en mano o por carta, las ofertas se realizaban en periódicos y todo se almacenaba en carpetas con toda la documentación a papel.

- **Evolución a la era de la información (Reclutamiento 2.0):**

Con la revolución de internet y el surgimiento de las redes sociales y portales de reclutamiento, los reclutadores obtienen una gran cantidad de datos e información que antes era imposible de obtener. De esta manera, realizando un análisis minucioso y usando las herramientas correctas, se podía confirmar si un candidato posee el talento buscado para ese puesto. Además se redujeron los costes y se aumentó el índice de certeza.

- **Escalado cualitativo en la adquisición de talento (Reclutamiento 3.0):**

La aparición de redes sociales profesionales, ya sea LinkedIn o Behance, provocan que el acceso a información y la eficacia ante el reclutamiento mejore sustancialmente. Esta evolución provoca que cada vez más empresas quieran tener su software especializado para RRHH, y así poder llevar una mejor gestión en la captación de talento.

- **La revolución para captar talento (Reclutamiento 4.0):**

En la actualidad estamos situados en esta etapa, en donde tanto candidatos como empresas se centran mucho en lo que denominamos como marca personal. Esta marca personal hace que los reclutadores puedan ver la evolución en tiempo real de los candidatos, haciendo que se pueda valorar mejor sus valías y aptitudes de cara al proceso de selección.

Esta evolución nos lleva a la conclusión de que es necesario que todas las empresas tengan un sistema eficiente para así poder obtener a los mejores candidatos y los que mejor se adapten a las necesidades de la empresa.

Tras consultar con ciertos empleados de RRHH de la empresa en la que realice mis prácticas curriculares, detecté que existían ciertas necesidades en este ámbito. Por ello, decidí enfocar mi TFG en una aplicación web que solvete todas sus necesidades y mejore el proceso tan importante como es el de adquisición de talento.

Con el desarrollo de este proyecto los reclutadores obtendrán las siguientes ventajas:

- Mejora en la eficiencia y ahorro de tiempo.
- Centralización.

- Mejora en el análisis de datos y toma de decisiones.
- Adaptabilidad y escalabilidad.

1.3. Objetivos

Los objetivos principales para la realización de este Trabajo Fin de Grado son los siguientes:

- Análisis del estado del arte.
- Investigación sobre las tecnologías a usar.
- Ingeniería del Software.
- Desarrollo de una primera versión de la aplicación web.

2. ESTADO DEL ARTE

A continuación vamos a realizar un estudio sobre las principales soluciones que existen en el mercado actualmente. Además, mediante este estudio recogeremos algunos posibles requisitos que tendrá nuestra aplicación para así poder generar la solución más completa posible.

A la hora de investigar sobre el estado del arte actual, podemos encontrar 3 tipos de soluciones:

2.1. Sistemas HRMS (Human Resources Management System)

[2] Dentro de este tipo de soluciones podemos encontrar aplicaciones que están diseñadas para facilitar y automatizar las tareas relacionadas con la administración del personal, es decir, gestión de nóminas, reclutamiento, evaluaciones de desempeño, gestión de los empleados, etc..

Las soluciones que actualmente están mejor valoradas de este tipo son:

2.1.1. Factorial HR



Ilustración 1. Logotipo de Factorial HR

[3] Factorial es un software de RRHH desarrollado para digitalizar y facilitar todos los procesos de gestión humana que tiene una empresa.

Las funcionalidades más importantes que nos aporta esta aplicación son:

- Control de horario y turnos
- Gestión de vacaciones
- ATS y Evaluación del desempeño
- Gestión documental y firma electrónica
- Portal del empleado
- Contratación en inducción de personal

2.1.2. Sage HMRS



Ilustración 2. Logotipo de Sage HMRS

[4] Sage HMRS es una solución de recursos humanos integral que aumenta la productividad al instante con una serie de ventajas.

- Incorporar nuevos empleados
- Automatización de los procesos de RRHH
- Obtención de información útil mediante el uso de filtros
- Gestión de vacaciones
- Control horario

2.1.3. Conclusiones

Tras realizar un estudio sobre este tipo de aplicaciones llegamos a la conclusión de que nuestra solución no entra dentro de este esquema ya que estas están más generalizadas en el ámbito de gestión de empleados.

Mi idea es generar una aplicación que solo gestione el reclutamiento y adquisición de talento ya que actualmente en la empresa Petroprix existe una aplicación HMRS para la gestión de empleados.

Sin embargo, a la hora de realizar el documento de requerimientos de la aplicación se tendrán en cuenta estas aplicaciones pero solo en el marco del reclutamiento de personal, obviaré las demás funciones de la aplicación.

2.2. Sistemas TMS (Talent Management System)

[5] Este tipo de software engloba aquellas aplicaciones que abordan los 4 pilares de la gestión del talento (Adquisición, desarrollo, retención y gestión del rendimiento del talento).

Las soluciones TMS ofrecen todas las herramientas necesarias para realizar una gestión integral del talento en una organización. Las empresas mediante estas herramientas pueden maximizar el valor de sus empleados y promover un entorno de trabajo productivo y satisfactorio gracias a sus prácticas de reconocimiento y recompensa y crecimiento dentro de la organización.

2.2.1. SAP SuccessFactors



Ilustración 3. Logotipo de SAP SuccessFactors

[6] Solución para la gestión de talento basada en la nube que brindan a sus profesionales o gerentes de RRHH todas las herramientas para realizar

una selección de candidatos idónea y que estos empleados tengan una evolución y desarrollo dentro de la organización.

2.2.2. Oracle HCM Cloud



Ilustración 4. Logotipo de Oracle HCM Cloud

[7] Solución completa de gestión de talento desarrollada por Oracle que ayuda a tomar el control en cada etapa del ciclo de vida de la adquisición de talento para atraer los mejores candidatos, aumentar la productividad y mejorar decisiones. Es una aplicación que estudia, recluta, integra, gestiona el rendimiento/compensaciones, respalda el aprendizaje continuo y desarrolla las habilidades del talento, todo en uno.

2.2.3. Conclusiones

Lo mismo que ocurría con los sistemas HMRS, los sistemas TMS no son el tipo de solución que estamos buscando para la resolución del problema.

Esto se debe a que aunque este sistema contiene el enfoque y la funcionalidad que buscamos, este ocupa un espectro más amplio de procesos del ciclo de vida de los empleados. Actualmente, la idea es una aplicación que solo gestione el proceso de reclutamiento y contratación.

2.3. Sistemas ATS (Applicant Tracking System)

[8] Los sistemas ATS son aplicaciones que ayudan a los responsables de RRHH en única y exclusivamente la gestión de información a la hora de realizar el reclutamiento de personal.

Su principal característica es la capacidad para filtrar información importante sobre los candidatos para así reducir automáticamente el número de candidatos potenciales a cierto puesto de trabajo.

2.3.1 Bizneo HR



Ilustración 5. Logotipo Bizneo HR

[9] Bizneo HR es una aplicación con varios módulos disponibles para la autogestión de procesos enfocados a las tareas de los responsables de RRHH. Dentro de esos módulos, el único que nos interesa es el de Reclutamiento.

El software de reclutamiento que nos aporta Bizneo tiene como principales ventajas la publicación de ofertas de empleo en todos los portales conocidos, y la recepción de las ofertas automáticamente (Tanto portales premium como gratuitos). Otra característica interesante es la denominada “Killer questions” en donde se da la posibilidad de que los candidatos respondan a una serie de preguntas de exclusión para así descartar aspirantes por sus respuestas.

2.3.2 Taleo



Ilustración 6. Logotipo Taleo

[10] Taleo Enterprise Edition es una plataforma de gestión en donde las organizaciones podrán realizar y automatizar todo el proceso de reclutamiento.

Las características más importantes de esta plataforma son la contratación configurable, en donde la solución ofertada puede ser modificada a las necesidades del cliente, la facilidad de uso de la herramienta y el ecosistema de socios, en donde se da la posibilidad de acceder a la gran red de socios que nos aporta Oracle.

Debido a que esta aplicación es gestionada por Oracle, se nos da la posibilidad de evolucionar en el futuro de Taleo a Oracle HCM, o lo que es lo mismo, pasar de una aplicación ATS a una TMS.

2.3.3. Conclusiones

Tras realizar el estudio sobre este tipo de sistemas, la solución ha de ser del tipo ATS, ya que este sistema es el que mejor se adapta a las necesidades dispuestas por la empresa a la que va dirigida la aplicación.

Sin embargo, aunque las aplicaciones que se han investigado en el apartado anterior son muy interesantes y cumplen con los requisitos mínimos, creo que la mejor opción es realizar una aplicación propia debido a los siguientes factores:

- Control total en el diseño, funcionalidad y características específicas.
- Mejor adaptación a los procesos internos y a los problemas o requisitos solicitados por la empresa.
- No dependencia en terceros para la actualización o resolución de problemas. Todo lo lleva a cabo un equipo dentro de la propia empresa, por lo que el trato y comunicación puede ser llevado a cabo de una manera más cordial.
- Menor coste a la larga. Aunque al principio los costes sean más altos debido a la necesidad de tener un equipo de desarrollo realizando el proyecto y manteniéndolo, poco a poco los costes se asemejan a tener una aplicación contratada a terceros debido a sus políticas de suscripción o limitaciones a futuro.

En conclusión, tras realizar el estudio del estado del arte, he decidido realizar una aplicación propia con enfoque en el sistema ATS y que cumpla con los requisitos mínimos solicitados. También se tendrá en cuenta este estudio para posibles actualizaciones futuras con funcionalidades de estas aplicaciones investigadas.

3. TECNOLOGÍAS UTILIZADAS

Para la elaboración de este Trabajo Fin de Grado, será necesario realizar un estudio de todas las tecnologías y herramientas disponibles para así poder seleccionar aquellas tecnologías y herramientas que mejor se adapten a las necesidades de este proyecto.

En este apartado no solo me enfocaré en seleccionar las tecnologías para el desarrollo de la aplicación, sino que también realizaré un estudio sobre otras herramientas que me puedan ser de gran utilidad a la hora de realizar este Trabajo Fin de Grado.

3.1. Estudio sobre tecnologías actuales

3.1.1. Editores de código

3.1.1.1. Visual Studio Code

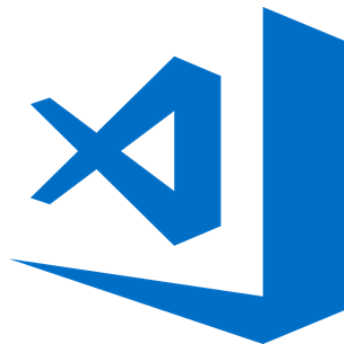


Ilustración 7. Logotipo Visual Studio Code

Editor de código gratuito desarrollado por Microsoft y lanzado en el año 2015. La característica más destacable de este editor es que tiene una gran cantidad de extensiones, por lo que podemos configurarlo totalmente a nuestro gusto y habilitar una gran cantidad de extensiones que nos facilitaran la vida a la hora de programar.

Esta característica comentada anteriormente, junto con la facilidad de uso que tiene para nuevos programadores, hace que sea una de los editores más populares a nivel mundial.

3.1.1.2. Sublime Text



Ilustración 8. Logotipo Sublime Text

[11] Editor de texto que fue desarrollado en 2007 por Jon Skinner, un antiguo ingeniero de Google. La principal característica de este editor es que nos simplifica totalmente el proceso de organizar y escribir el código, por lo que lo hace un buen editor para aquellos usuarios más experimentados que necesitan de múltiples herramientas y atajos para la navegación, escritura y búsqueda de código usando única y exclusivamente el teclado.

Este editor no es de código abierto, por lo que para obtener una licencia es necesario pagar una tarifa de 80\$, o usar la versión gratuita durante un tiempo limitado.

Al igual que VS Code, este editor tiene la posibilidad de instalar plugins adicionales para tener más funcionalidades, mediante la instalación y uso del denominado “Package Control”.

3.1.1.3. Atom

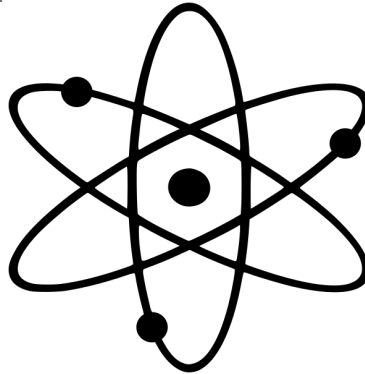


Ilustración 8. Logotipo Atom

[12] Atom es un editor de código que apareció en el año 2015 disponible para todos los sistemas operativos. Este editor es mantenido por su comunidad y es de código libre .

Al igual que los editores de código comentados anteriormente este tiene la posibilidad de instalar paquetes que contienen más funcionalidades y utilidades, por lo que lo hace totalmente configurable al gusto del usuario. Además, como este editor ha sido desarrollado por miembros de la plataforma GitHub, tiene la capacidad de sincronizarse automáticamente con nuestro repositorio de Git.

3.1.1.4. Conclusión y elección

Tras realizar un estudio y haber probado los tres editores he conseguido sacar las siguientes conclusiones[13]:

- A nivel de rendimiento, Sublime Text es la mejor opción ya que es un editor de texto con un gran rendimiento en lo que se refiere a velocidad y eficiencia. El siguiente sería VSCode que a lo largo del tiempo ha ido mejorando considerablemente su rendimiento. Finalmente tendríamos el editor Atom que se caracteriza por ser

algo lento y consumir una gran cantidad de recursos lo que hace que en ciertas ocasiones se ralentice nuestro ritmo de trabajo.

- Sobre las licencias y el precio, tenemos a Sublime Text como un claro perdedor, ya que es el único de los tres editores que no es de código libre o abierto, lo cual es una gran desventaja.
- Hablando de la personalización, en los tres editores he señalado la característica de la instalación de extensiones o paquetes. Sin embargo, VS Code es la mejor opción ya que es la que más cantidad de extensiones tiene disponibles actualmente.

Por ello, debido a las ventajas y desventajas puntualizadas anteriormente, llego a la conclusión de que los tres editores están a la par, no existen diferencias muy notorias entre uno u otro. Sin embargo, debido a lo familiarizado que estoy con VSCode, he decidido que este será el editor de código a usar para el desarrollo de este proyecto

3.1.2. Control de versiones

3.1.2.1. Git



Ilustración 9. Logotipo Git

Git [14] es un sistema de control de versiones creado por Linus Torvald y lanzado en el año 2007. Es de código abierto y actualmente es el más usado en el mundo.

Este sistema de control de versiones [6] tiene como principal característica la distribución de repositorios, o lo que es lo mismo, cada uno de los integrantes de un proyecto tienen una copia del repositorio en su ordenador. Esto hace que cada uno de los integrantes del proyecto pueda trabajar de manera independiente sin necesidad de conexión a Internet (Realizar commits sin conexión a Internet).

Además, teniendo el repositorio en una ubicación remota, conseguimos que los desarrolladores puedan sincronizar fácilmente el repositorio local con el remoto y nos aseguramos de que si el desarrollador elimina el repositorio de su ordenador por error, no le ocurrirá nada al repositorio remoto.

3.1.2.2. Subversion



Ilustración 10. Logotipo Subversion

Subversion es un sistema de control de versiones creado por CollabNet y lanzado en el año 2000. Es un software de código abierto bajo licencia de tipo Apache/BSD.

Es un sistema de control de versiones [6] del tipo centralizado, por lo que el proyecto está disponible en un repositorio y los distintos desarrolladores acceden al mismo para obtener los archivos necesarios. Para realizar las modificaciones y confirmarlas, es necesario tener conexión a internet

3.1.2.3. Mercurial



Ilustración 11. Logotipo Mercurial

Mercurial [16] es un sistema de control de versiones creado por Mat Mackall, fruto de la discontinuación del software BitKeeper. Es multiplataforma y libre.

Es un sistema de control de versiones [6] del tipo distribuido, por lo que tiene una gran escalabilidad y rendimiento. Además, es extensible por lo que podemos añadirle más funcionalidades mediante extensiones.

3.1.2.4. Conclusión y elección

Comparando las tres herramientas, llego a la conclusión de que la mejor elección es Git ya que tiene las siguientes ventajas [15]:

- Es un sistema distribuido, por lo que es más rápido, permite seguir trabajando aunque el repositorio remoto esté caído debido a que realizamos una copia total en local y permite trabajar sin conexión.
- Al comunicarnos de vez en cuando con el repositorio remoto, se consigue que las operaciones a realizar sean rápidas y ligeras.
- Su potencial a la hora de realizar ramificaciones, ya que favorece al trabajo de los desarrolladores, que no deben de preocuparse por realizar

cambios sobre el código funcional. Además, estas son fáciles de fusionar por lo que facilita aún más el trabajo entre desarrolladores.

- Gracias a la funcionalidad de solicitud de extracción, se facilita a los líderes de un proyecto en la revisión de los cambios realizados en una rama secundaria antes de fusionarla con una rama principal, así evitando problemas graves.
- Al ser la herramienta más popular actualmente, no es necesario explicar a nuevos integrantes del proyecto el flujo de trabajo ya que con casi total seguridad estarán familiarizados a Git.

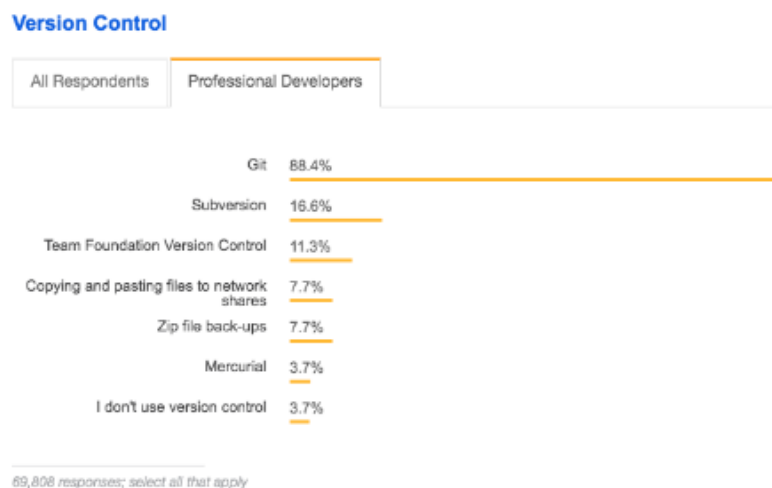


Ilustración 12. Imagen comparativa de herramientas de control de versiones [6].

Como podemos ver en la imagen anterior, casi el 90% de los desarrolladores tienen como primera opción el uso de Git.

3.1.3. Gestión de proyectos

3.1.3.1. Trello



Ilustración 13. Logotipo Trello

Trello [17] es una herramienta pensada para gestionar y/o administrar actividades y proyectos de una manera sencilla gracias a su formato simple. Tiene como principal objetivo la organización de información para así ayudar a los equipos de trabajo a realizar sus tareas de una manera más sencilla.

Tiene como principales características la facilidad de organización de la información mediante recursos visuales, la inclusión de herramientas como la “lluvia de ideas”, recoge el progreso del cumplimiento de un objetivo y permite el acceso de varios usuarios simultáneamente para así tener a todo un equipo de trabajo coordinado.

Sobre el precio de esta herramienta, existen distintos planes, por lo que dependiendo del proyecto y del tamaño de equipo, junto a las necesidades del mismo, se podrá elegir el plan que mejor se adapte. Estos son los planes:

FREE	STANDARD	PREMIUM	ENTERPRISE
\$0 USD Gratuito para todo el equipo Para personas o equipos que quieren organizar cualquier proyecto que se les ocurra. Empezar ahora mismo	\$5 USD por usuario al mes con facturación anual (\$6 con facturación mensual) Para los equipos pequeños que necesitan gestionar el trabajo y escalar la colaboración. Regístrate ahora	\$10 USD por usuario al mes con facturación anual (\$12.50 con facturación mensual) Para equipos que necesiten supervisar y visualizar varios proyectos de varias maneras, como en tableros, cronogramas, calendarios, etc. Probar gratis	\$17.50 USD por usuario al mes con facturación anual (\$210.00 precio anual por usuario) Para organizaciones que necesitan conectar el trabajo de distintos equipos con mayor seguridad y más controles. Coste estimado para <u>50</u> usuarios Contacto con el departamento de ventas

Ilustración 14. Imagen con los distintos planes de Trello

3.1.3.2. Jira



Ilustración 15. Logotipo Jira

Jira [18] es una herramienta que al igual que Trello, facilita la cooperación entre equipos para realizar una serie de tareas dentro de un proyecto. Es una herramienta que se puede usar tanto en el ámbito de desarrollo de software como en cualquier otro ámbito, gracias a su facilidad en la optimización del flujo de trabajo dentro de un equipo.

Como principales funcionalidades, podemos mencionar la facilidad para planificar y dividir una gran tarea en pequeños fragmentos, lo que favorece a una mejor gestión, inclusión de funcionalidades para la supervisión del trabajo, la generación automática de informes que ayuda a tener un mejor rendimiento del equipo y la automatización que nos aporta la herramienta.

Al igual que Trello, es una herramienta que contiene distintos planes, uno de ellos gratuito. Estos son los planes que tiene Jira:

Free	Standard	Premium	Enterprise
\$0 Siempre gratuito para 10 usuarios, Solo suscripción mensual Consíguela ahora	\$7.75 por usuario (estimado) \$77.50 al mes Iniciar versión de prueba	\$15.25 por usuario (estimado) \$152.50 al mes Iniciar versión de prueba	Facturación anual Pasa al ciclo de facturación anual para ver precios de Enterprise. Contacto con el departamento de ventas
✓ Hasta 10 usuarios ✓ Tableros de proyecto ilimitados ✓ Backlog y cronograma ✓ Informes e información útil ✓ 2 GB de almacenamiento ✓ Soporte de la comunidad	Todas las funciones de Free más: ✓ Hasta 35 000 usuarios ✓ Funciones y permisos de usuario ✓ Registros de auditoría ✓ Residencia de datos ✓ 250 GB de almacenamiento ✓ Soporte en horario comercial local	Todas las funciones de Standard más: ✓ Hojas de ruta avanzadas ✓ Espacio aislado y canales de publicaciones ✓ Archivado de proyectos ✓ SLA con tiempo de actividad garantizado ✓ Almacenamiento ilimitado ✓ Soporte Premium ininterrumpido	Todas las funciones de Premium más: ✓ Sitios ilimitados ✓ Controles de seguridad centralizados ✓ Suscripciones de usuario centralizadas ✓ Soporte Enterprise ininterrumpido

Ilustración 16. Imagen con los distintos planes de Jira

3.1.3.3. Conclusión y elección

En este caso, se ha seleccionado Jira como herramienta debido a que esta nos aporta mejores funcionalidades como la automatización de procesos y el seguimiento del progreso en tiempo real. Además, Jira nos aporta mejores informes y resúmenes detallados del progreso.

Otra causa importante por la que me he decantado por esta herramienta es la familiarización que tengo con ella, ya que, actualmente, es la herramienta que se está usando dentro de mi equipo de desarrollo de mi actual empresa.

3.1.4. Frameworks de desarrollo

En el mundo del desarrollo de software, el uso de frameworks se ha convertido en algo necesario para poder agilizar y simplificar el proceso de desarrollo de aplicaciones ya que estas nos aportan de base un conjunto de bibliotecas y herramientas útiles para el desarrollo de manera ágil y fácil.

Por ello, a continuación se va a realizar un estudio sobre los frameworks actuales y cuál va a ser seleccionado para el desarrollo de este proyecto. Hay que tener en cuenta, que mi aplicación web estará separada en dos partes:

- El front-end o capa superior, que se relaciona con todo lo que ven las personas o usuarios
- El back-end o capa inferior, donde ocurre todo lo que tiene que ver con la funcionalidad de la aplicación, optimización, conexión con Bases de datos,etc...

Al existir dos partes, será necesario encontrar dos frameworks que se adapten a las necesidades del proyecto.

3.1.4.1. Framework Back-End

En la selección de framework back-end existe un requisito impuesto por la empresa a la que va dirigida la aplicación. Este requisito es que el lenguaje de programación ha de ser PHP ya que todos los proyectos están realizados en este lenguaje. Por ello, se reducen drásticamente las opciones de framework back-end a seleccionar.

A continuación, se van a exponer dos de los framework PHP más utilizados, junto con sus ventajas y desventajas para así poder seleccionar aquel que se adapte mejor a las necesidades del proyecto.

3.1.4.1.1. Symfony



Ilustración 17. Logotipo Symfony

Symfony [19] es un framework de desarrollo de aplicaciones web basado en el patrón Modelo, Vista, Controlador y lanzado el año 2005, justo cuando se liberó PHP 5. Es un framework de código abierto el cual te ayuda a tener una mejor productividad gracias a la optimización que tiene en ciertas partes del desarrollo de aplicaciones.

Las principales ventajas de este framework son:

- **Compatibilidad:** Se puede instalar en cualquier sistema operativo. Esto es una gran ventaja ya que dentro de un equipo de

desarrollo puede haber integrantes que prefieran usar un sistema operativo u otro.

- **Intuitivo:** Es un framework que se hace muy sencillo para aquellos desarrolladores principiantes gracias a sus herramientas para generar código fácil y rápido. Además existen templates que permiten crear aplicaciones de una manera más sencilla para aquellos desarrolladores más principiantes.
- **Comunidad activa:** Actualmente es el proyecto PHP más activo por lo que existe una gran cantidad de documentación y tutoriales que facilita el desarrollo con este framework. Esto también hace que exista una gran cantidad de paquetes creados por otros usuarios que nos aportan infinidad de funcionalidades aplicables a nuestro proyecto.
- **Flexibilidad:** Este framework da la posibilidad de solo instalar aquellos módulos o paquetes que nos sea necesario, evitando así tener que instalar todo el framework.
- **Rendimiento:** Gracias a sus bajos requerimientos para funcionar, es un framework con un funcionamiento bastante ágil y bastante más rápido que otros entornos de la competencia.
- **Mantenibilidad y ampliabilidad:** Estas características son muy importantes ya que para el desarrollo de una aplicación, es necesario que el framework a utilizar de facilidades para mantenerlo y ampliar funcionalidades de una manera muy sencilla, sin tener que dedicar una gran cantidad de tiempo en resolver posibles errores a la hora de realizar modificaciones en la lógica.

Sin embargo, aunque existen una gran cantidad de ventajas, hay algunas características no tan ventajosas a tener en cuenta:

- **Rendimiento en versiones PHP anteriores:** Aunque una de las ventajas actuales del framework Symfony es el rendimiento, esto no era así en versiones anteriores de PHP. Para poder disfrutar de un rendimiento óptimo es necesario usar una versión de PHP igual o superior a 7, lo cual ya hace que sea un gran impedimento si nuestra aplicación va a estar basada en versiones de PHP más antiguas.
- **Tiempo de dedicación:** Symfony es un framework bastante amigable para aquellos usuarios principiantes, gracias a lo que comentamos anteriormente. Sin embargo, si queremos subir el nivel y tener una aplicación más profesional, será necesario dedicar una gran cantidad de tiempo para poder dominar todos los conceptos.

Como podemos ver, Symfony es una gran opción de framework para el desarrollo del Back-End de nuestra aplicación web.

3.1.4.1.2. Laravel

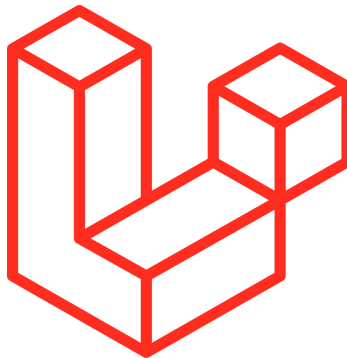


Ilustración 18. Logotipo laravel

Laravel [20] es un framework PHP creado en 2011 que ofrece funcionalidades que ayudan a la creación de aplicaciones web con código organizado y manejable de manera muy sencilla.

Los beneficios o ventajas de este framework son:

- **Facilidad de uso:** Es un framework que tiene como principal característica su sencillez a la hora de aprenderlo. Gracias a su extensa documentación, junto a la gran cantidad de tutoriales, hacen que el Laravel sea muy sencillo de utilizar para todo tipo de proyectos.
- **Arquitectura modular:** Su arquitectura modular permite a los desarrolladores tener un mayor control y capacidad para modificar y extender código gracias al uso de componentes.
- **Testing:** Es un framework que favorece la implementación de pruebas gracias a su soporte con PHPUnit, el framework de testing PHP más importante actualmente.
- **Gran comunidad:** Actualmente, tanto PHP como Laravel están muy recomendados ya que existe una gran comunidad que aporta y ayuda en la evolución y sostenibilidad tanto del lenguaje PHP como del framework Laravel.

Las desventajas son:

- **Lento:** Según se dice, las aplicaciones desarrolladas con Laravel, no destacan por su velocidad, aunque existen algunas maneras de mejorar en este ámbito.
- **Problemas en actualizaciones:** También existe el problema de que cuando una nueva versión de Laravel es lanzada, es necesario realizar una gran cantidad de cambios o parches para que así nuestro proyecto siga funcionando de manera correcta. Esto es un gran problema ya que no ayuda a la mantenibilidad del proyecto.

- **Versiones de PHP:** En este framework, al contrario que con Symfony, es necesario tener una versión superior a PHP 5.4. En mi caso no es un problema ya que no voy a usar una versión anterior a PHP 5.4 para el desarrollo de este proyecto.

3.1.4.1.3. Comparativa

Tras exponer las ventajas y desventajas de cada una de las opciones de framework Back-End, voy a realizar una pequeña comparativa [21] entre las dos opciones, para así llegar a la conclusión de cuál se adapta mejor a mis necesidades.

Entre Laravel y Symfony existen 3 diferencias notables:

- **Rapidez en el desarrollo:** Si la idea es generar un proyecto pequeño y rápido, se recomienda utilizar Laravel ya que es mucho más flexible y menos exigente que Symfony. Además, Laravel ha sido diseñado para que el usuario tenga las menos complicaciones posibles, por lo que lo convierte en un framework más sencillo que Symfony.
- **Potencia y rendimiento:** Symfony tiene un core más potente que Laravel. A esto si le sumamos la mejora obtenida en el rendimiento si usamos Symfony con PHP superior a 7, lo convierte en la mejor opción si lo que buscamos es la potencia y el rendimiento de nuestra aplicación.
- **Trabajo con Bases de Datos:** Symfony tiene la ventaja de la automatización de migraciones, la cual en Laravel es de manera manual.

En conclusión, tras exponer las características de cada una, junto con su comparativa, he llegado a la conclusión de que el framework que mejor se adapta a mis necesidades es Symfony. Aunque es verdad que la simplicidad

del framework Laravel llama mucho la atención, he optado por elegir Symfony ya que es preferible la escalabilidad y solidez que nos aporta este framework.

3.1.4.2. Framework Front-End

Al igual que he hecho en la selección de framework Back-End, voy a realizar un estudio sobre los frameworks Front-End más populares, exponiendo sus características.

Actualmente, existen dos opciones muy interesantes para el desarrollo Front de aplicaciones, React y Angular.

3.1.4.2.1. React

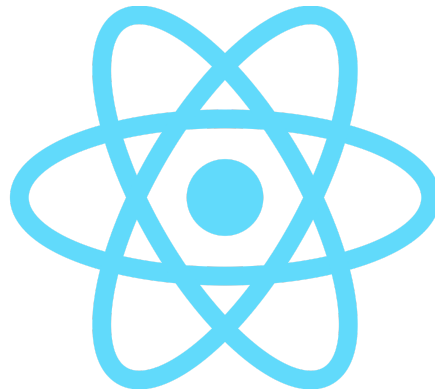


Ilustración 19. Logotipo React

React [22] es la librería de JavaScript más conocida actualmente en el desarrollo de aplicaciones web desarrollada por Facebook. Como he dicho anteriormente, React se considera a sí misma como una librería y no como un framework. Esto se debe a que React necesita de varios módulos para obtener las mismas funcionalidades que un framework de desarrollo Front-End.

Aunque lo comentado anteriormente parezca una desventaja, es todo lo contrario ya que esto lo convierte en una herramienta muy simple y fácil de aprender, y añadiendo algunos módulos extra puedes conseguir que React se convierta en un framework a la par que la competencia.

A continuación se van a exponer las ventajas que nos aporta React a la hora de desarrollar una aplicación web.

- **Fácil de usar:** Si eres un desarrollador con un poco de experiencia con JavaScript, puedes aprender a usar React en un plazo corto de tiempo ya que React es una librería Javascript.

- **Reutilización de componentes:** Mediante la implementación de componentes, podemos reutilizar otras componentes ya implementadas en otros proyectos, ya sean propios o de la comunidad, haciendo así que se reduzcan los tiempos en el desarrollo de la aplicación.

- **Facilidad a la hora de programar:** Gracias a la integración de JSX, podemos crear componentes mezclando Javascript con etiquetas HTML de una manera sencilla.

- **Buen rendimiento:** Gracias a la representación del DOM de forma virtual, hacemos que el proceso de actualización de la aplicación tras un cambio sea lo más ligero posible. Esto se debe a que gracias al Virtual Dom, solo actualizamos aquellos nodos de la aplicación que sean estrictamente necesarios, ahorrando así recursos en el procesamiento.

Sin embargo, aparecen una serie de inconvenientes que nos hacen pensar sobre la elección o no de esta librería.

- **Velocidad en las actualizaciones:** Aunque esto parezca una ventaja, no lo es cuando en el desarrollo de tu aplicación necesitas actualizar cada cierto tiempo partes del código ya desarrollado porque ha aparecido una nueva actualización de cierto módulo, que nos hace modificar gran parte del código ya implementado. Esto provoca también que exista poca documentación debido a su constante evolución.

- **Demasiada flexibilidad:** React, al no tener de base una estructura a la que adaptarnos, deja un poco de libertad a cada usuario para implementar su proyecto y estructurarlo según le convenga. Esto puede ser un problema cuando en un equipo entran y salen cada poco tiempo nuevos integrantes, provocando que tengan que adaptarse a la estructura establecida en ese proyecto.

- **No es un framework:** Aunque, como comente anteriormente, por una parte es una ventaja, ya que podemos añadir sólo aquellos módulos que nos interesen para crear nuestra aplicación, tiene su parte negativa. Esta es que si por ejemplo, tomamos un framework con una arquitectura MVC de base, este va a tener una mejor estructura de código y con patrones de diseño, mientras que React, al solo controlar la parte de la Vista, no va a ser tan estructurado ni fácil de entender y mantener.

3.1.4.2.2. Angular



Ilustración 20. Logotipo Angular

Angular [23] es un framework de JavaScript desarrollado por Google, de código abierto y escrito en TypeScript. Es usado para el desarrollo de aplicaciones web del tipo SPA (Single Page Application).

Este framework utiliza la arquitectura MVC, siendo Angular el modelo, la vista el código HTML y el controlador el código Javascript. De esta manera se ayuda a los desarrolladores a reducir la cantidad de código a implementar.

Este framework tiene una serie de ventajas:

- **Seguridad y funcionalidad:** Al ser un framework, la gran mayoría de funcionalidades ya vienen por defecto, haciendo que no sea necesario implementar librerías de terceros que pueden ser más inseguras y de menor calidad que las aportadas por el equipo de Angular.
- **Arquitectura robusta:** Como he comentado anteriormente, el seguir el patrón MVC provoca una mejor estructura de código y una mejor colaboración entre desarrolladores, mejorando en velocidad y escalabilidad.
- **Reutilización de código:** Se permite crear componentes, las cuales pueden ser llamadas desde cualquier parte de la aplicación, mejorando la presentación gracias a la encapsulación de código.
- **Gran comunidad:** La documentación, junto con la gran comunidad detrás de este framework, hace que se facilite un poco el aprendizaje.

Sin embargo, aparecen un par de desventajas que pueden echar para atrás a la hora de seleccionar a Angular como framework para desarrollo:

- **Curva de aprendizaje:** Si eres principiante de Angular, puede ser algo costoso el aprendizaje al principio debido a sus características complejas. Esto se hace más liviano gracias a la documentación.

- **Tamaño:** Al ser un framework, las aplicaciones desarrolladas serán más pesadas debido a las dependencias que esta trae por defecto. Esto puede llegar a provocar un problema en el rendimiento de ciertas aplicaciones.

3.1.4.2.3. Comparativa

Tras realizar el estudio de estas dos opciones para el desarrollo del front de la aplicación, se va a realizar una breve comparativa entre React y Angular, para así ver cuál se adapta mejor a mis necesidades.

- **Curva de aprendizaje:** La curva de aprendizaje es un factor a tener muy en cuenta, ya que los tiempos de entrega pueden variar drásticamente entre la selección de una u otra opción. En este caso, React es una mejor opción ya que es muchísimo más simple.
- **Comunidad y documentación:** Aunque como se ha dicho anteriormente, Angular tiene una gran documentación y comunidad, esta ha ido yendo cada vez a menos debido a la aparición de React.

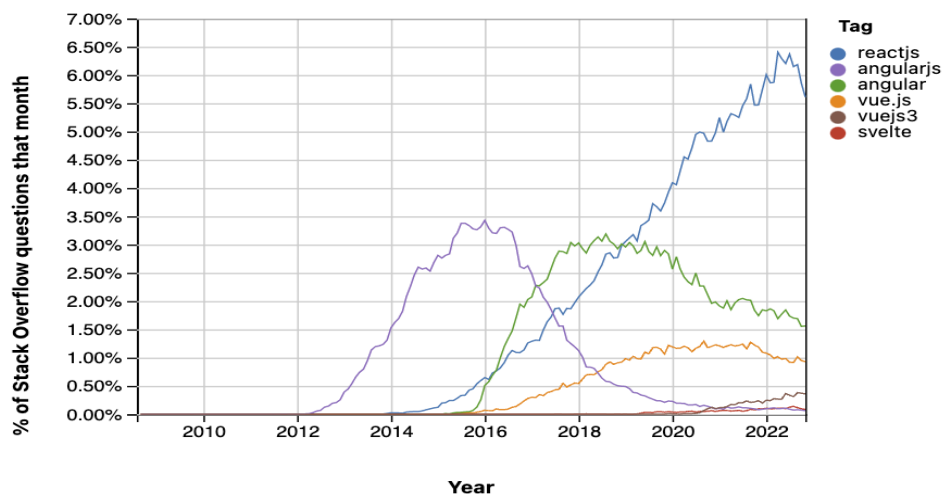


Ilustración 21. Comparativa de popularidad de frameworks Front-End

Aunque en el gráfico solo se muestra hasta el año 2022, actualmente React tiene una popularidad del 40% frente al 24% que tiene Angular.

- **Rendimiento:** Ambas herramientas tienen un rendimiento similar, siendo React algo mejor en proyectos más ambiciosos gracias al virtual Dom.
- **Sencillez:** Angular, al ser un framework completo, tiende a ser más difícil de integrar, frente a React que aunque necesita implementar varias librerías atiende a ser más sencillo de integrar.
- **Integridad:** Angular, gracias a su arquitectura MVC, tiene una solidez mayor, lo cual es una gran ventaja frente a React la cual para implementar esta arquitectura necesita de varias librerías que lo hace más difícil de implementar.

En conclusión, tras realizar este estudio, he decidido seleccionar React para el desarrollo de mi aplicación. Pienso que React se puede adaptar mejor a mis necesidades ya que mi aplicación no necesita tener todas las funcionalidades que me puede aportar el framework Angular.

4. INGENIERÍA DEL SOFTWARE

4.1. Introducción

En este apartado, voy a explicar que es la ingeniería del software y cómo ha ido evolucionando hasta llegar a la actualidad, en donde se ha convertido en una de las áreas más importantes dentro de las compañías para el desarrollo de software.

Antes de comentar su aparición y evolución, hay que definir el concepto de Ingeniería del software. Se puede decir que la ingeniería de software [24] es una disciplina o área de la informática que ofrece métodos y herramientas para facilitar y ayudar en el desarrollo de software de calidad.

Para poder explicar mejor qué es la ingeniería del software y su importancia, debemos de conocer su aparición y evolución.

[25] El concepto de Ingeniería de Software aparece por primera vez en 1968, en una conferencia en Garmisch (Alemania) en donde se expuso el problema conocido como “Crisis del Software”. La Crisis del Software fue una etapa comprendida entre los años 1960 y 1980 en donde todos los desarrollos de nuevo software tenían problemas durante el proceso de desarrollo debido a una mala planificación. Ante esta situación, se decidió buscar una solución que aplicar en todos los nuevos desarrollos para así poder solventar los problemas.

En primer lugar, aparece el denominado Modelo en Cascada en 1970, en donde todo desarrollo de software debería de seguir un ciclo de vida. Este ciclo de vida se dividía en varias etapas, siendo necesario finalizar una etapa para poder pasar a la siguiente, sin poder volver atrás.

Tras esto, pasan un par de décadas en donde muchos investigadores intentan buscar soluciones para solventar completamente la Crisis del

Software, creando nuevas tecnologías y prácticas para aplicar en el desarrollo de software. No es hasta el año 1985, donde surge el ciclo de vida del software, el cual sigue siendo el ciclo a seguir dentro de la Ingeniería del Software.

Finalmente, entre el año 1990 y la actualidad, ocurren ciertos eventos, como la aparición de metodologías ágiles, el abandono de las metodologías pesadas basadas en el modelo en cascada, aparición de la integración continua, el desarrollo del manifiesto ágil y la aparición de metodologías híbridas.

Como se puede observar, el concepto de Ingeniería del Software lleva siendo muy importante desde casi el principio de la programación, siendo cada vez más usado y necesario en todos los desarrollos de software. Por ello, es necesario dedicarle un apartado especial en este trabajo, ya que va a ser muy importante su aplicación en el desarrollo, para así poder tener una aplicación sencilla y de calidad.

En los siguientes apartados se va a explicar detalladamente cada una de las etapas del ciclo de vida del desarrollo de software y su aplicación en el proyecto a desarrollar.

4.2. Definición de requisitos

Dentro del ciclo de vida del desarrollo del software, nos encontramos con la etapa de definición de requisitos. En esta etapa, como bien indica su nombre, se realiza el estudio y se recogen todas las características funcionales y no funcionales que ha de tener el software final.

Es necesario que todos los requisitos formulados sigan una serie de reglas que van a ser enumeradas a continuación:

- **No ambiguo** → El requisito sólo puede tener una interpretación posible.
- **Conciso** → No utilizar tecnicismos, debe de ser comprensible.
- **Consistente** → Debe de existir independencia entre requisitos, uno no puede solapar a otro.
- **Completa** → El requisito debe de contener toda la información en él.
- **Alcanzables** → Es necesario que el requisito pueda llegar a ser conseguido.
- **Verificables** → Debe de existir formas para comprobar que el requisito ha sido solventado correctamente.

A continuación se va a realizar la definición de los requisitos que va a tener la aplicación. Se va a diferenciar entre requisitos funcionales y no funcionales.

4.2.1. Requisitos funcionales

Este tipo de requisitos se refieren a aquellas funcionalidades o servicios que prestarán el sistema y como responderá ante estímulos del exterior. Para la aplicación a desarrollar se han obtenido los siguientes requisitos funcionales:

- El usuario podrá realizar un inicio de sesión en la aplicación web mediante el uso de unas credenciales otorgadas por un administrador.
- El usuario podrá visualizar en la pantalla principal un resumen informativo con los datos más relevantes.
- El usuario siempre tendrá acceso a un menú superior y lateral para poder moverse entre los distintos apartados de la aplicación.
- El usuario podrá añadir nuevas ofertas de trabajo que serán publicitadas en la aplicación de petroprix.

- El usuario podrá añadir candidatos mediante el uso de un formulario y asociarlos a una o varias ofertas de trabajo.
- El usuario podrá gestionar a los candidatos de una oferta en el proceso de selección mediante el uso de un tablero Kanban.
- El usuario tendrá acceso a un apartado denominado Talent Pool, en donde se almacenarán aquellos perfiles de usuario que pueden ser interesantes a futuro y es necesario mantener un seguimiento en ellos.
- El sistema estará preparado para recibir candidatos aunque no existan ofertas de empleo, almacenándolos en el Talent Pool para ser gestionados a futuro.
- El usuario podrá realizar búsquedas de ofertas de empleo, ya estén finalizadas o no para así mantener un historial.
- El usuario podrá buscar entre los candidatos y consultar la información del mismo.
- El usuario podrá modificar la información almacenada de los candidatos.
- El sistema enviará automáticamente correos a los candidatos para mantenerlos informados de su estado en el proceso de selección.

Además de los requisitos nombrados anteriormente, existen otros requisitos opcionales, los cuales no va a ser posible completar para esta primera versión de la aplicación. Estos son los requisitos:

- El sistema será capaz de comunicarse con distintas APIS de los portales más importantes de empleabilidad para así publicar automáticamente las ofertas en estos portales.
- Desarrollo de un apartado “Talento” en la aplicación petroprix.com

- El sistema podrá enviar formularios con una batería de preguntas a los candidatos para así poder seleccionar aquellos candidatos que más interesen.
- El sistema implementará una IA la cual ayudará en el proceso de selección y obtención de candidatos.
- Generar un rol ADMIN que tendrá acceso a un apartado especial para generar nuevos reclutadores y enviar sus credenciales por correo.

4.2.2. Requisitos no funcionales

Se tratan de restricciones que no especifican nada sobre la funcionalidad del sistema, sino más a las propiedades que va a tener, como la escalabilidad, rendimiento, portabilidad,...

Para la aplicación existen estos requisitos no funcionales:

- El sistema tendrá una interfaz sencilla e intuitiva para el usuario.
- La aplicación debe de poder ser visualizada en cualquier tipo de dispositivo, con distintos tamaños de pantalla.
- El sistema ha de estar protegido contra accesos no autorizados.
- El sistema debe de poder tener varios usuarios conectados simultáneamente sin perder rendimiento.
- El sistema ha de estar disponible cuando sea necesario.

4.3. Planificación

Etapa del ciclo de vida del desarrollo de software en donde se trazan objetivos y se valoran las necesidades del proyecto, es decir, se realizan tareas como el análisis de costos, estimación de recursos, asignación de tareas, etc... Esta etapa está al mismo nivel que la definición de requerimientos, por lo que hasta no realizar la planificación ni finalizar la documentación de especificaciones de requerimientos no podremos avanzar en el ciclo de vida.

4.3.1. Selección del enfoque de Ingeniería del Software

En la etapa de planificación es necesario tomar una decisión muy importante, decidir si tomar una planificación enfocada en metodologías tradicionales o por el contrario, una metodología ágil.

Para saber qué metodología usar, es necesario en primer lugar saber qué son y qué características tiene cada una de ellas.

- **Metodologías tradicionales** → Son metodologías donde se planifica todo el proyecto de antemano, es decir, no es proactivo a cambio de requisitos u adaptación a las necesidades que surjan a lo largo del desarrollo de software. Son además metodologías con un alto costo de implementación. Actualmente, todos los proyectos necesitan flexibilidad a lo largo del desarrollo, además de tener restricciones en el tiempo, lo que ha provocado que actualmente se hayan dejado de lado este tipo de metodologías.
- **Metodologías ágiles** → Metodologías basadas en la división de las fases del proceso de desarrollo en otras aún más pequeñas, denominadas sprints. De esta manera se obtiene mayor versatilidad en el proceso de desarrollo. Además, se caracterizan por ser metodologías adaptativas a las necesidades actuales del proyecto.

Para el desarrollo de este proyecto, va a ser necesario seguir el enfoque de las metodologías ágiles, ya que actualmente es necesario involucrar al cliente dentro del proceso de desarrollo. Para ello, el cliente continuamente recibirá y podrá observar los avances del producto desde etapas tempranas del desarrollo, obteniendo de esta manera feedback para la mejora continua en la etapa de desarrollo del siguiente sprint.

Dentro de las metodologías ágiles podemos encontrar 3 metodologías muy usadas por las empresas. Su elección dependerá de la naturaleza del proyecto y la experiencia de la empresa en este tipo de metodologías. Estos son los tipos de metodologías ágiles que podemos encontrar:

4.3.1.1. Elección de metodología ágil a usar

- **Scrum:**

Scrum es una metodología cuyo enfoque consiste en que los equipos se organizan por cuenta propia para así alcanzar un objetivo común. Para ello se realizan una serie de reuniones y se utilizan una serie de herramientas para que las entregas de los proyectos sean lo más rápidas y eficientes posible.

Esta metodología se basa en 6 principios fundamentales:

1. Control de los procesos empíricos → Las entregas frecuentes ayudan a que el producto sea probado y mejorado en cada una de las entregas realizadas.
2. Autoorganización → Todo miembro del equipo debe de participar por completo en el proceso, por ello se necesita darle a cada uno independencia y autoorganización.

3. Colaboración → El trabajo en conjunto es una de las características más importantes en este marco de trabajo. Sin ella, no se puede realizar la entrega del mejor producto posible.
4. Priorización → La flexibilidad aportada por Scrum hace que los equipos evalúan y priorizan tareas según conveniencias en el momento.
5. Asignación de tiempo → Los cronogramas estipulados han de ser cumplidos estrictamente, tanto entregas como las reuniones diarias.
6. Desarrollo iterativo → El desarrollo iterativo provoca que el producto obtenga una mejora constante, además de una gran flexibilidad, obteniendo un producto de calidad.

- **Kanban:**

Kanban se basa en el trabajo en equipo y el flujo de tareas permanente. La planificación se basa en la visualización de tareas mediante el uso de un tablero, simplificando de esa manera esta etapa de planificación. Las tareas se sitúan en tres etapas posibles (Pendientes, en curso, finalizadas), mediante el uso del tablero. Además, solo se pueden tener 3 tareas en curso al mismo tiempo. Hasta no finalizar una no se puede empezar una nueva. Esto ayuda a que el flujo de trabajo sea ágil.

De esta forma, se consigue reducir el tiempo de realización de un proyecto gracias a la mejora continua del flujo de trabajo.

- **Extreme Programming:**

Metodología ágil que divide el desarrollo del proyecto en fases. Cada una de las fases de desarrollo es como un ciclo de vida completo, es decir, cada fase tiene su propia planificación, diseño, desarrollo y pruebas.

Es una metodología enfocada a proyectos con requisitos aún sin definir completamente, en donde pueden surgir muchos cambios en las necesidades. Es la metodología más adaptativa al cambio ya que se pueden aceptar cambios hasta en etapas tardías del ciclo de vida del proyecto.

Para el proyecto a realizar, pienso que la metodologías que mejor se adaptan son Scrum y Kanban ya que los requerimientos están definidos y son fijos, por lo que no se aprovechará completamente el enfoque de Extreme Programming.

Entre Scrum y Kanban, pienso que la mejor metodología a usar será Scrum debido a las desventajas que nos aporta Kanban.

1. Kanban no es una metodología que se pueda aplicar de forma óptima a todos los tipos de proyectos. Si tenemos eventos inesperados o variaciones, puede que el producto final no sea el mejor.
2. Aunque actualmente el equipo de desarrollo solo sea una persona, a futuro este proyecto seguirá evolucionando, entrando en él un equipo de desarrollo con varias personas. Al tener limitadas el número de tareas abiertas a solo 3, es posible que algunos trabajadores queden desocupados mientras que los otros finalizan sus tareas debido a la restricción de solo 3 tareas a la vez.
3. El almacenamiento del sistema Kanban en proyectos medio-grandes es muy costoso.

Por ello, el enfoque de la metodología Scrum es el elegido para desarrollar la planificación del proyecto. Para ello se aplicarán todas sus características y el proceso del mismo.

4.3.1.2. Aplicación de la metodología seleccionada

Dentro de la metodología Scrum existen una serie de características, etapas y procesos a seguir. A continuación se van a exponer y especificar cada una de ellas.

4.3.1.2.1. Roles

Dentro de los equipos de Scrum, existen tres roles específicos:

- **Scrum Master** → Persona experimentada en el framework que tiene la responsabilidad de que se comprenda y aplique. Es el encargado de que se maximice la eficiencia del equipo. En mi caso, debido a que el desarrollo de la aplicación se está realizando para la empresa donde estoy realizando las prácticas, el Scrum Master será mi Tutor Profesional Antonio Morillo Moreno el cual está muy experimentado en este framework y es el encargado de ser el Scrum Master dentro de los proyectos de esta empresa.
- **Product Owner** → Esta persona es la representante del negocio, los clientes y guía a los desarrolladores para construir el proyecto. Es decir, es el que organiza las tareas, las clarifica y prioriza según las necesidades. En mi caso, debido a que este proyecto lo estoy realizando solamente yo, el PO sería yo.

- **Desarrolladores** → Se encargan de la realización de cada uno de los incrementos del proyecto. En mi caso también sería yo ya que soy el único desarrollador actualmente.

4.3.1.2.2. Artefactos

Los artefactos en Scrum son aquellos elementos físicos que aparecen al aplicar la metodología. Estos son los principales artefactos:

- **Product Backlog** → Lista de todos los trabajos que se van a realizar para obtener el producto final. En él aparecerán todas las historias de usuario, funcionalidades, tareas y trabajos que deben ser realizadas. No es necesario que aparezcan todas las tareas desde el principio, se pueden ir añadiendo en el Product Backlog según aparezcan las necesidades.
- **Sprint Backlog** → Lista de todas las tareas que van a ser realizadas durante un sprint. Este artefacto es gestionado por el equipo de desarrollo durante el desarrollo del sprint para así tener un control de quien realiza las tareas, cuales están en desarrollo y cuáles quedan pendientes.
- **Incremento** → Llamamos incremento a la entrega que se consigue al finalizar un sprint y que es dada al cliente para recibir el feedback del mismo.

4.3.1.2.3. Reuniones

Para aplicar correctamente la metodología Scrum, es necesario realizar los siguientes eventos de forma regular y sin saltarse ninguno, ya que esto puede llevar a que el resultado final del producto no sea el mejor.

Existen 5 eventos principales en la aplicación de la metodología Scrum:

- **Sprint Planning** → Reunión realizada al principio del Sprint donde el equipo completo selecciona qué tareas del Product Backlog van a ser pasadas al Sprint Backlog y realizadas en ese Sprint. En esta reunión solo participa el Product Owner y el equipo de desarrollo, en donde se expone qué se va a realizar a lo largo del Sprint y cómo se va a realizar, estimando las tareas y pensando la mejor forma de llevar a cabo esas tareas. En mi caso, al ser el único integrante, esta reunión la realizaré yo solo, realizando ambos papeles con sus respectivas tareas.
- **Daily Scrum** → Reunión diaria en donde se expone al equipo de desarrollo lo que se hizo el día anterior, lo que se va a realizar hoy y problemas encontrados durante el desarrollo de las tareas. En mi caso esta reunión no se va a realizar, ya que al ser el único integrante no tiene mucho sentido.
- **Sprint Review** → Reunión realizada al final del Sprint donde se expone a los clientes el incremento obtenido. Previamente a esta reunión, el incremento ya debe de haber sido testeado y finalizado, siendo en esta reunión una demostración del producto en un entorno de producción. En mi caso, estas reuniones si se realizaran al finalizar cada Sprint, siendo los Stakeholders los integrantes de Recursos Humanos de mi empresa, donde les mostrare el avance de la aplicación y recibiré por su parte el feedback, actualizando de esa manera el Product Backlog con las nuevas necesidades.
- **Sprint Retrospective** → Reunión que se realiza también al finalizar el Sprint, antes o después del Sprint Review. En ella se realiza una reflexión sobre el último sprint y se exponen las ideas o problemas que los desarrolladores han detectado. De esta forma se intenta mejorar para que el próximo Sprint todo vaya mejor. Al ser un único integrante, esta Sprint Retrospective se realiza antes que la Review, ya que a lo

largo del Sprint se habrán detectado los posibles problemas y no es necesario exponerlos a nadie

- **Sprint Refinement** → Reunión donde se intenta actualizar y dejar en el mejor estado posible el Product Backlog. De esta forma siempre las tareas están claras y están preparadas para ser añadidas al Sprint Backlog sin que los desarrolladores tengan problemas en entenderlas. Este refinamiento se realizará por mi parte siempre al finalizar la Sprint Review, ya que con casi total seguridad aparecerán nuevas tareas para el Product Backlog, por lo que aprovecharé a actualizarlo completamente.

4.3.1.2.4. Divisiones del proyecto en Sprints

Como se ha comentado anteriormente, vamos a aplicar la metodología Scrum, por ello es necesario dividir el proyecto en distintos Sprints. Aunque no se sepa aún qué tareas del Product Backlog se van a realizar en cada uno de los Sprints, se va a realizar esta división para saber del tiempo que se dispone y de cuantas semanas serán cada uno de los Sprints, ya que dependiendo del proyecto, estos pueden durar desde 1 semanas hasta 4 semanas.

En mi caso, el desarrollo del proyecto comenzará el día 1 de febrero y finalizará el 31 de agosto. Por ello se disponen de 7 meses para el desarrollo completo del proyecto, tanto de documentación como de la aplicación en sí.

La idea es que los Sprints tengan una duración de 2 semanas, ya que es tiempo suficiente para realizar un incremento suficiente como para mostrar nuevas funcionalidades y características de la aplicación a los integrantes de recursos humanos.

De esta forma, la planificación se quedará de la siguiente manera:

Sprint	Tarea	Fecha
1	Estudio sobre el estado del arte y tecnologías a utilizar. Recolección de requisitos realizando reuniones con los integrantes de RRHH. Elección de las metodologías a utilizar y realización de tutoriales.	1 de febrero al 10 de febrero
2	Desarrollo de bocetos del front de la aplicación. Desarrollo inicial del proyecto Front, con las primeras vistas.	13 de febrero al 24 de febrero
3	Realización de la segunda etapa en el desarrollo del front	27 de febrero al 10 de marzo
4	Realización de la tercera etapa en el desarrollo del front	13 de marzo al 24 de marzo
5	Realización de la cuarta etapa en el desarrollo del front	27 marzo al 7 de abril
6	Realización de la quinta etapa en el desarrollo del front	10 de abril 21 de abril
7	Desarrollo de la primera etapa de la API	24 de abril 5 de mayo
8	Desarrollo de la segunda etapa de la API	8 de mayo al 19 de mayo
9	Desarrollo de la primera versión de la documentación a entregar	22 de mayo al 2 de junio
10	Desarrollo de la tercera etapa de la API	5 de junio al 16 de junio
11	Desarrollo de la cuarta etapa de la API	19 de junio al 30 de junio
12	Desarrollo de últimas funcionalidades tanto en el front como en el back, junto con últimas pruebas de la solución	3 de julio al 14 de julio

13	Desarrollo de documentación	17 de julio al 28 de julio
14	Desarrollo final de la documentación a entregar	14 de agosto al 25 de agosto
15	Revisión final de la documentación y últimos retoques	28 de agosto al 1 de septiembre

Tabla 1. Planificación Sprints

Como se puede ver, se ha tenido en cuenta en la planificación una semana de vacaciones en donde no se realizó nada sobre el proyecto, además de introducir un último sprint de solo una semana para revisión y corrección de la versión final de la documentación.

Hay que tener en cuenta que se han dedicado 2 horas diarias, siendo el desarrollo de un Sprint una media de 20 horas.

En apartados posteriores, se realizará una explicación más extensa de cada uno de los Sprint, mostrando cada una de las tareas realizadas en ese Sprint, y la evolución del Sprint Backlog y del Product Backlog.

4.3.1.2.5. Estimación de costes del proyecto

Para la realización de la estimación del coste de desarrollo de este proyecto, se va a utilizar el método COSMIC. Este método se basa en la medición de las tareas mediante el uso de puntos de función COSMIC. Con estos puntos, y obteniendo el precio medio que tendrá cada uno de los puntos, podemos llegar a la estimación del coste del proyecto junto con una estimación de tiempo.

Para la realización de la estimación de puntos COSMIC, hay que analizar cada uno de los requisitos funcionales y no funcionales del proyecto. Como no es necesario realizar un estudio tan exhaustivo del coste, sino que

solo necesitamos una aproximación, llego a la conclusión que realizando únicamente el estudio de uno de los requisitos, será suficiente para poder estimar más o menos los demás requisitos ya que todos los requisitos impuestos tienen una dificultad parecida.

Para el caso del requisito **"El usuario podrá añadir candidatos mediante el uso de un formulario y asociarlos a una o varias ofertas de trabajo"** aparecen los siguientes movimientos de datos:

- Entrada: Selección en el apartado de candidatos, la opción de añadir candidato
- Salida: Muestra en pantalla de un formulario completo para añadir el candidato.
- Entrada: Especificamos los datos completos del candidato
- Entrada: Adjuntamos el currículum aportado por el candidato
- Entrada: Seleccionamos la oferta de trabajo al que queremos asociarlo
- Escritura: Creacion de un registro candidato
- Escritura: Creación de un registro candidato_oferta
- Salida: Mostramos en pantalla una confirmación de operación
- Salida: Mostramos en la pantalla, la tabla con los candidatos, en donde aparecerá este nuevo candidato.

Como se puede observar, para este caso, se estiman 9 puntos COSMIC para esta funcionalidad.

Por ello, para todos los requisitos, estimamos una media de 9 o 10 puntos dependiendo del requisito, siendo en total una media de 150 puntos COSMIC.

Ahora, hay que calcular qué valor tiene cada uno de los puntos COSMIC para nuestro proyecto en concreto. Para obtener ese valor, es necesario

comparar con otro proyecto ya realizado y obtener qué coste real tuvo el desarrollo de ese proyecto.

Tras consultar a la empresa, he obtenido el dato que de media, para la realización de un proyecto de este tipo, con un equipo de desarrollo de 3 o 4 personas, se suelen realizar una media de 80 puntos COSMIC al mes. En mi caso, al ser solo una persona, de media podré realizar unos 20 puntos COSMIC al mes.

Hay que tener en cuenta, que lo más normal es que un equipo de desarrollo realice varios proyectos en paralelo, por lo que de media, podían dedicar unas 2-3 horas al día por desarrollador para cada proyecto como máximo.

Es decir, para el desarrollo de este proyecto, implementando todos los requisitos solicitados en esta primera versión, se estima que necesitaré una media de 7 meses y medio solo en el desarrollo.

Con respecto al coste, se estima que el coste medio al mes del equipo de desarrollo fue de 8000€. Como he dicho anteriormente, no era el único proyecto en desarrollo, por lo que el coste medio del equipo, solo en el desarrollo de ese proyecto fue de 2000€ mensuales.

Como se han desarrollado 80 puntos COSMIC al mes, llegamos a la conclusión de que el valor de cada punto COSMIC es de $2000/80 = 25\text{€}$. Por ende, para la implementación total del proyecto se estima que el coste será de $150 * 25 = 3750\text{€}$ solo en el apartado de desarrollo.

Como he comentado anteriormente, se estiman 7 meses y medio de desarrollo para tener todos los requisitos impuestos. Debido a que en la planificación de los sprints, solo hay disponible 5 meses y medio de desarrollo de la aplicación, por lo que va a ser imposible tener todos los requisitos completos para esta primera versión del proyecto. Por ello, he decidido que se

van a intentar completar todos los requisitos pero de una manera parcial, es decir, solo se va a realizar el desarrollo de las funcionalidades básicas de cada requerimiento, dejando las funcionalidades más complejas para futuros desarrollos.

4.4. Análisis

En esta etapa, como su nombre bien indica, se va a realizar un análisis conceptual del proyecto a realizar mediante el uso de diagramas.

4.4.1. Diagrama de clases

El diagrama de clases es un tipo de diagrama que describe la estructura o el diseño que tendrá una aplicación y permite modelar las relaciones que tendrán las distintas entidades de la misma.

Para la primera versión de esta aplicación, el diagrama de clases es el siguiente:



Ilustración 22. Diagrama de clases

Como he dicho anteriormente, este diagrama de clases representará la aplicación para esta primera versión. En futuras versiones, el diagrama aumentará, introduciendo nuevas entidades como “Empresa”, “Localidad” u otras que sean necesarias para futuros requisitos.

4.4.2. Diagrama de casos de uso

El diagrama de casos de uso es un tipo de diagrama cuyo principal objetivo es representar los distintos tipos de actores o roles tendrá la aplicación y cómo estos interactúan con el sistema. Es una buena forma para obtener requisitos del sistema.

En mi caso, he generado un único diagrama de casos de uso que engloba todas las interacciones del usuario en el sistema.

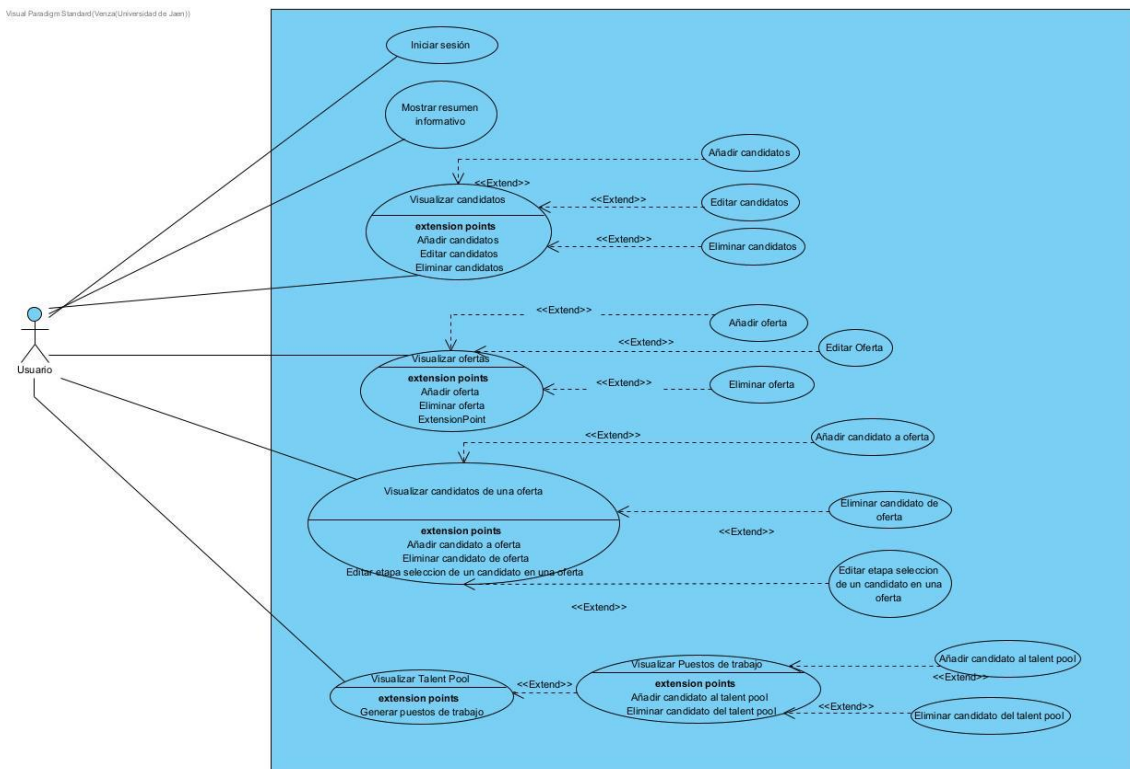


Ilustración 23. Diagrama de casos de uso

Tal y como comentamos en la etapa de definición de requisitos, las principales funcionalidades que podrá realizar el usuario en el sistema son:

- Inicio de sesión

- Mostrar resumen informativo
- Visualizar/Añadir/Editar/Eliminar candidatos
- Visualizar/Añadir/Editar/Eliminar ofertas
- Visualizar/Añadir/Eliminar candidatos de una oferta
- Actualizar estado de un candidato en una oferta (Envío de correos automático)
- Visualizar/Añadir/Editar/Eliminar trabajos en el talent pool
- Visualizar/Añadir/Eliminar candidatos de una trabajos en el talent pool

4.4.3. Diagrama de secuencia

Los diagramas de secuencia son un tipo de diagrama que ilustra la forma en que un sistema interactúa internamente para realizar una funcionalidad.

A continuación se van a mostrar y explicar cada uno de los diagramas de secuencia generados.

4.4.3.1. Diagrama de secuencia de Inicio de Sesión

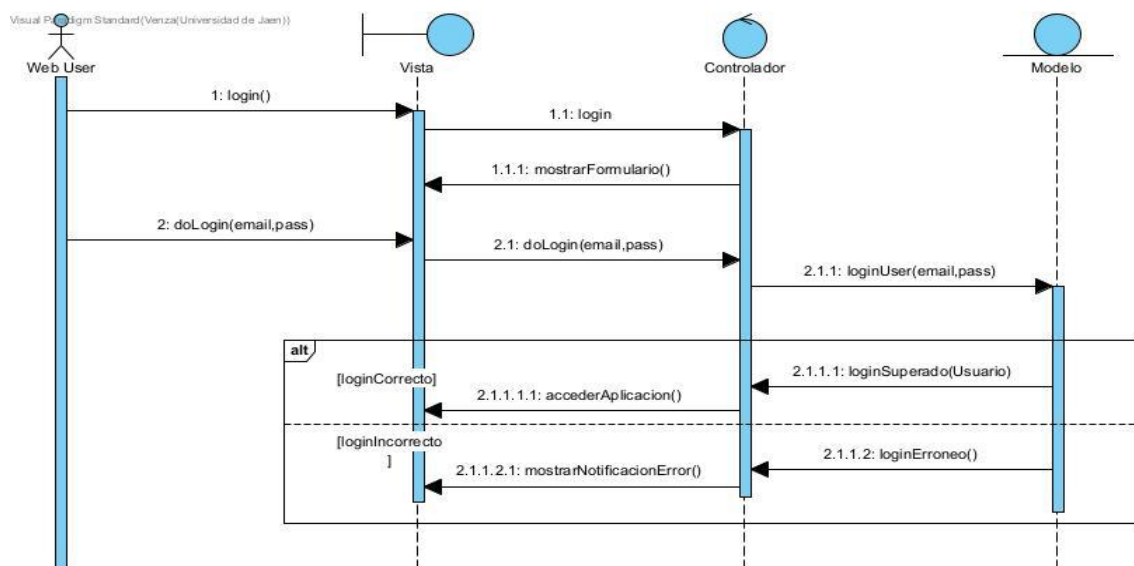


Ilustración 24. Diagrama de secuencia de Inicio de Sesión

Como comentamos anteriormente en este trabajo, la aplicación a desarrollar no tiene un apartado de registro en la aplicación. La única forma para acceder a esta aplicación es mediante unas credenciales generadas por un administrador. En el momento que tengamos esas credenciales podremos realizar la interacción que se explica en este diagrama de secuencia.

El usuario accede a la aplicación mediante la dirección web asociada. En ese momento, el formulario de login es mostrado. Mediante este formulario el usuario podrá introducir las credenciales y pulsar en iniciar sesión. Si el login es correcto, se le dará acceso a la aplicación, mostrando la vista principal. Además, se almacenará en el localStorage el token obtenido en este inicio de sesión. Mediante este token podremos realizar las peticiones a nuestra API. Si en algún momento perdemos este token o caduca, deberemos de volver a realizar el inicio de sesión. Si por el contrario el login ha sido erróneo, se mostrará una notificación de error en la parte inferior izquierda, indicando que hay algún problema y que será necesario volver a introducirlas.

En esta primera versión, se realizará una validación total a nivel de Back-end, pero en la vista solo se mostrará este error general. En futuras versiones, en este diagrama de secuencia se mostrarán distintas opciones de error, para que en la vista podamos tener un control mayor de errores y comunicación con el usuario.

4.4.3.2. Diagrama de secuencia de candidatos

- Visualizar candidato:

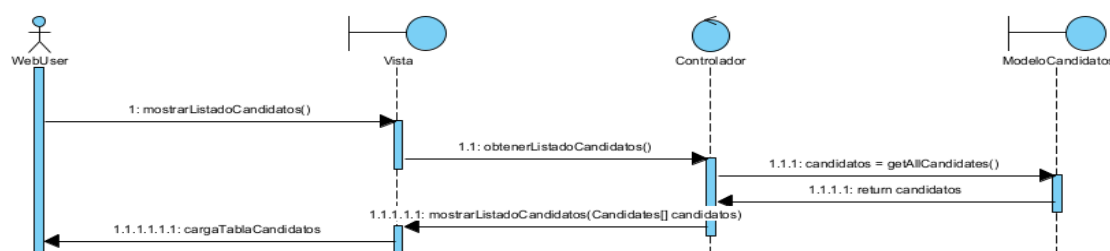


Ilustración 25. Diagrama de secuencia de Candidatos (Visualizar)

El usuario, al acceder a la vista Candidatos, se realizará este proceso en el sistema, para cargar los datos de los candidatos en la tabla que el usuario visualizará.

- Añadir candidato:

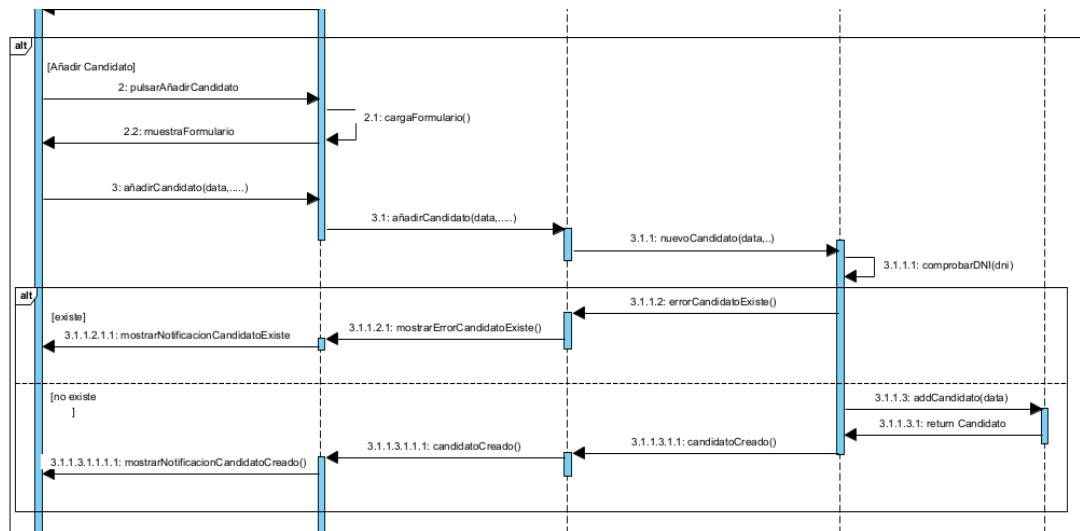
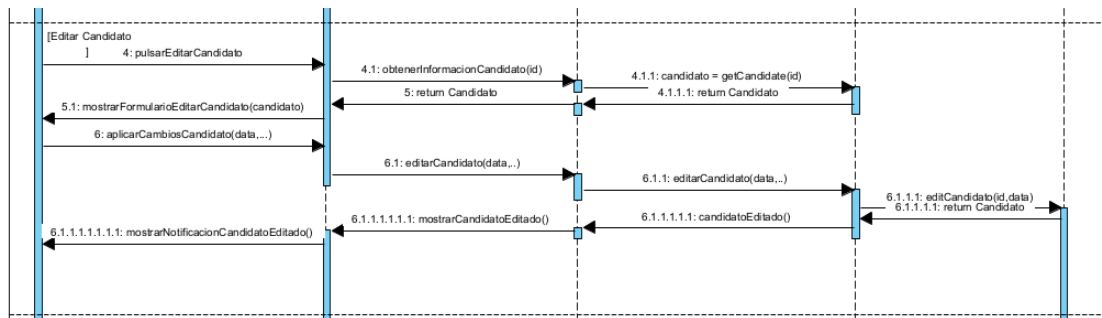
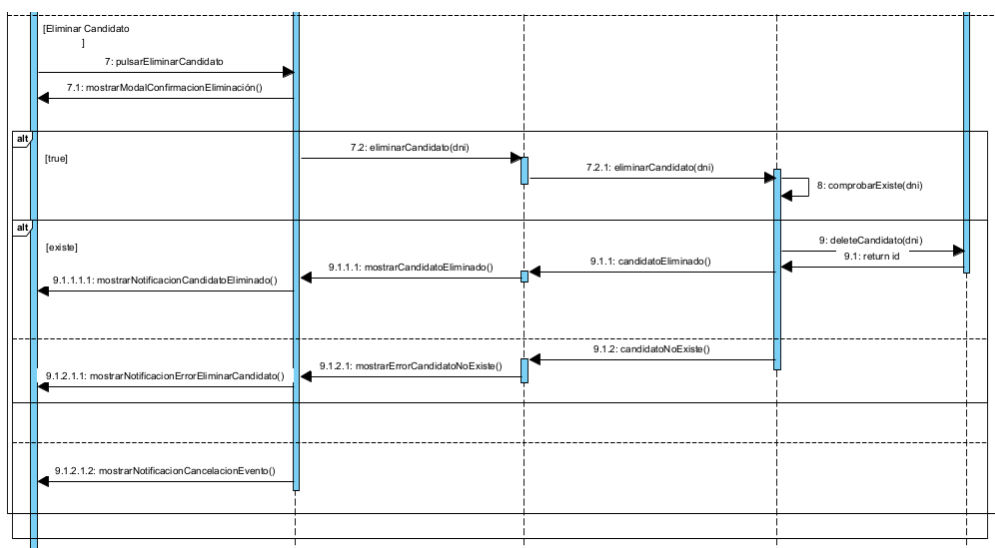


Ilustración 26. Diagrama de secuencia de Candidatos (Añadir)

El usuario, al pulsar el botón “Añadir candidato” de la barra superior, mostrará un modal con un formulario. Si el usuario rellena este formulario y pulsa el botón enviar, el sistema realizará el proceso de añadir. En este proceso, se comprueba si el dni aportado por el usuario ya existe o no en el sistema. Si no existe se añade y mostraremos una notificación de “Candidato añadido correctamente”. Por el contrario, si este usuario no existe o ha ocurrido algún posible error en el proceso, se mostrará una notificación de “Error al añadir candidato”.

- Editar candidato:**Ilustración 27. Diagrama de secuencia de Candidatos (Editar)**

El usuario, de la misma forma que en el proceso de añadir candidato, pulsara en el botón editar, mostrando el mismo modal y formulario, pero esta vez se mostrarán los datos del candidato. Si editamos algún valor y pulsamos el botón editar, el sistema realizará el proceso de edición del candidato seleccionado.

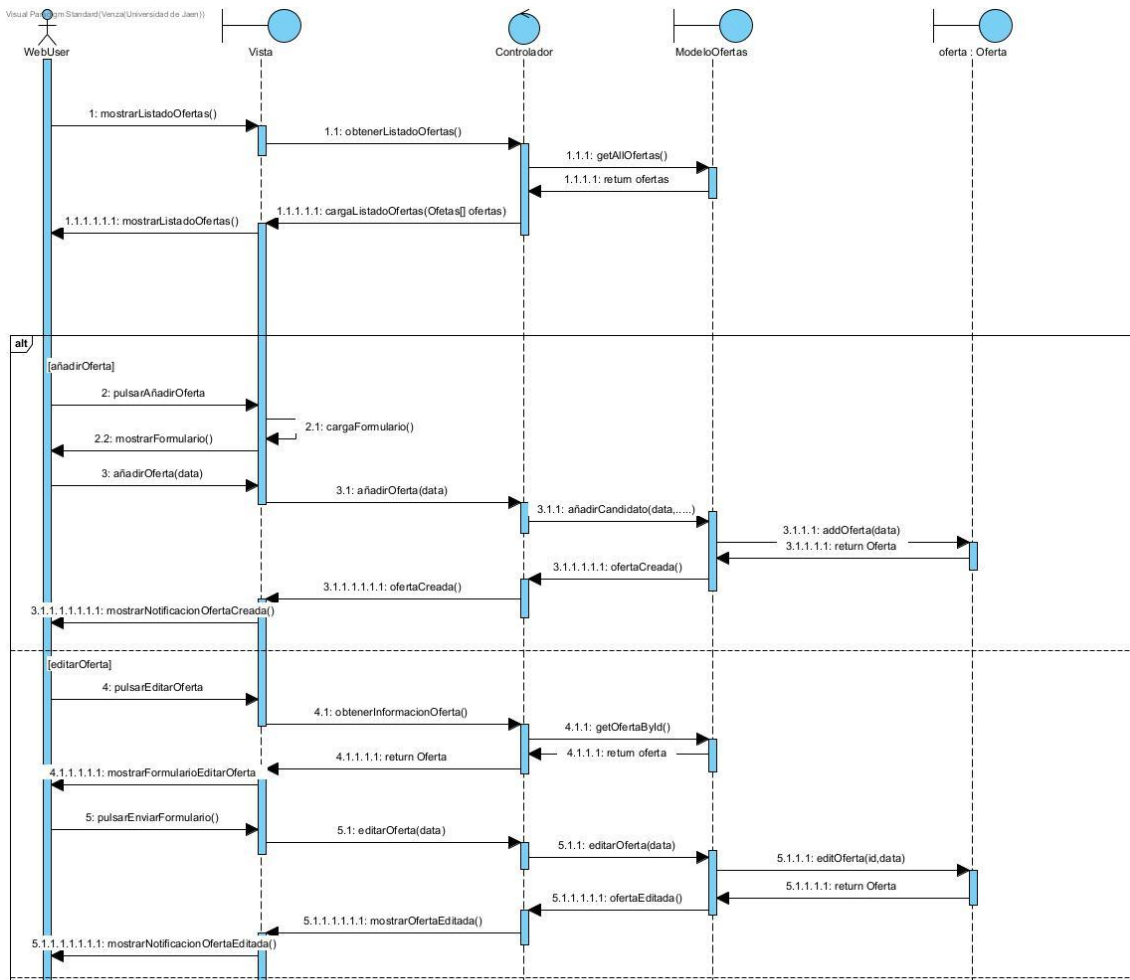
- Eliminar candidato:**Ilustración 28. Diagrama de secuencia de Candidatos (Eliminar)**

Si el usuario pulsa sobre el botón eliminar de alguna de las filas de la tabla candidatos, iniciará el proceso de eliminación de candidatos. En este proceso, en primer lugar, se mostrará un mensaje de confirmación de eliminación. Si éste confirma, el sistema se encargará de buscar el candidato y eliminarlo. Si el proceso finaliza correctamente, se mostrará un mensaje de

confirmación de eliminación o un mensaje de error si este proceso no finaliza sin errores. Si no confirmamos la eliminación, se mostrará un mensaje de confirmación de cancelación.

4.4.3.2. Diagrama de secuencia de ofertas

Este diagrama tiene una funcionalidad similar a la de candidatos, por lo que solo mostraré el diagrama.



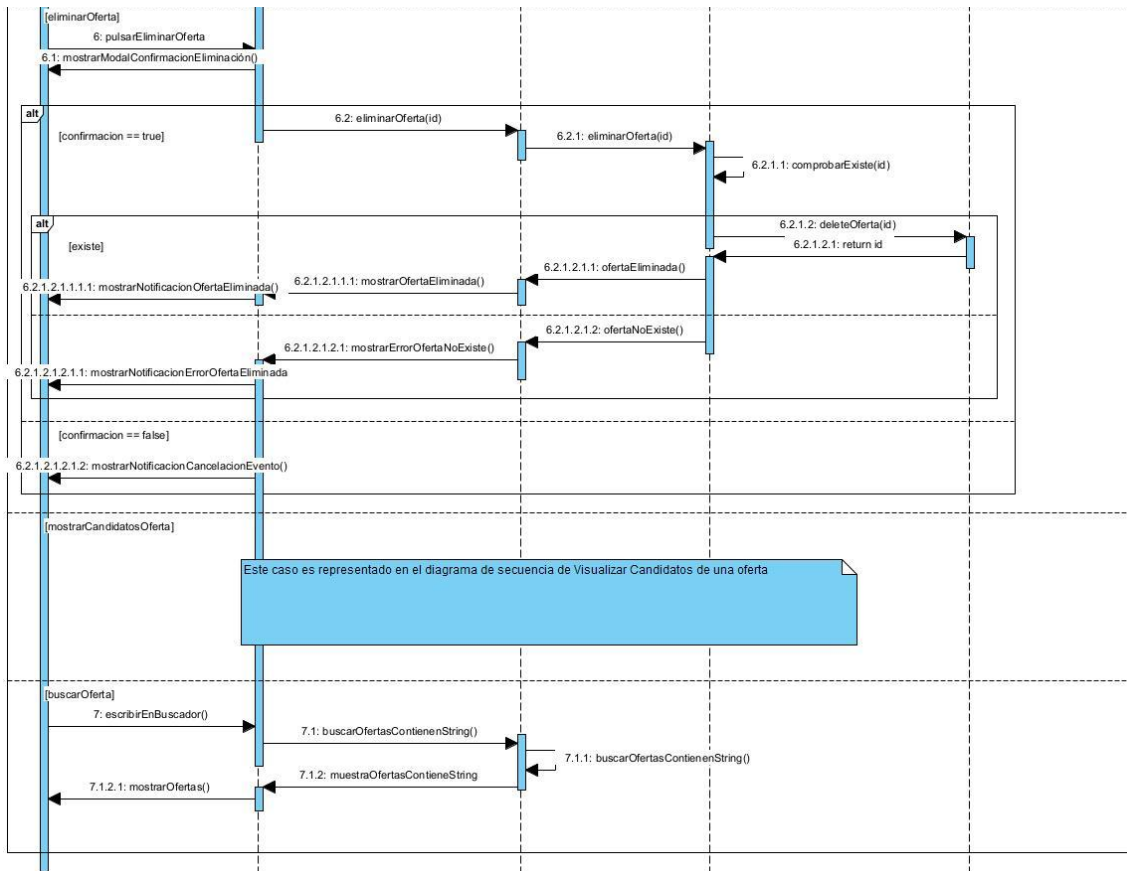
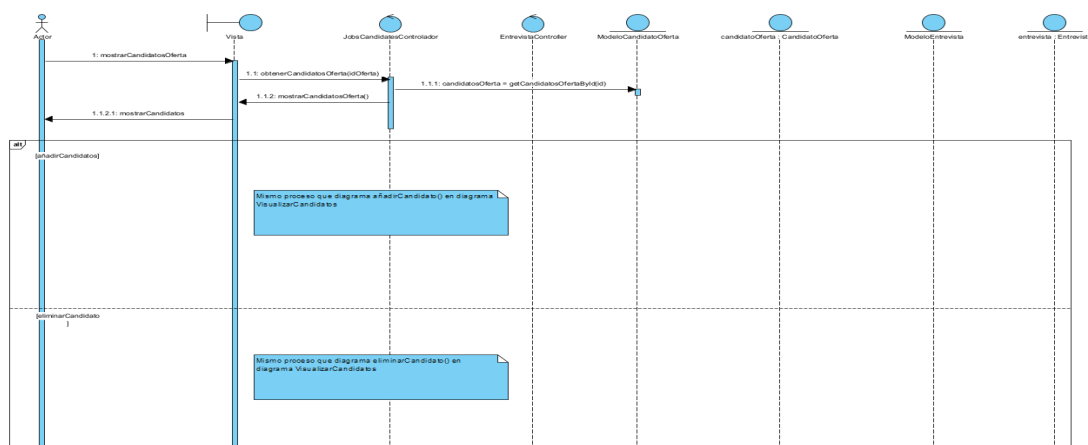


Ilustración 29. Diagrama de secuencia de Ofertas

4.4.3.3. Diagrama de secuencia del listado de candidatos para una oferta

Tanto el proceso de visualización como el de añadir y eliminar es similar a los diagramas anteriores, el único proceso a ser explicado será el de modificación de etapa del candidato



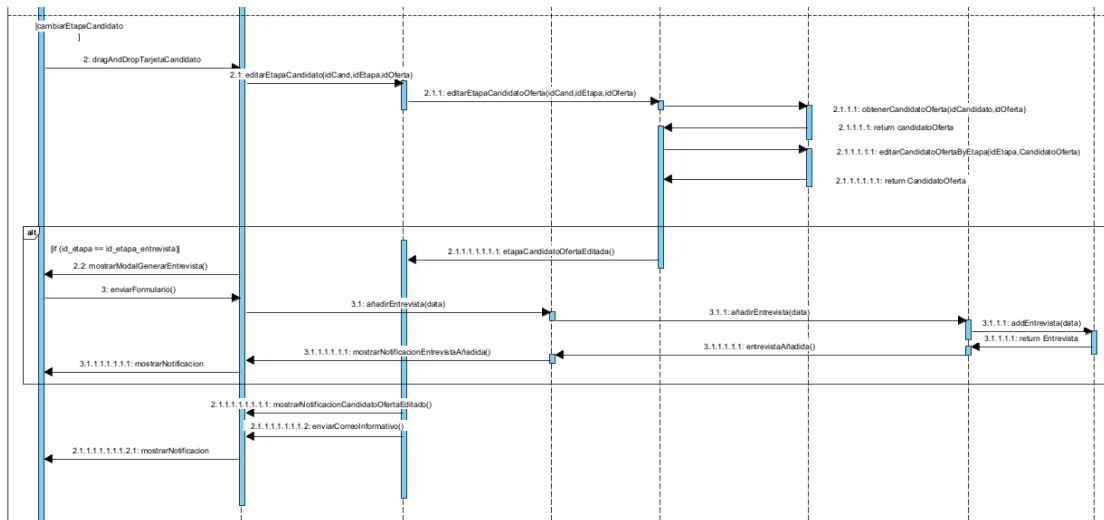
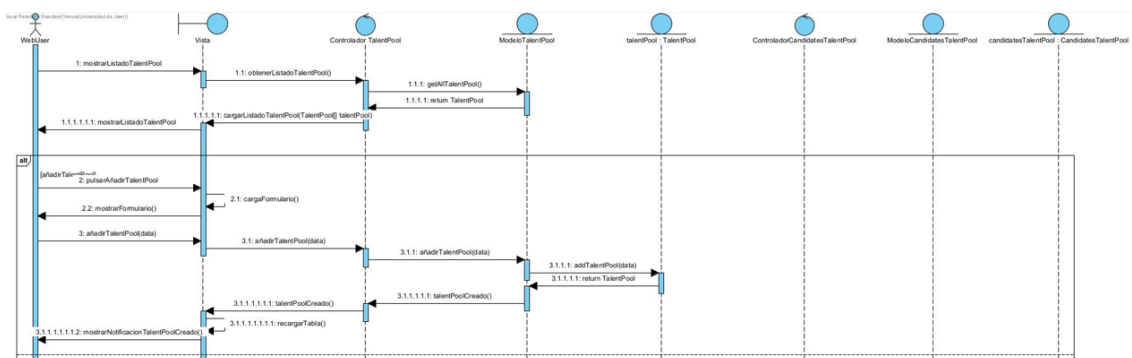


Ilustración 30. Diagrama de secuencia de candidatos de una oferta

El usuario, en primer lugar, cogerá una de las tarjetas mostrada en una de las columnas y realizará el proceso de Drag and Drop entre columnas. Al soltar la tarjeta, el sistema obtendrá el identificador del candidato en esa oferta y el identificador de la etapa a la que ha sido asignado, realizando el proceso de edición con la nueva etapa. Si este proceso ha concluido correctamente, la tarjeta aparecerá en la nueva columna de etapa. Por el contrario, si este proceso ha terminado mal, la tarjeta no se moverá de la columna.

Además, internamente, el sistema automáticamente mandará un correo al candidato para especificarle que ha avanzado o retrocedido en el proceso de selección.

4.4.3.4. Diagrama de secuencia del Talent Pool



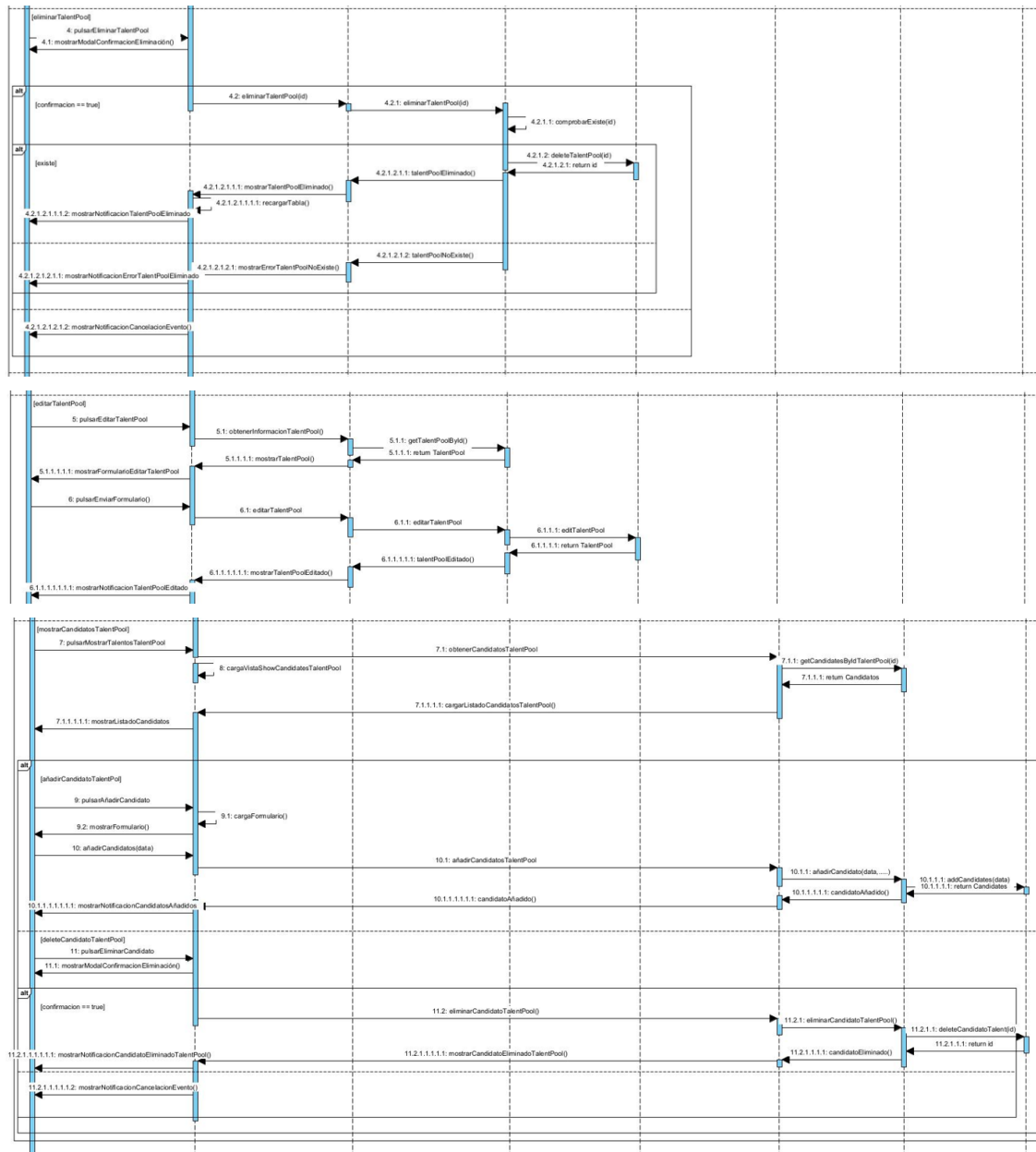


Ilustración 31. Diagrama de secuencia de talent pool

Este diagrama tiene todas las funcionalidades similares a los otros diagramas de secuencia comentados anteriormente (Añadir, eliminar, editar, borrar), por lo que omitiré su explicación para no ser repetitivo.

4.5. Diseño

Etapa del ciclo de vida del desarrollo de software en donde se crea la interfaz de usuario de la aplicación y se determinan todas las funcionalidades. Es decir, en esta etapa se va decidir la estructura que tendrá nuestra aplicación y la apariencia de la misma, para así evitar problemas de diseño, funcionalidad o estructura antes de realizar la implementación.

Para el desarrollo de aplicaciones, la etapa de diseño se subdivide en 3 fases.

4.5.1. Sketch

Un sketch es el primer boceto que recoge las ideas generales que tenemos sobre nuestra aplicación web. En esta etapa plasmamos las ideas que tenemos sobre el proyecto y responderemos a preguntas como, donde situaremos los elementos más importantes, donde estará la zona de navegación, que distribución tendrá nuestra aplicación, etc...

A continuación se van a mostrar una serie de imágenes con cada uno de los Sketch realizados para cada una de las vistas que tendrá la aplicación.

- Vista principal

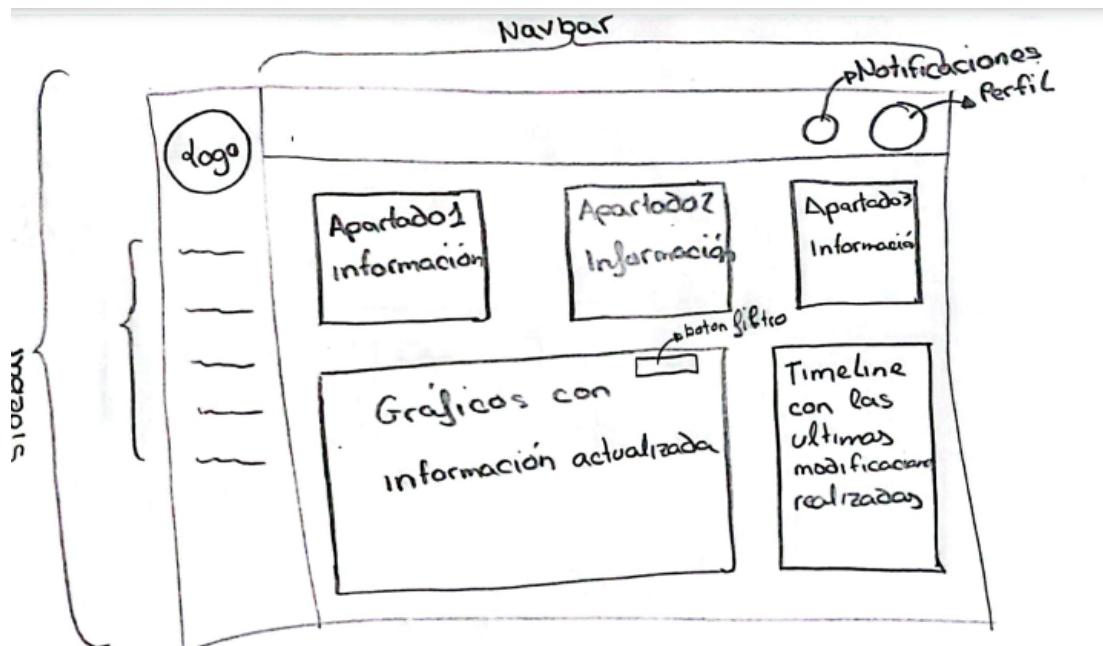


Ilustración 32. Sketch del dashboard/vista principal

La vista principal o dashboard, al igual que todas las vistas de la aplicación, tendrán su correspondiente menú lateral y menú superior. En el menú lateral nos encontramos botones para acceder a cada una de las vistas de la aplicación y el logo de la aplicación. En la barra superior aparecerán dos botones, uno para mostrar notificaciones y otro para editar nuestro perfil o cerrar sesión.

El dashboard está dividido en 2 partes. En la parte superior se mostrará la información más importante o relevante de la aplicación, es decir, el número de candidatos nuevos que han entrado, el número de ofertas de trabajo activas e información sobre el Talent Pool. En la parte inferior se mostrará un gráfico que representará visualmente más información sobre los candidatos y un timeline, en donde veremos el historial de operaciones realizadas en la aplicación, como puede ser añadir un nuevo candidato, modificación del mismo, etc...

- Vista Información candidatos

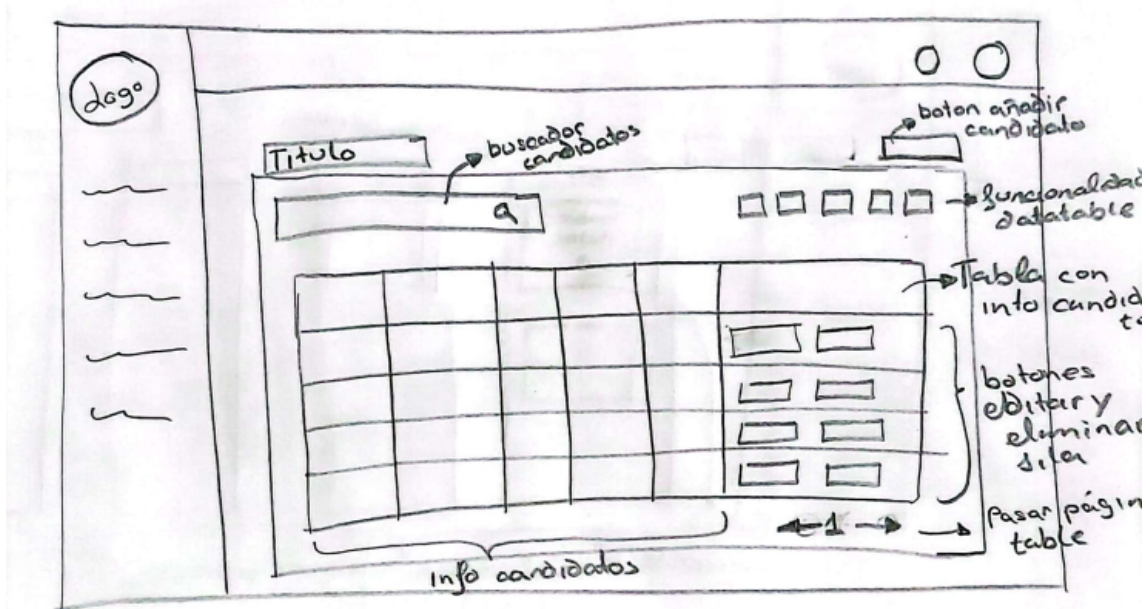


Ilustración 33. Sketch de la vista candidatos

En la vista candidatos se mostrará una tabla con la información de cada uno de los candidatos que tenemos almacenados en nuestra BBDD. Esta tabla tendrá varias columnas para mostrar la información más relevante del candidato. En la última columna de la tabla se mostrarán dos botones, el primero para editar/eliminar un candidato y otro para mostrar toda la información del candidato. Tendremos un paginador de tabla para poder pasar entre las distintas páginas de la tabla y un selector del número de filas a mostrar por página.

En la parte superior de la tabla aparecerá un botón para añadir candidatos manualmente y un buscador de candidatos.

- Vista ofertas de trabajo

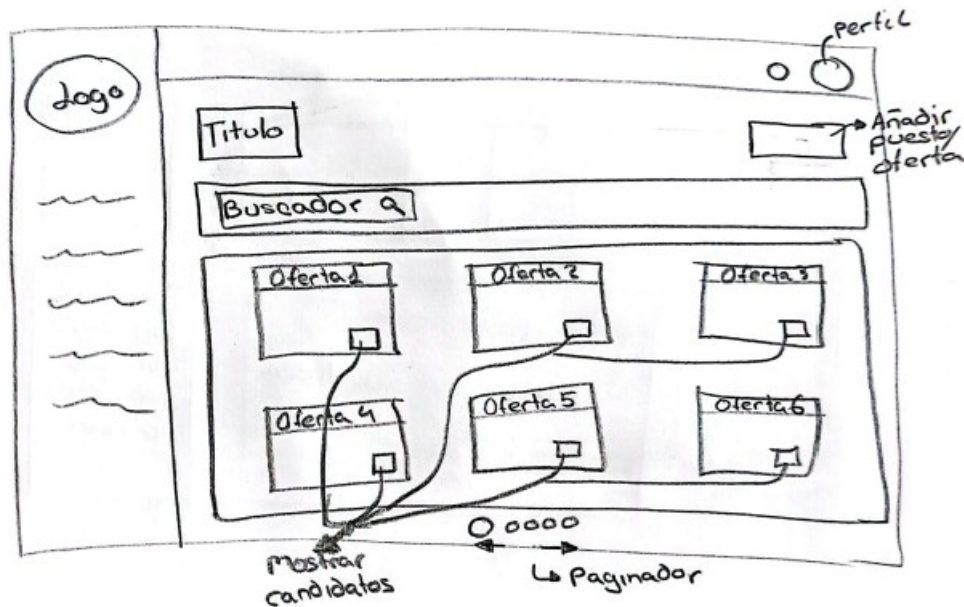


Ilustración 34. Sketch de la vista ofertas de trabajo

La vista ofertas de trabajo tiene una distribución similar a la de candidatos. Tenemos una primera parte superior donde aparecen un buscador de ofertas y un botón para añadir nuevas ofertas y una parte inferior donde mostraremos tarjetas por cada una de las ofertas de trabajo.

Cada tarjeta representa una oferta de trabajo. En ella aparecerá una imagen o icono representativo a la oferta, un título y una breve descripción. Finalmente aparecerá un botón para mostrar todos los candidatos inscritos a esta oferta de trabajo y en la etapa del proceso de selección que ocupa.

- Vista etapas proceso selección

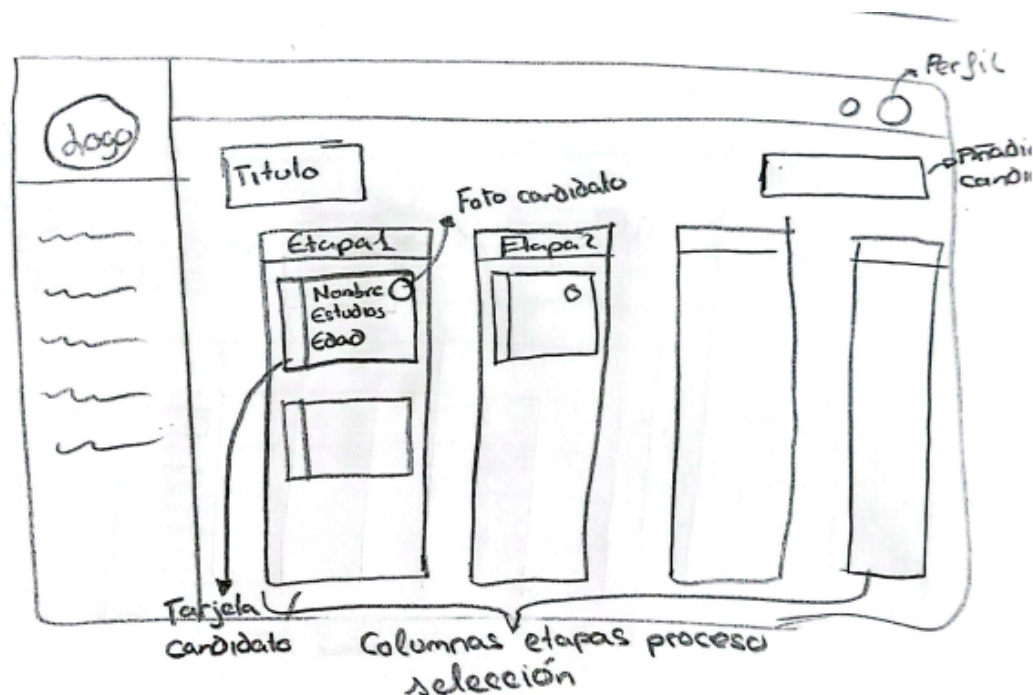


Ilustración 35. Sketch de la vista etapas del proceso de selección

La vista Etapa del proceso de selección aparece al pulsar el botón “Mostrar candidatos” explicado anteriormente. En esta vista se muestra un tablero tipo Kanban en donde se mostrarán varias columnas, cada una representando una etapa del proceso de selección.

Cada una de las columnas podrá tener o no una o varias tarjetas, que representarán a cada uno de los candidatos inscritos en esa oferta. Se podrán mover a los candidatos entre las distintas etapas arrastrando las tarjetas entre las distintas columnas.

Además, como en las vistas anteriores, existe un botón en la parte superior para añadir a un candidato a esa oferta de forma manual.

- Vista Talent Pool

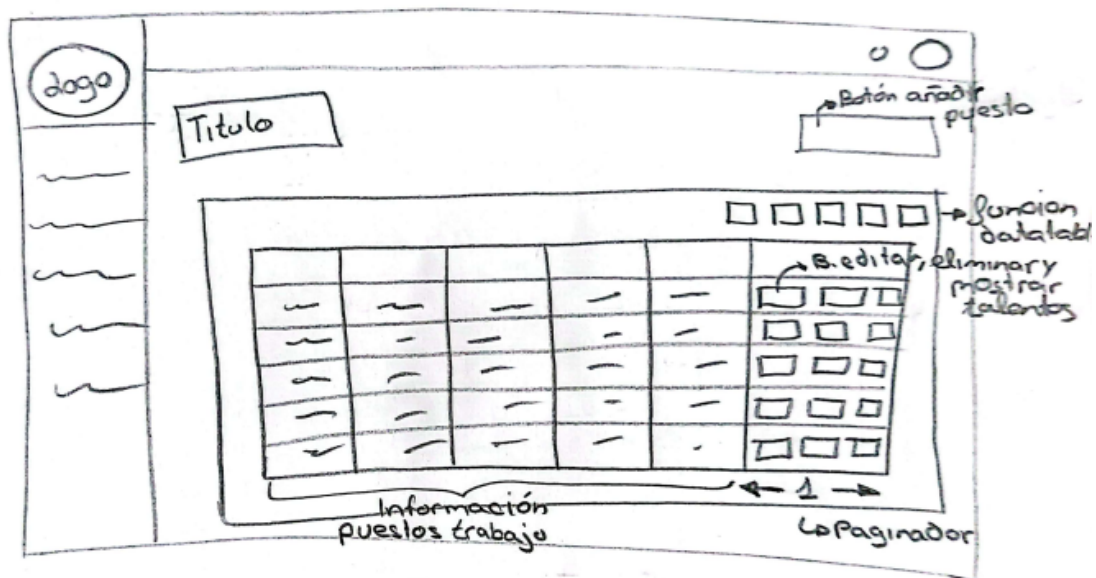


Ilustración 36. Sketch de la vista talent pool

En la vista Talent Pool se muestra una tabla similar a la de candidatos en donde mostraremos los distintos puestos de trabajo que tenemos en la empresa. Mediante los botones generados en la última columna podremos añadir, eliminar o mostrar los talentos de cada uno de los puestos de trabajo.

4.5.2. Wireframe

Un wireframe es una representación visual de cada una de las distintas vistas de una aplicación web, en donde se establecerá de una forma más seria la estructura y colocación de cada uno de los elementos que componen una vista.

La diferencia entre un sketch y un wireframe es que en los sketch plasmamos la idea inicial, mientras que en wireframe ya comenzamos a ordenar nuestras ideas, realizando un diseño más ordenado y definitivo.

A continuación se va a mostrar cada uno de los wireframe realizados para cada una de las vistas.

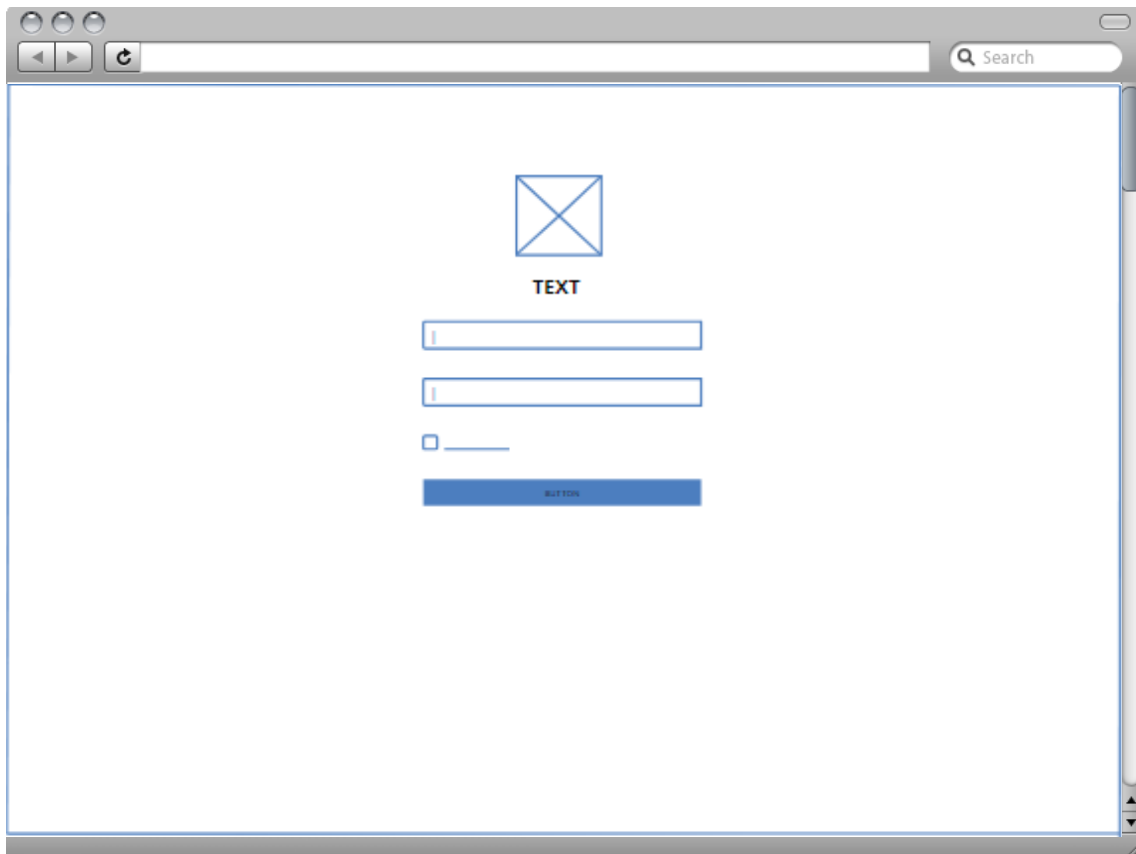


Ilustración 37. Wireframe pantalla de inicio

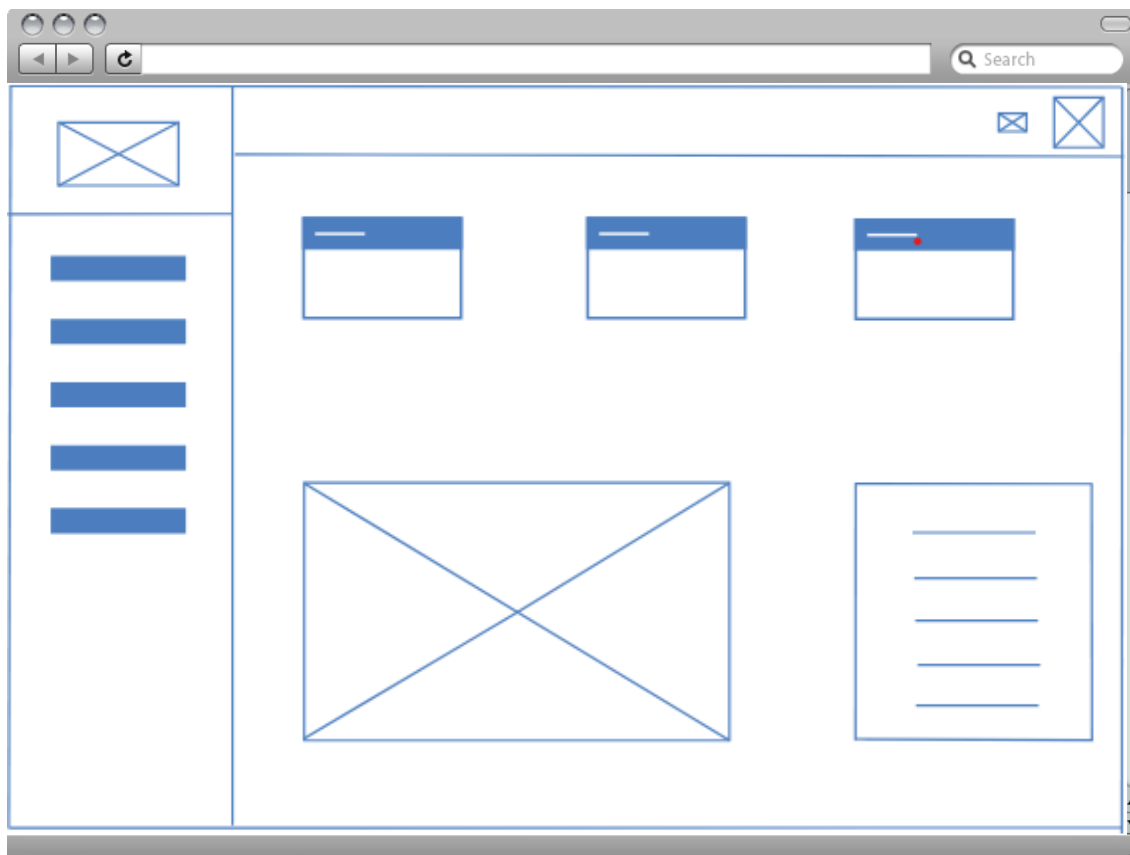


Ilustración 38. Wireframe del panel/vista principal

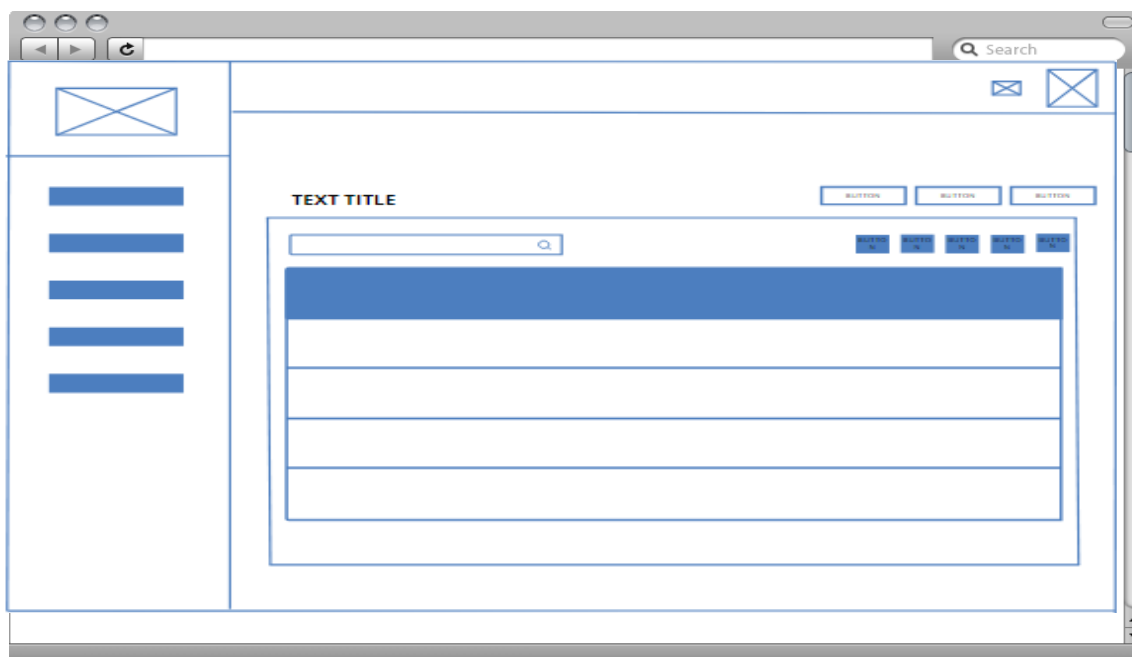


Ilustración 39. Wireframe de la vista de listado de candidatos

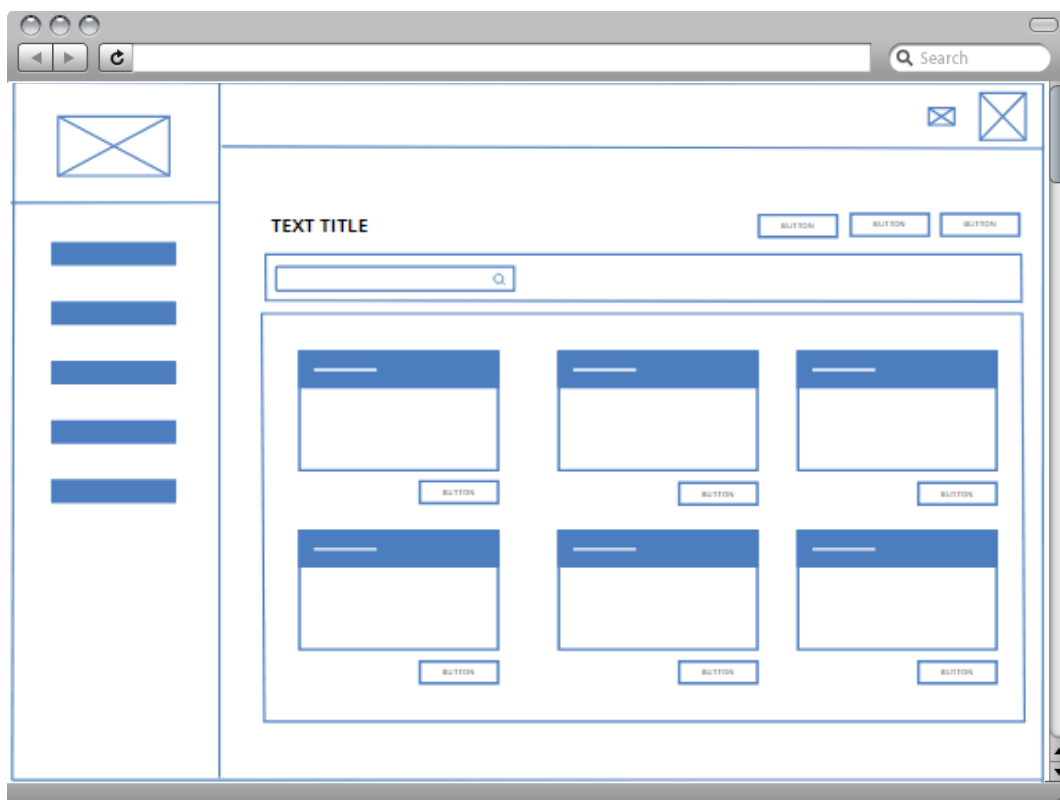


Ilustración 40. Wireframe de la vista de ofertas

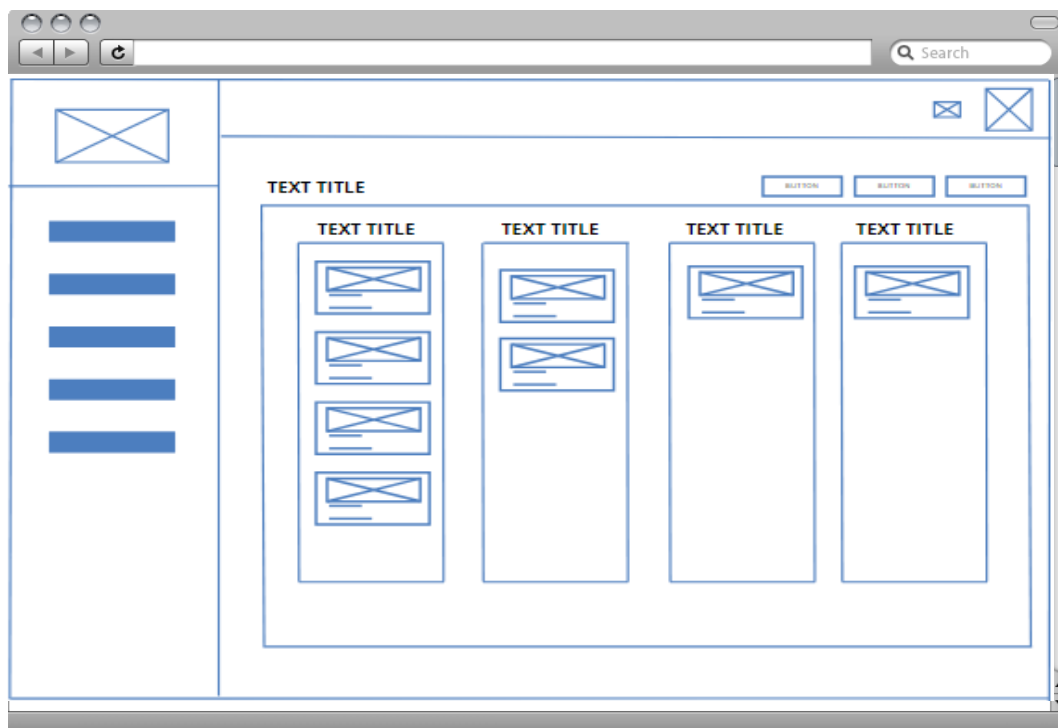


Ilustración 41. Wireframe de la vista para modificar estado de candidatos en una oferta de empleo

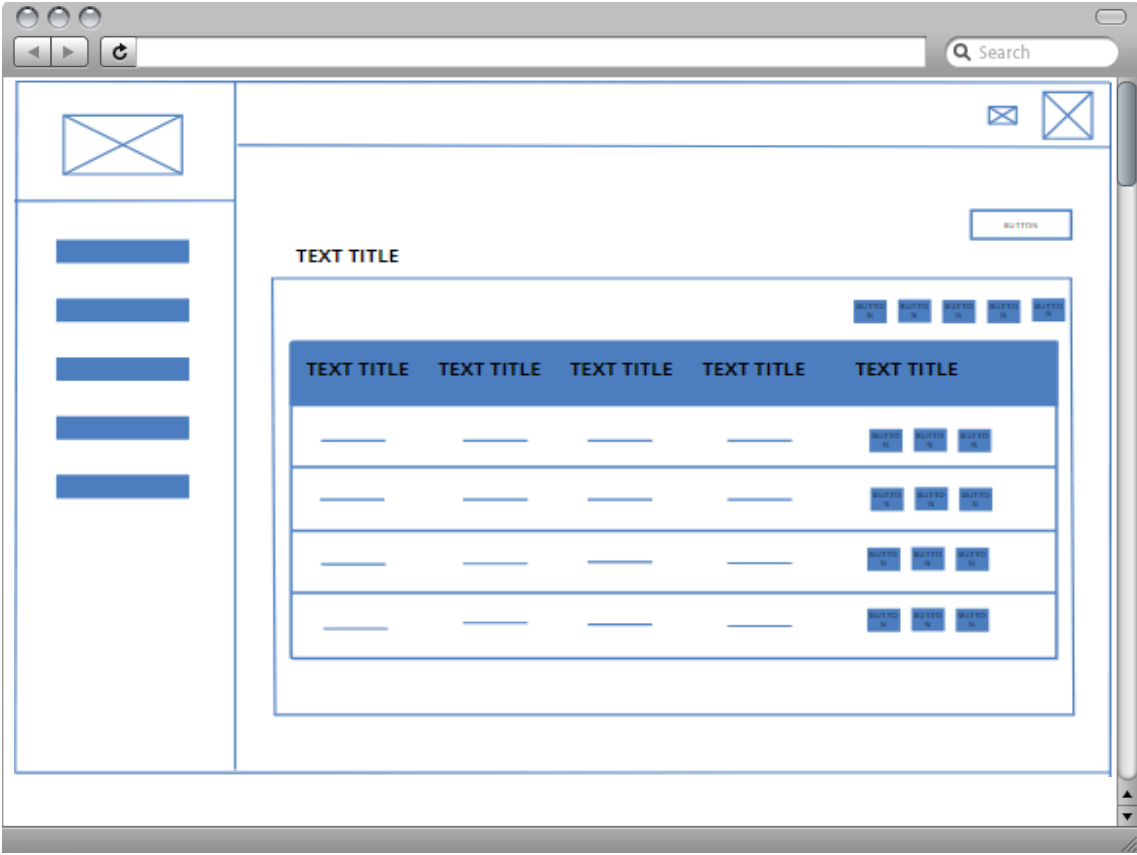


Ilustración 42. Wireframe de la vista para gestionar el talent pool

4.5.3. Mockup

Un mockup es un diseño más avanzado de la aplicación web que se suele utilizar para mostrar a los clientes. Se diferencian de los wireframe en que se realiza un diseño con muchísimo detalle, viendo ya imágenes, textos reales, paletas de colores, iconos,...

Existen dos posibilidades, realizar unos mockups totales, es decir, que ya serían casi definitivos y lo más parecidos posibles a la solución final, o realizar unos mockups parciales, dejando la posibilidad de realizar una cuarta fase para realizar un prototipo de la aplicación. Debido a temas de tiempo, se ha decidido realizar solamente los mockups completos de la aplicación, obviando la realización de un prototipo de la aplicación.

A continuación se van a mostrar cada una de las vistas con su diseño mockup.

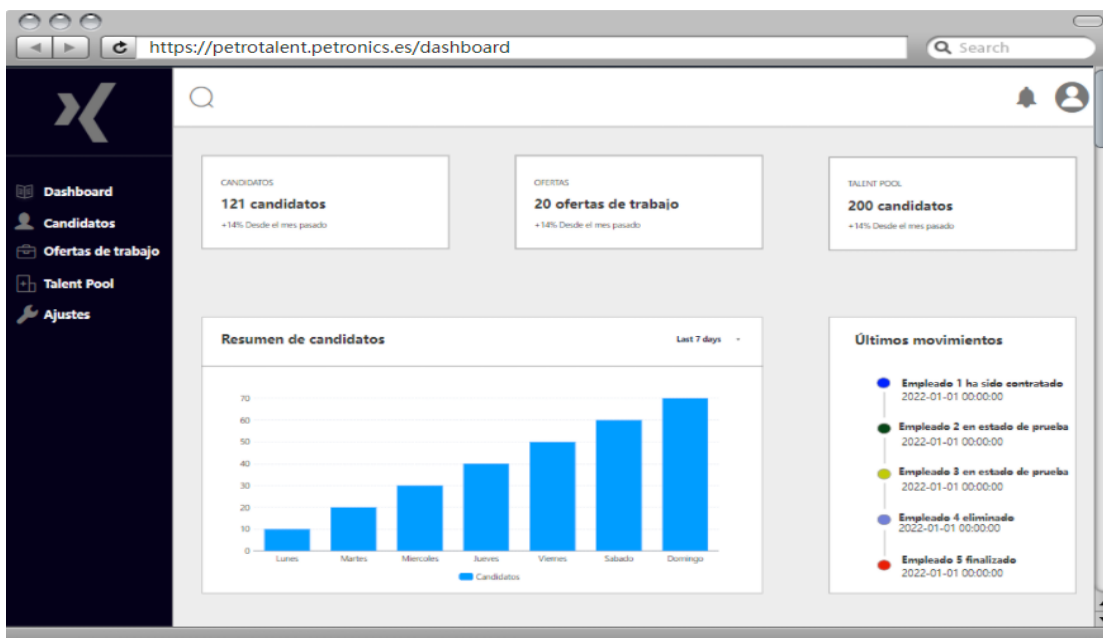


Ilustración 43. Mock Up del panel/vista principal

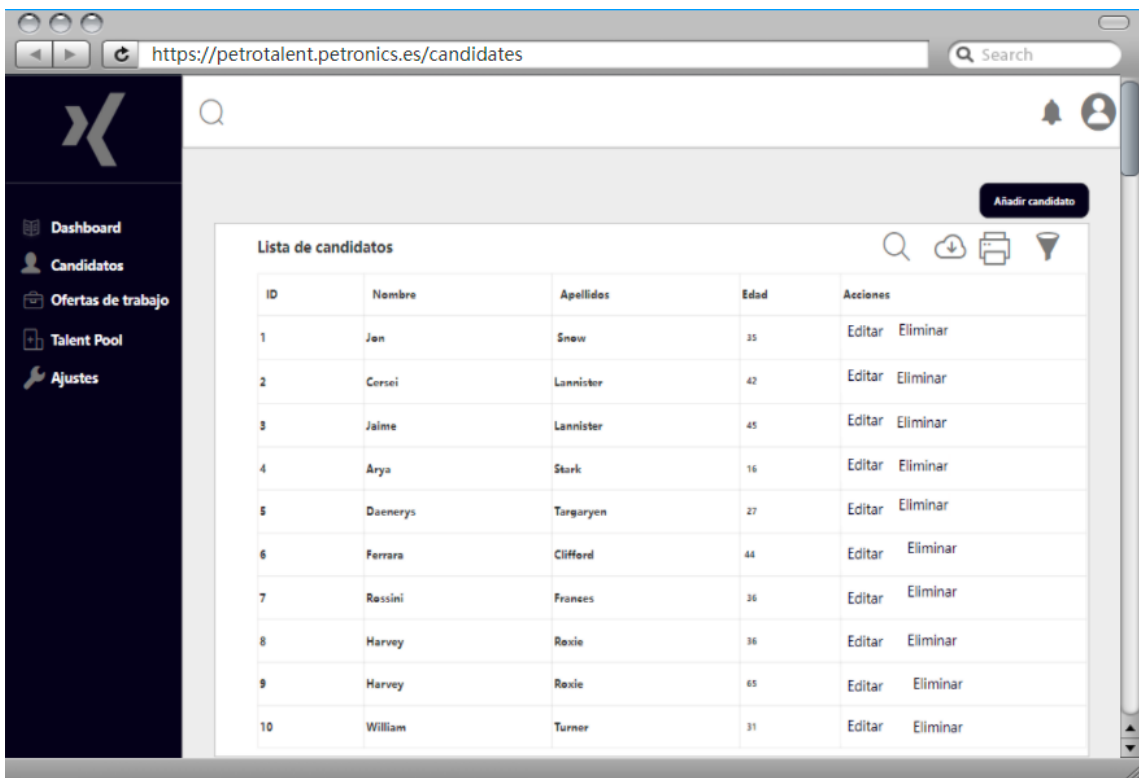


Ilustración 44 . Mock Up de la vista de listado de candidatos



Ilustración 45 . Mock Up de la vista con las ofertas de empleo

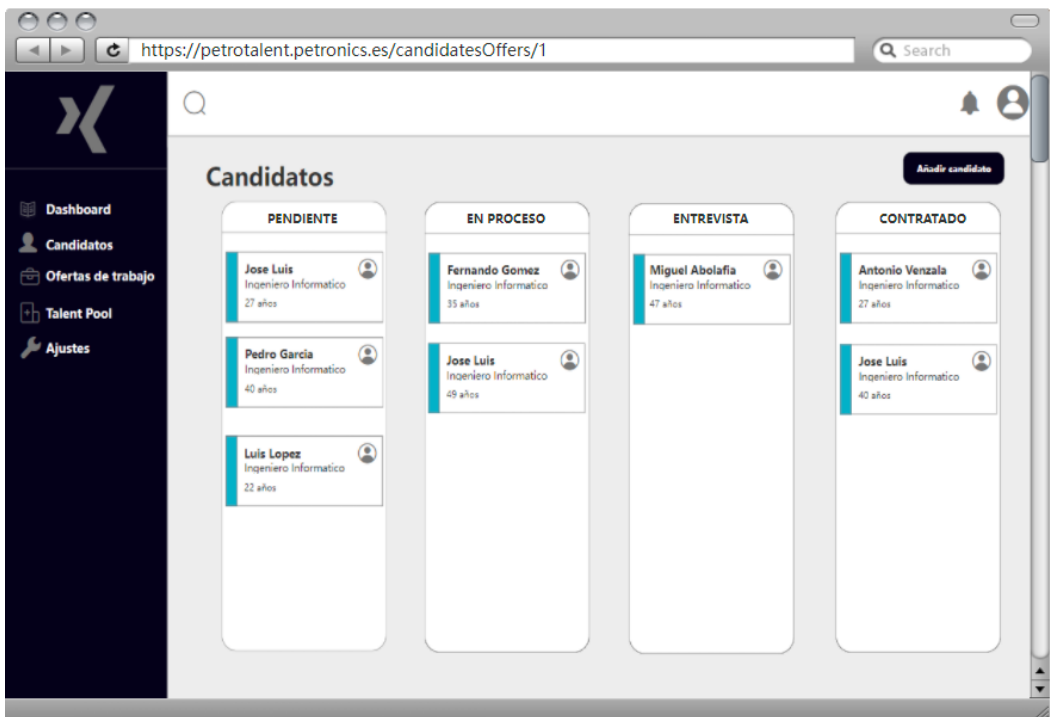


Ilustración 46 . Mock Up de la vista para modificar estado de candidatos en una oferta de empleo

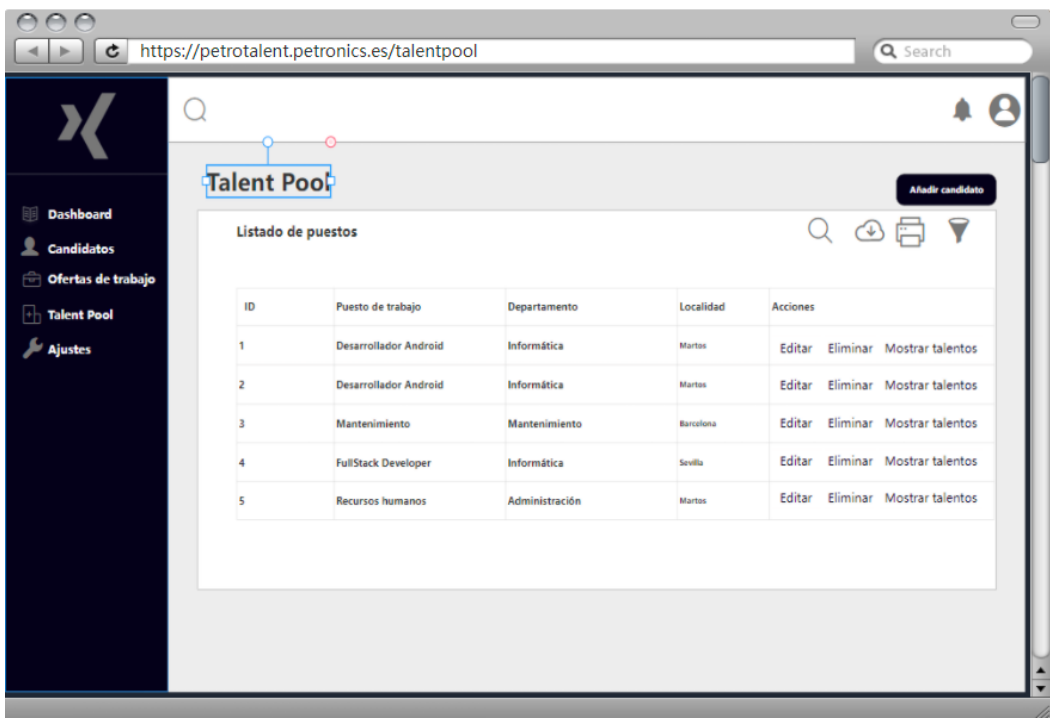


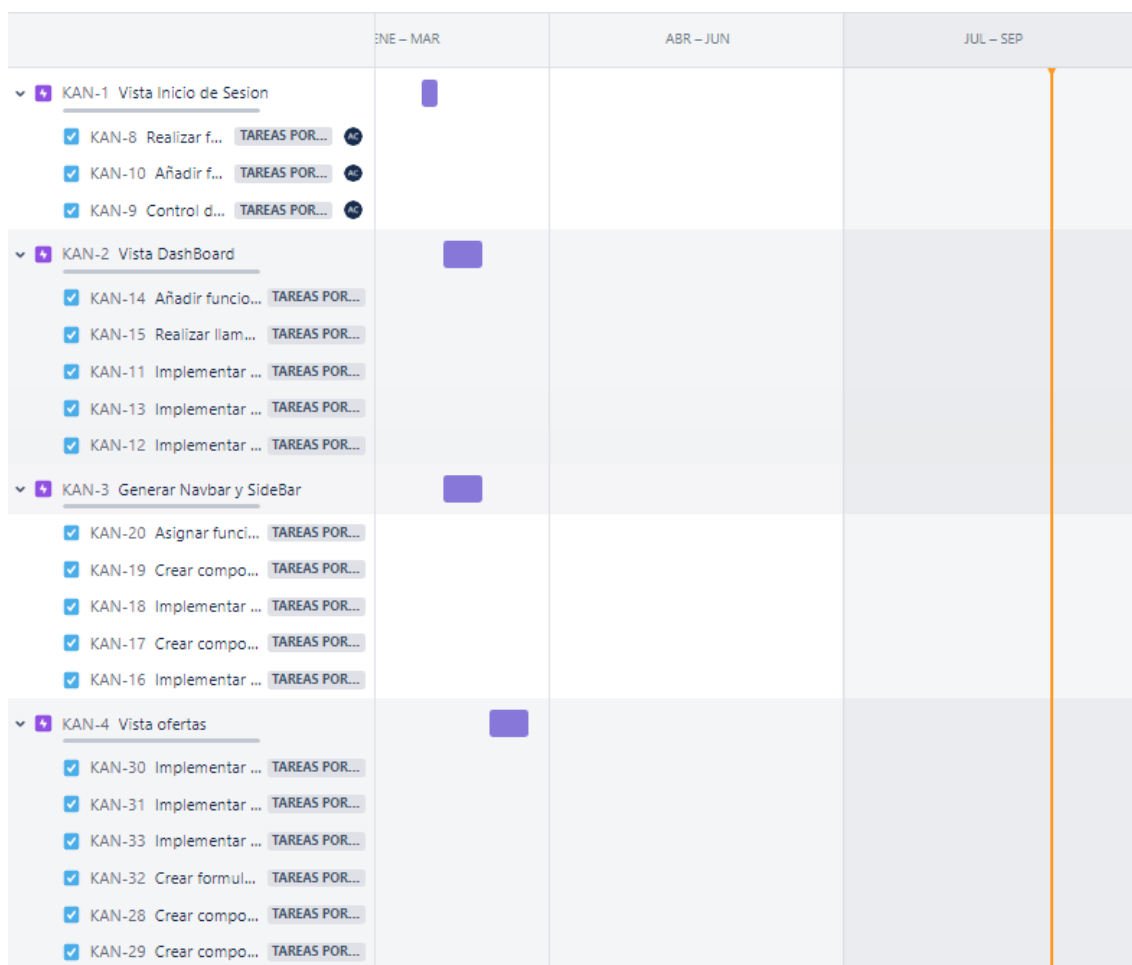
Ilustración 47 . Mock Up de la vista para gestionar el talent pool

4.6. Implementación

Etapa del ciclo de vida del desarrollo de software en donde se realiza el desarrollo de la aplicación, mediante el uso de las técnicas de desarrollo comentadas en apartados anteriores.

Este apartado va a ser dividido en distintos sub-apartados, coincidiendo cada uno de ellos con un Sprint. De esta manera se observará detalladamente cómo ha ido evolucionando el proyecto y la aplicación de la metodología SCRUM.

A continuación, se mostrará el cronograma inicial de la etapa de implementación, en donde aparecerán reflejados los sprints del desarrollo, tanto de la parte Front-End, como la de Back-End.



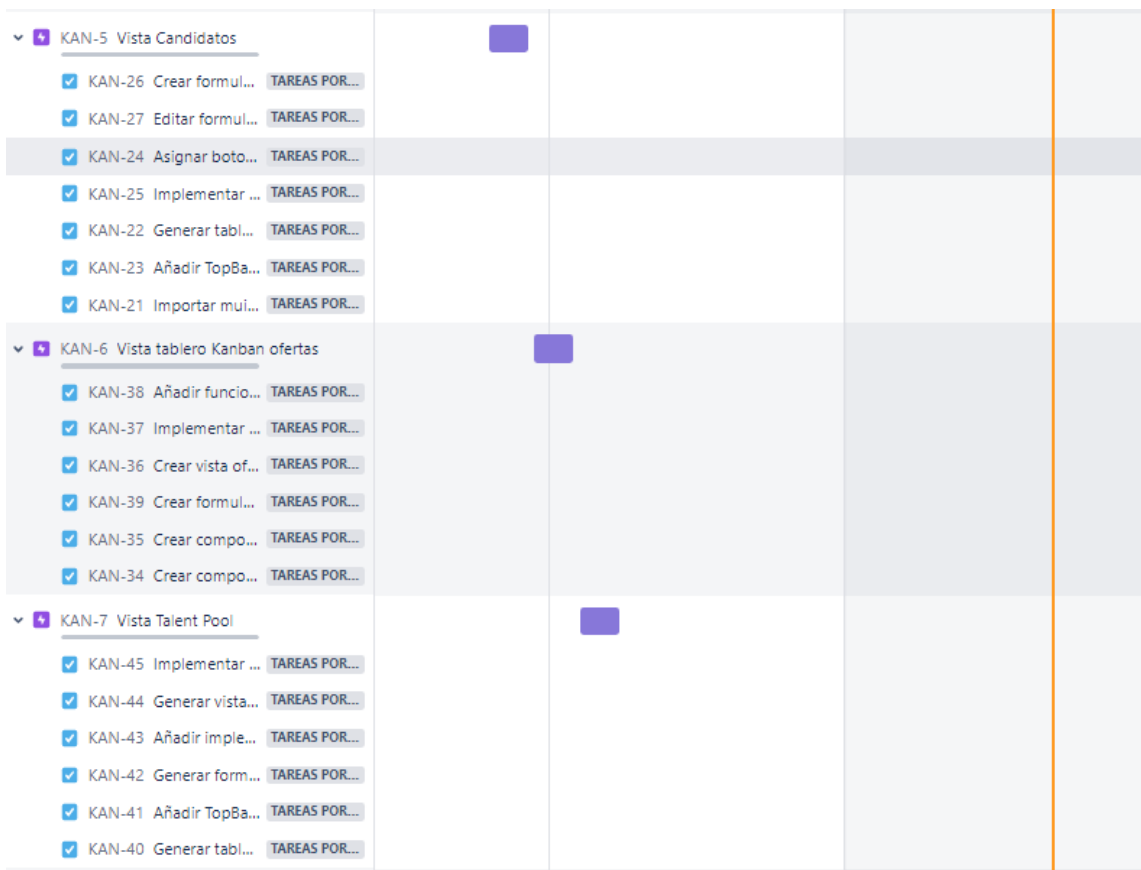
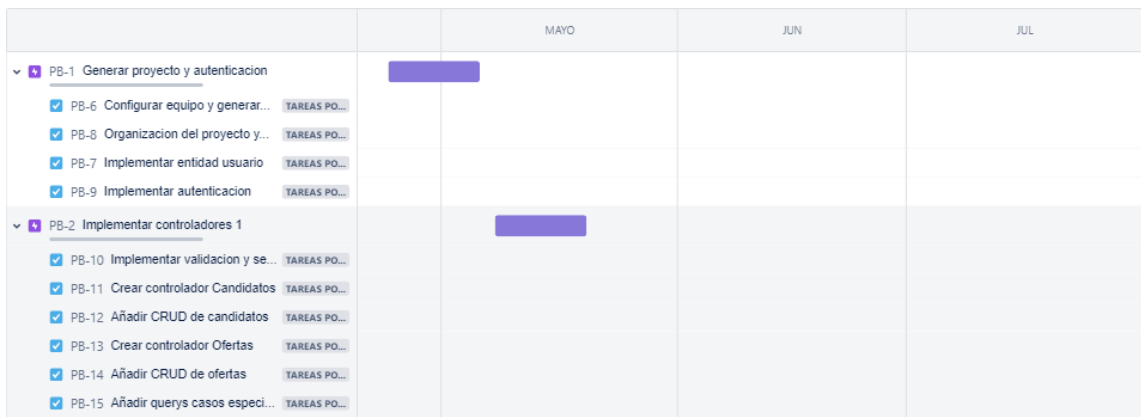


Ilustración 48 . Cronograma inicial del desarrollo Front-End



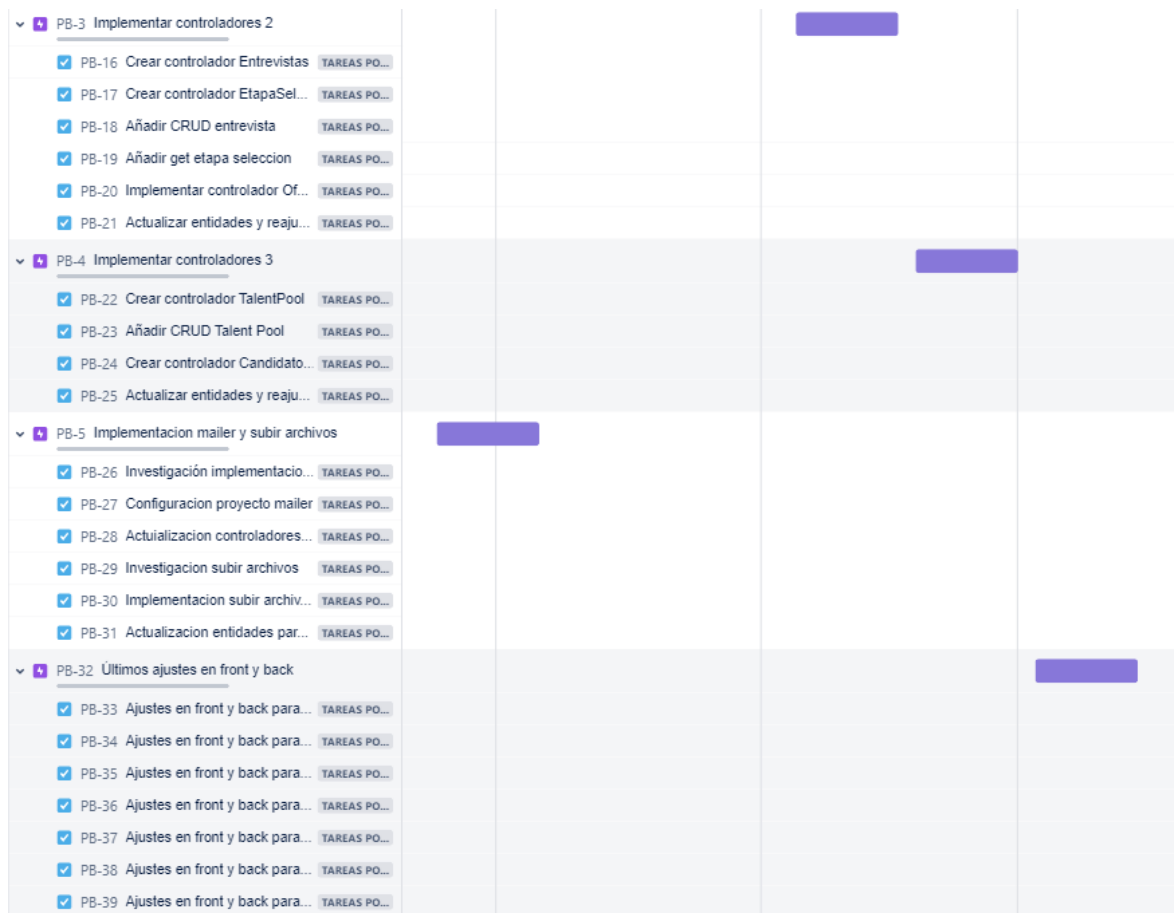


Ilustración 49 . Cronograma inicial del desarrollo Back-End

4.6.1. Sprint 2 - Desarrollo de la vista inicio de sesión

La implementación de la aplicación comenzó la segunda semana del sprint 2, justo después de realizar el diseño y puesta en común con el equipo de recursos humanos en una reunión que tuvo lugar el día 17 de febrero, siendo este diseño validado.

Justo al principio del desarrollo, se realizó la revisión del Product Backlog, obteniendo que para este Sprint había que realizar las tareas asociadas al desarrollo de la vista de inicio de sesión.

A mitad del Sprint el tablero estaba en este estado:

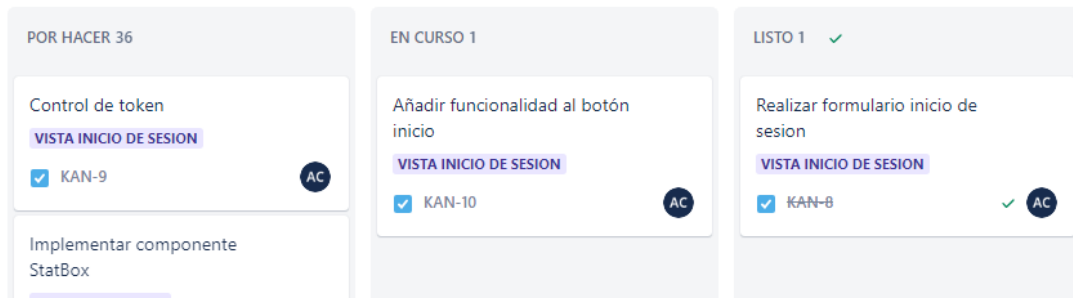


Ilustración 50 . Tablero a mitad Sprint 2

Como se puede observar, ya se había generado el proyecto inicial y terminado el formulario de inicio de sesión, teniendo a medias la finalización de la implementación del botón de inicio de sesión (Al no tener la API terminada, la petición se dejó comentada hasta que estuviese implementado).

Al finalizar el Sprint el tablero quedó en este estado:

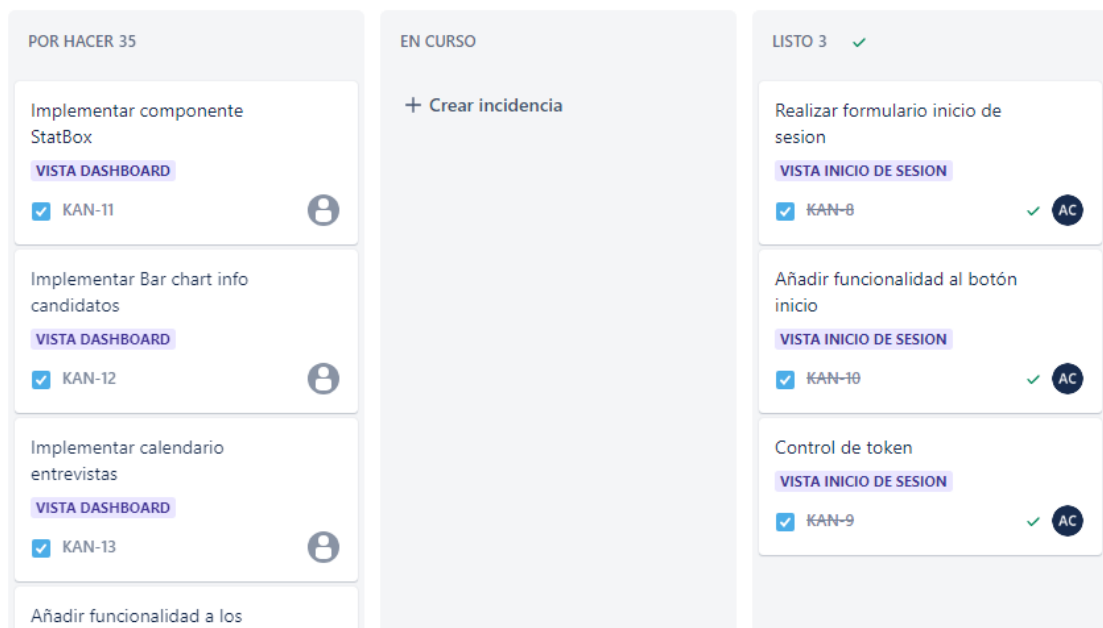
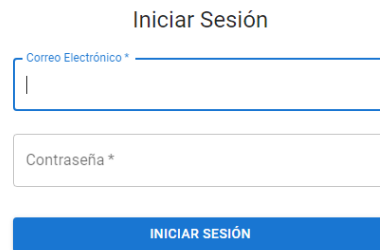


Ilustración 51 . Tablero a final Sprint 2

En el sprint review, al haber ya mostrado los diseños y validados en una reunión a mitad del sprint, no se pudo mostrar un gran incremento, debido a que la aplicación en este punto era solo un formulario de inicio de sesión. Este

fue validado, por lo que no aparecieron nuevas tareas para añadir en el Product Backlog y no hubo que realizar un Sprint Refinement.

A continuación se muestra una imagen con el formulario de inicio de sesión desarrollado.



El formulario de inicio de sesión (Login) tiene el título "Iniciar Sesión" centrado. Debajo del título hay dos campos de entrada de texto: el primero está etiquetado como "Correo Electrónico *" y el segundo como "Contraseña *". Ambos campos tienen un borde gris y un cursor de texto visible. Debajo de los campos hay un botón rectangular azul con el texto "INICIAR SESIÓN" en mayúsculas blancas.

Ilustración 52 . Imagen formulario Login

4.6.2. Sprint 3 - Desarrollo del dashboard y menubars

Justo al comenzar el Sprint se realizó el Sprint Planning, en donde se seleccionaron para desarrollar las tareas del las Epic Vista Dashboard y generar Navbar y Sidebar.

Por ello, aunque en los diagramas no aparezca una columna Sprint Backlog, supondremos que las tareas de estas épicas están en esta columna.

En la primera semana de desarrollo se consiguió realizar las siguientes implementaciones.

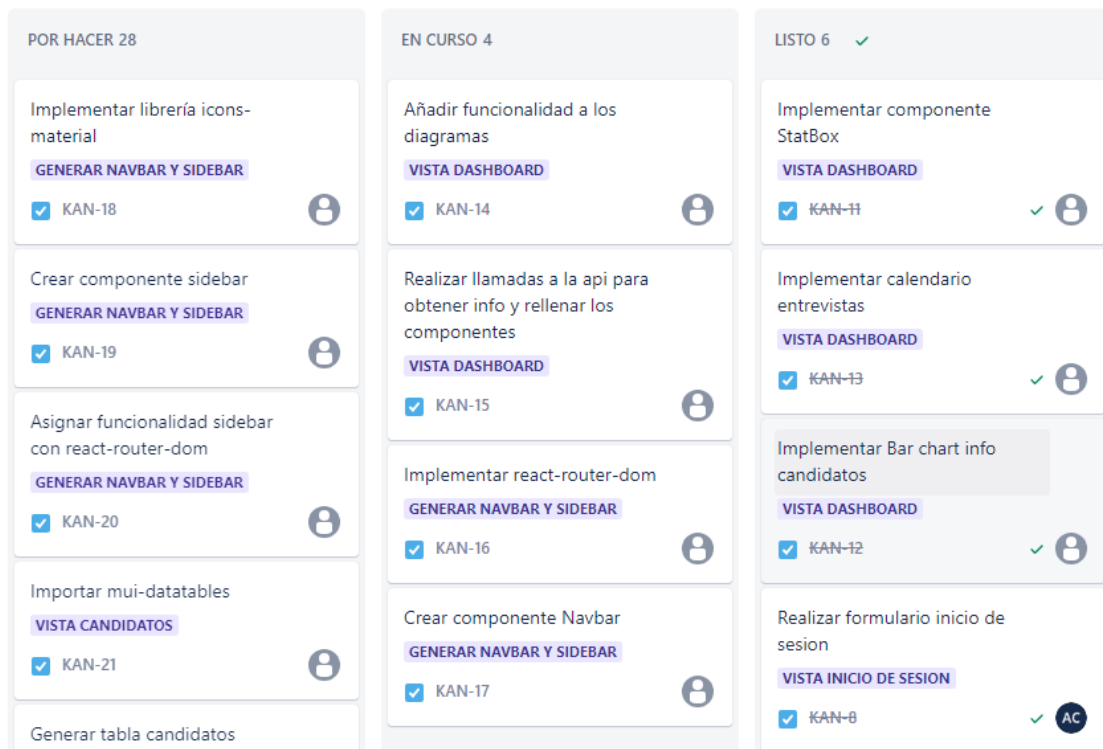


Ilustración 53 . Tablero a mitad Sprint 3

Se consiguió implementar el componente Statbox, que será el encargado de mostrar la información del número de candidatos han entrado el último mes, el número de ofertas en ese mes y el número de nuevos candidatos insertados en el Talent Pool.

Además, se implementó el calendario de las entrevistas mediante la librería “react-big-calendar” y una gráfica de barras mediante la librería “react-chartjs-2”.

Se quedaron a mitad de desarrollo las tareas asociadas a los Sidebars y Navbars, las cuales fueron desarrolladas antes de finalizar el Sprint.

Además, se realizó una pequeña modificación en el diseño, eliminando el apartado de “Timeline”, y añadiendo un calendario para las entrevistas. Esta modificación se debe a que para esta primera versión es más importante

mostrar información sobre las entrevistas, dejando ese timeline para desarrollos futuros.

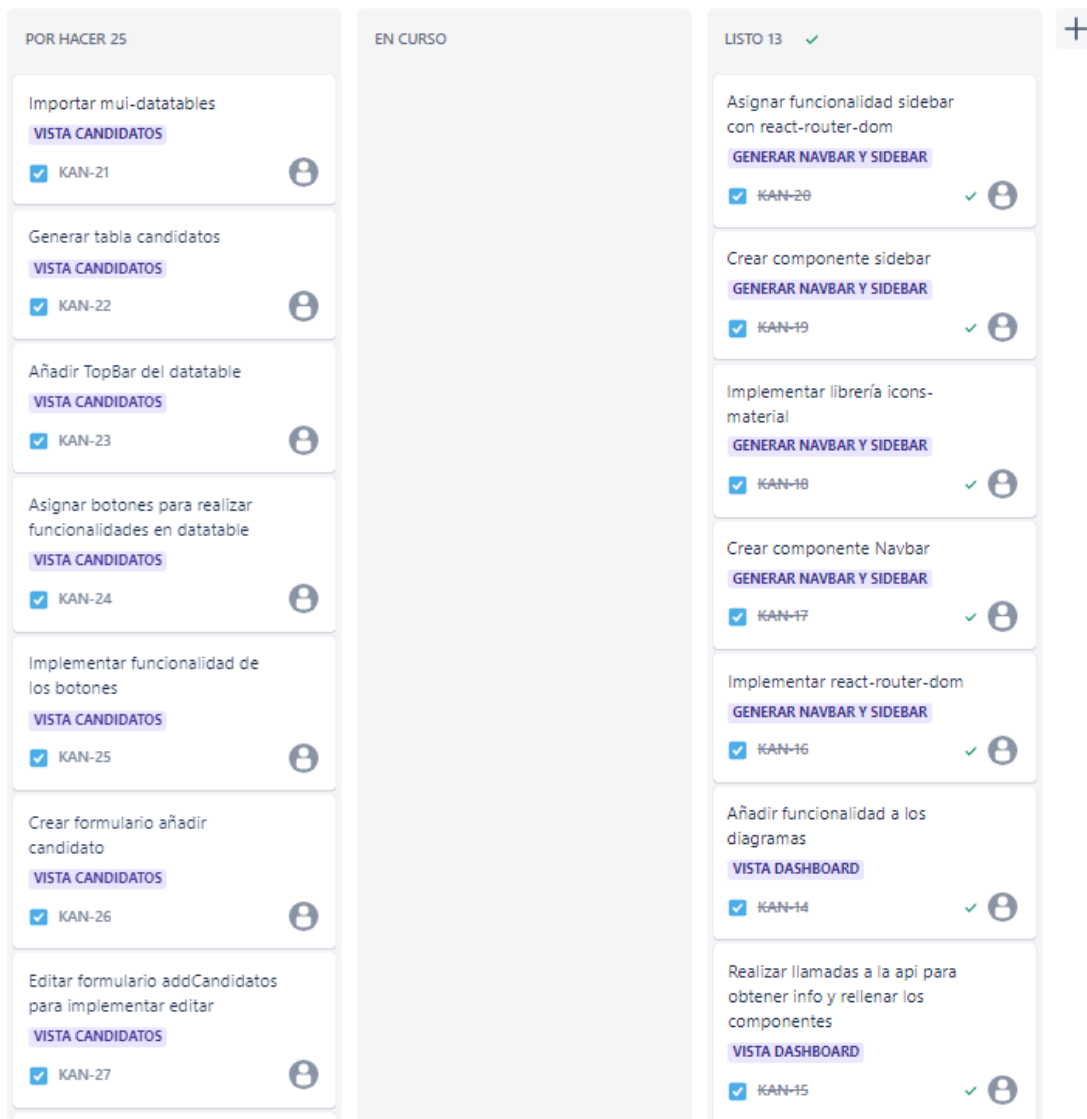


Ilustración 54 . Tablero a final Sprint 3

Tras finalizar el desarrollo de este Sprint se realizó una reunión el día 10 de marzo, mostrando el incremento desarrollado. No encontraron ningún requerimiento más, por lo que estos apartados quedaron desarrollados para esta primera versión, a falta también de descomentar las llamadas a la API para obtener datos.

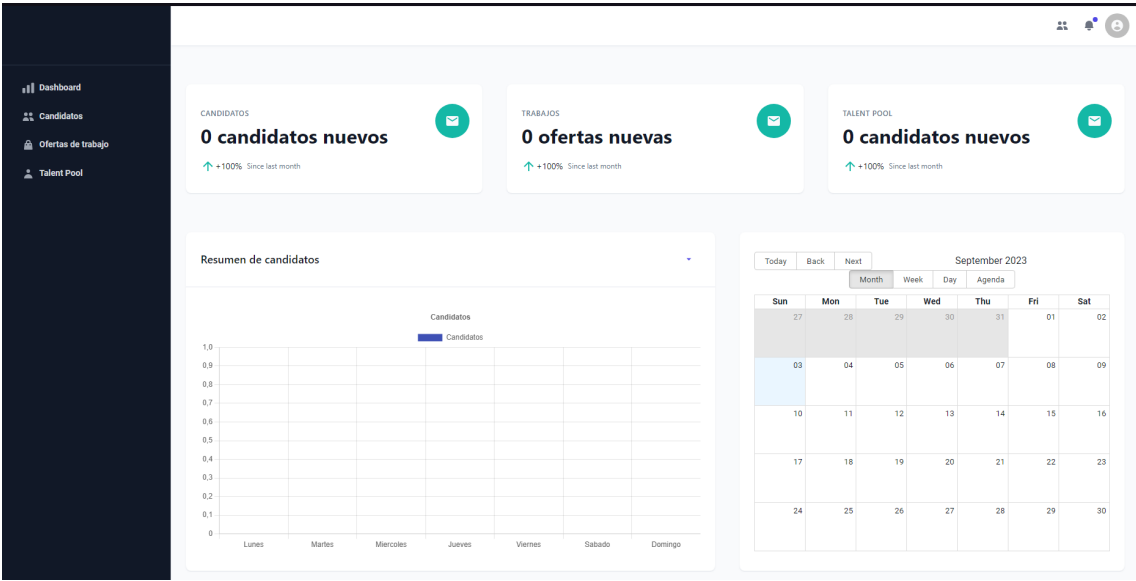


Ilustración 55 . Imagen Dashboard

4.6.3. Sprint 4 - Desarrollo de la vista ofertas y candidatos

Al igual que en Sprints anteriores, en primer lugar se realizó el Sprint Planning. En él decidí que se iban a realizar las tareas de desarrollo de las vistas Ofertas y Candidatos.

Se actualizó el Sprint Backlog y se comenzó a desarrollar estas nuevas tareas.

A mitad del Sprint, el estado de las tareas era el siguiente:

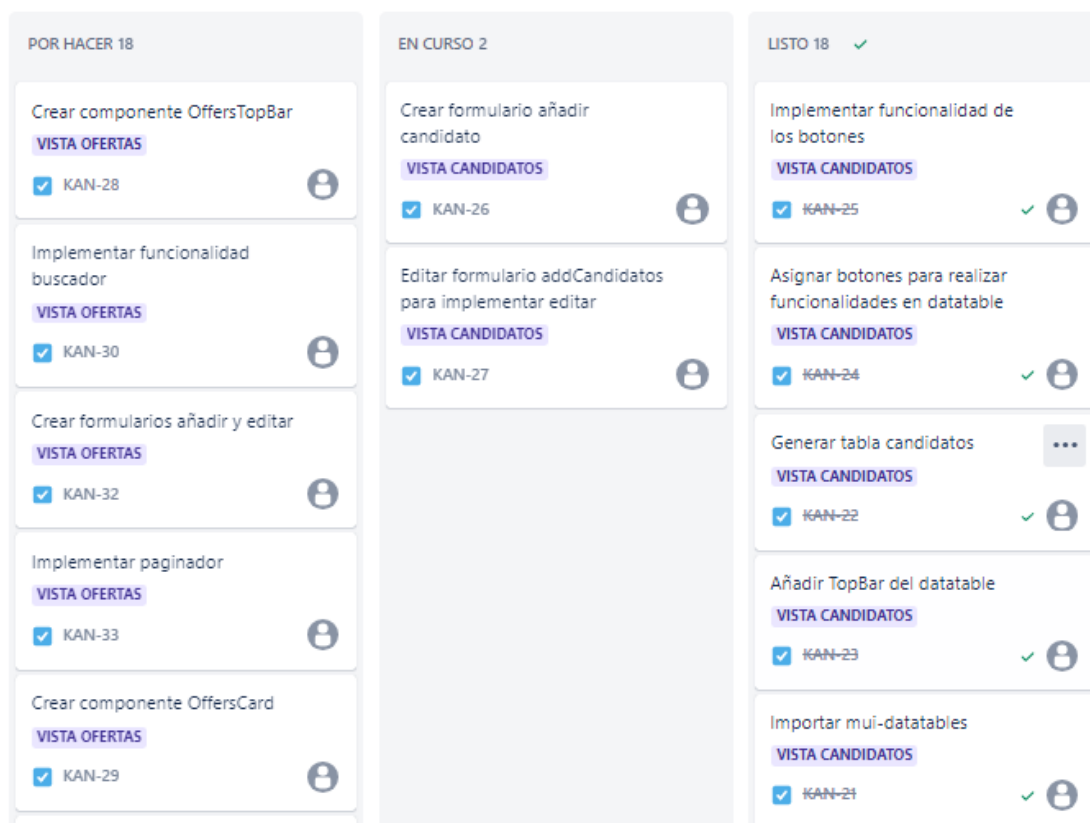


Ilustración 56 . Tablero a mitad Sprint 4

Como se puede observar, casi todas las tareas de la vista de candidatos habían sido finalizadas, a falta de terminar los formularios de añadir y eliminar. A lo largo de la siguiente semana se fueron resolviendo estas tareas pendientes y se comenzó con la épica Ofertas.

El estado de las tareas al finalizar el Sprint era el siguiente:

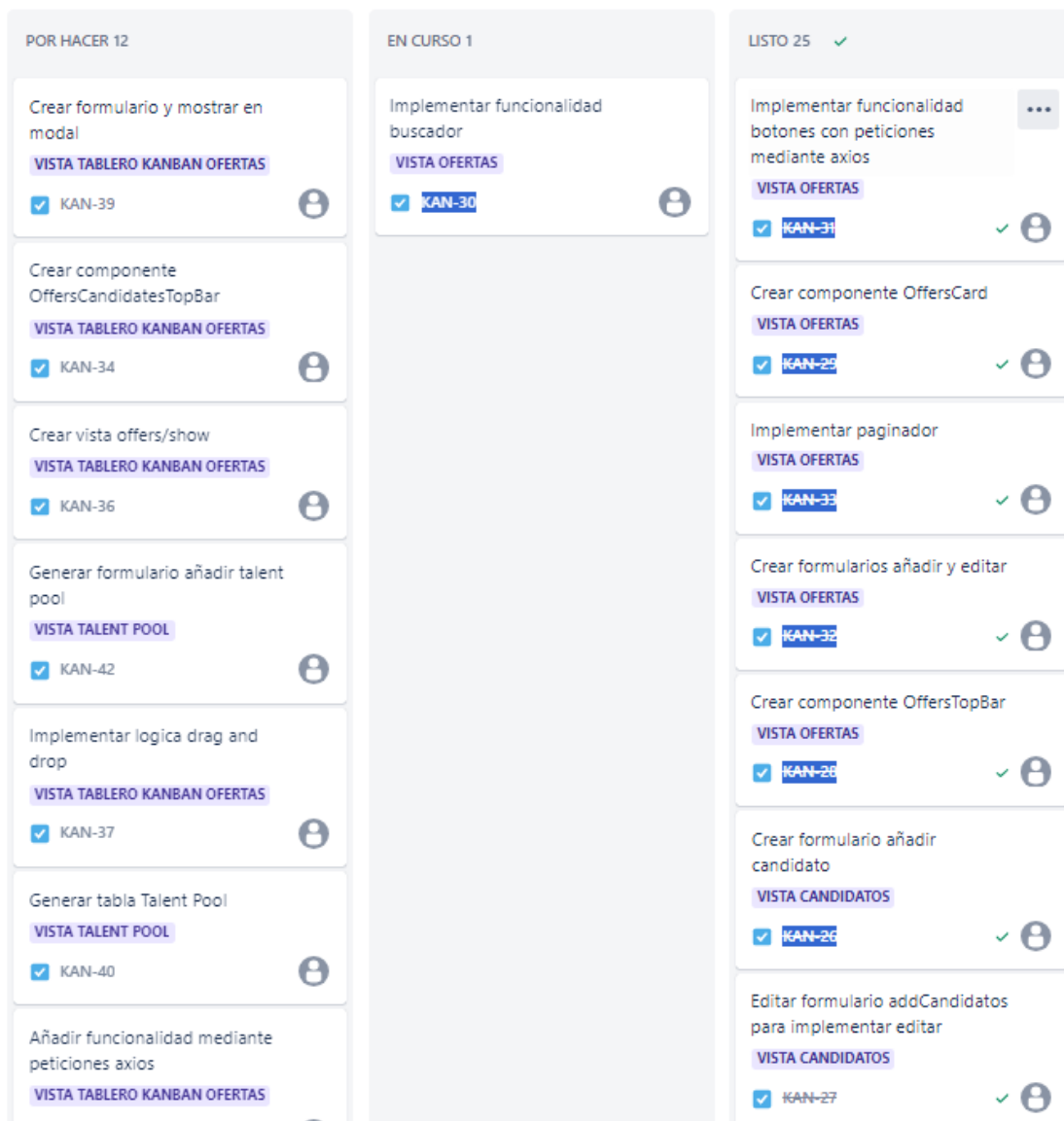


Ilustración 57 . Tablero a final Sprint 4

En este Sprint no dio tiempo a implementar la tarea del buscador, la cual fue devuelta al Product Backlog para ser desarrollada en futuros Sprints.

Luego se realizó la reunión para mostrar el incremento de la aplicación, en donde no se obtuvieron nuevos requisitos. Se reclamó que era necesario tener implementado este buscador, por lo que debería ser necesario asignarla para tenerlo en el siguiente incremento a ser posible.

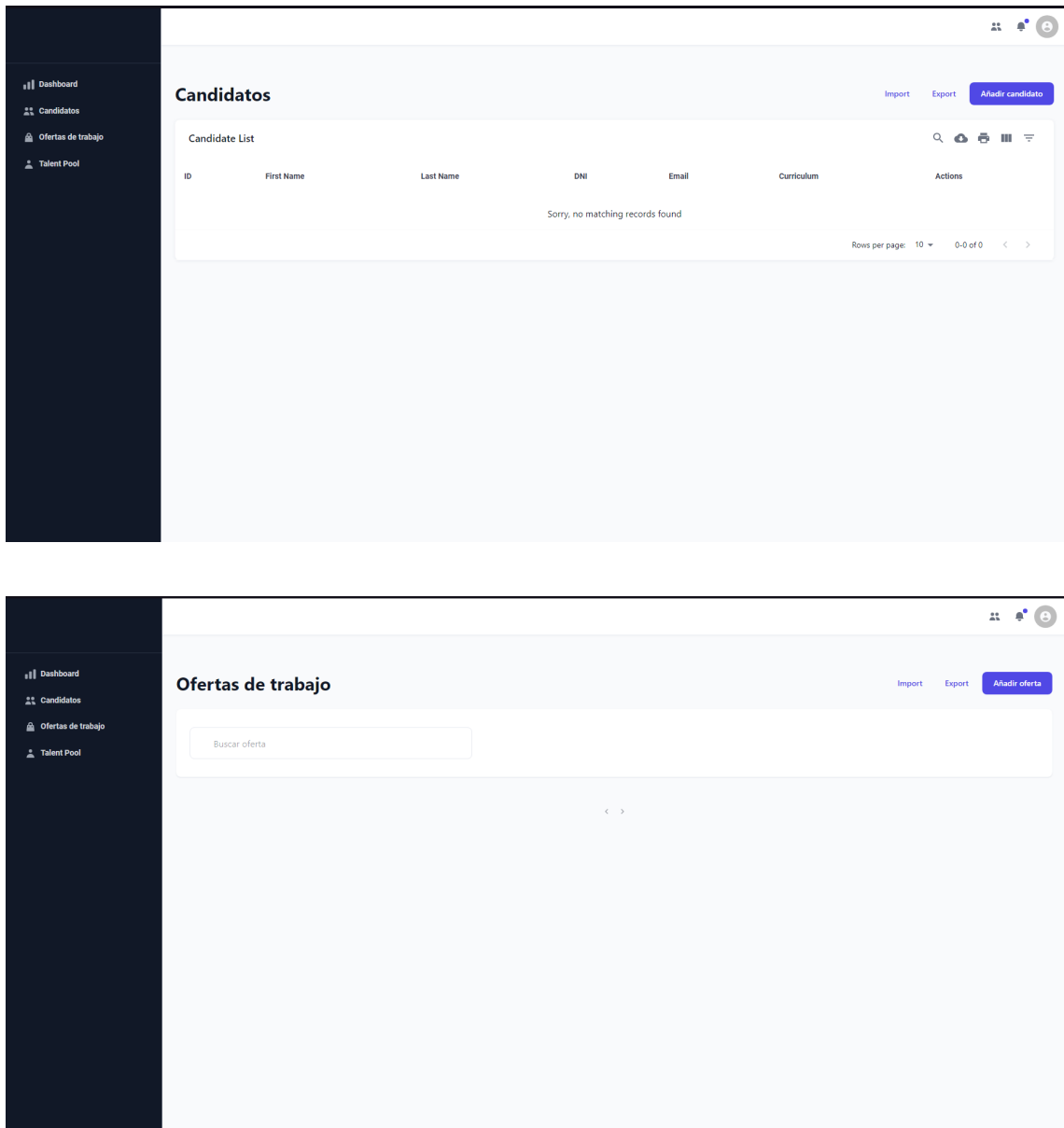


Ilustración 58 . Vistas candidatos y ofertas

4.6.4. Sprint 5 - Desarrollo del tablero Kanban para administrar candidatos de una oferta

En este sprint, se seleccionó en el Sprint Planning que era necesario realizar las tareas asociadas a la épica Vista tablero Kanban ofertas y la tarea pendiente del sprint anterior.

Para mitad de Sprint, el tablero estaba en este estado:

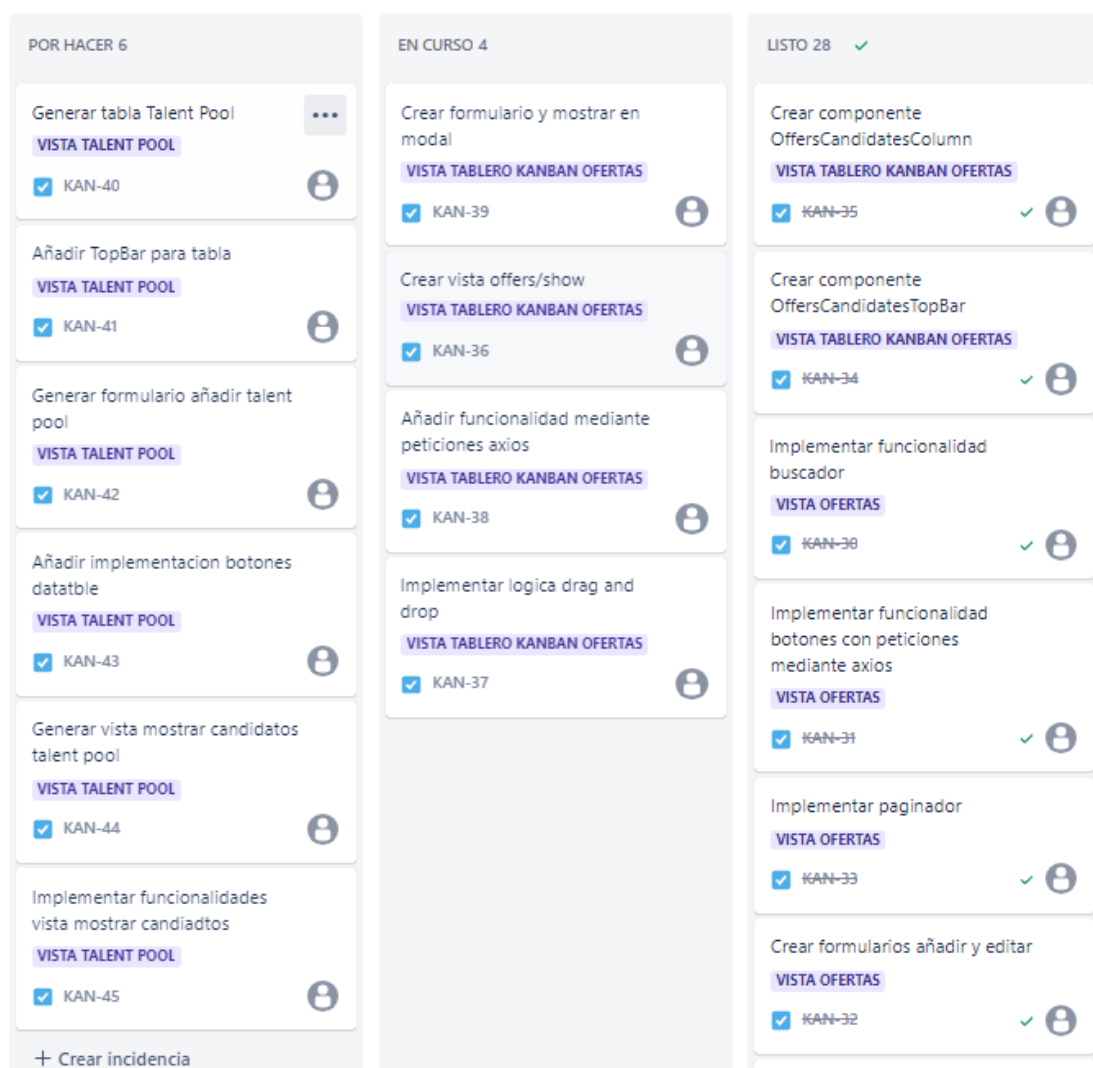


Ilustración 59 . Tablero a mitad Sprint 5

En la primera semana de desarrollo se estuvieron implementando los componentes `OffersCandidatesColumns`, que representará cada una de las columnas de mi tablero Kanban y el componente `OffersCandidatesTopBar`,

donde se implementará la cabecera de la vista. Además, se terminó de implementar el buscador de ofertas.

A lo largo de la siguiente semana se fueron desarrollando las demás tareas, teniendo al final del Sprint el siguiente estado de tablero.

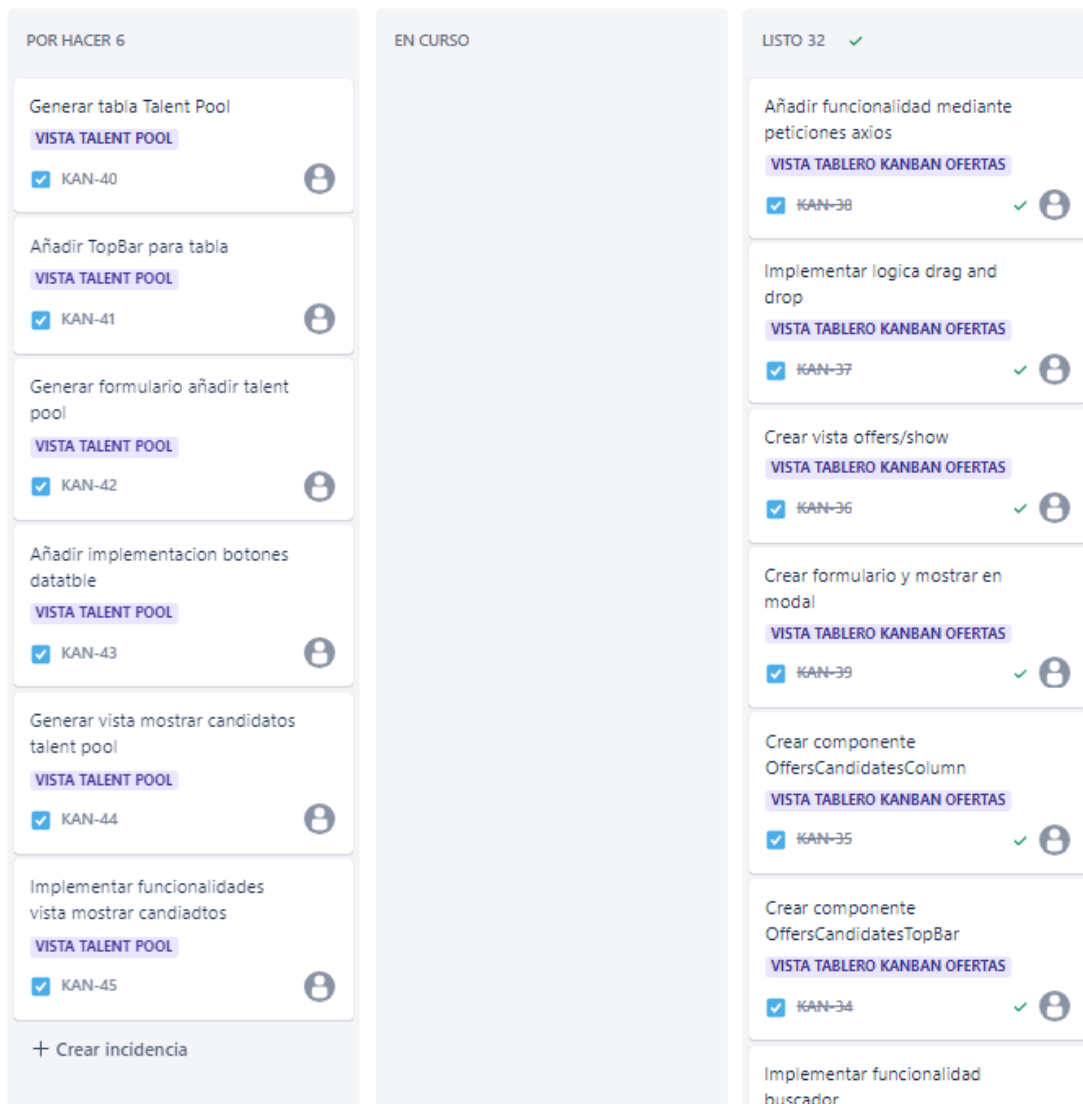


Ilustración 60 . Tablero a final Sprint 5

En este Sprint, dio tiempo a implementar todas las tareas, asignadas en el Sprint Review.

Tras realizar la reunión con el equipo de RRHH, quedaron muy contentos con esta implementación ya que cumplía a la perfección sus necesidades en este apartado.

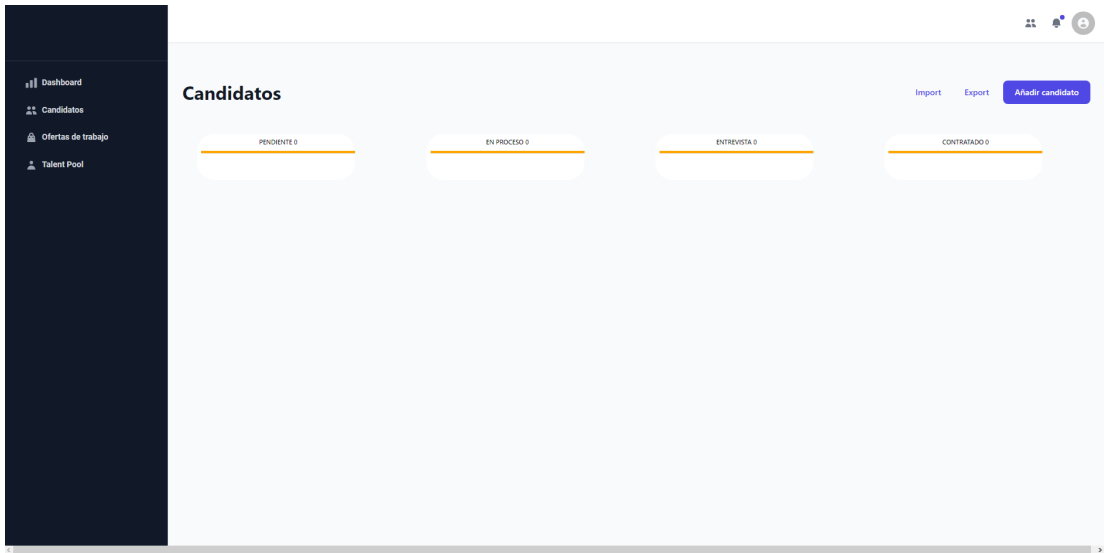


Ilustración 61 . Vista Offers Candidates

4.6.5. Sprint 6 - Desarrollo del tablero Kanban para administrar candidatos de una oferta

En este último sprint de desarrollo de la aplicación front, se realizó la última épica asignada, la vista Talent Pool.

Como en anteriores Sprint, se comenzó el desarrollo y a mitad del sprint se hizo una revisión del estado del tablero, teniendo las siguientes tareas finalizadas.

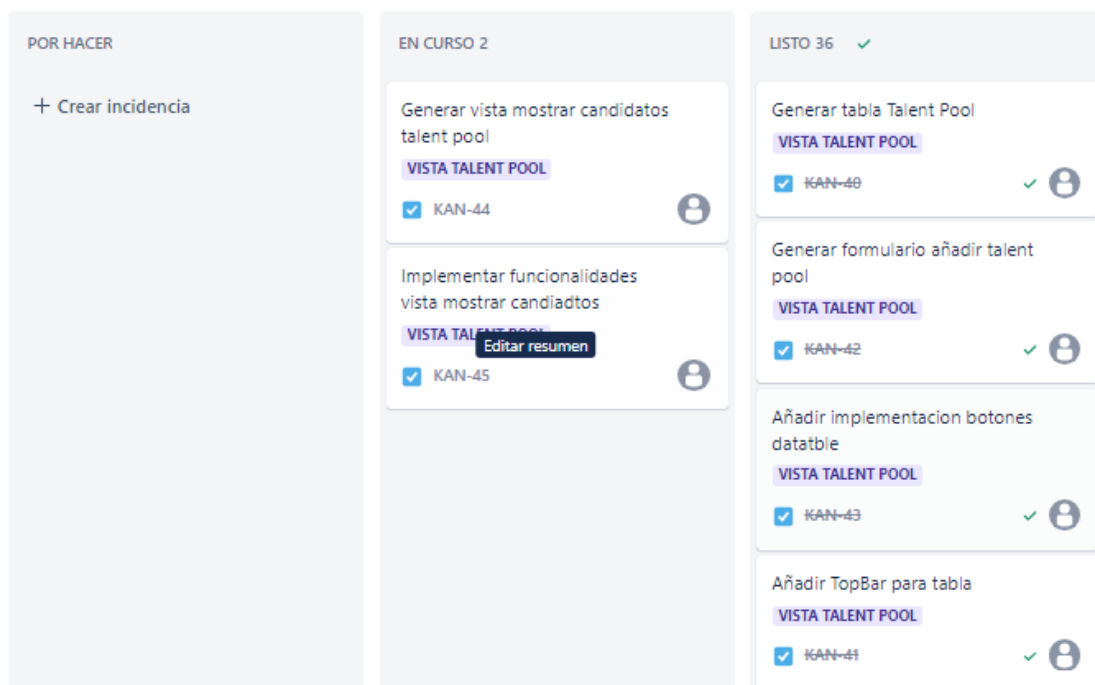


Ilustración 62 . Tablero a mitad Sprint 6

Esta vista era similar a la vista candidatos, por lo que la implementación de las tablas básicas y formularios fueron sencillos de desarrollar.

Al finalizar el sprint, este era el estado del tablero:

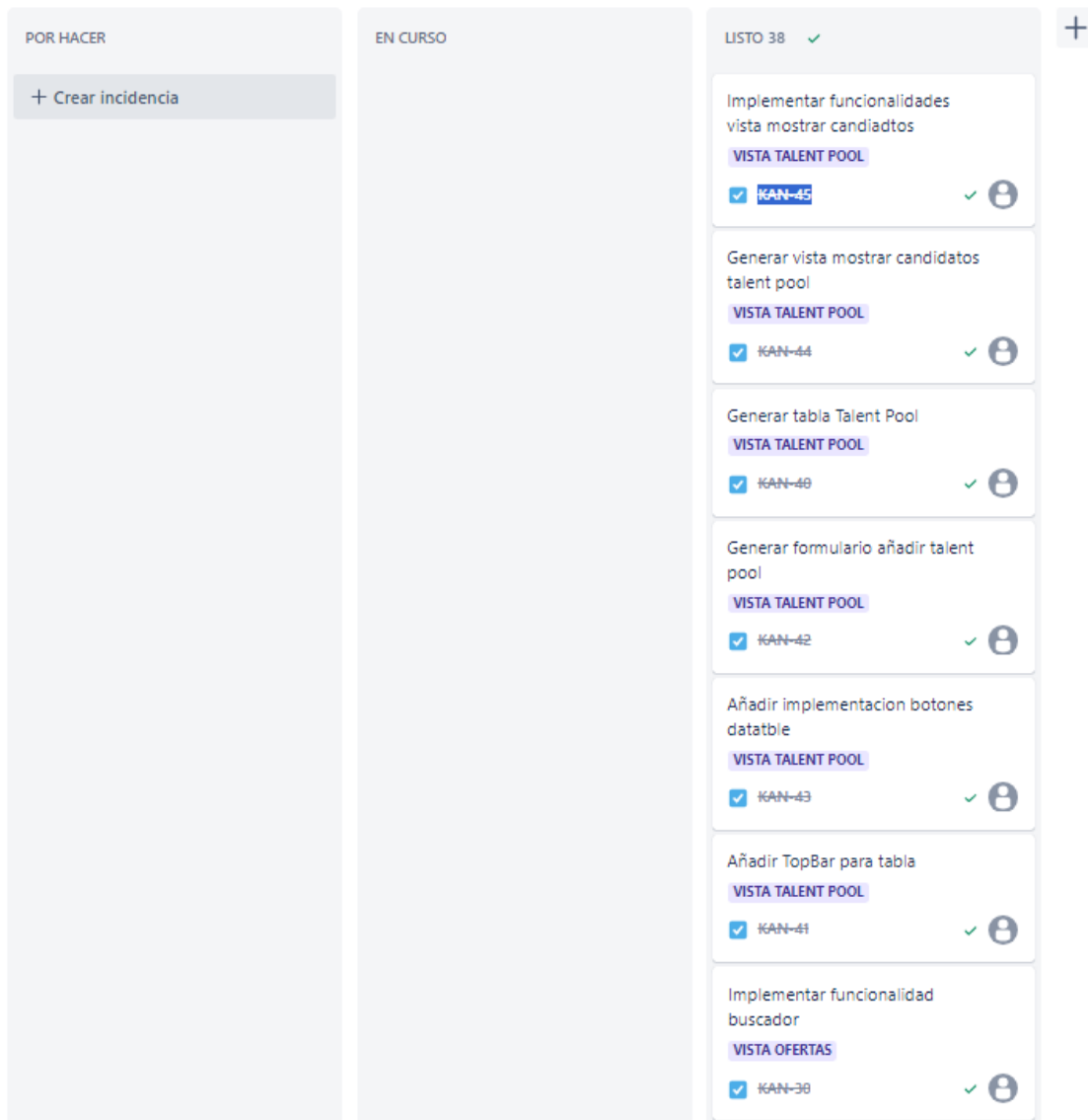


Ilustración 63 . Tablero a final Sprint 6

Al final del Sprint se pudo tener todas las tareas finalizadas, cumpliendo con las estimaciones de tiempo para el desarrollo del Front-End. El desarrollo de las tareas pendientes de este Sprint se resolvieron también de forma sencilla ya que esta vista reciclaba componentes y funcionalidades ya implementadas en otras vistas.

En la reunión del Sprint Review se mostró el incremento de la aplicación,

en la cual se indicaron algunos cambios que eran necesarios realizar. Estos cambios serán realizados en el Sprint 12, el cual está enfocado en el desarrollo de estos ajustes de la aplicación.

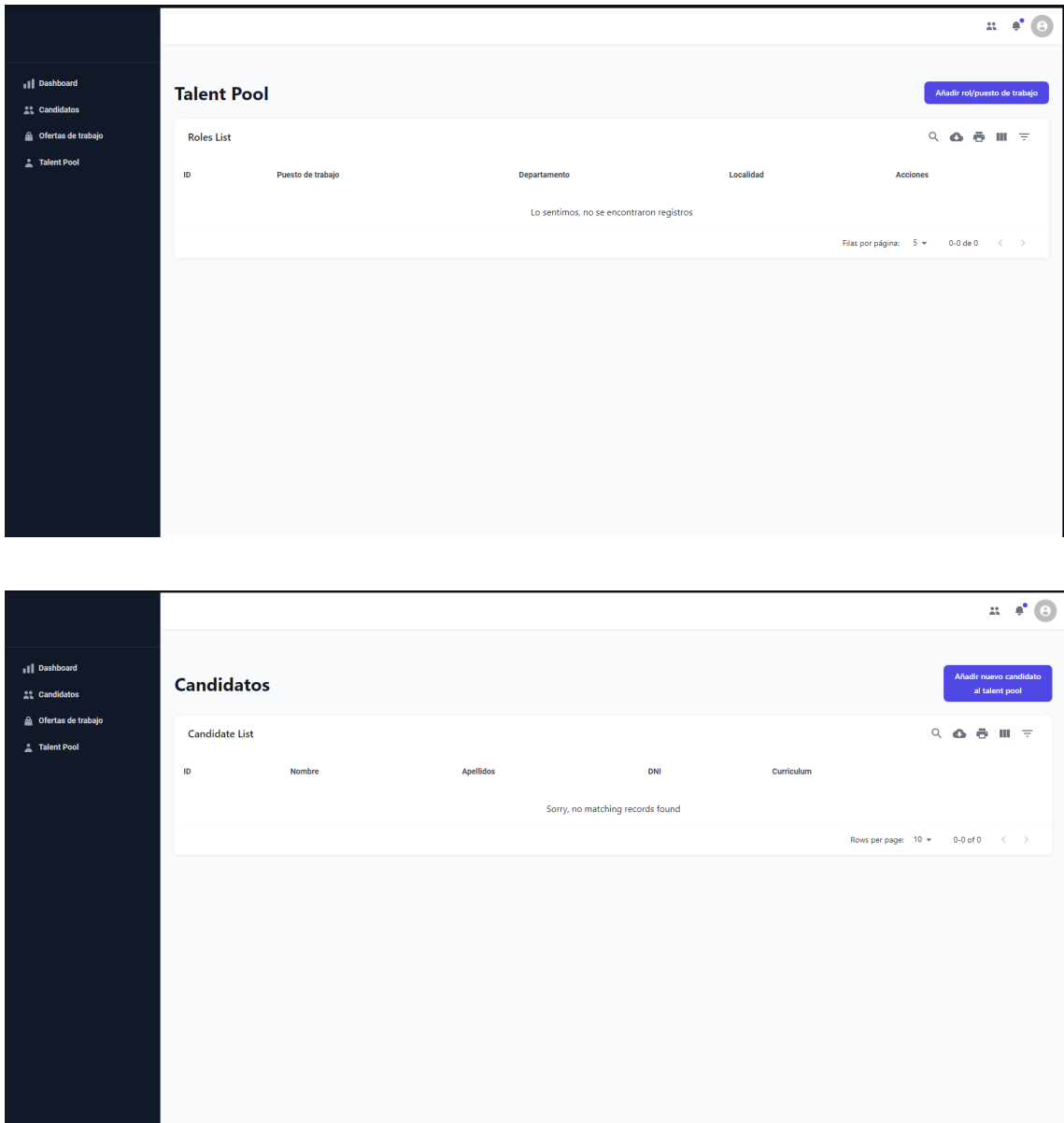


Ilustración 64 . Vista Talent Pool

4.6.6. Sprint 7 - Generar proyecto e implementación de autenticación.

Este sprint es el primero enfocado en el desarrollo de la aplicación backend. Al igual que en los sprints para la aplicación front, voy a ir documentando el desarrollo de esta aplicación, mostrando el tablero de Jira a mitad y a final del Sprint.

En el desarrollo de la aplicación backend, aunque no se documenta en las tareas de Jira, también se realizan ajustes en la aplicación front, descontando todas las líneas de peticiones axios a la API, para ir probando el correcto desarrollo de esta API, además del uso de Postman.

En este primer Sprint vamos a realizar la configuración inicial del proyecto, junto con la implementación de la autenticación mediante el uso de JSON WEB TOKEN.

Además, se implementarán las dependencias necesarias para la implementación de un sistema de envío de correos, junto con la posibilidad de subir archivos en local.

Tras la primera semana de desarrollo, el tablero tiene el siguiente estado.

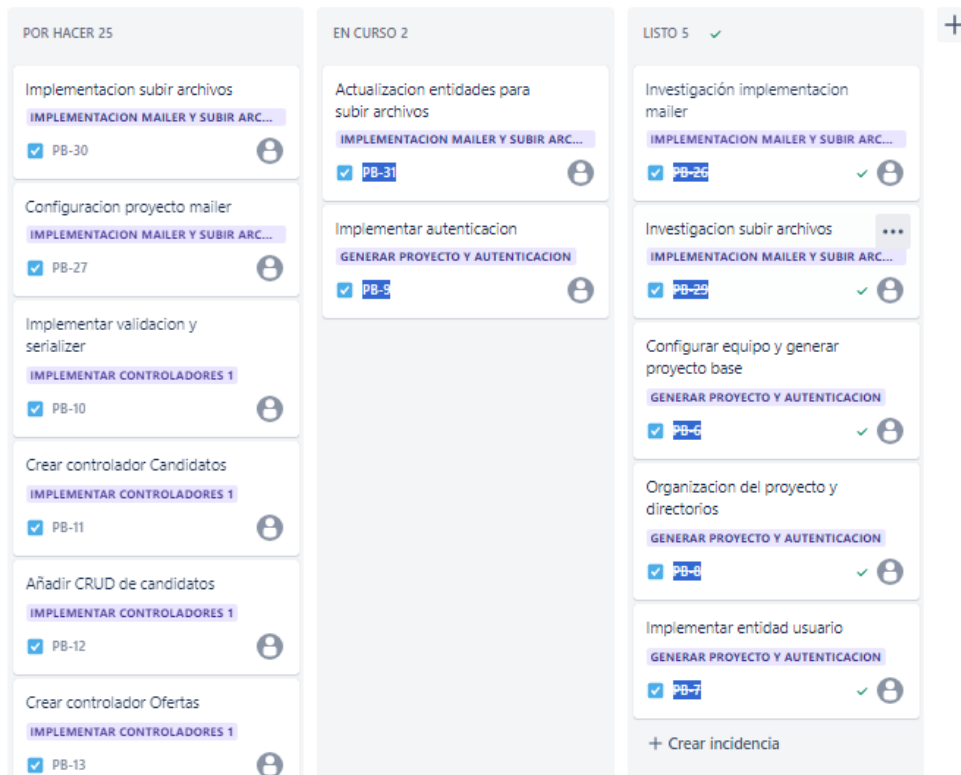


Ilustración 65 . Tablero a mitad Sprint 7

Como se puede observar, en la primera semana de desarrollo, se ha implementado el proyecto base, y la configuración de directorios. Además se ha investigado en la documentación de Symfony y se ha obtenido información para implementar un mailer y una forma de subir archivos en local.

En esta segunda parte del Sprint se van a realizar las implementaciones totales de estas investigaciones, junto con la implementación del JWT, obteniendo de esta manera una forma de realizar login seguro en mi aplicación.

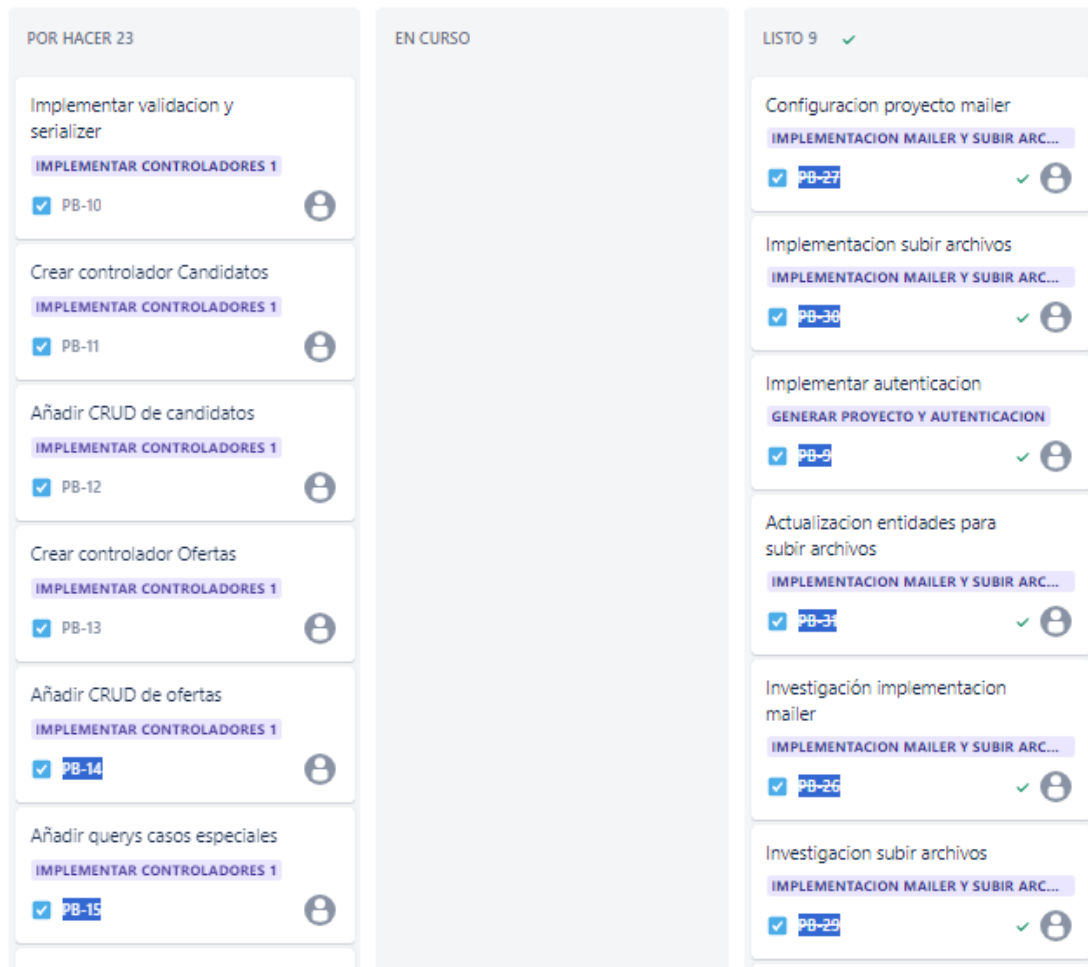


Ilustración 66 . Tablero a final Sprint 7

Al finalizar el sprint, se realizó una reunión rápida, mostrándoles esta funcionalidad de autenticación, junto el desarrollo del envío de correo y subida de archivos a nivel local.

Se solicitó por parte de RRHH la posibilidad de subir archivos a la nube mediante el uso de Amazon S3, y el envío de correos mediante servidores SMTP. Estas solicitudes quedaron apuntadas para futuros desarrollos ya que esto se salía de los límites de tiempo de los Sprint.

4.6.7. Sprint 8 - Implementación de controladores, entidades, modelos I

En este Sprint se han asignado al Sprint Backlog las tareas de desarrollo de los primeros y principales controladores de mi aplicación. Estos son el controlador de candidatos y el de ofertas.

Además de los controladores, se implementarán toda la estructura básica, como son las entidades, repositorio, etc...

En la primera semana de desarrollo se consiguió implementar la validación de formularios y el serializer. El componente Serializer en Symfony es el encargado de convertir objetos en un formato específico, ya sea json, xml, etc...

Además, se implementó el primer controlador, junto con todas sus peticiones y desarrollo de un CRUD básico.

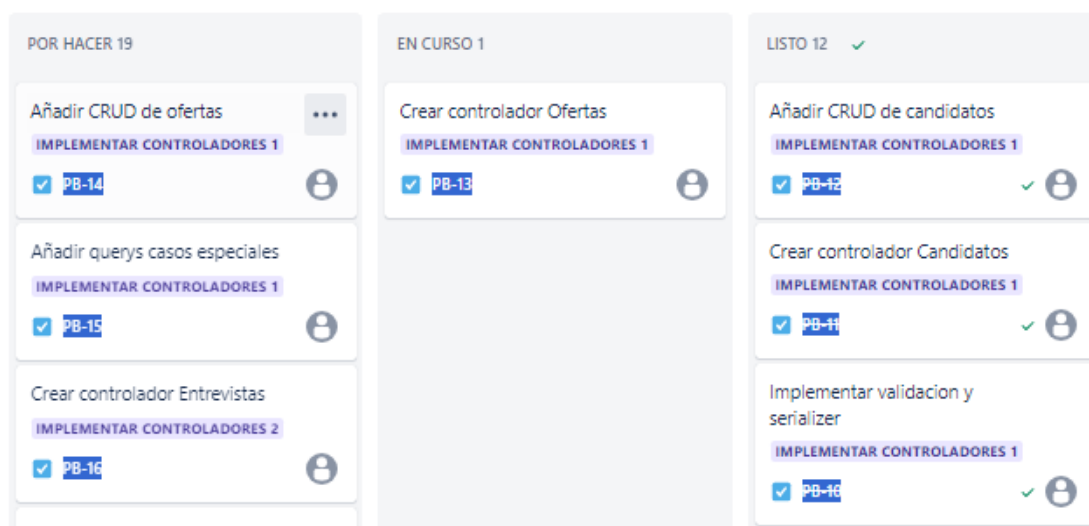


Ilustración 67 . Tablero a mitad Sprint 8

Para la siguiente semana del Sprint quedaría pendiente el desarrollo del controlador Ofertas y la implementación de querys y peticiones get especiales,

como por ejemplo la de obtener un conjunto de candidatos creados entre dos fechas.

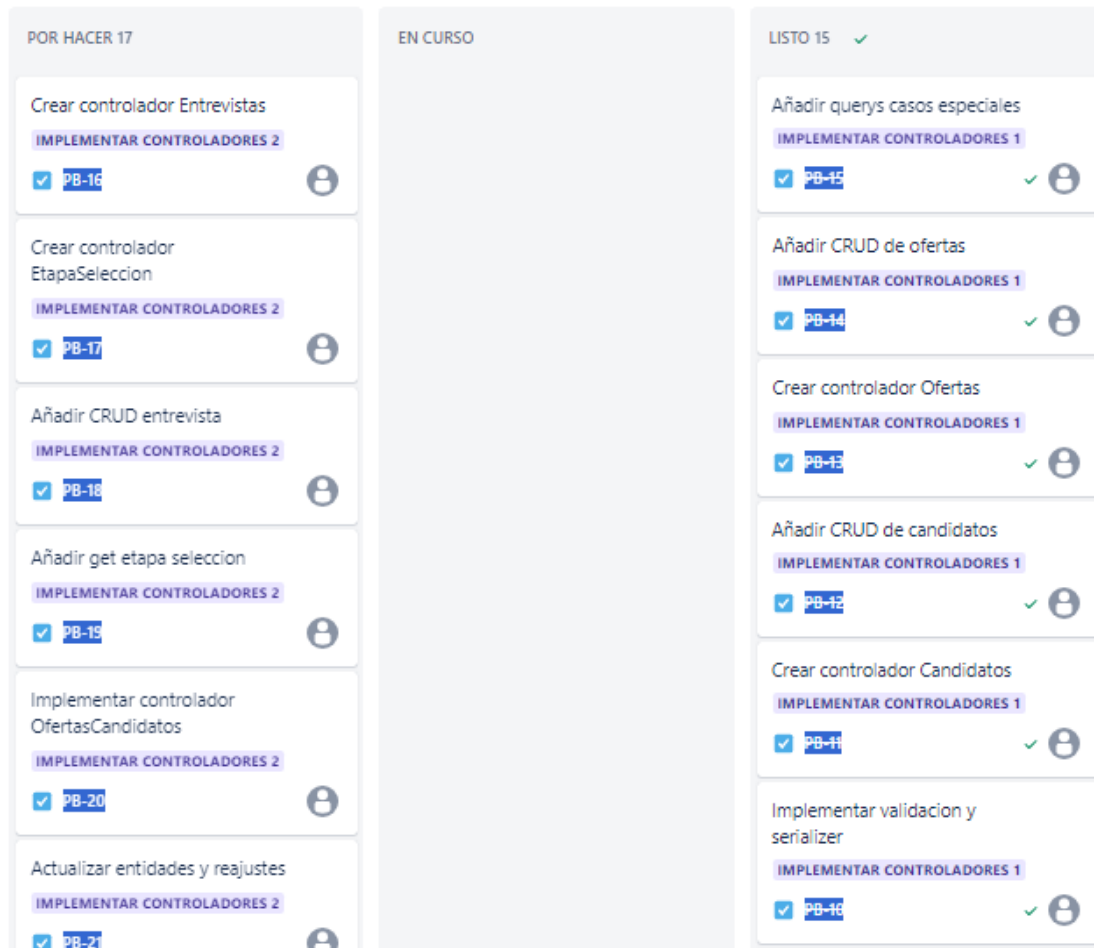


Ilustración 68 . Tablero a final Sprint 8

Al finalizar el Sprint, se cumplieron con los objetivos, desarrollando por completo estos controladores. En la reunión del Sprint Review se mostraron las distintas vistas de la aplicación, pero esta vez mostrando datos almacenados en BBDD, pudiendo realizar operaciones básicas con ellos.

4.6.8. Sprint 10 - Implementación de controladores, entidades, modelos 2

En este Sprint se continúa con la implementación de los controladores y entidades expuestos en el diagrama de clases. En este Sprint se realizará la implementación de la entidad entrevista, etapa selección y la tabla de oferta candidatos.

En la primera semana de desarrollo se consiguió implementar completamente los controladores de entrevista y etapa, junto con sus operaciones CRUD. Sin embargo no se consiguió finalizar el controlador OfertasCandidatos, ya que se necesitó de más tiempo para entender como Symfony implementa las relaciones muchos a muchos de 2 entidades.

Este era el estado del tablero en ese punto.

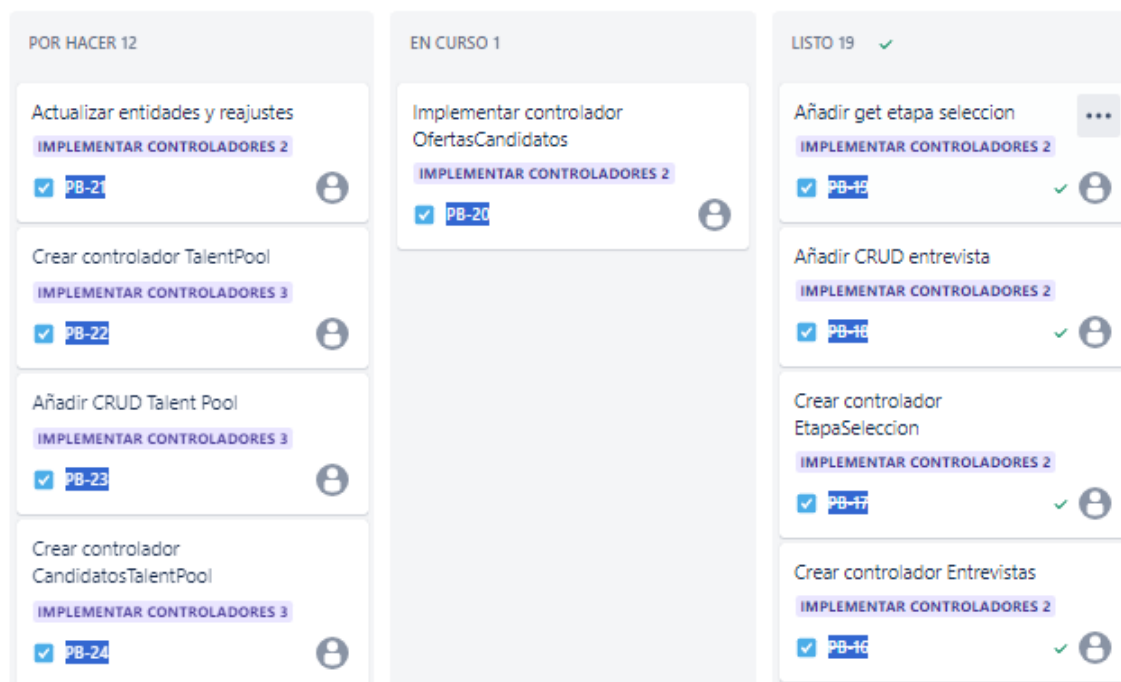


Ilustración 69 . Tablero a mitad Sprint 10

Al finalizar la semana se consiguió actualizar todas las entidades desarrolladas para implementar la relación entre Candidatos y Ofertas mediante el uso de la entidad Ofertas-Candidatos.

Finalmente el Sprint Backlog quedó vacío al finalizar este Sprint.

En la reunión del Sprint Review se mostró el calendario del dashboard con información almacenada en BBDD y se consiguió realizar una demostración del tablero Kanban implementado totalmente funcional.

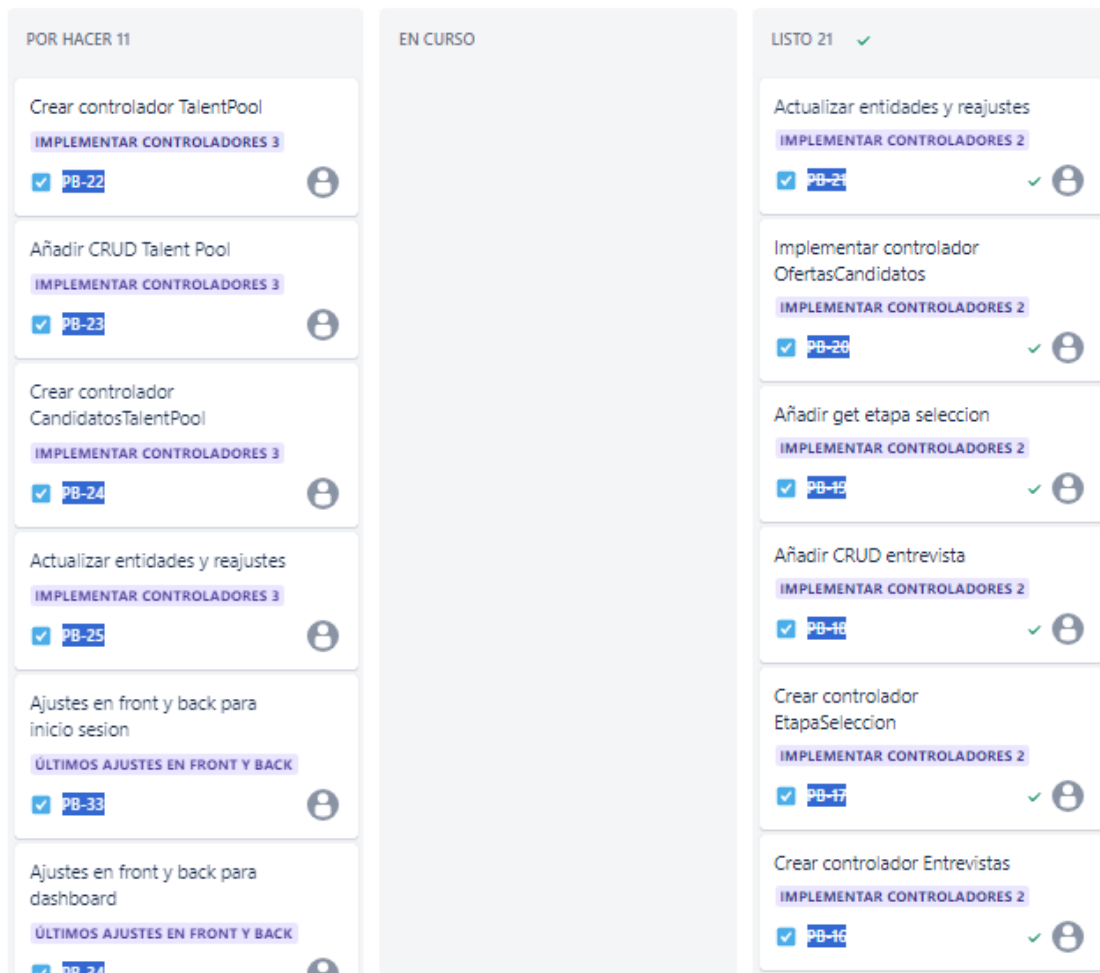


Ilustración 70 . Tablero a final Sprint 10

4.6.9. Sprint 11 - Implementación de controladores, entidades, modelos 3

En este Sprint se siguió implementando las entidades restantes, para darle funcionalidad al apartado del Talent Pool. Al finalizar la primera semana se consiguió implementar el controlador del Talent pool dejando para la siguiente la implementación de la entidad Candidates Talent Pool.

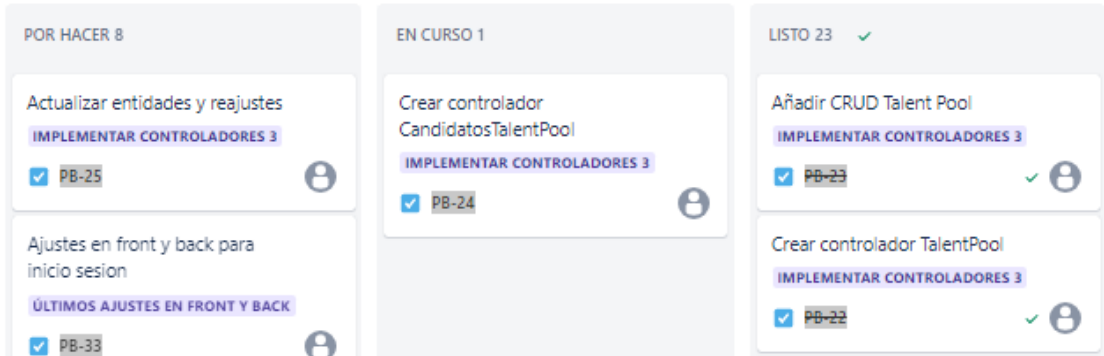


Ilustración 71 . Tablero a mitad Sprint 11

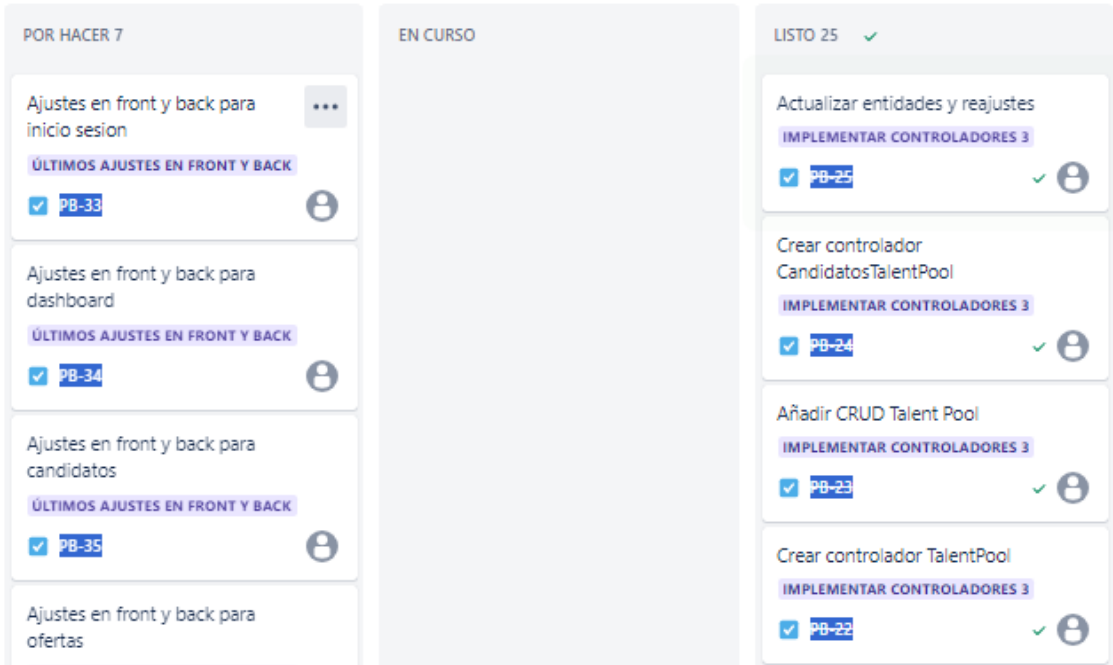


Ilustración 72 . Tablero a finalSprint 11

4.6.10. Sprint 12 - Últimos Ajustes

En este sprint se realizarán aquellos pequeños ajustes comentados por parte de RRHH para que esta primera versión de la aplicación esté totalmente a su gusto. La mayoría de estos ajustes son visuales, aunque si fuese necesario se realizarán modificaciones en la API.

La primera semana del sprint se dedicó totalmente al apartado de Dashboard, candidatos y ofertas, realizando ajustes como por ejemplo las distintas funcionalidades que se muestran en el calendario de entrevistas o apartados que aparecía información en inglés debido a los componentes importados de otras librerías.

Además se generó un logo para la aplicación que ha sido añadido en el apartado de login y en el Sidebar.

El tablero de esta primera semana se quedaría de la siguiente manera.

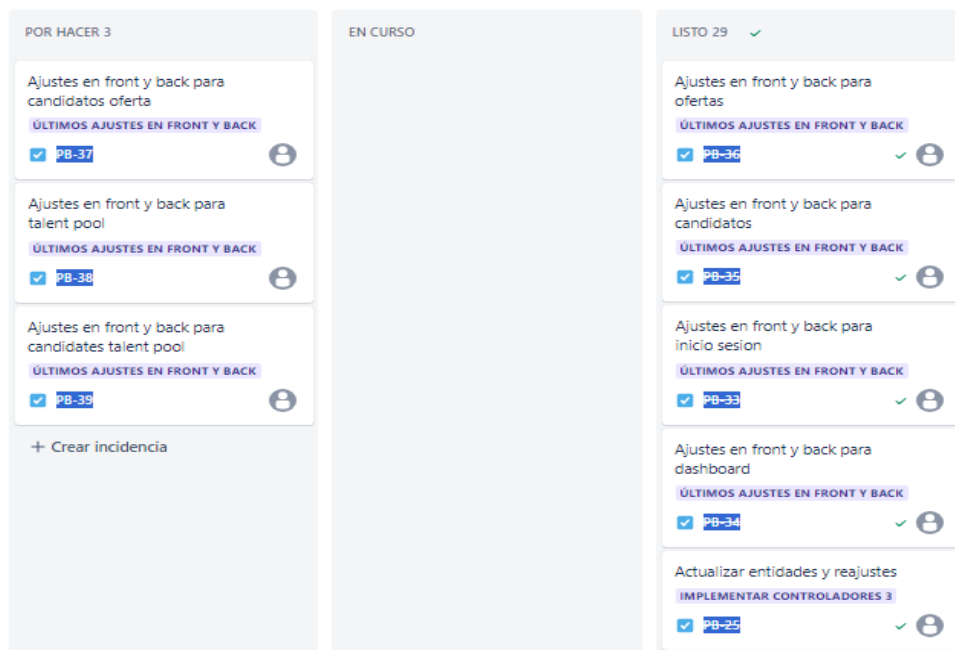


Ilustración 73 . Tablero a mitad Sprint 12

De la misma manera, la siguiente semana se realizaron ajustes y se solventaron errores en el Drag and Drop de la vista Candidatos-Oferta. Además se añadieron atributos faltantes a ciertas tablas, y se modificaron las tablas que muestran los Curriculumms para informar al usuario si cierto candidato ha entregado o no, o este se ha perdido.

El tablero al finalizar el Sprint se quedó de la siguiente manera.

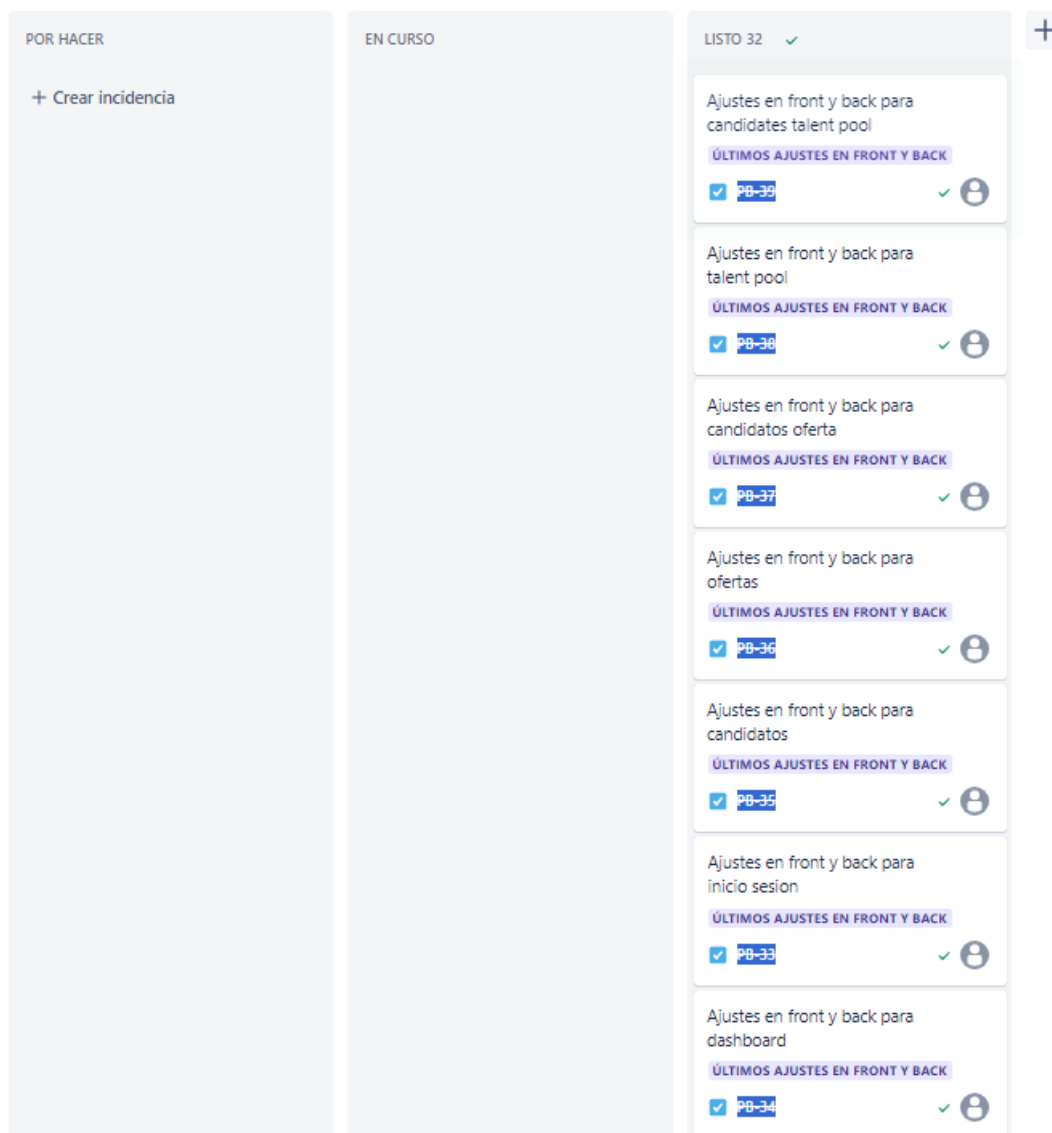


Ilustración 74 . Tablero a final Sprint 12

4.7. Pruebas/Resumen aplicación

Tras haber completado la implementación de la primera versión funcional de la aplicación a desarrollar, es necesario realizar una serie de pruebas para confirmar que el desarrollo ha ido conforme a lo planeado y se cumplen con todos los requisitos impuestos al principio del desarrollo de este trabajo.

A la misma vez que se realizan estas pruebas, se va a ir mostrando todos los apartados y funcionalidades de la aplicación, para tener un manual de usuario de uso de la aplicación.

Para realizar estas pruebas, se van a ir adjuntando imágenes de las distintas vistas de la aplicación, mostrando el comportamiento de cada una de las funcionalidades de la aplicación y las respuestas a las mismas.

De esta forma, podremos utilizar este apartado como manual de usuario, ya que se van a mostrar todos los apartados de la aplicación y cómo funcionan.

Aunque se van a realizar las pruebas de la API mediante el uso de la aplicación Front, se pone a disposición una colección de postman con todas las peticiones posibles que se pueden realizar actualmente a la API. Esta colección se puede obtener mediante el link que aparece en el apartado ANEXO.

4.7.1. Inicio de sesión

Al acceder a la aplicación, la primera vista que nos encontraremos es la de Inicio de Sesión. Al no tener el token de acceso a la aplicación, será necesario rellenar el formulario con las credenciales correspondientes para poder obtenerlo y que este se almacene en el localStorage que nos aportan los navegadores web. En este formulario existen dos posibles casos:

- Inicio de sesión correcto:

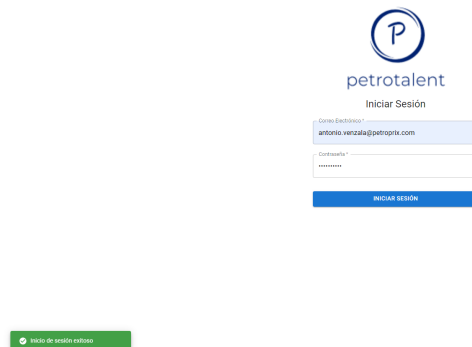


Ilustración 75 . Vista inicio de sesión correcto

- Inicio de sesión incorrecto:

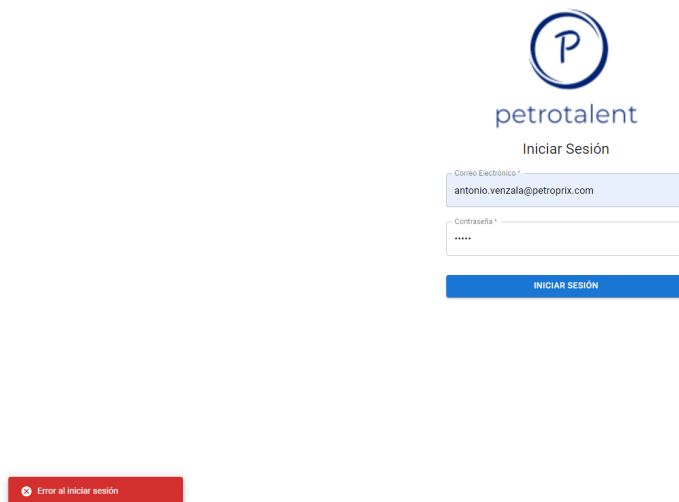


Ilustración 76 . Vista inicio de sesión incorrecto

Como se puede observar, dependiendo de si el login ha sido satisfactorio o no, se mostrará una notificación en el margen inferior izquierdo. La implementación de esta notificación se está realizando mediante el uso del componente Snackbar de la librería Material UI, de donde se han obtenido una gran cantidad de componentes para la realización de esta aplicación.

Actualmente, existe un límite de validez para un token generado, haciendo que si no hemos hecho cierre de sesión durante 1 hora, este pasará a ser invalido, provocando que la aplicación vuelva a este apartado y solicite de nuevo las credenciales de acceso. Otro posible caso es que se pierda el token del localStorage, desembocando en el mismo proceso.

4.7.2. Dashboard

Si el inicio de sesión ha sido satisfactorio, accederemos a la vista Dashboard, en donde se muestra un resumen rápido de la información más importante de la página.

En esta vista, al igual que en todas las demás, disponemos de un menú superior, donde podremos cerrar sesión en la aplicación y un menú lateral, donde podremos navegar entre las distintas secciones de la aplicación.

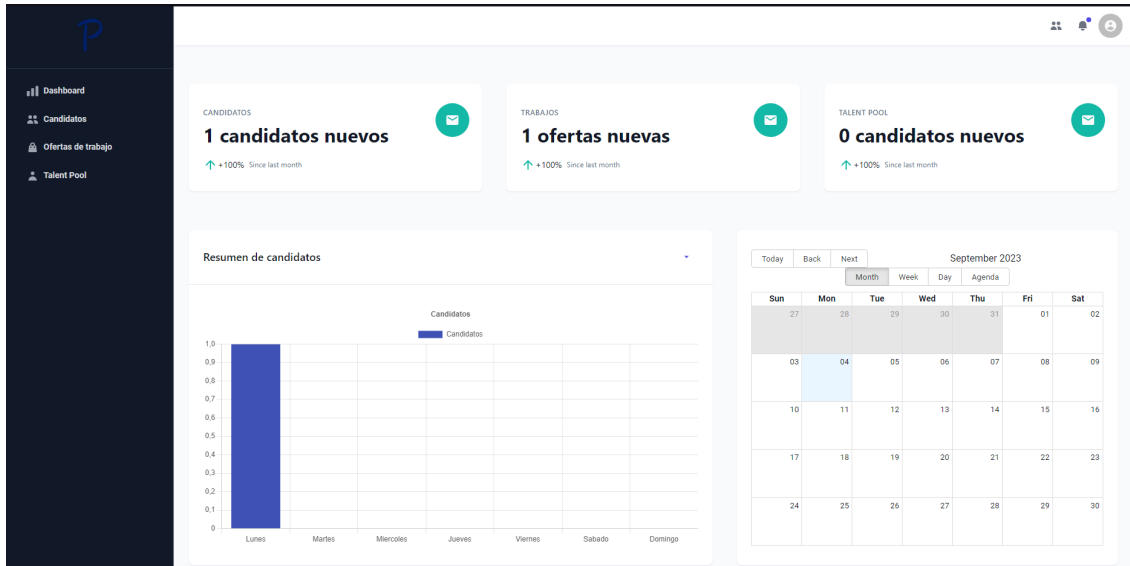
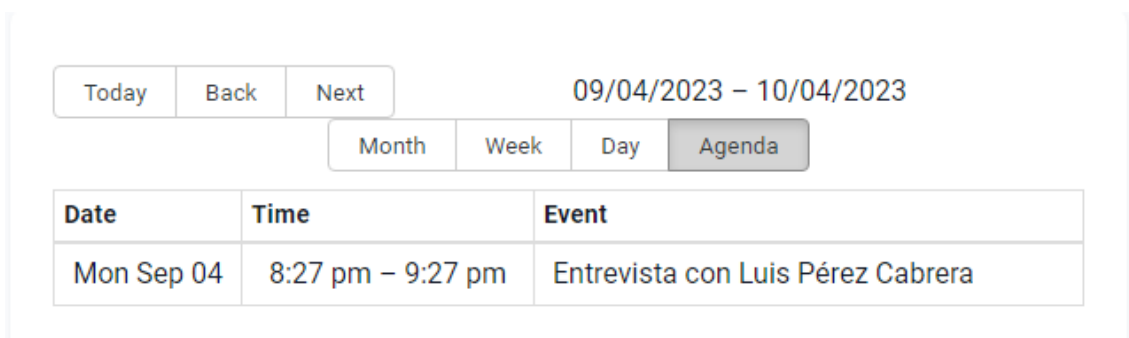


Ilustración 77 . Vista dashboard

En el desarrollo de esta vista, se han implementado una serie de componentes para mostrar la información más sensible. Entre estos datos nos podemos encontrar:

- Número de candidatos nuevos en el mes actual.
- Número de ofertas nuevas en el mes actual.
- Número de candidatos nuevos en el talent pool en el mes actual.
- Incremento del número de candidatos/ofertas con respecto al mes anterior.
- Gráfico de barras que muestra un resumen de los candidatos nuevos por cada día de la semana.
- Calendario donde se mostrarán las entrevistas actualmente asignadas.

Dentro del apartado calendario, se podrán utilizar todas las funcionalidades que nos aporta la librería react-big-calendar. La más interesante es el botón Agenda, en donde se mostrará un listado de todas las entrevistas a realizar a lo largo del mes o el día seleccionado.

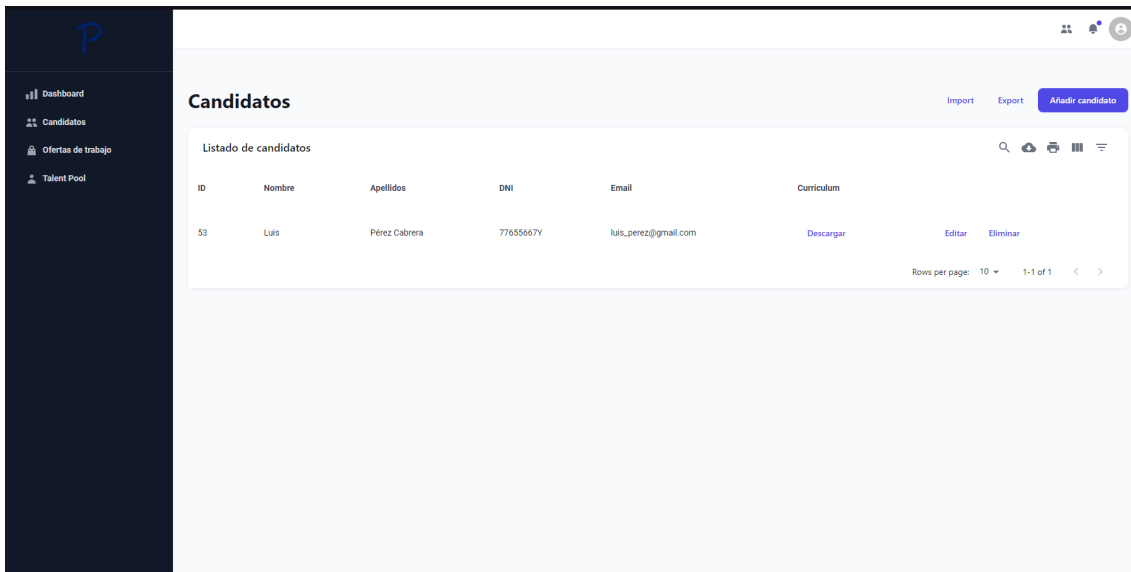


The screenshot shows the 'Agenda' view of a calendar. At the top, there are navigation buttons: 'Today', 'Back', and 'Next'. To the right, the date range '09/04/2023 - 10/04/2023' is displayed. Below these, there are view toggle buttons: 'Month', 'Week', 'Day', and 'Agenda' (which is currently selected and highlighted). The main content is a table listing events.

Date	Time	Event
Mon Sep 04	8:27 pm – 9:27 pm	Entrevista con Luis Pérez Cabrera

Ilustración 78 . Ejemplo menú calendar (Agenda)

4.7.3. Apartado candidatos



ID	Nombre	Apellidos	DNI	Email	Curriculum
53	Luis	Pérez Cabrera	77655667Y	luis_perez@gmail.com	Descargar

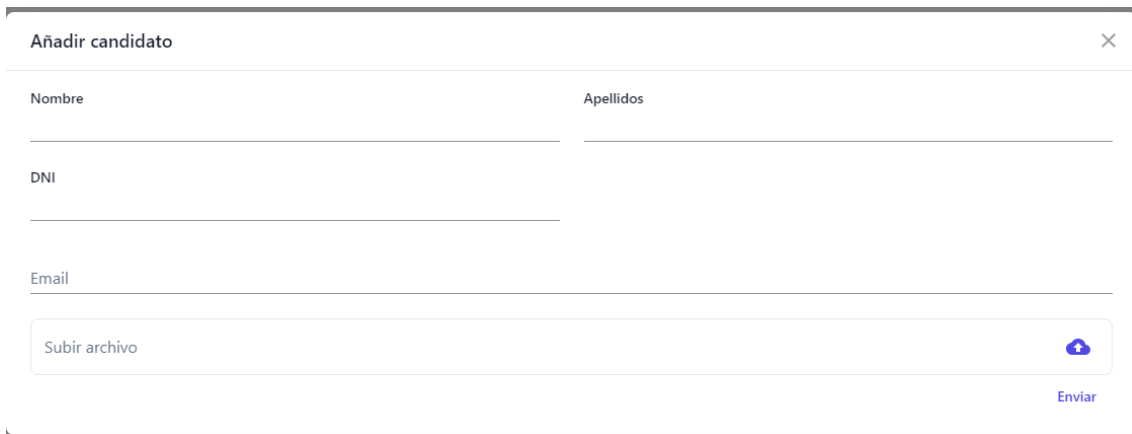
Ilustración 79 . Vista candidatos

En la barra lateral, si pulsamos sobre la opción Candidatos, se nos redirigirá al apartado mostrado en la imagen anterior. En él, nos encontramos con una lista de candidatos mostrada mediante una tabla, implementada mediante la librería “mui-datatables”.

Esta tabla de datos permite realizar una serie de funcionalidades muy interesantes, como el filtrado de columnas, selección del número de filas a mostrar por página, paginación, y generar un csv con los datos o imprimirlos.

Además de estas funcionalidades que trae de por sí el componente Datatable de la librería anteriormente mencionada, se pueden realizar otras operaciones, como “Añadir candidatos”, “Editar Información de los candidatos”, “Eliminar Candidatos” y “Descargar Curriculum”.

Si pulsamos sobre añadir un nuevo candidato, nos aparecerá el siguiente formulario.

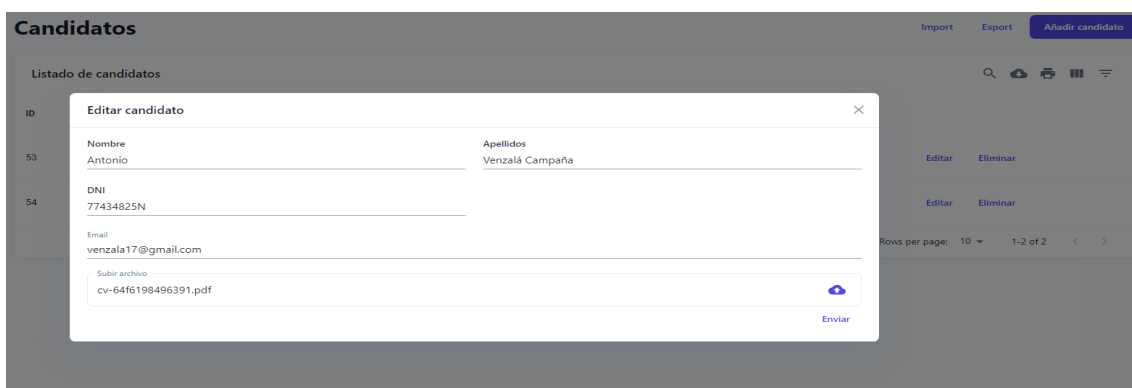


A screenshot of a web application modal titled "Añadir candidato". The modal has a close button (X) in the top right corner. It contains several input fields: "Nombre" and "Apellidos" (split into two columns), "DNI", and "Email". Below these fields is a file upload section with a "Subir archivo" button and a cloud icon. At the bottom right, there is a blue "Enviar" button.

Ilustración 80 . Modal Añadir Candidatos

Es necesario rellenar todos los campos del formulario para que sea validado y almacenado en la base de datos. Si al enviar el formulario se muestra una notificación de color verde con el mensaje “Candidato añadido correctamente”, significara que el proceso ha sido fructifero y que el candidato ha sido validado y almacenado. Por el contrario, si aparece una notificación roja de error, significa que algún dato aportado es erróneo o no cumple con los requisitos, por lo que será necesario revisarlo y re-enviarlo.

Al igual que en el proceso de añadir candidato, si pulsamos sobre el botón Editar, se nos mostrará un formulario con los datos del candidato. El proceso de edición es el mismo que el de añadir candidato, por lo que se realizan las mismas validaciones.



A screenshot of a web application showing a list of candidates and an "Editar candidato" modal. The modal is titled "Editar candidato" and has a close button (X). It contains input fields for "Nombre" (Antonio), "Apellidos" (Venzalá Campaña), "DNI" (77434825N), and "Email" (venzala17@gmail.com). Below these fields is a file upload section with a "Subir archivo" button and a cloud icon, showing a file named "cv-64f6198496391.pdf". At the bottom right, there is a blue "Enviar" button. In the background, a table lists candidates with columns for ID, Name, Surname, DNI, Email, and Actions (Editar, Eliminar). The table shows two candidates with IDs 53 and 54.

Ilustración 81 . Modal Editar Candidatos

En el proceso de eliminar candidatos, en el momento de pulsar el botón se mostrará una alerta de confirmación. Dependiendo de la opción que elijamos, se eliminará o no el candidato seleccionado.

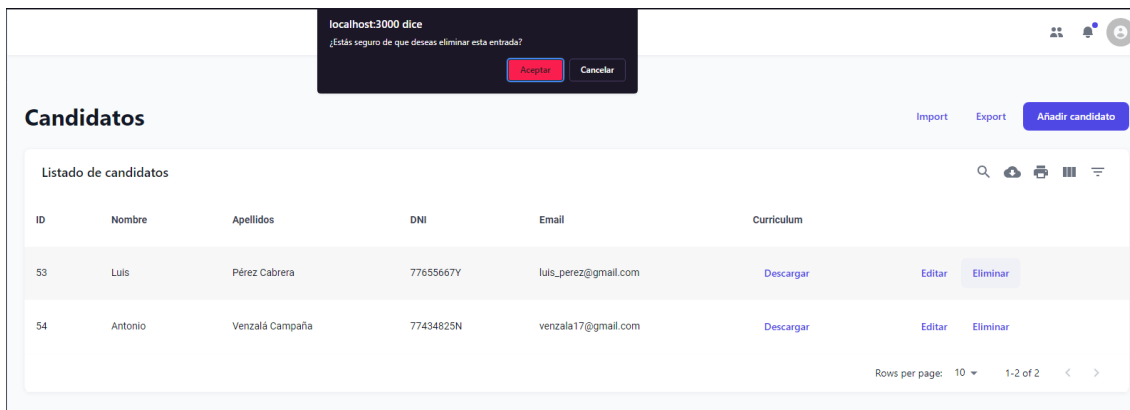


Ilustración 82 . Eliminar candidatos

Por último, todos los candidatos añadidos, salvo que el currículum adjunto se haya perdido o borrado, aparecerán con la opción "Descargar" activa. Si pulsamos sobre el botón se comenzará la descarga de este documento.

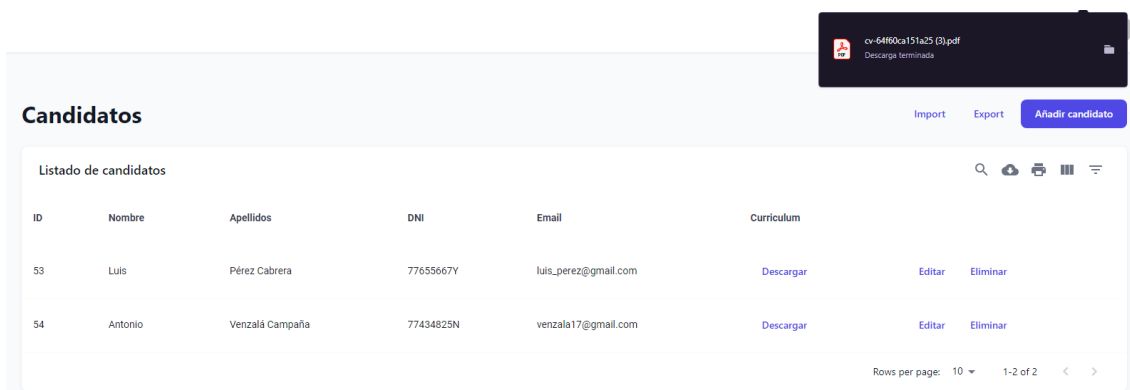


Ilustración 83 . Descargar curriculum vitae

Actualmente, el almacenamiento de estos archivos se está realizando en el mismo directorio del proyecto. En un futuro se implementará un

almacenamiento en la nube usando el S3 de amazon para almacenar esta y más archivos o información importante para la aplicación.

Tras terminar de explicar las funcionalidades que nos aporta el apartado Candidatos, se va a hablar sobre el apartado Ofertas. Si accedemos, en primer lugar se nos mostrará una vista como la que se va a adjuntar a continuación.

4.7.4. Vista ofertas - Candidatos por oferta

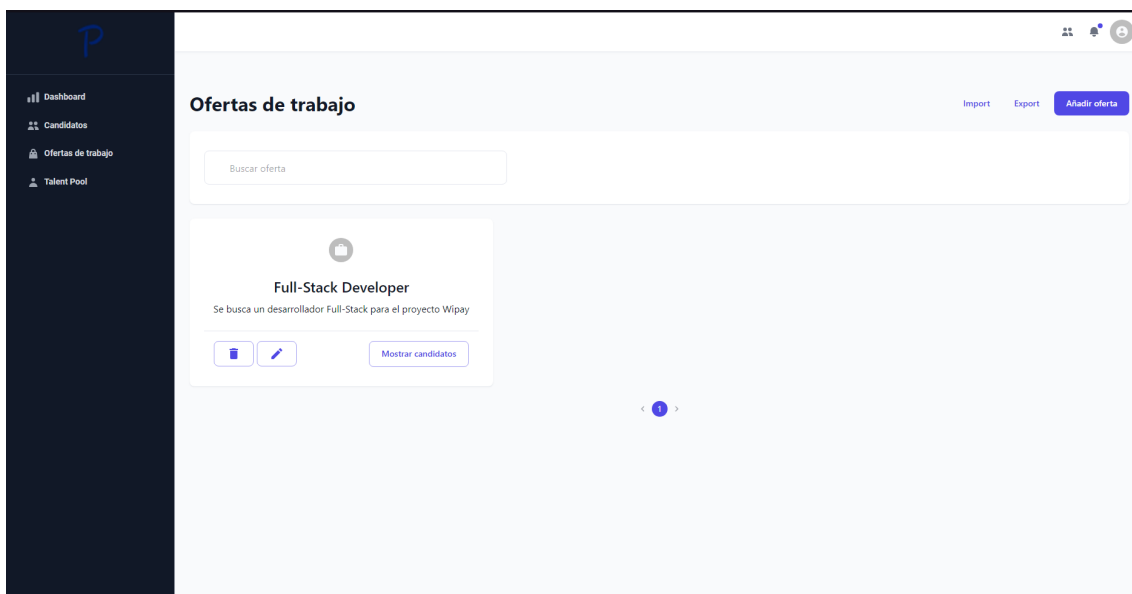


Ilustración 84 . Vista Ofertas de empleo

Dentro de este apartado nos podemos encontrar con un listado de ofertas, mostrado mediante el uso de tarjetas. Además, al igual que con la tabla de datos del apartado anterior, se nos permite realizar las mismas funcionalidades básicas.

Tanto el botón de “Añadir candidato” como el botón de “Editar”, nos abrirá un modal con un formulario simple. Igualmente, se realizará una validación de los datos enviados por parte de nuestra API, mostrando una notificación de confirmación o error dependiendo de cómo haya finalizado el proceso.

The image shows two side-by-side form modals. The left modal is titled 'Añadir oferta' (Add offer) and contains two input fields: 'Nombre de la oferta' (Offer name) and 'Descripción' (Description). The right modal is titled 'Editar oferta' (Edit offer) and contains the same two input fields, but with pre-filled text: 'Full-Stack Developer' for the name and 'Se busca un desarrollador Full-Stack para el proyecto V' for the description. Both modals have a blue 'Enviar' (Send) button at the bottom right.

Ilustración 85 . Formularios Ofertas de empleo

El botón eliminar abrirá el mismo mensaje de confirmación, realizando la misma operación que el caso anterior.

The image is a screenshot of the PetroTalent web application. At the top, there is a dark notification banner from 'localhost:3000' asking for confirmation to delete an offer, with 'Aceptar' (Accept) and 'Cancelar' (Cancel) buttons. Below the banner, the page title is 'Ofertas de trabajo' (Job offers). On the right, there are links for 'Import', 'Export', and a blue 'Añadir oferta' (Add offer) button. A search bar labeled 'Buscar oferta' is present. The main content area displays a job offer card for 'Full-Stack Developer' with the description 'Se busca un desarrollador Full-Stack para el proyecto Wipay'. The card includes icons for deleting and editing the offer, and a 'Mostrar candidatos' (Show candidates) button. At the bottom, there is a pagination control showing '1'.

Ilustración 86 . Formularios Ofertas de empleo

Si accedemos a la opción “Mostrar candidatos de una oferta”, se nos redirigirá al tablero Kanban asociado a la oferta seleccionada. En este tablero se nos mostrarán las distintas etapas de selección de candidatos junto con los candidatos asociados a esa oferta.

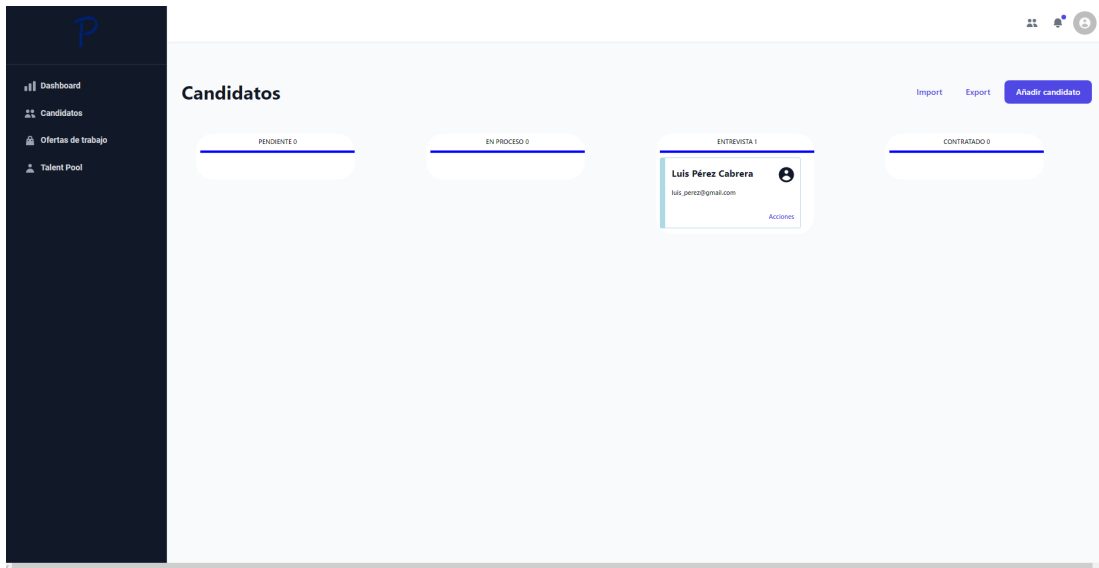


Ilustración 87 . Vista Candidatos por oferta de empleo

Como podemos observar, se han generado una serie de columnas que representan cada una de las etapas del proceso de selección de talento. En esta vista se nos da la posibilidad de editar la etapa en la que se encuentran cada uno de los candidatos, para mantener un control del estado de cada uno. Para llevar esto a cabo, se ha implementado la lógica de Drag and Drop, en donde cada una de las tarjetas puede ser arrastrada entre las distintas etapas de selección.

Cada vez que arrastramos y soltamos una tarjeta en una columna distinta a la actual, se nos notificará una alerta de confirmación. Si esta es confirmada, se comenzará el proceso de edición del candidato. Si todo ha finalizado correctamente se nos notificará con un mensaje de confirmación de edición y se mostrará al candidato en su nueva columna. Por el contrario, si

este proceso no finaliza correctamente, el candidato no se moverá de la columna y se notificará un mensaje de error.

También, en el momento de mover a los candidatos entre las distintas etapas del proceso, se notificará al candidato mediante el envío de un correo electrónico al correo facilitado informando del estado de su solicitud. Actualmente, al no tener un servicio SMTP donde implementar el envío de correos real, se ha usado la aplicación Mail Trap para simular y testear el envío de correos.

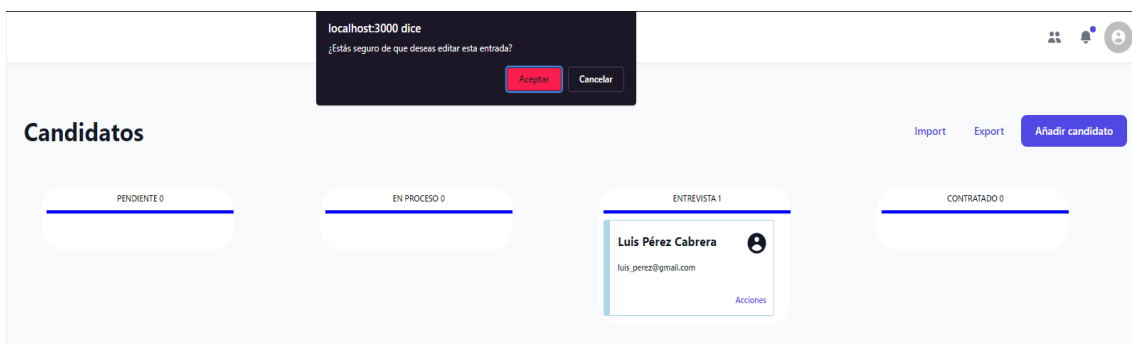


Ilustración 88 . Modificar etapa proceso selección

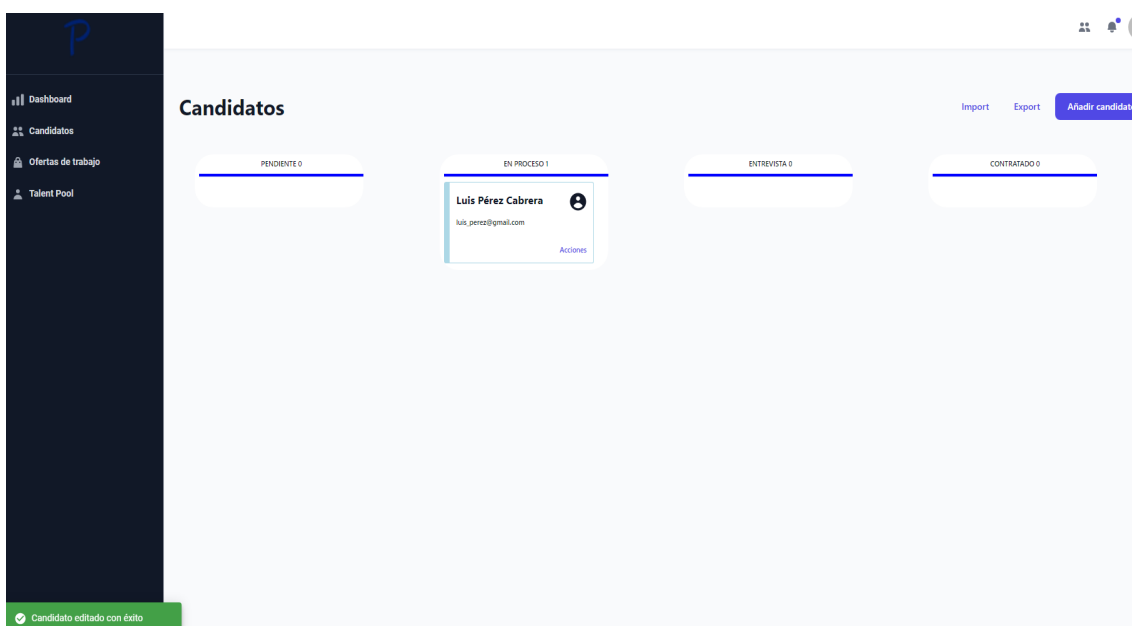


Ilustración 89 . Confirmación modificación etapa proceso selección

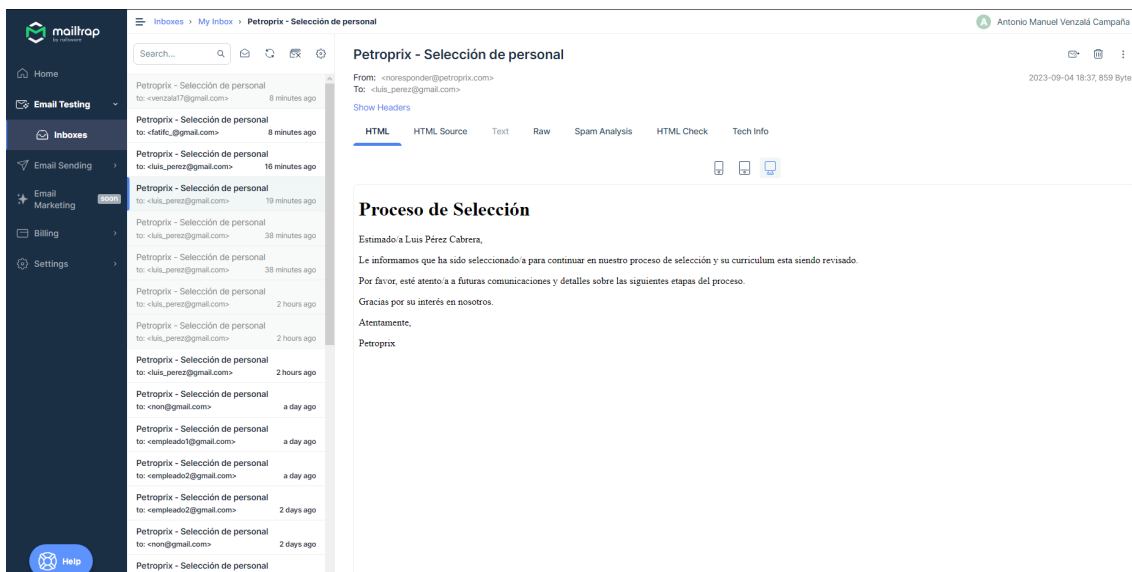


Ilustración 90 . Mail enviado a correo asociado

Existe el caso de mover al candidato a la etapa de entrevista, donde se nos mostrará un formulario para generar una entrevista. Esta entrevista es opcional, y puede ser omitida en el proceso.

A screenshot of a web form titled 'Añadir entrevista' (Add interview). The form has two input fields: 'Fecha' (Date) with a placeholder 'dd/mm/aa' and a calendar icon, and 'Hora' (Time) with a placeholder '--:--' and a clock icon. Below these fields is a blue button labeled 'Enviar' (Send). The form is enclosed in a light gray border with a close button (X) in the top right corner.

Ilustración 91. Formulario entrevista

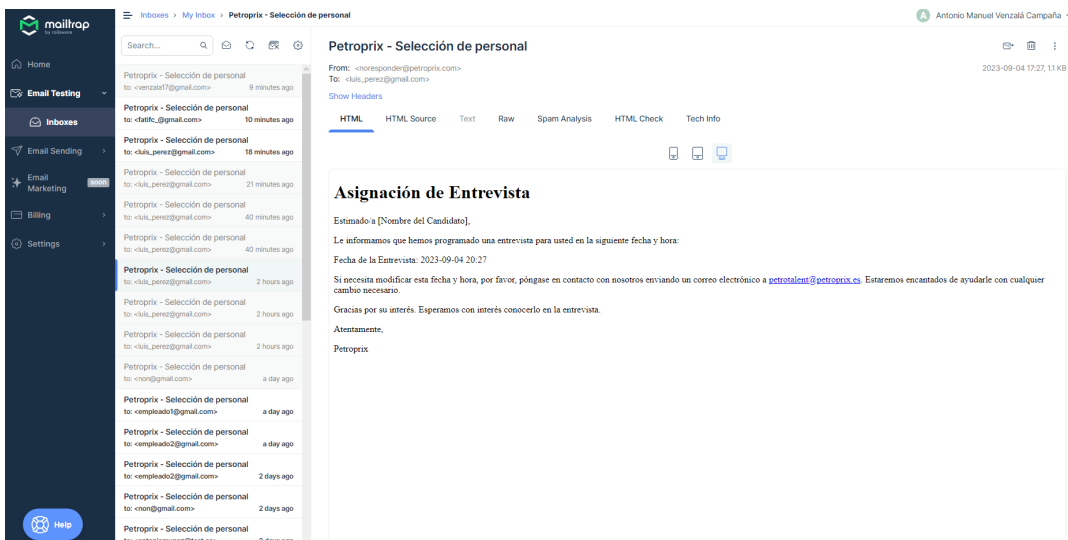


Ilustración 92. Correo notificando entrevista

Además de implementar todas estas funciones, se da la posibilidad al usuario de añadir nuevos candidatos a la oferta de empleo, mediante el uso de una tabla de datos con un selector.

Para añadir candidatos, será suficiente con seleccionar en la tabla aquellos candidatos que queramos asociar a esa oferta y pulsar sobre Añadir. Si todo el proceso ha finalizado correctamente, las tarjetas de los candidatos aparecerán en la etapa “Pendiente”

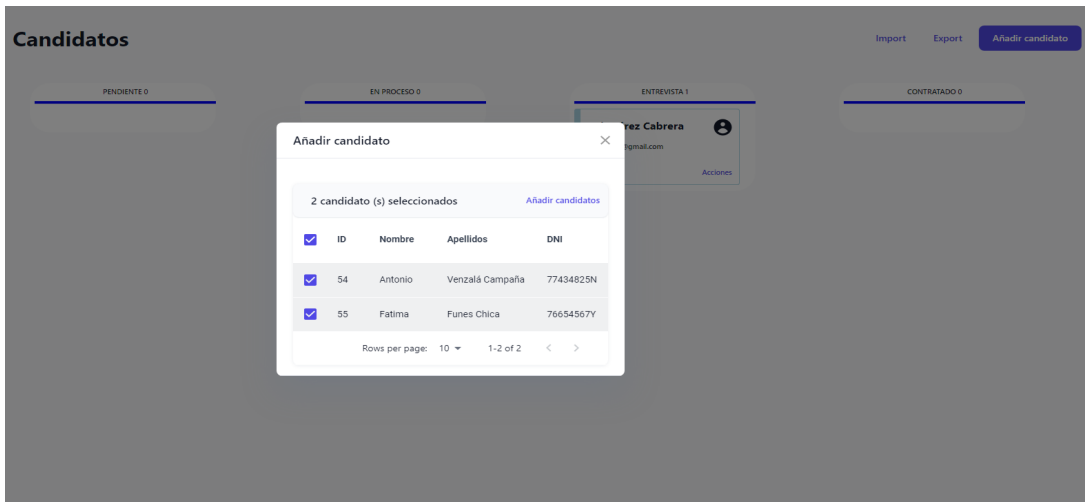


Ilustración 93 . Formulario añadir candidatos a ofertas de empleo

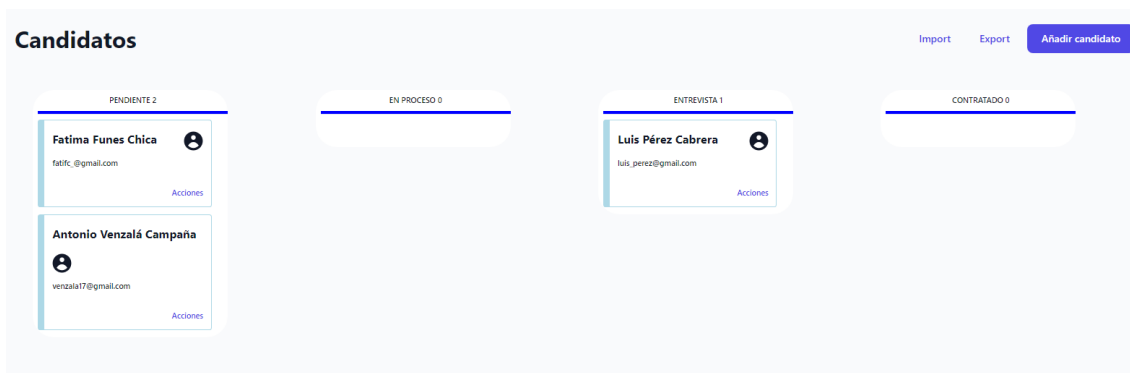


Ilustración 94 . Vista candidatos de una oferta al finalizar añadir

Dentro de las tarjetas de candidatos aparece un botón denominado acciones. Este botón al ser presionado, desplegará un listado de posibles funciones. Actualmente solo está disponible la de eliminar. De la misma manera que en casos anteriores, si confirmamos la eliminación el candidato desaparecerá de la oferta.

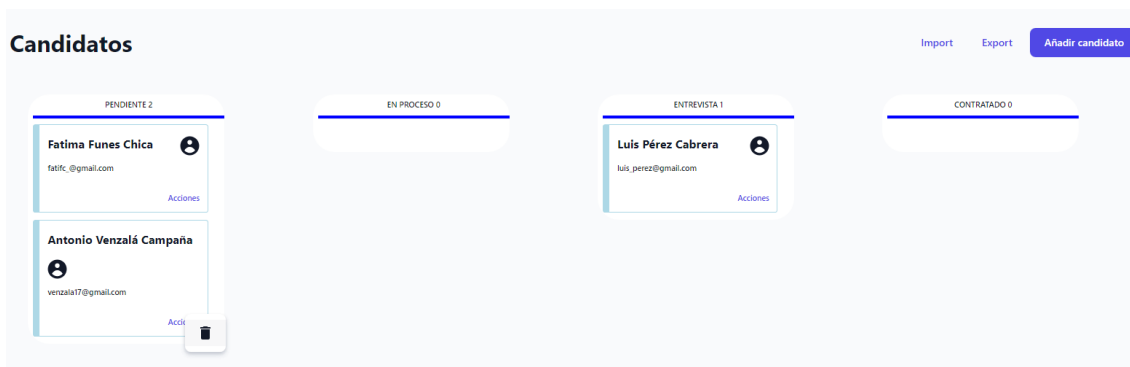


Ilustración 95 . Botón eliminar candidato

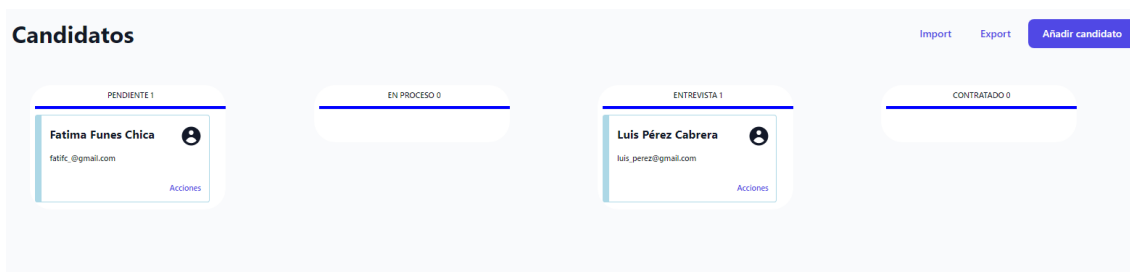
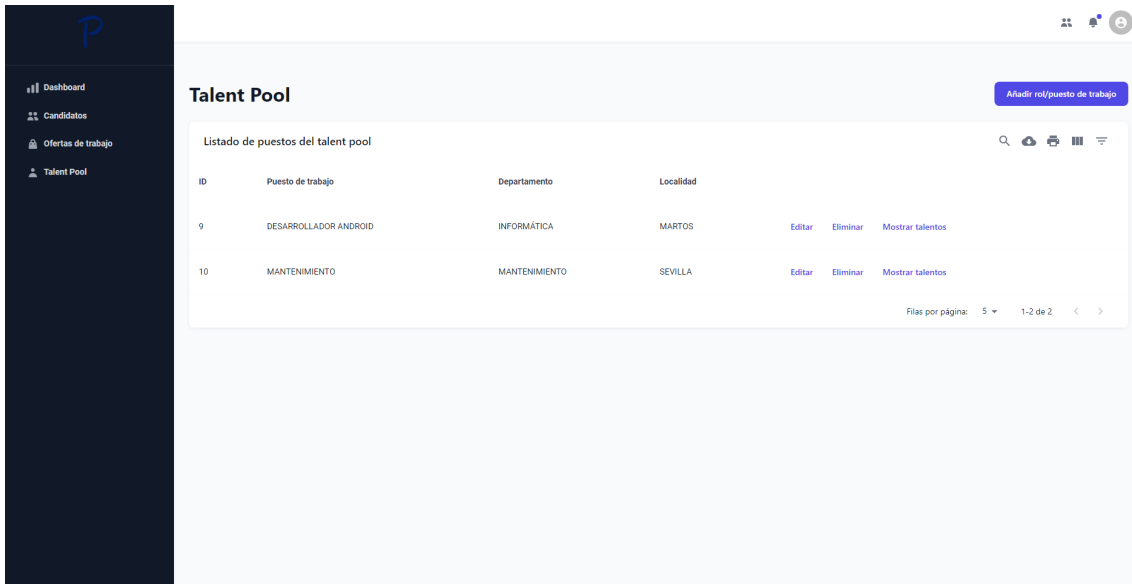


Ilustración 96 . Candidato eliminado de la oferta

4.7.5. Apartado Talent-Pool y Candidatos de un puesto

Por último, si accedemos al apartado del Talent Pool, se mostrará en primer lugar una tabla con todos los puestos de trabajo disponibles en la empresa. Mediante este listado, podremos almacenar aquellos candidatos potenciales, para mantener sus datos de contacto por si a futuro fuese necesario solicitar sus aptitudes y talento.

Al igual que con los candidatos y las ofertas, se aportan las mismas funciones, pudiendo añadir, eliminar, editar los puestos o mostrar los candidatos asociados a esos puestos del talent pool.



ID	Puesto de trabajo	Departamento	Localidad	
9	DESARROLLADOR ANDROID	INFORMÁTICA	MARTOS	Editar Eliminar Mostrar talentos
10	MANTENIMIENTO	MANTENIMIENTO	SEVILLA	Editar Eliminar Mostrar talentos

Ilustración 97 . Listado de puestos del talent pool

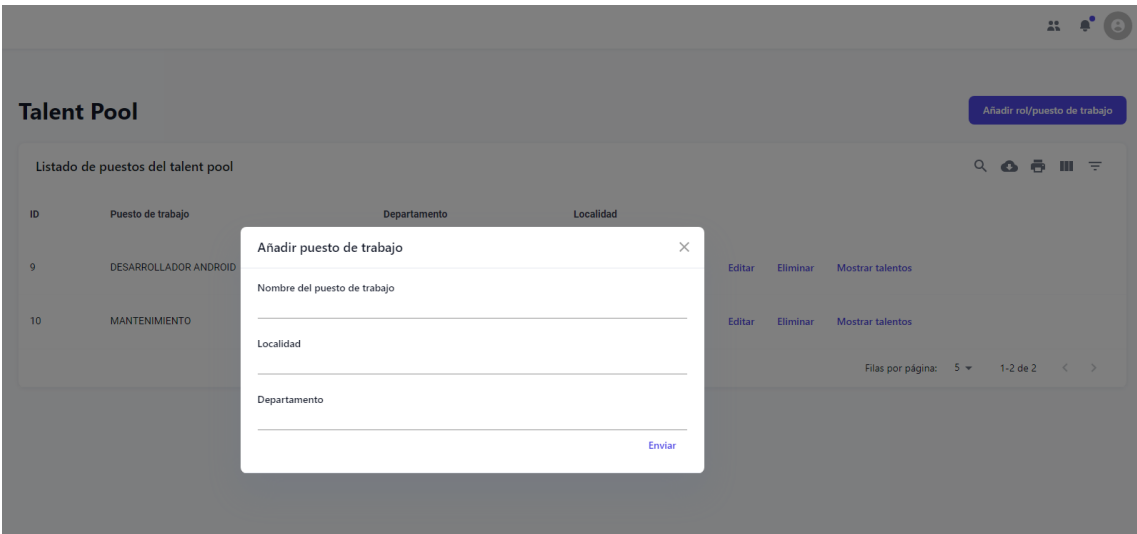


Ilustración 98 . Añadir puesto al talent pool

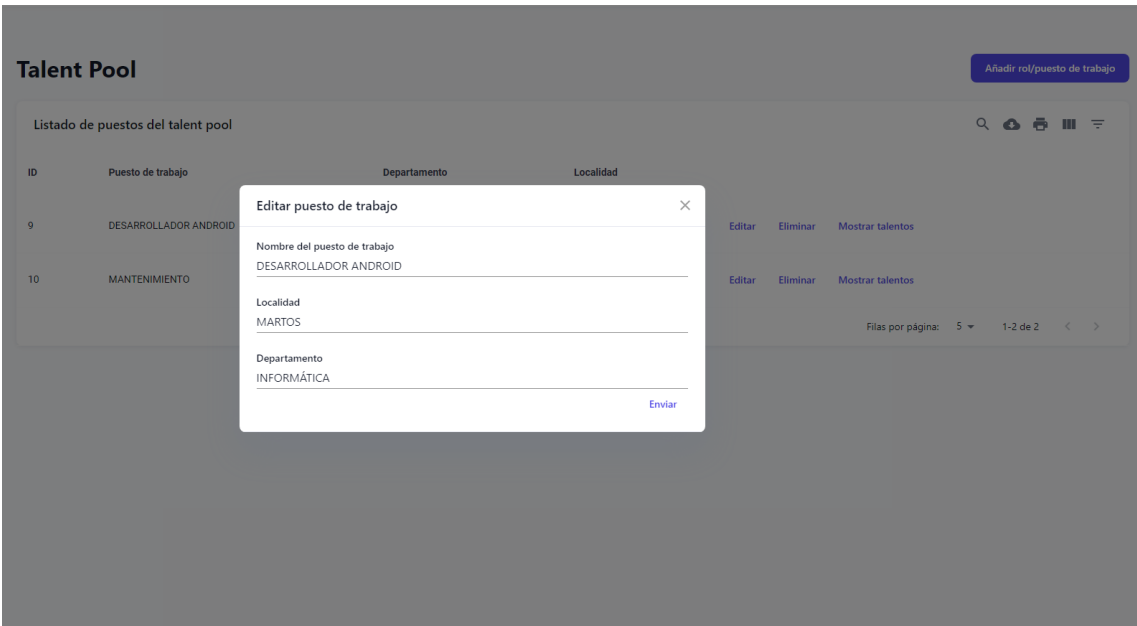


Ilustración 99 . Editar puesto del talent pool

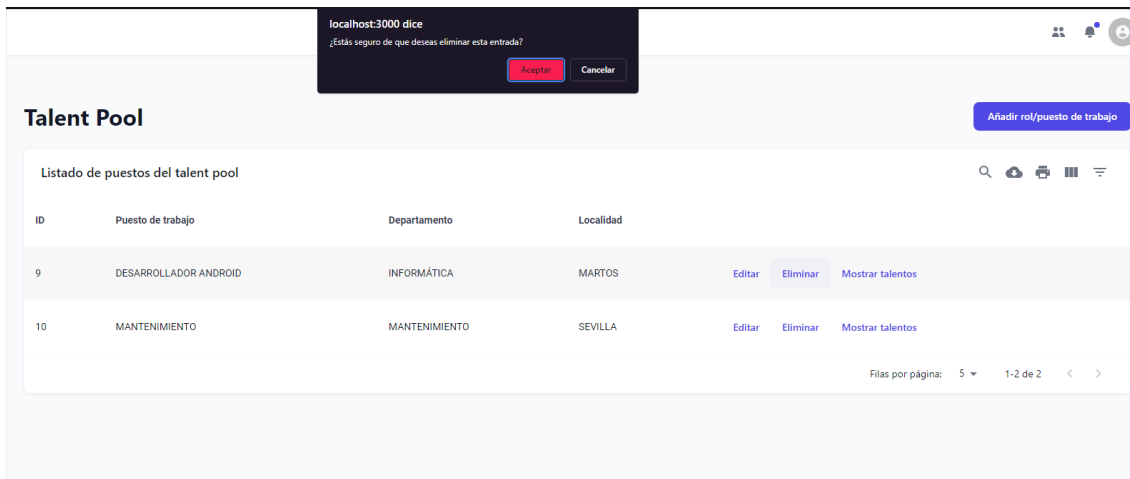


Ilustración 100. Eliminar puesto del talent pool

En el desarrollo de esta aplicación se ha intentado mantener un diseño equilibrado, por ello la mayoría de vistas intentan mantener la misma estética, para que así el usuario pueda familiarizarse mucho más rápido con la aplicación y no tener menús complicados que compliquen la tarea al usuario. Es por eso, que todas las vistas utilizan un datatable para la gestión de datos.

Aparte de las funcionalidades típicas, se da acceso a otro apartado mediante los botones de Mostrar Candidatos. Al pulsar sobre él, se nos cargará una vista con los candidatos almacenados para ese tipo de puesto. De esta manera será mucho más fácil para el responsable encontrar usuarios ya estudiados y realizarles una oferta de forma más rápida y sencilla

Al igual que las demás vistas, esta tiene un diseño parecido, con funcionalidades similares. A continuación se muestran imágenes con cada una de ellas.

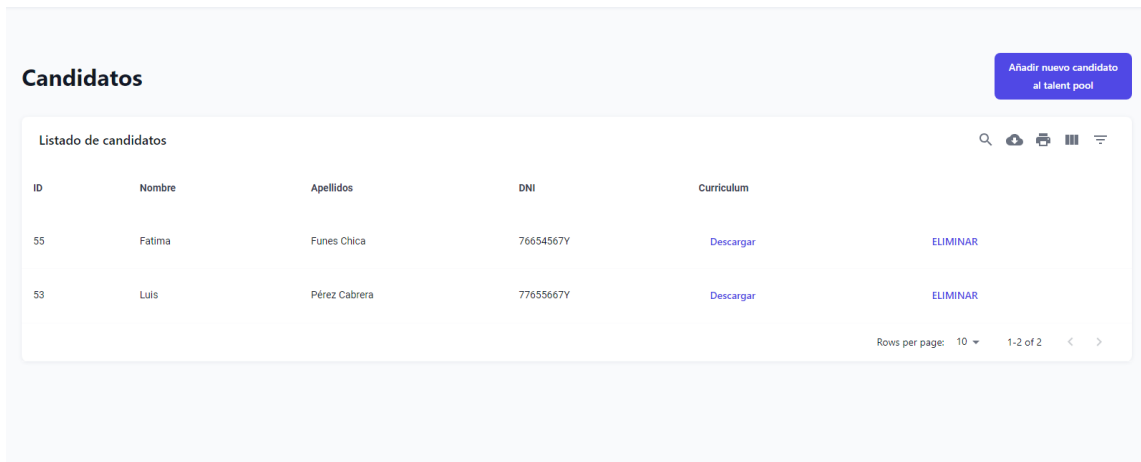


Ilustración 101 . Vista candidatos por puesto del talent pool

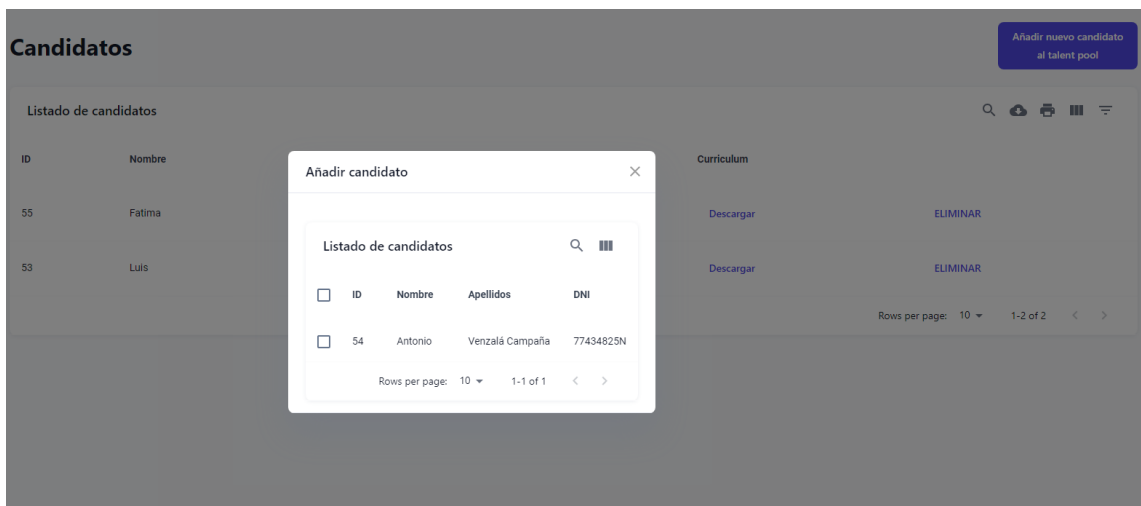


Ilustración 102 . Añadir candidatos a puesto del talent pool

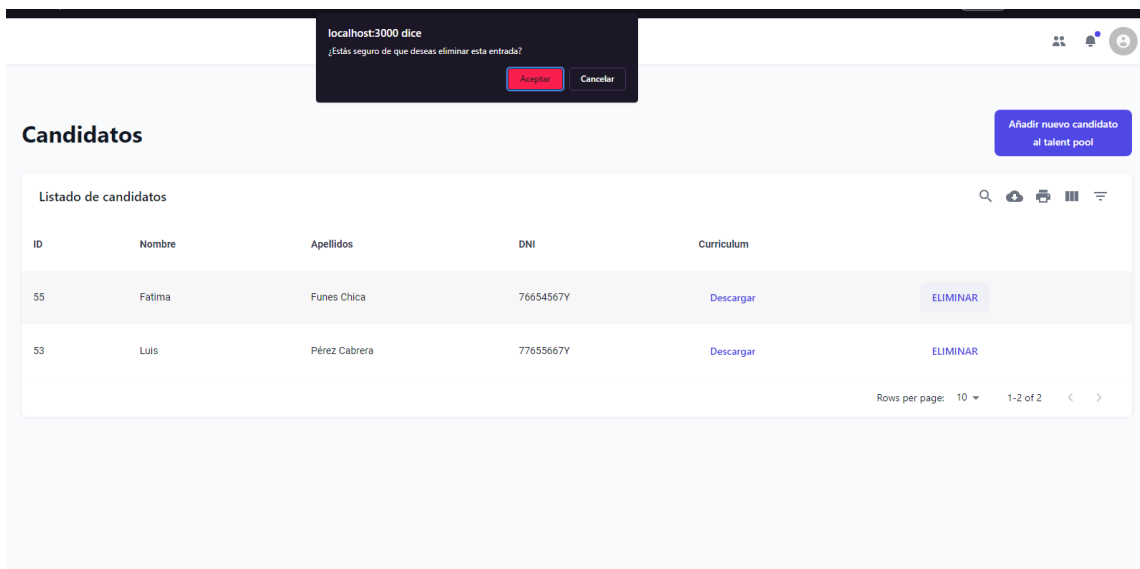


Ilustración 103 . Eliminar candidato de puesto del talent pool

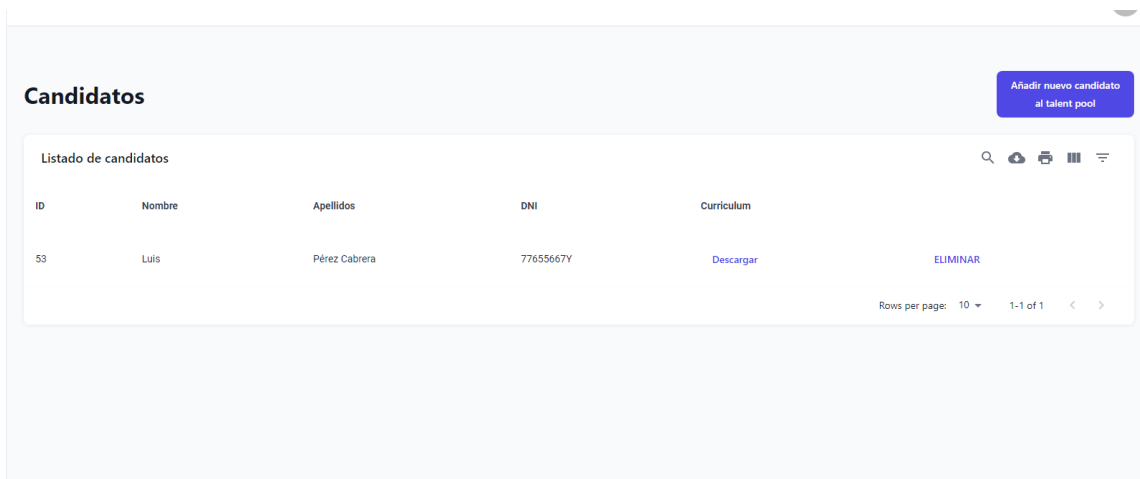


Ilustración 104 . Confirmación eliminación candidato puesto del talent pool

5. CONCLUSIÓN

5.1. Conclusiones

Para terminar, me gustaría comentar que el desarrollo de este Trabajo Fin de Grado ha sido un proceso muy enriquecedor para mí ya que me ha ayudado a entender un poco mejor todo el proceso tan costoso y difícil que tiene detrás el desarrollo de una aplicación, además de haber podido reciclar los conocimientos sobre las tecnologías actuales de desarrollo web.

Por otro lado, el desarrollo de esta aplicación me ha servido para poder aplicar en una situación real todos los conocimientos obtenidos sobre la gestión de proyectos y la Ingeniería del Software en un proyecto real, entendiendo la importancia de la misma.

Si no se aplican los principios de la Ingeniería del Software de forma efectiva, es posible que el proyecto no hubiese finalizado de forma exitosa. Es por ello que es necesario destacar la importancia de la disciplina, planificación y la atención a los detalles, si queremos obtener soluciones de calidad que satisfagan las necesidades de los clientes.

Remarcar que este no es el final del proyecto, sino que este se irá desarrollando y mejorando, obteniendo más funcionalidades y mejoras, pero siempre intentando seguir y aplicando los principios de la Ingeniería del Software.

5.2. Posibles desarrollos futuros

Como es normal, en este tramo de tiempo tan corto no es posible realizar una aplicación web robusta y con infinidad de funcionalidades. Por ello, muchas de las funcionalidades que se pensaron al principio del proyecto no han podido ser desarrolladas y se han quedado para realizar en futuras versiones. Entre estos desarrollos nos encontramos:

- **Desarrollo de un apartado “Talento” en la aplicación web Petroprix:**

Como es normal, para que esta aplicación tenga sentido y sea más automática, es necesario que los propios usuarios sean capaces de enviarnos su información y curriculum para entrar a las ofertas de empleo. Por ello, se pensó en desarrollar un nuevo apartado en la aplicación de petroprix, en donde se daría acceso a un gran listado de ofertas actuales y un formulario básico asociado a cada oferta. De esta forma, los usuarios que rellenen y envíen los formularios automáticamente generarían su registro en la aplicación PetroTalent y los responsables de Recursos Humanos podrían obtener su información para ser tratada.

- **Implementación de APIs de los portales de empleo más conocidos**

Otra idea que se barajó como posible requisito fue la implementación de estas APIs, de forma que mediante nuestra aplicación sería posible realizar ofertas de empleo que se suban automáticamente a estos portales, recibiendo de ella los candidatos y creando un ecosistema entre estos portales y la aplicación.

Este desarrollo se descartó para esta primera versión ya que era necesario en primer lugar tener una base funcional y eficiente, dejando esto como un desarrollo secundario.

Sin embargo, es muy importante e interesante implementar esto ya que facilita mucho el trabajo de los responsables de RRHH, teniendo que sólo gestionar una aplicación en vez de consultar varias.

- **Implementación de servicios de almacenamiento en la nube y envío de correo.**

Otro desarrollo interesante y muy necesario es el de almacenamiento y gestión de servicios en la nube. En mi caso, la opción del servicio S3 de Amazon Web Services es la más interesante, ya que permite el almacenamiento de cualquier tipo de información u objeto. Mediante este servicio se permite almacenar desde la información de la aplicación, como currículums, hasta el almacenamiento de copias de seguridad de nuestros datos, implementación de análisis de datos, etc..

En la actualidad, con la gran carga de datos que se enfrentan las aplicaciones, es necesario implementar y depender de un servicio de este calibre.

Por otro lado, al igual que para el almacenamiento, es necesario implementar algún servicio o API que nos permita enviar correos en producción. En mi caso, ya que estamos basando el almacenamiento en los servicios de Amazon, sería interesante implementar la API Amazon SES para el envío de correos.

Esta API, denominada Simple Email Service, como bien indica su nombre nos facilita la vida a la hora de realizar el envío de correos de producción.

Solamente con la implementación de esta API, que existe la documentación de Amazon, y utilizando sus métodos, es posible enviar correos electrónicos a cualquier persona.

- **Tareas de reclutamiento realizadas por Inteligencias Artificiales.**

Como ya se ha expuesto a lo largo del trabajo, las tareas de reclutamiento son muy difíciles y costosas de realizar, debido al tratamiento de una gran cantidad de datos. De esta manera, se propone facilitar la ayuda de los responsables mediante la implementación de Inteligencia Artificial para desarrollar alguna de las tareas.

Entre estas tareas nos podemos encontrar:

- Lectura e interpretación de CV → Mediante una IA, desarrollada mediante el campo del procesamiento del lenguaje natural (PLN), es posible que en pocos segundos, la tecnología sea capaz de analizar y organizar todos los datos y curriculums de nuestra aplicación. De esta manera se reduciría notablemente la carga de trabajo de los responsables.
- Motor de búsqueda de candidatos → De la misma manera que en el caso anterior, gracias a la tecnología se podrá realizar en el Talent Pool de nuestra aplicación un buscador de candidatos, en donde introduciendo una serie de filtros, en pocos segundos aparecerán algunos perfiles que podrían satisfacer las necesidades impuestas.

Como podemos observar, esta implementación es muy interesante pero se dejará para desarrollos futuros, ya que es secundaria actualmente.

ANEXO

En la entrega, se han adjuntado los siguientes contenidos:

- Zip “Petroprix Talent Front” que contiene el código fuente de la aplicación React.
- Zip “ Petroprix Talent Back” que contiene el código fuente la API Symfony.
- Archivo JSON con la colección de peticiones de Postman.
- Memoria del proyecto en formato PDF.
- Documentación de entrega del proyecto.

Si se quisiese desplegar las aplicaciones, se recomienda seguir estos dos videos en donde se explican como instalar todos los componentes necesarios para poder desplegarlo.

- Symfony → <https://www.youtube.com/watch?v=9OUvACXdxFs>
- React → <https://www.youtube.com/watch?v=wMDBblo0zEo>

Además, es necesario realizar una migración de la BBDD para poder hacer funcionar la API. Se tienen que realizar los siguientes comandos en consola:

- `php bin/console make:migration`
- `php bin/console doctrine:migrations:migrate`

Por último, para poder realizar login en la aplicación es necesario desplegar ambos proyectos, y en la tabla user generada mediante la migración, se introducirá un correo y password.

Para generar el password se usará el comando siguiente:

- `php bin/console security:hash-password`

De esta manera, obtendremos un usuario con unas credenciales válidas para acceder a la aplicación y poder seguir el manual expuesto en el punto 4.7. Pruebas/Manual.

6. BIBLIOGRAFÍA

[1] Evolución del reclutamiento de talento a lo largo de la historia:
<https://www.gbsrecursoshumanos.com/blog/evolucion-del-reclutamiento/> (2 de Febrero 2023)

[2] ¿Qué son los sistemas HRMS? Herramientas HRMS:
<https://www.docuSign.mx/blog/HRMS> (5 de Febrero 2023)

[3] Herramienta HRMS 1. Factorial
<https://elreferente.es/directorio/factorial-recursos-humanos/#:~:text=Factorial%20es%20un%20software%20de.personal%2C%20control%20de%20gastos%20y> (5 de Febrero 2023)

[4] Herramienta HRMS 2. Sage
[https://www.sage.com/es-es/sage-business-cloud/hr/?utm_gachannelgroup=PaidSearch&utm_source=GOOGLE&utm_medium=PaidSearch&utm_campaign=ES%7CGoogle%7CSmall%7CGeneric_SageHR/Main\(MaxConv-NA\)PerformanceMax_SMASAHr_&utm_delivery=Paid&gad=1&gclid=CjwKCAjwUOIBhBREiwA7agf1h65kOPZHEtEOhF6_ZRuDIO1DndcyBD6LNz-QfVc7o426uopO3S-SBoCL5EQAvD_BwE&gclidsrc=aw.ds](https://www.sage.com/es-es/sage-business-cloud/hr/?utm_gachannelgroup=PaidSearch&utm_source=GOOGLE&utm_medium=PaidSearch&utm_campaign=ES%7CGoogle%7CSmall%7CGeneric_SageHR/Main(MaxConv-NA)PerformanceMax_SMASAHr_&utm_delivery=Paid&gad=1&gclid=CjwKCAjwUOIBhBREiwA7agf1h65kOPZHEtEOhF6_ZRuDIO1DndcyBD6LNz-QfVc7o426uopO3S-SBoCL5EQAvD_BwE&gclidsrc=aw.ds)
(6 de Febrero 2023)

[5] ¿Qué son los sistemas TMS?
https://en.wikipedia.org/wiki/Talent_management_system (7 de Febrero 2023)

[6] Herramienta TMS 1. SAP
https://www.sap.com/spain/products/hcm/talent-management.html?campaigncode=crm-ya22-int-1517073&source=ppc-es-google_ads-search-71700000104590096-58700008223232861-hxm_hxm-x-x-x&dfa=1&gad=1&gclid=CjwKCAjw-b-kBhB-EiwA4fvKrOqA4jcuPbKzy879488xe-aTmUz-Dg_LplJJ_AKu_bQzZSYoDfjG0xoCFEgQAvD_BwE&gclidsrc=aw.ds (7 de Febrero 2023)

[7] Herramienta TMS 2. Oracle HCM Cloud
<https://www.oracle.com/es/human-capital-management/talent-management/> (7 de Febrero 2023)

[8] ¿Qué son los sistemas ATS?
<https://empresas.infoempleo.com/hrtrends/ats-reclutamiento/> (8 de Febrero 2023)

[9] Herramienta ATS 1. Bizneo
<https://www.bizneo.com/reclutamiento-seleccion-ats/> (8 de Febrero 2023)
<https://www.bizneo.com> (9 de Febrero 2023)

[10] Herramienta ATS 2. Taleo
<https://www.oracle.com/es/human-capital-management/taleo/> (9 de Febrero 2023)

[11] Información sobre el editor de código Sublime Text

https://es.wikipedia.org/wiki/Sublime_Text (22 de Mayo 2023)

[12] Información sobre el editor de código Atom

<https://www.drauta.com/atom-un-ide-para-el-desarrollador-web> (23 de Mayo 2023)

[13] Comparativa entre VSCode y Sublime Text

<https://www.versuante.com/visual-studio-code-o-sublime-text-que-editor-es-mejor/> (22 de Mayo 2023)

[14] Herramienta Control de versiones. Git :

<https://www.sivsa.com/site/ventajas-controlador-versiones-git-en-programacion/> (23 de Mayo 2023)

[15] Ventajas de Git

<https://www.campusmvp.es/recursos/post/que-es-git-ventajas-e-inconvenientes-y-por-que-deberias-aprenderlo-bien.aspx#:~:text=Git%20ofrece%20varias%20ventajas%20frente,operaciones%20de%20control%20muy%20rápidas.> (24 de Mayo 2023)

[16] Herramienta Control de versiones. Mercurial :

<https://www.adictosaltrabajo.com/2010/09/17/mercurial/> (24 de Mayo 2023)

[17] Herramienta de gestión de proyectos. Trello:

<https://trello.com/es/tour#:~:text=¿Qué%20es%20Trello%3F,las%20necesidades%20de%20tu%20equipo.> (24 de Mayo 2023)

[18] Herramienta de gestión de proyectos. Jira:

<https://www.atlassian.com/es/software/jira/pricing> (24 de Mayo 2023)

[19] Frameworks Back-End. Symfony:

<https://symfony.es/documentacion/> (20 de Julio 2023)

<https://www.kiwop.com/blog/symfony-todo-sobre-este-potente-framework-php> (25 de Mayo 2023)

[20] Frameworks Back-End. Laravel:

<https://levelup.gitconnected.com/should-you-use-php-laravel-in-2023-a7930d4afd0b>
<https://www.kiwop.com/blog/cuando-escoger-symfony-como-framework-para-tu-proyecto-web> (25 de Mayo 2023)

[21] Frameworks Back-End. Comparativa:

<https://www.kiwop.com/blog/la-comparativa-definitiva-laravel-vs-symfony> (25 de Mayo 2023)

<https://kinsta.com/es/blog/symfony-vs-laravel/#:~:text=Symfony%20es%20más%20adecuado%20para,un%20plazo%20más%20corto.> (25 de Mayo 2023)

[22] Frameworks Front-End. React:

https://www.hostinger.es/tutoriales/que-es-react#¿Por_que_utilizar_React (27 de Mayo 2023)

<https://www.solbyte.com/blog/react-js-ventajas-e-inconvenientes/> (27 de Mayo 2023)

[23] Frameworks Front-End. Angular:

https://www.hostinger.es/tutoriales/que-es-angular#¿Que_es_AngularJS (27 de Mayo 2023)

<https://www.campusmvp.es/recursos/post/las-5-principales-ventajas-de-usar-angular-para-crear-aplicaciones-web.aspx> (27 de Mayo 2023)

[24] ¿Qué es la Ingeniería del Software?

https://es.wikipedia.org/wiki/Ingeniería_de_software (29 de Mayo 2023)

[25] Evolución de la Ingeniería del Software

<https://www.sutori.com/es/historia/evolucion-de-la-ingenieria-de-software--GiNbtP6biBDQZ8XP8QZvjDkD> (29 de Mayo 2023)