Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA DE SEGOVIA

Grado en Ingeniería Informática de Servicios y Aplicaciones

App para la asignación arbitral SegoFutsal

Alumno: Ismael García Palomino

Tutor: Luis Ignacio Sebastián Martín

App para la asignación arbitral SegoFutsal

Ismael García Palomino

17 de junio de 2025

Índice general

Lista de figuras v

Lista de tablas vii

Resumen xv

Abstract xvii

I Memoria del Proyecto 1

1. Descripción del Proyecto 3
1.1. Introducción . . . 3
1.2. Objetivos del trabajo . . . 4
1.2.1. Objetivos generales . . . 4
1.2.2. Objetivos específicos . . . 4
1.2.3. Alcance y limitaciones . . . 5
1.3. Estado del arte . . . 5
1.4. Beneficios . . . 6

2. Metodología 7
2.1. Proceso de desarrollo . . . 7
2.2. Herramientas utilizadas . . . 8
2.2.1. Microsoft Lists . . . 9
2.2.2. Microsoft Power Apps . . . 10
2.2.3. Microsoft Power Automate . . . 10
2.2.4. Microsoft Sharepoint . . . 11
2.2.5. Microsoft Copilot . . . 11
2.2.6. Microsoft Excel . . . 12
2.3. Arquitectura . . . 13
2.3.1. Arquitectura Lógica . . . 13
2.3.2. Arquitectura Física . . . 13
2.4. Definición de siglas y abreviaturas . . . 14

3. Planificación **17**
3.1. Estimación del esfuerzo 17
3.2. Planificación temporal 20
3.3. Presupuesto económico 22
3.3.1. Hardware 22
3.3.2. Software 22
3.3.3. Recursos humanos 24
3.3.4. Presupuesto total 24
3.4. Riesgos 25

4. Conclusiones **29**
4.1. Conclusión 29
4.2. Mejoras 30

II Documentación técnica **33**

5. Análisis **35**
5.1. Actores 35
5.2. Requisitos 35
5.2.1. Requisitos de negocio 36
5.2.2. Reglas de negocio 36
5.2.3. Requisitos de usuario 36
5.2.4. Requisitos funcionales 37
5.2.5. Requisitos no funcionales 37
5.2.6. Restricciones 40
5.2.7. Suposiciones 40
5.3. Casos de uso 41
5.3.1. Subsistema gestionar árbitros 42
5.3.2. Subsistema ver partidos 46
5.3.3. Subsistema asignar de árbitros 50

6. Diseño **57**
6.1. Arquitectura lógica 57
6.2. Arquitectura física 58
6.3. Diagrama de entidad-relación 58
6.4. Modelo lógico de datos 58

7. Implementación **63**
7.1. Pantallas 63
7.1.1. Home 63
7.1.2. Gestión Árbitros 65
7.1.3. Nuevo Árbitro 65
7.1.4. Ver Árbitro 67

7.1.5. Editar Árbitro 68
7.1.6. Gestión Partidos 68
7.1.7. Ver Partidos 70
7.1.8. Asignar Árbitros 70
7.2. Flujos Lógicos 73
7.2.1. Crear Partidos 73
7.2.2. Asignaciones 74
7.2.3. Enviar Correos 78
7.3. Conclusión 79

8. Pruebas **83**
8.1. Pruebas de caja negra 83
8.2. Pruebas de caja blanca 92
8.3. Pruebas de seguridad 94
8.4. Pruebas de rendimiento 95

III Manuales de la Aplicación **97**

9. Manual de Instalación **99**

10. Manual de Usuario **101**
10.1. Inicio 101
10.2. Gestión de partidos 101
10.3. Gestión de árbitros 102
10.4. Asignar árbitros a partidos 103

Bibliografía **109**

Índice de figuras

2.1. Modelo de desarrollo incremental 8
2.2. Logotipos de la tecnología utilizada 12
2.3. Arquitectura física 14

3.1. Cronograma 23

5.1. Subsistema gestionar árbitros 46
5.2. Subsistema ver partidos 50
5.3. Subsistema asignar árbitros 55

6.1. Modelo Entidad-Relación 59

7.1. Conexiones en Microsoft Power Apps 74
7.2. Inicio del flujo asignar árbitros 75
7.3. Fijar fecha a los partidos 75
7.4. Paralelización de los tres horarios de los partidos 76
7.5. División de árbitros por categoría y comprobación de si hay suficientes . . 77
7.6. Designación arbitral 78
7.7. Designación arbitral para los partidos de categoría benjamín 79
7.8. Copia de seguridad de la jornada 80
7.9. Ejemplo de una copia de seguridad 80
7.10. Flujo para enviar los correos a los árbitros 81
7.11. Detalle del correo enviado 82

8.1. Tiempo de ejecución del flujo de asignaciones 95

9.1. Compartir lista con otros usuarios 99
9.2. Compartir SegoFutsal con otros usuarios 100

10.1. Vista home de SegoFutsal 102
10.2. Vista de la gestión de partidos 103
10.3. Vista de detalle de los partidos 104
10.4. Vista de la gestión de los árbitros 104
10.5. Vista del formulario para agregar árbitros 105
10.6. Vista de detalle de los árbitros 105

10.7. Vista de las designaciones arbitrales 106
10.8. Pop-up durante la ejecución del flujo 106
10.9. Pop-up una vez ha finalizado el flujo 107

Índice de cuadros

2.1. Abreviaturas y sus significados 15

3.1. Puntos de función sin ajustar 19
3.2. Factores de ajuste 20
3.3. Resumen de la planificación temporal 22
3.4. Presupuesto del hardware 24
3.5. Presupuesto del software 24
3.6. Presupuesto de los recursos humanos 25
3.7. Presupuesto final del proyecto 25
3.8. R-01. Pérdida de datos 26
3.9. R-02. Falta de conformidad con requisitos 26
3.10. R-03. Problemas técnicos con la integración 27
3.11. R-04. Sobrecarga del servidor 27
3.12. R-05. Alteraciones en el calendario de desarrollo 28
3.13. R-06. Cambios reglamentarios 28

5.1. Actores del sistema 35
5.2. Requisitos de negocio 36
5.3. Reglas de negocio 36
5.4. Requisitos de usuario 37
5.5. Requisitos funcionales 38
5.6. Requisitos de eficiencia 38
5.7. Requisitos de usabilidad 39
5.8. Requisitos de mantenibilidad 39
5.9. Requisitos de seguridad 39
5.10. Requisitos de disponibilidad 39
5.11. Requisitos de implementación 40
5.12. Requisitos de interfaces externas 40
5.13. Requisitos de información 41
5.14. Restricciones del sistema 41
5.15. Suposiciones del sistema 41
5.16. CU-01. Crear árbitro 42
5.17. CU-02. Modificar árbitro 43

5.18. CU-03. Ver árbitros 44
5.19. CU-04. Eliminar árbitro 44
5.20. CU-05. Buscar árbitros por nombre 45
5.21. CU-06. Recargar lista de árbitros 46
5.22. CU-07. Ver partidos de la temporada 47
5.23. CU-08. Filtrar partidos por categoría 47
5.24. CU-09. Filtrar partidos por liga 48
5.25. CU-10. Filtrar partidos por jornada 48
5.26. CU-11. Ver detalles de un partido 49
5.27. CU-12. Borrar filtros 50
5.28. CU-13. Seleccionar jornada 51
5.29. CU-14. Marcar árbitros disponibles 52
5.30. CU-15. Asignar árbitros a una jornada 53
5.31. CU-16. Asignar horario a partidos 53
5.32. CU-17. Notificar árbitros por correo 54
5.33. CU-18. Crear copia de seguridad 54

6.1. Entidad Árbitros 60
6.2. Entidad Partidos 60
6.3. Entidad Equipos 60
6.4. Relación entre Árbitros y Partidos 61
6.5. Relación entre Equipos y Partidos 61

8.1. PCN-01 Crear árbitro 83
8.2. PCN-02 Crear árbitro sin nombre 84
8.3. PCN-03 Crear árbitro sin email 84
8.4. PCN-04 Crear árbitro sin categoría 85
8.5. PCN-05 Crear árbitro con el email mal formado 85
8.6. PCN-06 Eliminar árbitro 86
8.7. PCN-07 Editar árbitro correctamente 86
8.8. PCN-08 Editar árbitro sin nombre 86
8.9. PCN-09 Editar árbitro sin email 87
8.10. PCN-10 Editar árbitro sin categoría 87
8.11. PCN-11 Editar árbitro con el formato del email incorrecto 87
8.12. PCN-12 Ver partidos sin filtros 88
8.13. PCN-13 Ver partidos con filtro de jornada 88
8.14. PCN-14 Ver partidos con filtro de categoría 88
8.15. PCN-15 Ver partidos con filtro de liga 89
8.16. PCN-16 Ver partidos con múltiples filtros 89
8.17. PCN-17 Búsqueda de árbitros por nombre correcta 90
8.18. PCN-18 Búsqueda de árbitros por nombre incorrecta 90
8.19. PCN-19 Asignación de árbitros cuando hay disponibilidad suficiente 90
8.20. PCN-20 Asignación de árbitros con falta de ellos para S1 91

8.21. PCN-21 Asignación de árbitros con falta de ellos para S2 91
8.22. PCN-22 Asignación de árbitros con falta de ellos para D1 92
8.23. PCB-01 Lógica interna de la creación, edición y búsqueda de árbitros . . . 92
8.24. PCB-02 Lógica interna del filtrado de partidos 93
8.25. PCB-03 Lógica interna de asignación de árbitros 93
8.26. PCB-04 Lógica interna de creación de la copia de seguridad 94
8.27. PCB-05 Lógica interna del envío de correos a árbitros 94
8.28. PS-01 Acceso a SegoFutsal 95
8.29. PR-01 Tiempo máximo en ejecutar el flujo de asignaciones arbitrales . . . 95

A mi yaya Carmen

Agradecimientos

En primer lugar, quería agradecer a los profesores de la Escuela de Ingeniería Informática de Segovia por demostrarme que, aunque estemos en la universidad, se puede aprender de una forma mucho más cercana. En especial, a mi tutor Luis Ignacio, por su cariño durante los últimos años y su comprensión.

Quiero también agradecer a todos mis compañeros de las prácticas en NTT Data. Todo mi aprendizaje con esta tecnología se lo debo a su tiempo, paciencia, y su buen hacer conmigo. Sin duda alguna, han sido una parte fundamental de este Trabajo de Fin de Grado, y es por ello que les estoy muy agradecido.

No me puedo olvidar tampoco de mis compañeros del doble grado, que han hecho de esta época un sueño hecho realidad. Y por supuesto, de toda la gente que he podido conocer durante mi tiempo en Segovia. Pese al poco tiempo que hemos podido compartir, ha sido el suficiente para que, si pudiera, volvería a repetir esta etapa desde el primer día una y otra vez. ¡Qué bien se vive en Segovia!

Y por último, quiero agradecer a mi familia, y en especial a mi mamá y a mi papá, por todo el amor que me han dado, y por todo el sacrificio que han hecho (y siguen haciendo) por mí. Os quiero mucho.

Resumen

En este Trabajo de Fin de Grado se recoge el desarrollo de una aplicación destinada a la asignación de árbitros a partidos de fútbol sala base de la provincia de Segovia, teniendo en cuenta una serie de restricciones. La aplicación permite gestionar árbitros (crear, modificar, visualizar y eliminar), consultar los partidos de la temporada mediante filtros, así como asignar horarios y árbitros a los encuentros de una jornada concreta. Tras la asignación, el sistema envía automáticamente un correo electrónico a cada árbitro con la información correspondiente y realiza una copia de seguridad con los datos esenciales de los partidos. Para su desarrollo se han utilizado herramientas de Microsoft 365, como Microsoft Power Apps, Microsoft Power Automate, Microsoft SharePoint y Microsoft Lists, lo que ha permitido crear una solución flexible, automatizada y fácilmente integrable en entornos empresariales.

Palabras claves: Fútbol sala, Gestión de árbitros, Asignación de partidos, Microsoft Power Apps, Microsoft Power Automate, Microsoft Sharepoint, Microsoft Lists, Notificaciones automáticas, Copia de seguridad.

Abstract

This work presents the development of an application designed to assign referees to youth futsal matches in the province of Segovia, taking into account a series of restrictions. The application allows users to manage referees (create, modify, view, and delete), view the season's matches using filters, and assign times and referees to the matches of a specific matchday. Once the assignments are made, the system automatically sends an email to each referee with the relevant information and creates a backup containing the essential match data. The application was developed using Microsoft 365 tools, including Microsoft Power Apps, Microsoft Power Automate, Microsoft SharePoint, and Microsoft Lists, enabling the creation of a flexible, automated, and easily integrable solution for business environments.

Keywords: Futsal, Referee Management, Match Assignment, Microsoft Power Apps, Microsoft Power Automate, Microsoft SharePoint, Microsoft Lists, Automatic Notifications, Backup.

Parte I

Memoria del Proyecto

Capítulo 1

Descripción del Proyecto

El Trabajo de Fin de Grado está organizado en cuatro partes: la primera de ellas, presenta el contexto y la planificación y objetivos del proyecto. La segunda, describe el análisis y los requisitos que debe tener la aplicación. La tercera incluye las guías para los usuarios y la instalación de la aplicación. Finalmente, se encuentra la bibliografía que contienen información adicional relevante y que se ha hecho referencia en distintas partes de la memoria.

1.1. Introducción

Este Trabajo de Fin de Grado nace de una necesidad común en el deporte local: el problema de la asignación de árbitros a los partidos. Resulta que en muchas competiciones locales sigue realizándose de forma manual, sin el apoyo de herramientas que lo automaticen o lo hagan más eficiente. Esta falta de digitalización genera una carga de trabajo considerable para los responsables, puede dar lugar a errores humanos, y, sobre todo, implica una gran pérdida de tiempo que podría evitarse con una solución automatizada.

Tanto mi tutor como yo compartimos una gran pasión por el fútbol, y eso nos permitió entender rápidamente el valor que tendría una herramienta sencilla pero eficaz que ayudase a automatizar este proceso. La idea inicial era aplicarla al fútbol provincial o incluso al regional, pero finalmente se ha optado por centrarse en el fútbol sala, ya que permitía acercarse más al caso práctico que ocurre en Segovia, donde el fútbol sala provincial tiene una estructura más manejable y mejor definida. Así, comenzó el planteamiento y el desarrollo de una aplicación que, ayudaría a gestionar las designaciones arbitrales a los partidos de fútbol sala que se juegan.

Su originalidad radica precisamente en eso: centrarse en un ámbito donde la tecnología todavía no ha llegado del todo, y en demostrar que no es necesario desarrollar un software extremadamente complejo para resolver un problema como la designación arbitral a nivel local. En estas competiciones, donde los recursos son mucho más limitados que en el deporte profesional, una herramienta sencilla, bien pensada y adaptada a la realidad del entorno puede marcar una diferencia significativa en la gestión diaria.

La aplicación que se ha desarrollado está diseñada para integrarse en federaciones regionales o locales de deporte, como es el caso de la Real Federación de Castilla y León de Fútbol [19], que gestiona el fútbol en la comunidad de Castilla y León. Una de las principales ventajas de esta aplicación es su escalabilidad, ya que puede adaptarse fácilmente a otros deportes y regiones, permitiendo su implementación en una variedad de contextos deportivos. Además, dado que la aplicación se basa en tecnologías de Microsoft, no hay problemas con respecto a la compatibilidad ni con el entorno utilizado, lo que garantiza una integración fluida y accesible dentro de las infraestructuras tecnológicas ya existentes en estas organizaciones.

1.2. Objetivos del trabajo

1.2.1. Objetivos generales

El objetivo principal de este Trabajo de Fin de Grado ha sido el desarrollo de una aplicación informática destinada a facilitar y optimizar la asignación de árbitros a partidos en el ámbito del deporte provincial o regional, sujeto a ciertas restricciones, para que federaciones, comités arbitrales, y servicios municipales puedan usarla, ya que a diferencia de las competiciones de alto nivel o del deporte profesional, donde se requieren sistemas complejos con algoritmos avanzados de planificación, estas organizaciones no necesitan soluciones tan potentes ni sofisticadas, sino herramientas funcionales, accesibles, escalables y adaptadas a su contexto deportivo. Sumado a ello, tradicionalmente este proceso se realiza de forma manual, con las complicaciones que ello conlleva en términos de carga de trabajo y riesgo de errores.

1.2.2. Objetivos específicos

- Implementar un sistema que permita gestionar el registro de árbitros, incluyendo funcionalidades para añadir, editar, ver, y eliminar sus datos personales.
- Permitir la visualización estructurada de la información de todos los partidos de fútbol sala de la provincia de Segovia en las categorías de alevín y benjamín, cada una compuesta por dos ligas (primera y segunda), con seis equipos por liga.
- Definir el horario de todos los partidos de una jornada.
- Designar a los árbitros dados de alta partidos en función del número de la jornada, teniendo en cuenta una serie de limitaciones.
- Aviso de errores en caso de que la designación arbitral no haya sido posible.
- Notificar automáticamente a los árbitros que han sido elegidos con la información básica del partido que deberán arbitrar.

- Realizar copias de seguridad con toda la información una vez ya estén los árbitros asignados a una jornada.

1.2.3. Alcance y limitaciones

Las principales funcionalidades de la aplicación han sido descritas anteriormente, destacando que el caso práctico sobre el que se trabaja corresponde a la liga regular de alevín y benjamín de fútbol sala de Segovia, utilizando los equipos reales que participan en dichas competiciones [18]. Por tanto, no se contemplan otras categorías, ligas ni equipos fuera de este ámbito.

En cuanto a las limitaciones de la tecnología utilizada será necesario tener una licencia activa de Microsoft 365 para poder desarrollar, mantener y usar la aplicación diseñada (en este caso, con la licencia que aporta la Universidad de Valladolid es suficiente).

Además, cabe destacar que la interfaz de la aplicación se ha desarrollado para móvil, aunque como es una aplicación web, es completamente funcional en ordenadores y portátiles, lo que garantiza una experiencia multiplataforma.

1.3. Estado del arte

En el ámbito de la gestión deportiva, especialmente en lo que respecta a la asignación de árbitros y la organización de partidos, existen múltiples soluciones tecnológicas que han ido evolucionando en función del nivel competitivo y los recursos disponibles. En las competiciones de alto nivel, como el fútbol profesional, se han desarrollado sistemas complejos que permiten optimizar hasta el más mínimo detalle del calendario deportivo.

A nivel nacional, la Real Federación Española de Fútbol utiliza una plataforma llamada *Fénix*, que integra la gestión de licencias, calendarios, actas arbitrales y otras funciones administrativas. Esta solución centraliza todos los procesos relacionados con la actividad federativa, facilitando el seguimiento y la planificación de las competiciones.

En el plano internacional, destacan herramientas como *RefAssist* [21], empleadas por federaciones de distintos deportes (como el baloncesto en Francia, el hockey en Países Bajos o el fútbol en Australia) para gestionar de forma automatizada las designaciones arbitrales. Estas plataformas incorporan funcionalidades avanzadas como el historial de actuaciones de los árbitros, sus preferencias, disponibilidad geográfica o conflictos de interés.

Un caso especialmente relevante es el de LaLiga, la entidad que gestiona la Primera y Segunda División del fútbol profesional en España. Esta organización ha desarrollado un sistema de planificación, capaz de analizar en segundos miles de combinaciones de calendarios alternativos cada temporada. Para ello, se apoya en un conjunto de reglas y restricciones (como derbis, festividades, eventos internacionales, etc.), procesadas mediante algoritmos de optimización y sistemas de alta computación. Además, LaLiga utiliza superordenadores como parte de sus herramientas de simulación y análisis predictivo que permiten evaluar escenarios posibles en función de los resultados deportivos.

Estas soluciones demuestran que, en los niveles más altos del deporte, la tecnología aplicada a la planificación y gestión es cada vez más sofisticada. Sin embargo, la mayoría de estas herramientas están fuera del alcance de federaciones modestas, clubes locales o comités arbitrales provinciales, que no disponen de los recursos técnicos ni económicos necesarios para adoptar este tipo de plataformas. En este contexto, surge la necesidad de soluciones más accesibles, como la propuesta en este trabajo.

1.4. Beneficios

A diferencia de los sistemas complejos empleados en el deporte profesional, SegoFutsal ha sido desarrollado utilizando tecnología *low code*, un enfoque moderno que permite construir aplicaciones funcionales mediante interfaces visuales y componentes predefinidos. Este paradigma no solo reduce significativamente la necesidad de escribir código manual, sino que también democratiza el desarrollo de soluciones digitales, permitiendo que usuarios sin formación especializada en programación puedan construir herramientas a medida.

El uso de la plataforma Power Platform de Microsoft aporta una ventaja clave: la integración inmediata con el ecosistema Microsoft 365, presente ya en muchas organizaciones educativas, federativas y administrativas. Esto significa que la adopción de SegoFutsal no requiere infraestructuras adicionales ni inversiones significativas en hardware o software, ya que puede desplegarse directamente en entornos existentes.

Otra de las ventajas fundamentales de esta solución es su escalabilidad. Aunque se ha diseñado específicamente para gestionar las competiciones de fútbol sala en categorías base en la provincia de Segovia, su estructura permite adaptarla fácilmente a otros deportes, niveles de competición o regiones geográficas. Esta versatilidad convierte a SegoFutsal en una herramienta útil no solo para el caso específico sobre el que se ha desarrollado, sino también como modelo de aplicación reutilizable para otras entidades deportivas con necesidades similares.

Además, al no depender de desarrollos a medida ni de proveedores externos, la aplicación permite a las propias federaciones mantener y evolucionar el sistema con mayor autonomía. Esto resulta especialmente valioso en entornos con presupuestos ajustados, donde la sostenibilidad técnica y económica es una condición indispensable para que una herramienta digital tenga continuidad en el tiempo.

Capítulo 2

Metodología

En este capítulo se detallan las metodologías usadas a lo largo del Trabajo de Fin de Grado.

2.1. Proceso de desarrollo

El desarrollo de SegoFutsal se ha llevado a cabo siguiendo un modelo de **desarrollo incremental**. Consiste en construir la aplicación por medio de entregas sucesivas de funcionalidad completa. En cada incremento se desarrolla, implementa y prueba una parte funcional. Se ha elegido esta metodología debido a la naturaleza modular de la plataforma, cuyas tres funcionalidades principales (gestión de árbitros, gestión de partidos, y asignación de árbitros a partidos) presentan un grado razonable de independencia entre sí. Aprovechando esta situación, se ha comenzado el desarrollo por los módulos funcionalmente más sencillos con el objetivo de adquirir experiencia en el proyecto, aprender de los errores y fallos que ocurren, y después abordar la designación de los árbitros a los partidos lo último por su mayor complejidad.

A continuación, se describen los principales bloques funcionales desarrollados y el orden en que fueron implementados:

- **Primero:** Diseño de la base de datos, definiendo las tablas, y campos necesarios. Posteriormente, se procedió a poblar las tablas con registros según se iban necesitando para las funcionalidades.

- **Segundo:** Desarrollo de la gestión de árbitros, que permite realizar operaciones de escritura, lectura, borrado, y modificación sobre los datos de los colegiados. Supuso un primer contacto con la lógica de la aplicación y permitió validar la integración con la base de datos.

- **Tercero:** A continuación, se implementó la gestión de partidos, enfocada en la visualización de los encuentros registrados y en la incorporación de filtros como la jornada, categoría, o liga.

- **Cuarto:** Se desarrolló la asignación de árbitros, funcionalidad central del sistema. Su implementación requirió utilizar flujos lógicos para tener en cuenta las restricciones y evitar conflictos.

- **Quinto:** Finalmente, fue realizada la copia de seguridad tras la asignación de árbitros a los partidos de una misma jornada. De este modo, se asegura la conservación del estado del sistema en ese punto, permitiendo su recuperación en caso necesario.

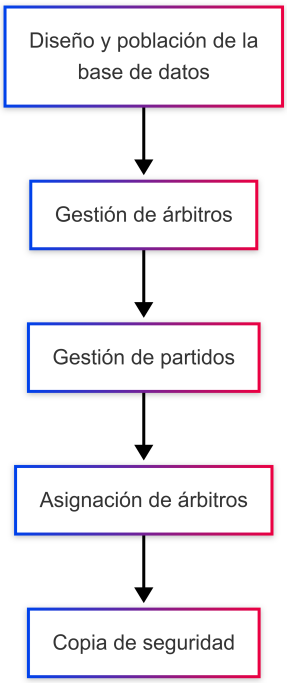


Figura 2.1: Modelo de desarrollo incremental

2.2. Herramientas utilizadas

A continuación, se detallan las tecnologías concretas que se han utilizado para desarrollar SegoFutsal, incluyendo los lenguajes de programación, la base de datos utilizada, y las herramientas empleadas para el desarrollo de la app.

Es importante señalar que el desarrollo de la aplicación ha coincidido temporalmente con mi periodo de prácticas en empresa, lo que ha influido directamente en la elección tecnológica. Durante esta etapa, he tenido la oportunidad de aprender y trabajar con Microsoft Power Platform [14], una plataforma *low code* que resulta especialmente adecuada para aplicaciones de este tipo y con esta finalidad. Esta circunstancia ha permitido, no solo aplicar conocimientos reales del entorno laboral, sino también mejorar la eficiencia y

calidad del código utilizando algunos consejos que me daban mis compañeros de trabajo sobre esta plataforma.

En cuanto a las plataformas *low code* se basan en el uso de herramientas visuales y componentes predefinidos para el diseño y desarrollo de aplicaciones, reduciendo considerablemente la necesidad de escribir código manual. Esta forma de desarrollo software acelera el proceso de desarrollo, reduce los costes, y facilita la creación de soluciones funcionales incluso por parte de usuarios sin tanta experiencia técnica. En entornos empresariales, este tipo de plataformas permiten implementar aplicaciones internas de forma ágil y mantenible, lo que las convierte en una opción ideal para ciertos proyectos orientados a la gestión de recursos, ya bien sean humanos o materiales.

No obstante, como contrapartida, existen situaciones en las que el uso de código puro resulta más adecuado y eficiente, especialmente cuando se requiere un control más detallado sobre la lógica de negocio o se desea implementar funcionalidades muy específicas que las herramientas visuales no permiten desarrollar fácilmente. Sumado a ello, se trata de una tecnología relativamente nueva, en constante evolución, con frecuentes actualizaciones que incluyen nuevos componentes, correcciones de errores y cambios en la interfaz gráfica. Esta continua transformación, si bien aporta mejoras constantes, también puede suponer un reto en términos de estabilidad y adaptación a largo plazo. Entre las plataformas *low code* más utilizadas actualmente destacan Microsoft Power Platform, Retool, y Salesforce Lightning.

2.2.1. Microsoft Lists

Microsoft Lists [7] es una herramienta de la suite de Microsoft 365 diseñada para la creación, gestión y seguimiento de información estructurada en formato de lista. Permite a los usuarios organizar datos de forma similar a una hoja de cálculo, pero con funcionalidades más avanzadas orientadas a la colaboración y la automatización de procesos. Cada lista puede contener columnas personalizables (texto, campos numéricos, opciones, imágenes, columnas de tipo lookup, entre otros), y reglas predefinidas para flujos de trabajo.

Se ha utilizado esta plataforma como base de datos de la aplicación, ya que la idea que hay tras estas listas no es otra que la de una tabla cualquiera como podría tener cualquier otro gestor de base de datos. Dado que se integra de forma nativa con otras herramientas del ecosistema Microsoft que se han usado, como Microsoft Power Apps o Microsoft Power Automate, la convierte en una solución especialmente útil para guardar toda la información de los árbitros, partidos y equipos de las dos categorías.

Es importante destacar que Microsoft posee otra herramienta, Microsoft Dataverse, un poco más potente y enfocada a ser base de datos con el uso de tablas en vez de listas. Sin embargo, esta opción se ha descartado porque su utilización está condicionada por el tipo de suscripción que uno disponga de Mircosoft 365. En el caso del entorno de la Universidad de Valladolid, no incluye Dataverse. De todos modos, para el objetivo de este Trabajo de Fin de Grado, con Micrososft Lists es suficiente y adecuada.

2.2.2. Microsoft Power Apps

Microsoft Power Apps [8] es la plataforma de desarrollo *low code* incluida en el ecosistema de Microsoft 365, que permite crear aplicaciones personalizadas de manera rápida y sencilla mediante el uso de componentes visuales y lógica declarativa. Su objetivo principal es facilitar la creación de interfaces de usuario funcionales que interactúan con diversas fuentes de datos, como puede ser en nuestro caso SharePoint, Microsoft Lists o Microsoft Power Apps, sin necesidad de escribir grandes cantidades de código.

Se ha utilizado Power Apps para el desarrollo completo de la aplicación SegoFutsal. Esta herramienta ha sido empleada principalmente para construir la interfaz gráfica con la que interactúa el usuario, permitiéndole acceder, visualizar y modificar los datos para la gestión de árbitros, partidos y designaciones. La aplicación centraliza toda la información relevante y facilita tareas como la asignación de árbitros o la consulta de encuentros.

Uno de los elementos más utilizados en Power Apps y poco conocido en la construcción de interfaces, han sido las galerías, un tipo de control visual que permite mostrar colecciones de datos de forma estructurada y repetitiva. Cada galería presenta un conjunto de registros provenientes de una fuente de datos (en este caso, de Microsoft Lists), mostrando en pantalla información clave mediante etiquetas, imágenes, o botones. Las galerías permiten aplicar fácilmente filtros, búsquedas u ordenaciones, adaptando dinámicamente la información mostrada según la lógica de la aplicación. En SegoFutsal, han sido utilizadas para mostrar los árbitros disponibles para arbitrar o para todos los partidos de una jornada, facilitando al usuario la navegación y gestión de forma intuitiva.

2.2.3. Microsoft Power Automate

Microsoft Power Automate [11] es la plataforma de automatización de flujos de trabajo *low code* que forma parte integral del ecosistema de Microsoft 365. Diseñada para facilitar la automatización de tareas repetitivas y procesos empresariales, Power Automate permite a los usuarios crear flujos automatizados que conectan diversas aplicaciones y servicios, tanto dentro como fuera de Microsoft. Estos flujos pueden ser utilizados para sincronizar archivos, enviar y recibir notificaciones, recopilar y procesar datos, y ejecutar una amplia variedad de tareas sin necesidad de intervención manual. Además, es especialmente útil para mejorar la eficiencia operativa de las organizaciones al reducir el tiempo y el esfuerzo necesarios para realizar tareas que, de otro modo, serían manuales y propensas a errores. Todo ello con una interfaz sencilla, utilizando bloques de acciones precargadas, propia de la naturaleza de programación *low code*. Es por ello por lo que, en el desarrollo del algoritmo de asignación de árbitros, Microsoft Power Automate ha sido la herramienta esencial para automatizar varios procesos críticos relacionados con la gestión de árbitros y partidos. En concreto, se han implementado dos flujos de trabajo principales para optimizar la gestión de datos y la asignación de árbitros.

El primero es un flujo auxiliar diseñado para poblar automáticamente la lista de PARTIDOS con datos provenientes de la lista EQUIPOS. Este flujo garantiza que cada partido tenga la información necesaria sobre los equipos participantes, lo que ha simplificado la

creación de estos registros.

El segundo flujo, y el más importante, es el algoritmo de designación arbitral. Este algoritmo ha sido diseñado utilizando Power Automate para asignar árbitros a los partidos de manera eficiente y equitativa. El flujo verifica que se cumplan las restricciones, como la disponibilidad de los árbitros, y determina los árbitros para todos los partidos de la jornada que previamente ha seleccionado el usuario. Finalmente, envía un correo electrónico a todos los colegiados que arbitran esa jornada con la información de los partidos. Gracias a Power Automate, este proceso se lleva a cabo de forma automatizada, reduciendo significativamente el tiempo y esfuerzo necesarios para la designación manual y minimizando el riesgo de errores humanos.

2.2.4. Microsoft Sharepoint

Microsoft SharePoint [15] es una plataforma de colaboración y gestión de contenido que forma parte del ecosistema de Microsoft 365. Diseñada para facilitar la colaboración en equipo y la gestión eficiente de documentos, SharePoint permite a las organizaciones crear sitios web personalizados donde los usuarios pueden compartir información, almacenar documentos, y colaborar en proyectos de manera segura y organizada. Con capacidades robustas para la gestión de documentos, flujos de trabajo, y aplicaciones empresariales, SharePoint se utiliza ampliamente para crear intranets corporativas, gestionar bibliotecas de documentos, y desarrollar soluciones empresariales personalizadas que integran datos y procesos de múltiples fuentes.

En el contexto de SegoFutsal, SharePoint ha sido utilizado como el repositorio central para almacenar toda la información relacionada con la aplicación, guardándola en una solución dentro de entorno de la UVa. Esto incluye la propia app, así como los flujos de trabajo automatizados creados con Power Automate. SharePoint también gestiona las conexiones necesarias para la operación de la aplicación, como la información de las cuentas de Office 365 y las configuraciones de correo electrónico. Además, es el lugar donde se guardan tanto las copias de seguridad como el esquema que deben seguir estas copias. Al centralizar estos datos y conexiones en SharePoint, se garantiza que la información esté siempre accesible y segura, facilitando la integración, e incluso la modificación o eliminación de conexiones en caso de que fuera necesario.

2.2.5. Microsoft Copilot

Microsoft Copilot [6] es una herramienta de inteligencia artificial desarrollada por Microsoft en colaboración con OpenAI, diseñada (en el contexto de la programación) para asistir a los desarrolladores en la escritura de código. Integrado en entornos de desarrollo como Visual Studio Code, Copilot utiliza modelos avanzados de aprendizaje automático para sugerir líneas de código, funciones completas, y soluciones a problemas de programación en tiempo real. Esta herramienta se basa en el análisis de miles de millones de líneas de código de repositorios públicos para ofrecer recomendaciones precisas y contextuales, ayudando a los desarrolladores a escribir código más rápido y con menos

Figura 2.2: Logotipos de la tecnología utilizada

errores.

Durante el desarrollo de la aplicación de asignación de árbitros, Microsoft Copilot ha sido utilizado como una herramienta de apoyo para escribir y optimizar el código en Microsoft Power Apps. Su capacidad para sugerir fragmentos de código y funciones completas ha permitido ahorrar tiempo y mejorar la calidad del código. Además, al ser una herramienta de Microsoft, Copilot ofrece ventajas adicionales cuando se trabaja con plataformas de Microsoft, como es en nuestro caso. También se ha utilizado ChatGPT [17] como apoyo complementario en tareas de desarrollo y resolución de dudas técnicas.

2.2.6. Microsoft Excel

Microsoft Excel se ha utilizado como herramienta principal para crear copias de seguridad de la información relacionada con los partidos de una jornada. La metodología aplicada consiste en el uso de una plantilla predefinida que se encuentra en Microsoft Sharepoint que contiene una tabla estructurada. Esta tabla se rellena con los datos específicos de cada jornada, como los equipos, el árbitro asignado, o el lugar del encuentro. Una vez completada la tabla, se genera un nuevo archivo de Excel que actúa como copia de seguridad para almacenar la información de manera organizada y accesible. Finalmente, se guarda en una carpeta de Backup de Microsoft Sharepoint.

2.3. Arquitectura

2.3.1. Arquitectura Lógica

La arquitectura lógica de **SegoFutsal** está organizada en varias capas y servicios proporcionados por Microsoft, cada uno desempeñando un papel crucial en el funcionamiento de la aplicación.

- **Capa de Presentación (Cliente):** Está constituida por la interfaz de usuario generada mediante Microsoft Power Apps. Los usuarios interactúan con la aplicación a través de componentes visuales ya definidos como formularios, galerías y botones. Microsoft Power Apps proporciona una interfaz visual que facilita el acceso y la gestión de los datos mediante conectores directos a Microsoft Lists o a Microsoft Power Automate.

- **Capa de Aplicación:** En este caso, Microsoft Power Automate actúa como el motor de los flujos de trabajo, procesando por ejemplo las designaciones arbitrales y ejecutando la lógica del negocio necesaria para asignar árbitros a los partidos, y creando una copia de seguridad de los datos. Los flujos se configuran para realizar tareas automatizadas, como enviar correos para notificar a los árbitros, o ayudar a poblar las listas con los datos iniciales.

- **Capa de Datos:** La base de datos está formada por Microsoft Lists. Sin embargo, es Microsoft SharePoint quien actúa como el repositorio central donde están los flujos, la aplicación, las conexiones necesarias para su funcionamiento, y las copias de seguridad, permitiendo una gestión eficiente y organizada.

2.3.2. Arquitectura Física

La arquitectura física de SegoFutsal describe cómo están distribuidos los componentes de la aplicación en el entorno de Microsoft y cómo se comunican entre sí.

- **Cliente:** El cliente es cualquier dispositivo con acceso a Internet que tenga permisos de uso de la aplicación. Esto incluye ordenadores de escritorio, portátiles o dispositivos móviles con la aplicación de Power Apps instalada.

- **Servidor:** La infraestructura de Microsoft se utiliza para alojar Power Apps, Power Automate y SharePoint. Estos servicios están basados en la nube, lo cual proporciona escalabilidad, disponibilidad y redundancia a nivel global.

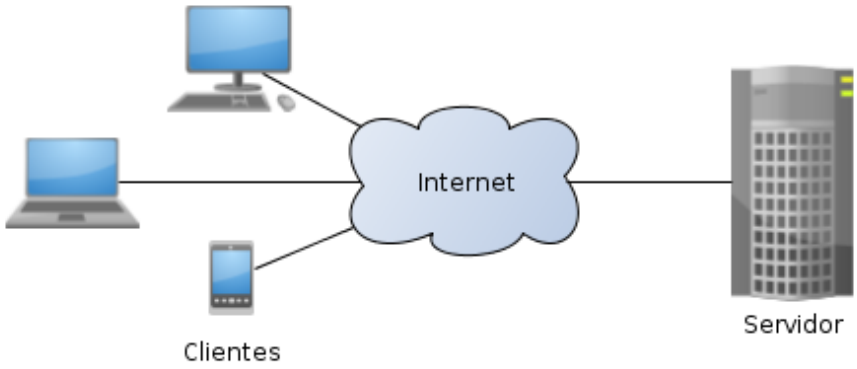


Figura 2.3: Arquitectura física

2.4. Definición de siglas y abreviaturas

En esta sección se presenta en el Cuadro 2.1 las abreviaturas y su significado utilizadas a lo largo de la memoria. De esta manera, se facilita la comprensión del documento, especialmente para aquellos lectores que puedan no estar familiarizados con ciertos términos técnicos o que puedan encontrar ambigüedad en el uso de algunos acrónimos.

Abreviatura	**Significado**
ACT	Actor
APP/app	Aplicación
D1	Horario de domingo a las 11:00
EI	External Input (Entradas Externas)
EIF	External Interface File (Ficheros de Interfaz Externa)
EQ	External Query (Salidas Externas)
EO	External Output (Salidas Externas)
FA	Factores de Ajuste
ILF	Internal Logical File (Ficheros Lógicos Internos)
PCN	Prueba de Caja Negra
PCB	Prueba de Caja Blanca
PF	Puntos de Función
PFA	Puntos de Función Ajustados
PFNA	Puntos de Función No Ajustados
PR	Prueba de Rendimiento
PS	Prueba de Seguridad
R	Riesgos
RES	Restricción
RF	Requisito Funcional
RgN	Regla de Negocio
RN	Requisito de Negocio
RNF	Requisito No Funcional
RU	Requisito de Usuario
S1	Horario de sábado a las 11:00
S2	Horario de sábado a las 13:00
SUP	Suposición
TFG	Trabajo de Fin de Grado

Cuadro 2.1: Abreviaturas y sus significados

Capítulo 3

Planificación

En este capítulo se abordarán las cuestiones relativas a la planificación del trabajo.

3.1. Estimación del esfuerzo

Para estimar el esfuerzo necesario en el desarrollo de la aplicación, se ha optado por utilizar el método de **puntos de función de Albrecht**. Este método permite cuantificar la funcionalidad ofrecida al usuario final de forma independiente de la tecnología empleada. A continuación, se describe en qué consiste esta metodología y cuáles son sus principales características.

Origen y Desarrollo

El método de puntos de función fue desarrollado por Allan Albrecht en IBM durante la década de 1970. Surgió como una respuesta a la necesidad de medir el tamaño del software de manera objetiva, dejando de lado la tecnología o lenguaje específico utilizado en su desarrollo.

Conceptos Clave

El método de puntos de función es una forma de estimar por descomposición, midiendo según la funcionalidad que ofrece al usuario, utilizando cinco características principales:

- **Entradas Externas (EI)**: Son las entradas que provienen del exterior del sistema, como datos que el usuario ingresa a través de formularios.
- **Salidas Externas (EO)**: Son las salidas producidas por el sistema y enviadas al exterior, como informes generados, resultados visualizados o archivos exportados.
- **Consultas Externas (EQ)**: Interacciones que permiten recuperar datos de una base de datos sin realizar cálculos complejos, como consultas de búsqueda o consultas basadas en parámetros.

- **Ficheros Lógicos Internos (ILF)**: Son archivos o bases de datos que el sistema mantiene y gestiona internamente.

- **Ficheros de Interfaz Externa (EIF)**: Son archivos o bases de datos mantenidos por otro sistema, pero que el sistema actual utiliza para obtener información.

Cada una de estas componentes se evalúa por su complejidad y se les asigna una cantidad de puntos de función. Estos puntos permiten estimar el esfuerzo necesario para desarrollar una aplicación o proyecto software.

Ventajas del Método

- **Independencia del Lenguaje**: Proporciona una medida que no depende de tecnologías específicas o de la cantidad de líneas de código.

- **Orientación al Usuario**: Se centra en lo que el sistema hace realmente para el usuario, no solo en cómo está programado.

- **Estandarización**: Permite a las organizaciones y desarrolladores utilizar una medida común para evaluar y comparar proyectos de software.

Inconvenientes del Método

- **Subjetividad**: La clasificación de las complejidades puede variar y ser subjetiva entre diferentes evaluadores.

- **Necesidad de Detalle**: Requiere un entendimiento detallado del sistema para identificar y clasificar adecuadamente cada componente funcional.

- **Limitaciones para Sistemas Complejos**: Puede no capturar todos los matices de sistemas en tiempo real o altamente interactivos donde la lógica de negocio es profunda.

Aplicación a SegoFutsal

Una vez entendido el fundamento del método de puntos de función, se procede a utilizar este método para estimar el esfuerzo que requiere desarrollar SegoFutsal.

Dominio	**Complejidad**	**Peso**	**Nº Funciones**	**Total**
Entradas Externas	Baja	3	1	3
Entradas Externas	Media	4	4	16
Entradas Externas	Alta	6	1	6
Salidas Externas	Baja	4	1	4
Salidas Externas	Media	5	2	10
Salidas Externas	Alta	7	1	7
Consultas Externas	Baja	3	2	6
Consultas Externas	Media	4	3	12
Consultas Externas	Alta	6	1	6
Ficheros Lógicos Internos	Baja	7	0	0
Ficheros Lógicos Internos	Media	10	2	20
Ficheros Lógicos Internos	Alta	15	4	60
Ficheros de Interfaz Externa	Baja	5	0	0
Ficheros de Interfaz Externa	Media	7	1	7
Ficheros de Interfaz Externa	Alta	10	0	0
			Puntos de Función sin Ajustar (PFNA)	**157**

Cuadro 3.1: Puntos de función sin ajustar

Para completar la estimación por puntos de función, se deben considerar también los factores de ajuste. Estos factores permiten valorar el entorno técnico y las características funcionales que pueden influir en el esfuerzo de desarrollo. Van de 0 (ninguna influencia) a 5 (esencial) y se muestran en el Cuadro 3.2.

Una vez ya tenemos los Factores de Ajuste (FA) podemos aplicar los Puntos de Función ya ajustados con la siguiente fórmula

$$\text{PFA} = \text{PFNA} \times (0.65 + 0.01 \times \text{FA})$$

Que en nuestro caso es

$$\text{PFA} = 157 \times (0.65 + 0.01 \times 40) = 157 \times 1.05 = 164.85$$

Estos puntos de función muestran la estimación del esfuerzo para desarrollar esta aplicación.

Factor de Ajuste	Peso
Comunicación de datos	4
Funciones distribuidas	3
Prestaciones	4
Gran uso de la configuración	1
Velocidad de las transacciones	4
Entrada online de datos	2
Diseño para la eficiencia del usuario final	5
Actualización de datos online	3
Complejidad del proceso lógico interno	5
Reusabilidad del código	3
Facilidad de instalación	1
Facilidad de operación	1
Localizaciones múltiples	1
Facilidad de cambios	3
Total	**40**

Cuadro 3.2: Factores de ajuste

3.2. Planificación temporal

La planificación temporal es fundamental para asegurar el desarrollo organizado y eficiente del proyecto SegoFutsal. Esta sección ofrece una visión clara de las etapas y el cronograma previsto para la ejecución de cada fase del proyecto, permitiendo un seguimiento adecuado del progreso y facilitando ajustes según sea necesario, garantizando que el proyecto se mantenga alineado con sus objetivos temporales. A grandes rasgos, se ha supuesto que una semana de trabajo equivale a unas 16 horas, mientras que la duración total han sido de unas 17 semanas.

1. **Análisis**

   - **Propósito:** Identificar los requisitos funcionales del sistema.
   - **Duración:** *32* horas
   - **Tareas:**
     - Recolección de Requisitos (14 horas)
     - Evaluación de Requisitos (12 horas)
     - Revisión y Confirmación de Requisitos (6 horas)

2. Diseño

- **Propósito:** Elaborar la base de datos y el diseño detallado del sistema.
- **Duración:** *32* horas
- **Tareas:**
  - Diseño de la base de datos (14 horas)
  - Identificación de las vistas necesarias (13 horas)
  - Diseño de la Interfaz de Usuario (5 horas)

3. Implementación

- **Propósito:** Desarrollar los componentes del sistema.
- **Duración:** *96* horas
- **Tareas:**
  - Configuración del Entorno de Desarrollo (6 horas)
  - Desarrollo de Componentes Principales (40 horas)
  - Elaboración de los flujos de trabajo (50 horas)

4. Pruebas

- **Propósito:** Verificar y validar que el sistema cumpla con los requisitos establecidos.
- **Duración:** *48* horas
- **Tareas:**
  - Pruebas generales (8 horas)
  - Pruebas de la gestión de árbitros (8 horas)
  - Pruebas de la visualización de partidos (6 horas)
  - Pruebas de la asignación de árbitros (20 horas)
  - Pruebas de la copia de seguridad (6 horas)

5. Documentación y Entrega

- **Propósito:** Documentar el sistema y preparar la entrega final.
- **Duración:** *64* horas
- **Tareas:**
  - Documentación de la memoria (52 horas)
  - Presentación (12 horas)

Resumen de la planificación temporal

Fase	Duración (horas)	Porcentaje del total
Análisis	32	11.76 %
Diseño	32	11.76 %
Implementación	96	35.29 %
Pruebas	48	17.65 %
Documentación y Entrega	64	23.53 %
Total	**272**	**100 %**

Cuadro 3.3: Resumen de la planificación temporal

Conclusión

A modo de conclusión, la planificación temporal presentada permite visualizar de forma clara la distribución de las tareas a lo largo del desarrollo del proyecto. Esta estructura facilita el seguimiento del progreso y la gestión del tiempo. En la Figura 3.1 se muestra el cronograma del proyecto donde se detalla la distribución temporal de las distintas tareas planificadas, permitiendo observar su secuencia y solapamiento a lo largo del periodo establecido.

3.3. Presupuesto económico

En esta sección se detalla la estimación económica del proyecto, considerando los costes asociados al hardware y software utilizados, así como a los recursos humanos implicados según la planificación establecida.

3.3.1. Hardware

En el Cuadro 3.4 se muestra el presupuesto del hardware utilizado para el proyecto.

3.3.2. Software

En el Cuadro 3.5 se muestra el presupuesto del software utilizado para el proyecto. Es importante recalcar que el coste de la licencia de Microsoft 365 [5] normalmente es asumido por el cliente, ya que sin ella, el propio cliente no puede hacer uso de la app. Sin embargo, en este caso se asume este gasto en el coste del proyecto.

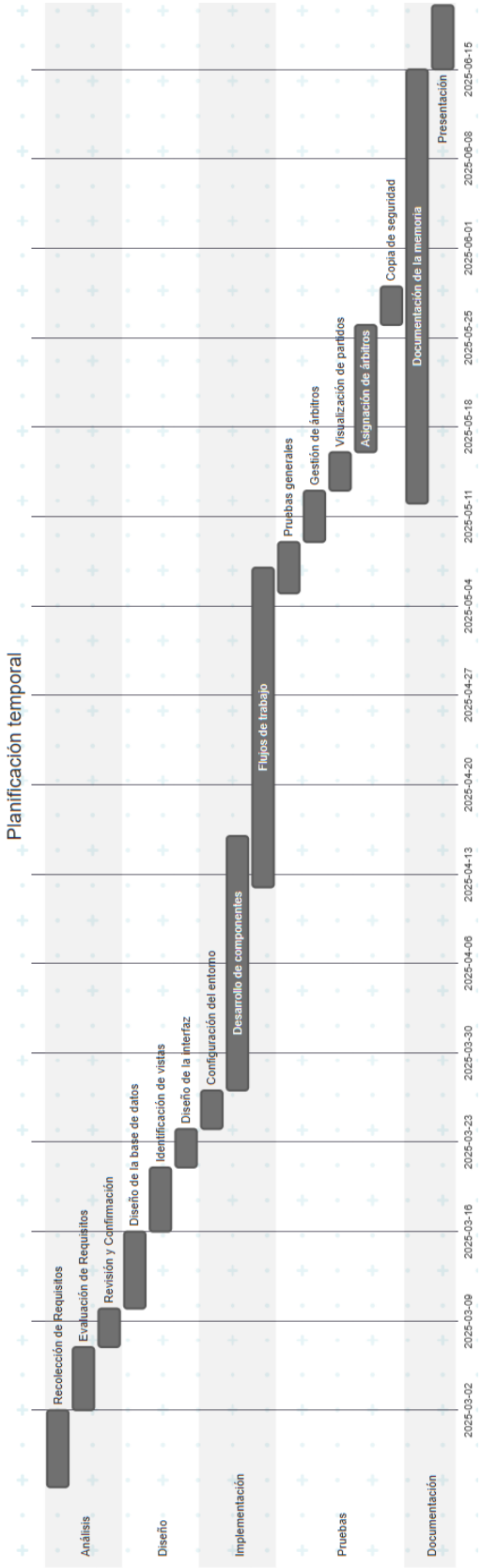


Figura 3.1: Cronograma

Componente	Precio	Vida útil	Uso en el proyecto	Porcentaje	Coste
Ordenador portátil	1100,00 €	78 meses	4 meses	5,13 %	56,41 €
Ratón	20,00 €	36 meses	4 meses	11,11 %	2,22 €
Monitor	250,00 €	78 meses	4 meses	5,13 %	12,82 €
Adaptador	5,00 €	36 meses	4 meses	11,11 %	0,56 €
Internet	66,59 €/mes	—	4 meses	30 %	79,90 €
				Total	**151,91 €**

Cuadro 3.4: Presupuesto del hardware

Componente	Precio	Vida útil	Uso en el proyecto	Porcentaje	Coste
Licencia de Microsoft 365 Empresa Básico	8,12 €/mes	—	4 meses	100 %	32,48 €
Windows 11	0,00 €	—	—	—	0,00 €
Mermaid	0,00 €	—	—	—	0,00 €
Google Chrome	0,00 €	—	—	—	0,00 €
Inteligencias Artificiales	0,00 €	—	—	—	0,00 €
				Total	**32,48 €**

Cuadro 3.5: Presupuesto del software

3.3.3. Recursos humanos

En el Cuadro 3.6 se muestra el presupuesto de los recursos humanos utilizados para el proyecto. Dado que la aplicación ha sido llevada a cabo por una sola persona y el tutor, se describen los roles que han sido necesarios con sus costes. Se ha supuesto que el salario es bruto durante una jornada laboral de 40 horas semanales.

3.3.4. Presupuesto total

Finalmente, en el Cuadro 3.7 se muestra un resumen de los costes totales del proyecto

Rol	**Salario Mensual**	**Coste por hora**	**Horas**	**Coste**
Analista	2.500 €	15,63 €	40	625,20 €
Programador	2.200 €	13,75 €	120	1.650,00 €
Ingeniero de pruebas	2.000 €	12,50 €	55	687,50 €
Tutor	1.800 €	11,25 €	20	225,00 €
			Coste Total	**3.187,70 €**

Cuadro 3.6: Presupuesto de los recursos humanos

Sección	**Coste Total**
Hardware	151,91 €
Software	32,48 €
Recursos humanos	3.187,70 €
Precio final del proyecto	**3.373,09 €**

Cuadro 3.7: Presupuesto final del proyecto

3.4. Riesgos

La gestión de riesgos es un proceso clave en el desarrollo de cualquier proyecto, ya que permite identificar, evaluar y planificar respuestas ante posibles eventos que puedan afectar negativamente su ejecución o resultados. El objetivo es minimizar la probabilidad y el impacto de estos riesgos mediante acciones preventivas (mitigación) y reactivas (contingencia), asegurando así una mayor estabilidad y control sobre el desarrollo del proyecto.

A continuación, se presentan los riesgos identificados para el desarrollo de SegoFutsal, junto con su análisis individual y las medidas propuestas para gestionarlos.

ID del Riesgo	R-01
Nombre	Pérdida de Datos
Descripción	Fallos técnicos o errores en el sistema que pueden conducir a la pérdida de datos importantes, como la información de los árbitros o partidos.
Probabilidad	Media
Impacto	Alto
Plan de Mitigación	Implementar copias de seguridad regulares y verificar la integridad de la base de datos.
Plan de Contingencia	Tener un sistema de recuperación preparado que pueda restaurar los datos a partir de las copias de seguridad más recientes en el menor tiempo posible.

Cuadro 3.8: R-01. Pérdida de datos

ID del Riesgo	R-02
Nombre	Falta de Conformidad con Requisitos
Descripción	La aplicación no cumple con todos los requisitos o expectativas marcadas, como funciones específicas mal implementadas.
Probabilidad	Baja
Impacto	Alto
Plan de Mitigación	Realizar pruebas frecuentes y recabar retroalimentación continua.
Plan de Contingencia	Contar con un equipo listo para realizar ajustes y revisiones.

Cuadro 3.9: R-02. Falta de conformidad con requisitos

ID del Riesgo	R-03
Nombre	Problemas Técnicos con la Integración
Descripción	Dificultades al integrar funciones como el envío automático de correos o la implementación de filtros avanzados.
Probabilidad	Alta
Impacto	Medio
Plan de Mitigación	Realizar pruebas de integración tempranas y capacitar al equipo en herramientas necesarias.
Plan de Contingencia	Mantener un cronograma flexible.

Cuadro 3.10: R-03. Problemas técnicos con la integración

ID del Riesgo	R-04
Nombre	Sobrecarga del Servidor
Descripción	La aplicación podría experimentar lentitud o fallos por un número inesperado de usuarios o peticiones simultáneas.
Probabilidad	Media
Impacto	Alto
Plan de Mitigación	Escalar recursos del servidor según demanda y optimizar los procesos de la aplicación, como la asignación de árbitros.
Plan de Contingencia	Aumentar temporalmente la capacidad del servidor y reducir carga de trabajo no esencial.

Cuadro 3.11: R-04. Sobrecarga del servidor

ID del Riesgo	R-05
Nombre	Alteraciones en el Calendario de Desarrollo
Descripción	Retrasos en el desarrollo del proyecto debido a problemas técnicos, cambios de alcance o disponibilidad del equipo.
Probabilidad	Media
Impacto	Medio
Plan de Mitigación	Estimar plazos con márgenes adecuados y planificar tareas con prioridades.
Plan de Contingencia	Reorganizar recursos humanos y técnicos para centrarse en funciones prioritarias.

Cuadro 3.12: R-05. Alteraciones en el calendario de desarrollo

ID del Riesgo	R-06
Nombre	Cambios Reglamentarios
Descripción	Cambios en normativas deportivas que afecten el funcionamiento de las reglas de asignación de árbitros.
Probabilidad	Baja
Impacto	Medio
Plan de Mitigación	Mantenerse informado de las novedades reglamentarias y diseñar en la medida de lo posible las reglas de asignación de manera modular y adaptable.
Plan de Contingencia	Actualizar la aplicación rápidamente en coordinación con los responsables.

Cuadro 3.13: R-06. Cambios reglamentarios

Capítulo 4

Conclusiones

4.1. Conclusión

La realización de este Trabajo de Fin de Grado ha sido una experiencia enriquecedora, permitiéndome aplicar los conocimientos adquiridos durante mis prácticas en la empresa (NTT Data Spain) al desarrollo de la aplicación. Ha sido gratificante observar la evolución del proyecto, especialmente considerando que tanto mi tutor como yo planteamos este TFG para abordar un problema real: la asignación de árbitros a partidos de fútbol en competiciones locales y/o regionales. Aunque el resultado final requiere ajustes y mejoras para poder solventar el problema real, sin duda alguna proporciona una base sólida que puede ser utilizada en futuros proyectos similares.

Una de las principales ventajas de este proyecto ha sido el uso de la tecnología *low code* de Microsoft. Esta herramienta me ha demostrado ser extremadamente útil, permitiendo desarrollar aplicaciones como la planteada en el Trabajo de Fin de Grado de manera sencilla y rápida. Además, gracias a ella he podido restar mi preocupación por aspectos críticos como la seguridad, la instalación o las herramientas de uso, permitiéndome centrar en la lógica de la aplicación y en el diseño de la interfaz.

Sin embargo, el uso de tecnología *low code* también me ha presentado ciertos desafíos. Actualmente, existe poca guía y documentación sobre el desarrollo *low code*, lo que ha dificultado mi aprendizaje y la implementación. He notado que este estilo de desarrollo requiere que el programador se adapte más a ellas que ellas al programador, lo cual es diferente de lo que ocurre con otros lenguajes de programación o herramientas de desarrollo más tradicionales, en las que tienes más libertad para resolver los problemas que se te aparecen.

En resumen, considero que enfrentarme al desarrollo de este proyecto desde cero ha sido una experiencia valiosa. De hecho, dependiendo del objetivo de otras aplicaciones, recomendaría considerar el uso de la tecnología *low code*, ya que ofrece una forma eficiente y accesible de desarrollar soluciones tecnológicas.

4.2. Mejoras

Con el objetivo de optimizar el proceso de designación arbitral y acercarlo más a la realidad, se han identificado una serie de posibles mejoras orientadas a reforzar las restricciones existentes en el sistema. Estas mejoras no solo buscan evitar situaciones de conflicto de interés, sino también promover un reparto equilibrado de los partidos, teniendo en cuenta aspectos como la carga de trabajo de los árbitros, su zona de residencia, historial de designaciones o posibles vínculos personales con los equipos. A continuación, se detallan varias propuestas que podrían ser incorporadas en futuras versiones de la aplicación para aumentar la fiabilidad y robustez del sistema de designación.

- **Evitar designaciones repetidas con el mismo equipo en un periodo corto:** Limitar que un árbitro arbitre a un mismo equipo más de una vez cada ciertas jornadas para evitar conflictos de interés y mejorar la percepción de imparcialidad.
- **Restricción por localización geográfica:** Impedir que un árbitro arbitre a equipos de su misma localidad o zona geográfica para preservar la neutralidad.
- **Historial disciplinario y conflictos previos:** Registrar incidencias pasadas y restringir la designación de árbitros en partidos de equipos con los que hayan tenido conflictos o informes negativos.
- **Vínculos personales o familiares con equipos:** Permitir que los árbitros declaren relaciones personales o familiares con determinados equipos, y bloquear su designación en dichos partidos.
- **Rotación geográfica y por tipo de partido:** Establecer un sistema de rotación que evite que un árbitro dirija siempre en la misma zona o tipo de competición (liga, copa, etc.), promoviendo la variedad en las designaciones.

Además de las restricciones en las designaciones arbitrales, se plantean a mayores una serie de mejoras enfocadas a ampliar las funcionalidades de la aplicación y cubrir de forma más completa todo el ciclo de gestión arbitral. Actualmente, la herramienta se centra en la asignación de árbitros, la administración del colectivo arbitral y la visualización de los partidos de cada jornada. No obstante, integrar funcionalidades tanto en la fase previa como posterior a los encuentros permitiría incrementar la utilidad práctica del sistema. A continuación, se presentan algunas de ellas.

- **Carga de partidos desde la aplicación:** Permitir introducir manualmente o importar los partidos de toda la temporada desde un archivo externo, facilitando así la planificación inicial.
- **Registro digital de actas arbitrales:** Habilitar un formulario para que los árbitros puedan completar y subir el acta del partido desde la propia aplicación, incluyendo incidentes, amonestaciones, resultados y observaciones.

- **Historial de partidos por árbitro:** Añadir una sección donde cada árbitro pueda consultar su historial de partidos arbitrados, incluyendo fechas, equipos y categoría.
- **Generación de informes:** Permitir la exportación de informes de actividad por árbitro, por jornada o por equipo, útiles para análisis o rendición de cuentas.
- **Gestión de incidencias y observaciones:** Incluir un apartado para reportar y consultar incidencias en partidos, tanto por parte del árbitro como del comité organizador.
- **Ampliación de información del colectivo arbitral:** Incorporar nuevos campos en el perfil de los árbitros para almacenar información adicional que pueda ser útil en la gestión y administración del colectivo arbitral. Estos campos permitirán tener un registro más completo y detallado de cada árbitro, facilitando tareas administrativas y mejorando la trazabilidad de los datos. Algunos de los campos que podrían añadirse son el DNI, la fecha de nacimiento, la dirección, o un estado de activo/inactivo.

Parte II

Documentación técnica

Capítulo 5

Análisis

Para abordar el análisis de SegoFutsal, es fundamental detallar los requisitos que guiarán su desarrollo y funcionalidad. Este capítulo se centra en la identificación y descripción de los elementos esenciales que definirán cómo interactúa el usuario con el sistema, asegurando que se cumplan los objetivos del negocio de manera eficiente.

5.1. Actores

En esta sección, se identifica el actor principal que interactúa con el sistema, esencial para comprender cómo se gestionan las diferentes tareas dentro de la aplicación y ayudan a orientar el desarrollo de funcionalidades específicas.

ID	**Actor**	**Descripción**
ACT-01	Usuario registrado (tiene los permisos necesarios para utilizar la aplicación)	Es el único actor del sistema. Este usuario, que sería el responsable de las asignaciones arbitrales de la federación de fútbol sala de Segovia, dispone de todas las funcionalidades de la aplicación.

Cuadro 5.1: Actores del sistema

5.2. Requisitos

A continuación, se detallan las especificaciones necesarias para abordar las necesidades operativas y funcionales de la aplicación. Estos requisitos están diseñados para asegurar que la aplicación cumpla con las expectativas del negocio y ofrezca un soporte robusto para las operaciones diarias que pueda tener la organización que use esta aplicación.

5.2.1. Requisitos de negocio

En el Cuadro 5.2 se muestran los requisitos de negocio de la app SegoFutsal.

ID	Requisito
RN-01	Reducir el tiempo y esfuerzo administrativo dedicado a la planificación y asignación de árbitros para cada jornada.
RN-02	Minimizar los errores humanos en la asignación de árbitros, garantizando imparcialidad y cumplimiento de las normas de la federación.
RN-03	Aumentar la eficiencia operativa del departamento arbitral mediante la automatización de procesos clave como notificaciones y copias de seguridad.
RN-04	Garantizar la transparencia en la gestión de designaciones arbitrales para reforzar la confianza de los clubes y árbitros en el sistema.
RN-05	Facilitar la trazabilidad y auditoría de las asignaciones realizadas, en caso de reclamaciones o revisiones por parte de algún equipo.

Cuadro 5.2: Requisitos de negocio

5.2.2. Reglas de negocio

En el Cuadro 5.3 se muestran las reglas de negocio de la app SegoFutsal.

ID	Descripción
RgN-01	La información de los partidos debe actualizarse inmediatamente después de la asignación de árbitros.
RgN-02	Los correos electrónicos de notificación deben enviarse automáticamente a los árbitros tan pronto como se complete su asignación de forma correcta.
RgN-03	La copia de seguridad debe crearse inmediatamente después de la asignación de árbitros a los partidos de una jornada.
RgN-04	Solo puede ejecutarse el flujo lógico de asignar árbitros a aquellas jornadas que no hayan tenido designación arbitral anteriormente.

Cuadro 5.3: Reglas de negocio

5.2.3. Requisitos de usuario

En el Cuadro 5.4 se muestran los requisitos de usuario de la app SegoFutsal.

ID	Requisito
RU-01	Crear perfil para un árbitro
RU-02	Modificar perfil de un árbitro
RU-03	Ver detalles de los árbitros
RU-04	Eliminar árbitros
RU-05	Asignar árbitros a partidos de una misma jornada
RU-06	Ajustar la disponibilidad de los árbitros antes de asignarlos a los partidos
RU-07	Acceder a la información de los partidos
RU-08	Filtrar los partidos por liga
RU-09	Filtrar los partidos por categoría
RU-10	Filtrar los partidos por jornada
RU-11	Seleccionar la jornada a asignar
RU-12	Automatizar notificaciones a los árbitros
RU-13	Crear copias de seguridad al asignar la jornada a arbitrar
RU-14	Establecer automáticamente el horario de los partidos al seleccionar una jornada para la asignación
RU-15	Confirmar manualmente la disponibilidad de los árbitros antes de realizar la asignación
RU-16	Mostrar un mensaje de error si no hay árbitros suficientes disponibles para cubrir todos los partidos de una jornada, indicando categoría y franja horaria afectada
RU-17	Visualizar la información de los árbitros ya asignados a partidos a través de la consulta de partidos por jornada

Cuadro 5.4: Requisitos de usuario

5.2.4. Requisitos funcionales

En el Cuadro 5.5 se muestran los requisitos funcionales de la app SegoFutsal.

5.2.5. Requisitos no funcionales

A continuación, se muestran una serie de requistos no funcionales de la app SegoFutsal.

Requisitos de eficiencia

En el Cuadro 5.6 se muestran los diferentes requisitos de eficiencia de SegoFutsal.

ID	Requisito
RF-01	Si el usuario solicita crear, ver, editar o eliminar un árbitro, el sistema deberá gestionar correctamente la información del árbitro seleccionado, además de poder buscarle por nombre entre todos los registrados.
RF-02	El sistema deberá mostrar mensajes claros de éxito o error al realizar operaciones de creación, edición o eliminación de árbitros, informando al usuario del resultado de cada acción.
RF-03	Cuando el usuario acceda a los partidos, el sistema mostrará la información disponible de la temporada permitiendo aplicar filtros por categoría, liga y jornada.
RF-04	Si el usuario selecciona una jornada con árbitros disponibles, el sistema asignará automáticamente horarios y árbitros a los partidos, respetando las restricciones definidas.
RF-05	Si el usuario selecciona una jornada con árbitros disponibles, el sistema asignará automáticamente horarios y árbitros a los partidos, respetando las restricciones definidas en el Cuadro 5.14.
RF-06	Una vez completada la asignación de árbitros a los partidos, el sistema enviará automáticamente un correo a cada árbitro con los detalles de sus partidos asignados.
RF-07	Al finalizar la asignación de una jornada, el sistema generará una copia de seguridad que incluirá la información de los partidos y los árbitros asignados.

Cuadro 5.5: Requisitos funcionales

ID	Requisito
RNF-EF-01	La aplicación debe tener una tasa de fallo de 5 % como máximo.
RNF-EF-02	Debe soportar la concurrencia de múltiples usuarios sin afectar el rendimiento del sistema.
RNF-EF-03	El sistema debe realizar la propia asignación de árbitros en menos de minuto y medio.

Cuadro 5.6: Requisitos de eficiencia

Requisitos de usabilidad

En el Cuadro 5.7 se muestran los diferentes requisitos de usabilidad de SegoFutsal.

ID	Requisito
RNF-US-01	La aplicación deberá ser fácil de utilizar y comprender para todos los usuarios.
RNF-US-02	Proporcionará una interfaz intuitiva para que los usuarios accedan a las funciones principales.

Cuadro 5.7: Requisitos de usabilidad

Requisitos de mantenibilidad

En el Cuadro 5.8 se muestran los diferentes requisitos de mantenibilidad de SegoFutsal.

ID	Requisito
RNF-MAN-01	La aplicación deberá ser capaz de integrarse fácilmente con futuros cambios y mejoras solicitadas.
RNF-MAN-02	El sistema deberá ser capaz de reiniciarse de manera sencilla tras la corrección de errores.

Cuadro 5.8: Requisitos de mantenibilidad

Requisitos de seguridad

En el Cuadro 5.9 se muestran los diferentes requisitos de seguridad de SegoFutsal.

ID	Requisito
RNF-SEG-01	No se permitirá el acceso a la aplicación sin previa autentificación.
RNF-SEG-02	No se permitirá la visualización de las copias de seguridad sin previa autentificación.

Cuadro 5.9: Requisitos de seguridad

Requisitos de disponibilidad

En el Cuadro 5.10 se muestran los diferentes requisitos de disponibilidad de SegoFutsal.

ID	Requisito
RNF-DISP-01	La aplicación estará disponible los 365 días del año para el acceso de los usuarios con conexión a Internet.

Cuadro 5.10: Requisitos de disponibilidad

Requisitos de implementación

En el Cuadro 5.11 se muestran los diferentes requisitos de implementación de SegoFutsal.

ID	Requisito
RNF-IMP-01	La aplicación se desarrollará utilizando Microsoft Power Apps como entorno de desarrollo.
RNF-IMP-02	Se utilizará Microsoft Power Automate para la designación arbitral.
RNF-IMP-03	La base de datos se desarrollará en Microsoft Lists.
RNF-IMP-04	Microsoft SharePoint será usado como repositorio del proyecto.

Cuadro 5.11: Requisitos de implementación

Requisitos de interfaces externas

En el Cuadro 5.12 se muestran los diferentes requisitos de interfaces externas de SegoFutsal.

ID	Requisito
RNF-IE-01	La aplicación debe integrarse con el sistema de correo electrónico para enviar notificaciones automáticas a los árbitros.
RNF-IE-02	La aplicación debe poder guardar las copias de seguridad en formato XSLS.
RNF-IE-03	La aplicación deberá interactuar con la plataforma Microsoft SharePoint para almacenar copias de seguridad.

Cuadro 5.12: Requisitos de interfaces externas

Requisitos de información

En el Cuadro 5.13 se muestran los diferentes requisitos de información de SegoFutsal.

5.2.6. Restricciones

En el Cuadro 5.14 se muestran las diferentes restricciones que debe cumplir la aplicación SegoFutsal.

5.2.7. Suposiciones

En el Cuadro 5.15 se muestran las diferentes suposiciones bajo las cuales se desarrolla la aplicación SegoFutsal.

ID	Requisito
RNF-INF-01	La información básica de un árbitro debe ser su nombre, email, teléfono, categoría asignada (alevín o benjamín) y foto de perfil.
RNF-INF-02	La información básica de un partido debe ser su jornada, equipo local y viistante, árbitro asignado, categoría, liga, fecha y lugar.

Cuadro 5.13: Requisitos de información

ID	Restricción
RES-01	Un árbitro no puede ser asignado a más de un partido que se juega a la misma hora en una jornada.
RES-02	Los árbitros deben ser seleccionados para partidos que correspondan a su categoría de arbitraje o menor.
RES-03	Las asignaciones deben tener en cuenta el número máximo de partidos arbitrados en una temporada (máximo 30).
RES-04	Las asignaciones deben tener en cuenta la disponibilidad actualizada del árbitro.
RES-05	Las asignaciones deben tener en cuenta posibles sanciones que suspendan temporalmente al árbitro de su función.
RES-06	Los equipos solo juegan un partido por jornada.

Cuadro 5.14: Restricciones del sistema

ID	Suposición
SUP-01	Existen dos categorías de arbitraje: alevín y benjamín, cada una con dos ligas diferentes (primera y segunda).
SUP-02	Cada liga consta de 6 equipos, lo que define la cantidad de partidos y árbitros necesarios por jornada.
SUP-03	Los horarios de los partidos están unificados, sin distinción entre franjas S1, S2 o D1.

Cuadro 5.15: Suposiciones del sistema

5.3. Casos de uso

Los casos de uso describen cómo interactúan los usuarios con la aplicación para lograr distintos objetivos. Son una herramienta clave para entender el funcionamiento del sistema desde el punto de vista del usuario.

En este proyecto, los casos de uso se han agrupado en tres subsistemas, que se detallan a continuación:

5.3.1. Subsistema gestionar árbitros

Este subsistema agrupa los casos de uso relacionados con la creación, modificación, consulta y eliminación de árbitros en la aplicación. Permite al usuario administrador mantener actualizada la información de los árbitros disponibles para las asignaciones de partidos.

Nombre e ID del CU	CU-01. Crear árbitro
Actor	Usuario
Descripción	El usuario introduce los datos del árbitro a insertar en la base de datos del sistema (nombre, email, categoría, etc.) y da su alta en el sistema.
Precondiciones	PRE-1. El usuario está autenticado y tiene permiso para utilizar la aplicación.
Postcondiciones	POST-1. El árbitro queda almacenado en la base de datos.
Flujo normal	FN1 El usuario selecciona “Gestión Árbitros” en la pantalla inicial. FN2 El usuario pulsa el botón “+” en la pantalla de gestión de árbitros. FN3 El sistema muestra el formulario de datos del árbitro. FN4 El actor rellena los campos y confirma. FN5 El sistema almacena el nuevo árbitro. FN6 El sistema notifica al actor del éxito de la operación.
Flujos alternativos	FA1 El actor decide cancelar la creación y el sistema descarta los datos introducidos.
Excepciones	E1 Error de conexión con la base de datos.
Prioridad	Alta
Observaciones	—

Cuadro 5.16: CU-01. Crear árbitro

Nombre e ID del CU	CU-02. Modificar árbitro
Actor	Usuario
Descripción	El usuario selecciona un árbitro existente, edita sus datos y guarda los cambios.

Precondiciones	PRE-1. El árbitro ya está dado de alta en el sistema.
Postcondiciones	POST-1. La información del árbitro queda actualizada.
Flujo normal	FN1 El usuario selecciona “Gestión Árbitros” en la pantalla inicial. FN2 El usuario selecciona el árbitro a modificar. FN3 El usuario pulsa el botón del lápiz en la pantalla que muestra la información del árbitro. FN4 El sistema muestra el formulario con los datos actuales. FN5 El actor realiza las modificaciones y confirma. FN6 El sistema valida y actualiza la información.
Flujos alternativos	FA1 El actor cancela la edición y el sistema mantiene los datos previos sin cambios.
Excepciones	E1 Error de conexión con la base de datos.
Prioridad	Media
Observaciones	—

Cuadro 5.17: CU-02. Modificar árbitro

Nombre e ID del CU	CU-03. Ver árbitros
Actor	Usuario
Descripción	El usuario visualiza la lista de todos los árbitros registrados, con posibilidad de búsqueda.
Precondiciones	PRE-1. Existen árbitros registrados en el sistema.
Postcondiciones	POST-1. Ver toda la información acerca de los árbitros
Flujo normal	FN1 El usuario selecciona “Gestión Árbitros” en la pantalla inicial. FN2 El usuario selecciona el árbitro a visualizar.
Flujos alternativos	FA1 No hay árbitros que cumplan el criterio de búsqueda.

Excepciones	E1 Error al recuperar datos de la base de datos.
Prioridad	Media
Observaciones	—

Cuadro 5.18: CU-03. Ver árbitros

Nombre e ID del CU	CU-04. Eliminar árbitro
Actor	Usuario
Descripción	El usuario selecciona un árbitro y lo borra del sistema.
Precondiciones	PRE-1. El árbitro está dado de alta en el sistema.
Postcondiciones	POST-1. El árbitro queda eliminado de la base de datos.
Flujo normal	FN1 El usuario selecciona “Gestión Árbitros” en la pantalla inicial. FN2 El usuario selecciona el árbitro a borrar. FN3 El usuario pulsa el botón de la papelera en la pantalla que muestra la información del árbitro. FN4 El sistema elimina el árbitro.
Flujos alternativos	—
Excepciones	E1 Error al eliminar en la base de datos.
Prioridad	Media
Observaciones	—

Cuadro 5.19: CU-04. Eliminar árbitro

Nombre e ID del CU	CU-05. Buscar árbitros por nombre
Actor	Usuario
Descripción	El usuario introduce un nombre (o parte del mismo) en la barra de búsqueda para filtrar la lista de árbitros.
Precondiciones	PRE-1. Existen árbitros registrados en el sistema.
Postcondiciones	POST-1. Se muestra una lista filtrada de árbitros cuyo nombre coincide con la búsqueda.

Flujo normal	FN1 El usuario accede a la pantalla de “Gestión Árbitros”. FN2 El usuario escribe un nombre en la barra de búsqueda. FN3 El sistema filtra y muestra los árbitros cuyo nombre coincide (parcial o totalmente).
Flujos alternativos	FA1 No se encuentran coincidencias y se muestra el mensaje “No se han encontrado árbitros”.
Excepciones	E1 Error al procesar la búsqueda o al acceder a la base de datos.
Prioridad	Baja
Observaciones	La búsqueda no distingue entre mayúsculas y minúsculas.

Cuadro 5.20: CU-05. Buscar árbitros por nombre

Nombre e ID del CU	CU-06. Recargar lista de árbitros
Actor	Usuario
Descripción	El usuario pulsa un botón (con icono de ruleta) para recargar manualmente la lista de árbitros visibles en pantalla.
Precondiciones	PRE-1. El usuario se encuentra en la pantalla de “Gestión Árbitros”.
Postcondiciones	POST-1. La lista de árbitros se actualiza con los datos más recientes.
Flujo normal	FN1 El usuario accede a la pantalla de “Gestión Árbitros”. FN2 El usuario pulsa el botón de recarga. FN3 El sistema obtiene los datos actualizados y muestra la lista de nuevo.
Flujos alternativos	FA1 Si no hay cambios en los datos, el sistema muestra la misma lista.
Excepciones	E1 Fallo de conexión a la base de datos.
Prioridad	Media

Observaciones	Esta acción puede utilizarse para forzar una sincronización tras otra operación (alta/modificación).

Cuadro 5.21: CU-06. Recargar lista de árbitros

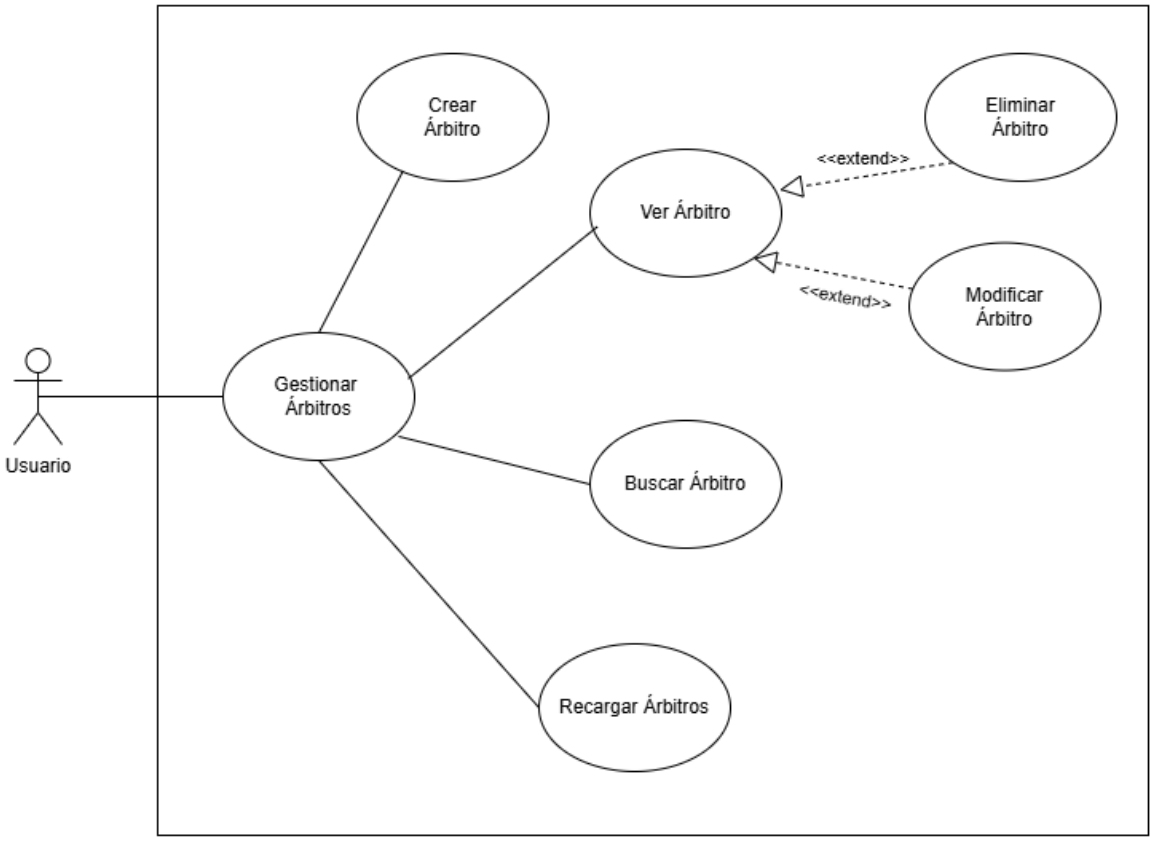


Figura 5.1: Subsistema gestionar árbitros

5.3.2. Subsistema ver partidos

Este subsistema permite al usuario consultar todos los partidos de la temporada registrados en SegoFutsal. Ofrece herramientas para filtrar los partidos por categoría, liga y jornada, así como para visualizar los detalles de un partido específico.

Nombre e ID del CU	CU-07. Ver partidos de la temporada
Actor	Usuario

Descripción	El usuario accede a la vista general de todos los partidos registrados en la temporada.
Precondiciones	PRE-1. Existen partidos registrados en el sistema.
Postcondiciones	POST-1. Se muestra la lista de partidos de la temporada.
Flujo normal	FN1 El usuario selecciona la opción “Gestión Partidos” desde el menú principal. FN2 El sistema muestra todos los partidos registrados en la temporada.
Flujos alternativos	—
Excepciones	E1 Error al acceder a la lista de partidos.
Prioridad	Media
Observaciones	—

Cuadro 5.22: CU-07. Ver partidos de la temporada

Nombre e ID del CU	CU-08. Filtrar partidos por categoría
Actor	Usuario
Descripción	El usuario filtra la lista de partidos según la categoría (alevín, o benjamín).
Precondiciones	PRE-1. Se ha accedido a la vista de partidos.
Postcondiciones	POST-1. Se muestra una lista de los partidos filtrada por la categoría seleccionada.
Flujo normal	FN1 El usuario selecciona una categoría del desplegable. FN2 El sistema filtra la lista de partidos por esa categoría.
Flujos alternativos	—
Excepciones	E1 Error al acceder a la lista de partidos filtrados por categoría seleccionada.
Prioridad	Baja
Observaciones	—

Cuadro 5.23: CU-08. Filtrar partidos por categoría

Nombre e ID del CU	CU-09. Filtrar partidos por liga
Actor	Usuario
Descripción	El usuario filtra la lista de partidos según la liga (primera o segunda).
Precondiciones	PRE-1. Se ha accedido a la vista de partidos.
Postcondiciones	POST-1. Se muestra una lista filtrada por la liga seleccionada.
Flujo normal	FN1 El usuario selecciona una liga (Primera o Segunda) del desplegable. FN2 El sistema filtra la lista de partidos por esa liga.
Flujos alternativos	—
Excepciones	E1 Error al acceder a la lista de partidos de una liga.
Prioridad	Baja
Observaciones	—

Cuadro 5.24: CU-09. Filtrar partidos por liga

Nombre e ID del CU	CU-10. Filtrar partidos por jornada
Actor	Usuario
Descripción	El usuario filtra la lista de partidos según el número de la jornada.
Precondiciones	PRE-1. Se ha accedido a la vista de partidos.
Postcondiciones	POST-1. Se muestra una lista filtrada por la jornada seleccionada.
Flujo normal	FN1 El usuario selecciona una jornada del desplegable (de la 1 a la 10). FN2 El sistema filtra la lista de partidos por esa jornada.
Flujos alternativos	—
Excepciones	E1 Error al acceder a la lista de partidos de una jornada.
Prioridad	Baja
Observaciones	—

Cuadro 5.25: CU-10. Filtrar partidos por jornada

Nombre e ID del CU	CU-11. Ver detalles de un partido
Actor	Usuario
Descripción	El usuario selecciona un partido específico para consultar sus detalles (equipos, fecha, lugar, árbitro, etc.).
Precondiciones	PRE-1. El partido se encuentra listado en pantalla.
Postcondiciones	POST-1. Se muestran los datos detallados del partido seleccionado.
Flujo normal	FN1 El usuario hace clic sobre un partido de la lista. FN2 El sistema despliega la información detallada del partido.
Flujos alternativos	—
Excepciones	E1 Error al cargar los detalles del partido.
Prioridad	Media
Observaciones	—

Cuadro 5.26: CU-11. Ver detalles de un partido

Nombre e ID del CU	CU-12. Borrar filtros
Actor	Usuario
Descripción	El usuario se encuentra en la pantalla donde se ve la infomación de los partidos y pulsa el botón de “Borrar filtros” para borrar los filtos.
Precondiciones	PRE-1. Anteriormente se ha aplicado algún filtrado por jornada, categoría o liga
Postcondiciones	POST-1. Se muestra la lista completa de los partidos de la temporada.
Flujo normal	FN1 El usuario hace clic sobre el botón “Borrar filtros”. FN2 El sistema actualiza la lista completa de los partidos de la temporada.
Flujos alternativos	—
Excepciones	E1 Error al borrar los filtros. E2 Error al mostrar de nuevo todos los partidos.
Prioridad	Baja

Observaciones	—

Cuadro 5.27: CU-12. Borrar filtros

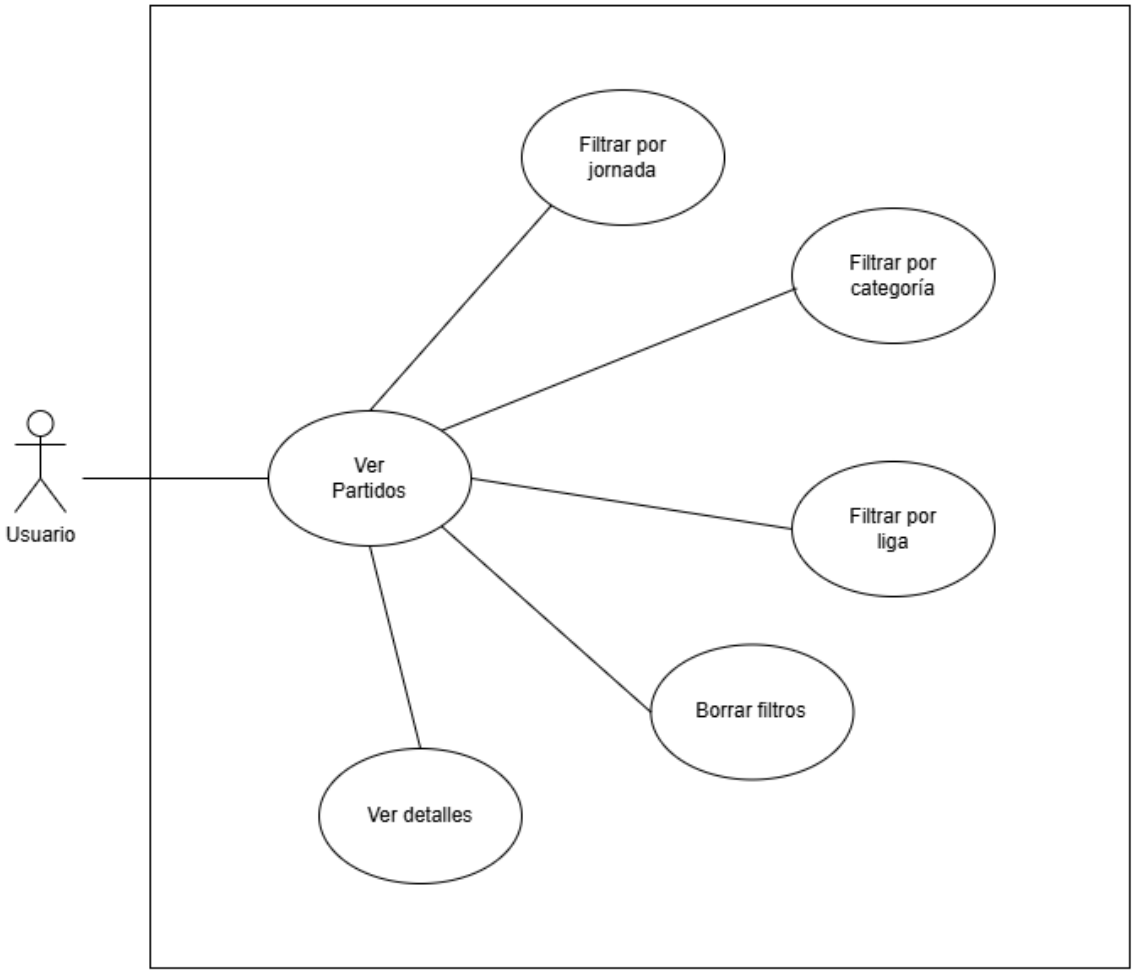


Figura 5.2: Subsistema ver partidos

5.3.3. Subsistema asignar de árbitros

Este subsistema gestiona la asignación de árbitros a los partidos de una jornada completa, teniendo en cuenta una serie de restricciones que están definidas en el Cuadro 5.14. El sistema realiza automáticamente las siguientes tareas: asignación de horario y árbitros a cada partido, notificación por correo y creación de la copia de seguridad con los datos de la jornada.

Nombre e ID del CU	CU-13. Seleccionar jornada
Actor	Usuario
Descripción	El usuario elige de un desplegable la jornada para la cual se asignarán árbitros.
Precondiciones	PRE-1. Se está en la pantalla de asignación de árbitros.
Postcondiciones	POST-1. La jornada queda fijada para el proceso de asignación.
Flujo normal	FN1 El sistema muestra un desplegable con todas las jornadas disponibles. FN2 El usuario selecciona una jornada. FN3 El sistema marca como activa la jornada seleccionada.
Flujos alternativos	—
Excepciones	E1 La jornada no queda marcada como activa.
Prioridad	Alta
Observaciones	—

Cuadro 5.28: CU-13. Seleccionar jornada

Nombre e ID del CU	CU-14. Marcar disponibilidad de árbitros
Actor	Usuario
Descripción	El usuario indica los turnos S1, S2, y D1 en los que los árbitros están disponibles para arbitrar.
Precondiciones	PRE-1. Se está en la pantalla de asignación de árbitros.
Postcondiciones	POST-1. El sistema registra el estado de disponibilidad de cada árbitro.
Flujo normal	FN1 El sistema muestra la lista de árbitros con casillas de verificación. FN2 El usuario marca o desmarca los árbitros disponibles. FN3 El sistema guarda la configuración de disponibilidad.
Flujos alternativos	—
Excepciones	E1 Error al guardar la disponibilidad en la base de datos.
Prioridad	Alta

Observaciones	Solo los árbitros marcados se tendrán en cuenta en la asignación.

Cuadro 5.29: CU-14. Marcar árbitros disponibles

Nombre e ID del CU	CU-15. Asignar árbitros a una jornada
Actor	Usuario
Descripción	El usuario inicia el proceso de asignación automática de árbitros a todos los partidos de una jornada.
Precondiciones	▪ PRE-1. El usuario se encuentra en la vista de asignación de árbitros. ▪ PRE-2. Existen partidos registrados en la jornada seleccionada. ▪ PRE-3. Existen árbitros registrados en el sistema. ▪ PRE-4. El usuario ha marcado la disponibilidad horaria de los árbitros (CU-14).
Postcondiciones	▪ POST-1. Cada partido de la jornada tiene asignado un árbitro disponible. ▪ POST-2. Los árbitros asignados han recibido una notificación por correo electrónico. ▪ POST-3. Se ha generado una copia de seguridad con la información de los partidos, árbitros y horarios asignados.
Flujo normal	FN1 El usuario pulsa el botón “Asignar”. FN2 El sistema ejecuta su algoritmo para determinar la asignación final teniendo en cuenta las restricciones. FN3 El sistema notifica al usuario el éxito de la operación.
Flujos alternativos	FA1 La asignación falla por algún motivo y se muestra un mensaje de error y su causa.

Excepciones	E1 Fallo al procesar la asignación por falta de árbitros en una categoría o en un horario. E2 Fallo al procesar la asignación automática por límite de tiempo.
Prioridad	Muy Alta
Observaciones	Las restricciones están descritas en el Cuadro 5.14.

Cuadro 5.30: CU-15. Asignar árbitros a una jornada

Nombre e ID del CU	CU-16. Asignar horario a partidos
Actor	Usuario
Descripción	El sistema asigna automáticamente el horario de cada partido de la jornada.
Precondiciones	▪ PRE-1. El usuario se encuentra en la vista de asignación de árbitros. ▪ PRE-2. Existen partidos registrados en la jornada seleccionada. ▪ PRE-3. Existen árbitros registrados en el sistema. ▪ PRE-4. El usuario ha marcado la disponibilidad horaria de los árbitros (CU-14).
Postcondiciones	POST-1. Cada partido tiene un horario asignado.
Flujo normal	FN1 El usuario pulsa el botón “Asignar”. FN2 El sistema ejecuta su algoritmo para determinar el horario de los partidos.
Flujos alternativos	—
Excepciones	—
Prioridad	Alta
Observaciones	Se integra con CU-15 para asignar árbitros.

Cuadro 5.31: CU-16. Asignar horario a partidos

Nombre e ID del CU	CU-17. Notificar árbitros por correo
Actor	Usuario

Descripción	El sistema envía un correo electrónico con la información del partido a cada árbitro asignado.
Precondiciones	PRE-1. Horario y asignaciones confirmadas y realizadas de manera exitosa (CU-15 y CU-16).
Postcondiciones	POST-1. Los árbitros han recibido la notificación con la información de los partidos que arbitran.
Flujo normal	FN1 El sistema genera un correo por partido dirigido al árbitro designado con todos los detalles. FN2 Envía los correos automáticamente.
Flujos alternativos	—
Excepciones	E1 Fallo en el envío de correo a un árbitro.
Prioridad	Alta
Observaciones	—

Cuadro 5.32: CU-17. Notificar árbitros por correo

Nombre e ID del CU	CU-18. Crear copia de seguridad
Actor	Usuario
Descripción	El sistema genera y almacena una copia de seguridad de los datos de la jornada.
Precondiciones	PRE-1. Horario y asignaciones confirmadas y realizadas de manera exitosa (CU-15 y CU-16).
Postcondiciones	POST-1. La copia de seguridad queda guardada en un archivo .xslx con el nombre “CopiaSeguridad_JornadaX”, donde X es el número de la jornada asignada.
Flujo normal	FN1 El sistema lee los datos de la jornada. FN2 Genera un fichero de copia de seguridad. FN3 Almacena el fichero en la ubicación configurada.
Flujos alternativos	—
Excepciones	E1 Error al escribir el fichero de respaldo.
Prioridad	Alta
Observaciones	—

Cuadro 5.33: CU-18. Crear copia de seguridad

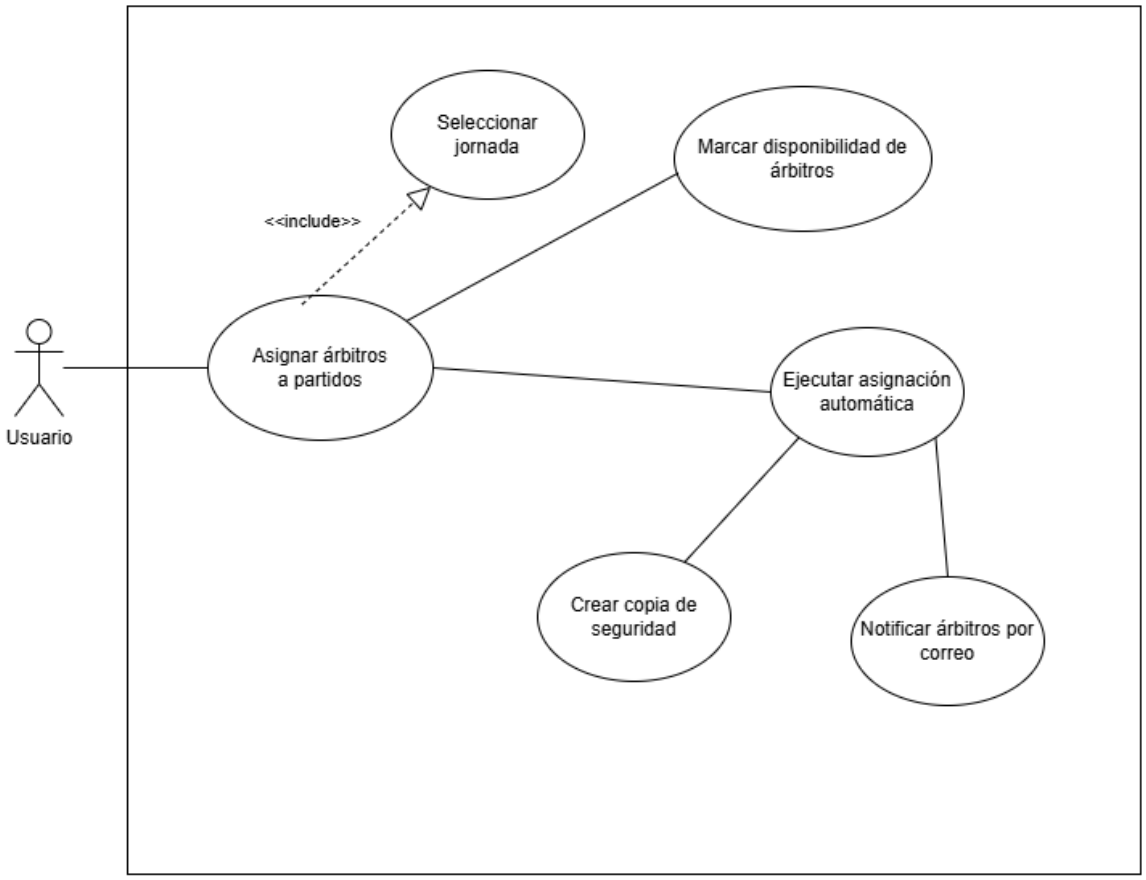


Figura 5.3: Subsistema asignar árbitros

Capítulo 6

Diseño

En este capítulo se detalla el diseño de SegoFutsal. Se describe la arquitectura lógica y física del sistema, el modelo entidad-relación utilizado para estructurar los datos y el modelo lógico de datos en forma de diccionario. Todo ello con el objetivo de garantizar una base sólida, escalable y coherente para el funcionamiento del sistema.

6.1. Arquitectura lógica

La arquitectura lógica define la organización estructural del sistema en capas, separando las diferentes responsabilidades funcionales para mejorar la mantenibilidad, escalabilidad y reutilización del software. Esta división permite que cada componente se desarrolle, modifique y pruebe de forma independiente, facilitando el trabajo colaborativo y la evolución del sistema a lo largo del tiempo.

En esta aplicación se han definido tres capas principales:

- **Capa de presentación**: Encargada de la interfaz de usuario y la interacción directa con el sistema. Se ha implementado utilizando Microsoft Power Apps, permitiendo a los usuarios acceder a la aplicación desde diferentes dispositivos de forma sencilla e intuitiva.

- **Capa de negocio**: Contiene la lógica principal del sistema, encargada de gestionar los árbitros, y las designaciones arbitrales con los flujos de trabajo. Esta capa se ha desarrollado utilizando Microsoft Power Apps y Power Automate, permitiendo la automatización de procesos.

- **Capa de persistencia**: Responsable del almacenamiento y recuperación de los datos. Para esta aplicación, se han utilizado *Microsoft SharePoint* y *Microsoft Lists* como soluciones de almacenamiento, integradas de forma nativa con el ecosistema de Microsoft 365 y compatibles con Microsoft Power Apps y Microsoft Power Automate.

6.2. Arquitectura física

La arquitectura física se refiere a la infraestructura tecnológica necesaria para el despliegue y funcionamiento de la aplicación, incluyendo servidores, bases de datos, redes y dispositivos. En este proyecto, una de las principales ventajas ha sido el uso del ecosistema de Microsoft Power Platform, que permite abstraerse de la gestión de la infraestructura física.

Gracias a esta tecnología, no es necesario configurar ni mantener servidores manualmente, ya que Microsoft proporciona todo lo necesario en la nube. Esto simplifica enormemente el desarrollo y despliegue, permitiendo centrarse en la lógica de negocio y el diseño funcional de la aplicación.

Para trabajar con esta arquitectura, únicamente se requiere un ordenador con conexión a Internet para desarrollar la solución, y posteriormente, cualquier dispositivo con acceso a la red (como un ordenador o un teléfono móvil) para utilizar la aplicación desde Power Apps.

6.3. Diagrama de entidad-relación

El modelo entidad-relación es una representación gráfica de la estructura lógica de la base de datos. Se utiliza para identificar las entidades relevantes del sistema, sus atributos y las relaciones entre ellas. Este diagrama sirve como base para el diseño de la base de datos, asegurando una organización coherente y eficiente de la información.

En la Figura 6.1 se muestra el diagrama entidad-relación diseñado para esta aplicación. Aunque a primera vista pueda parecer un modelo sencillo, es suficiente para cubrir los objetivos funcionales planteados, centrados en la designación arbitral. No se ha buscado una estructura excesivamente compleja, sino una representación clara, eficiente y adecuada a las necesidades del sistema actual. No obstante, en la Sección 4.2 se proponen una serie de posibles mejoras y ampliaciones de SegoFutsal que, de implementarse, requerirían una ampliación del modelo, lo que implicaría una mayor complejidad tanto a nivel de entidades como de relaciones.

6.4. Modelo lógico de datos

En esta sección se detalla el modelo lógico de datos a partir del diagrama entidad-relación. Se describen los atributos de cada entidad y relación, incluyendo sus nombres, tipos de datos, claves primarias y foráneas, así como su función dentro del sistema. Además, se especifica la cardinalidad entre entidades, reflejando cómo se estructura la información en la base de datos para garantizar su integridad y coherencia.

Cabe destacar que, al utilizar Microsoft Lists como sistema de almacenamiento, cada lista funciona como una tabla dentro de una base de datos. Estas listas incluyen automáticamente una serie de campos internos generados por la propia plataforma. De entre ellos, el único utilizado en esta aplicación es el campo ID, un número entero que se asigna

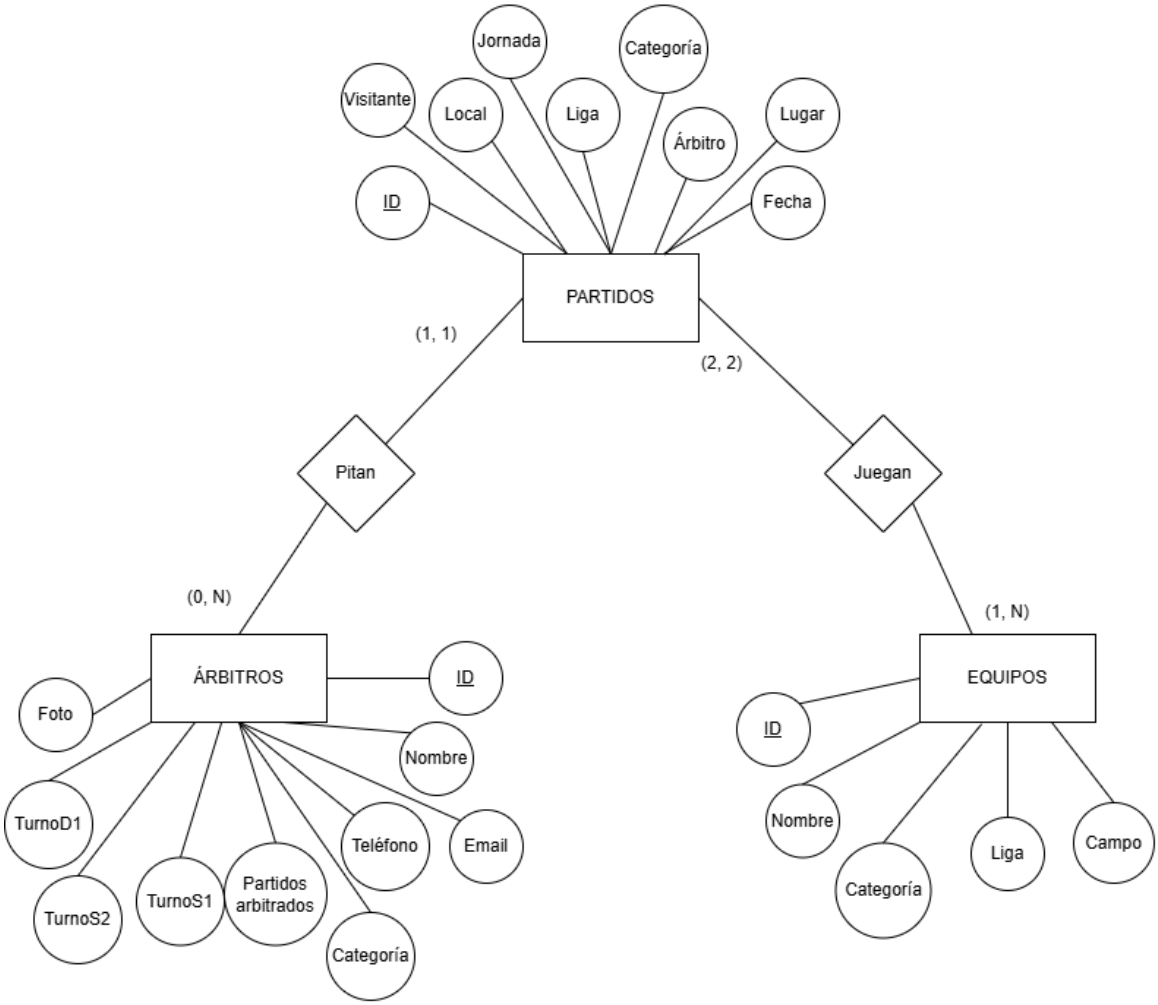


Figura 6.1: Modelo Entidad-Relación

automáticamente a cada nuevo registro y que actúa como identificador único dentro de la lista, cumpliendo la función de clave primaria.

Entidad Árbitros

Nombre	**Definición**	**Tipo**	**Restricciones**
ID	Identificador único del árbitro	int	Primary Key
Nombre	Nombre completo del árbitro	varchar	Not null
Email	Correo electrónico del árbitro	varchar	Not null
Telefono	Teléfono de contacto	varchar	—
Categoria	Nivel de arbitraje del árbitro	elección	Alevín o Benjamín

Nombre	Definición	Tipo	Restricciones
TurnoS1	Disponibilidad en sábado a las 11:00	boolean	Not null
TurnoS2	Disponibilidad en sábado a las 13:00	boolean	Not null
TurnoD1	Disponibilidad en domingo a las 11:00	boolean	Not null
Partidos arbitrados	Número de partidos ya arbitrados	int	Default 0
Foto	Imagen de perfil del árbitro	imagen	—

Cuadro 6.1: Entidad Árbitros

Entidad Partidos

Nombre	Definición	Tipo	Restricciones
ID	Identificador único del partido	int	Primary Key
Jornada	Número de jornada	int	Not null
Local	Nombre del equipo local	varchar	Not null
Visitante	Nombre del equipo visitante	varchar	Not null
Arbitro	Árbitro asignado al partido	int	Foreign Key
Categoria	Categoría del partido	elección	Alevín o Benjamín
Liga	Nombre de la liga	elección	Primera o Segunda
Fecha	Fecha del partido	elección	S1, S2, o D1
Lugar	Lugar donde se disputa el partido	varchar	Not null

Cuadro 6.2: Entidad Partidos

Entidad Equipos

Nombre	Definición	Tipo	Restricciones
ID	Identificador único del equipo	int	Primary Key
Nombre	Nombre oficial del equipo	varchar	Not null
Categoria	Categoría en la que compite el equipo	elección	Alevín o Benjamín
Liga	Nombre de la liga a la que pertenece	elección	Primera o Segunda
Campo	Campo donde juega el equipo	varchar	Not null

Cuadro 6.3: Entidad Equipos

Relación Pitar

Entidad	Participación	Cardinalidad
Árbitros	0	N
Partidos	1	1

Cuadro 6.4: Relación entre Árbitros y Partidos

Relación Jugar

Entidad	Participación	Cardinalidad
Equipos	1	N
Partidos	2	2

Cuadro 6.5: Relación entre Equipos y Partidos

Capítulo 7

Implementación

En este capítulo se expone el proceso seguido para llevar a cabo la implementación del sistema, detallando cómo se tradujeron en soluciones concretas los elementos previamente definidos en el diseño.

7.1. Pantallas

A continuación, se detallan las pantallas que componen la aplicación desarrollada en Microsoft Power Apps. Las principales vistas implementadas son las siguientes:

- Home (scrHome).
- Gestión Árbitros (scrGestionArbitros).
- Nuevo Árbitro (scrNuevoArbitro).
- Editar Árbitro (scrEditarArbitro).
- Ver Árbitro (scrVerArbitro).
- Gestión Partidos (scrGestionPartidos).
- Ver Partidos (scrVerPartidos).
- Asignar Árbitros (scrAsignarArbitros).

7.1.1. Home

Esta vista actúa como pantalla de inicio, dando la puerta de entrada a la aplicación. En esta pantalla se encuentran el mensjae personalizado de bienvenida y tres botones: Gestión Árbitros, Gestión Partidos y Asignar Árbitros, los cuales redirigen a sus respectivas secciones.

Para implementar el mensaje personalizado se utiliza una etiqueta de texto, y en su propiedad *Text* aparece lo siguiente

Text
```
"¡Bienvenido " & UsuariosdeOffice365.MyProfileV2().givenName & "!"
```

Se puede ver que se hace uso de la conexión que tiene Microsoft Power Apps con la propia cuenta de Microsoft Office 365 del usuario.

Otra de las propiedades más importantes que tienen los componentes es la propiedad *OnSelect*, que define el comportamiento cuando se clica en ese mismo componente. En concreto, cuando se pulsa el botón de "Gestión de partidos" se ejecuta:

OnSelect
```
Reset(ddwCategoria);;
Reset(ddwLiga);;
Reset(txtJornada);;
Navigate(scrGestionPartidos; ScreenTransition.Fade)
```

Para el de "Gestión de Árbitros":

OnSelect
```
Navigate(scrGestionArbitros; ScreenTransition.Fade);;
Reset(txtBarraBusqueda)
```

Y por último, para el de "Asignar Árbitros":

OnSelect
```
Set(var_fin_flujo; false);;
Navigate(scrAsignarArbitros; ScreenTransition.Fade)
```

Como se observa, todos tienen en común que para navegar entre las distintas vistas de la aplicación se usa el comando *Navigate()*.En concreto, esta forma es la que más se usa en Microsoft Power Apps para navegar entre las vistas o pantallas con animaciones. Otra forma (que no se ha optado por usar pero que es igual de útil) para navegar a la vista en la que hemos estado justo anteriormente es la siguiente:

OnSelect
```
Back()
```

7.1.2. Gestión Árbitros

Esta pantalla representa el punto de partida para toda la gestión de árbitros dentro de la aplicación. Desde aquí, el usuario puede visualizar de forma centralizada todos los árbitros registrados en la base de datos. Para ello, se utiliza una galería de Microsoft Power Apps (ya comentado en la Sección 2.2.2), un componente visual que permite mostrar listas de elementos (en este caso, árbitros) de forma dinámica y personalizable. La galería se conecta directamente a la lista donde están registrados y se actualiza en tiempo real.

En concreto, la propiedad *Items* de la galería contiene este fragmento de código para mostrar los árbitros en función de la barra de búsqueda que contiene esta vista.

Items

```
If(IsBlank(txtBarraBusqueda);
    ARBITROS;
    Search(ARBITROS; txtBarraBusqueda.Text; 'Nombre (NombreArbitro)')
)
```

Para ver cómo se vuelve a cargar la galería utilizando el botón de refrescar, basta con ver la propiedad *OnSelect* de dicho botón:

OnSelect

```
Refresh([@ARBITROS])
```

Si hacemos lo mismo para el botón de añadir, vemos su comportamiento, que es el siguiente:

OnSelect

```
Navigate(scrNuevoArbitro; ScreenTransition.CoverRight);;
ResetForm(frmNuevoArbitro);;
Set(var_intento_envio; false);;
NewForm(frmNuevoArbitro)
```

Por último, para poder ver toda la información de los árbitros, basta con pulsar en la flecha que aparece en cada registro, ya que su propiedad *OnSelect* es la siguiente:

OnSelect

```
Navigate(scrVerArbitro; ScreenTransition.CoverRight)
```

7.1.3. Nuevo Árbitro

Como se acaba de ver, si se pulsa el botón que indica que se desea añadir un nuevo árbitro en la pantalla anterior, nos redirige a esta, que permite al usuario añadir un nuevo

árbitro al sistema mediante un formulario estructurado. Se trata de una de las dos formas principales que ofrece Microsoft Power Apps para insertar registros en una base de datos (la otra forma se ve en la Sección 7.1.8). El formulario está conectado directamente a Microsoft Lists y facilita la introducción de la información necesaria, aplicando validaciones básicas para garantizar la calidad de los datos antes de su envío. Cabe destacar que una vez seleccionado el componente de formulario en Microsoft Power Apps, hace falta configurarlo con la lista y los campos que se desean, además de elegir si estos campos se quieren ver, modificar, o crear desde cero.

El botón para cancelar esta operación de añadir un árbitro nos devuelve de nuevo a la pantalla de gestión de árbitros:

OnSelect

```
Navigate(scrGestionArbitros; ScreenTransition.UnCoverRight)
```

Por último, si finalmente se registra un nuevo árbitro, se ejecuta

OnSelect

```
Set(var_intento_envio; true);;

If(
    frmNuevoArbitro.Valid && IsMatch(DataCardValue2.Text;
     "^[\w\.-]+@[\w\.-]+\.\w{2,4}$");
    SubmitForm(frmNuevoArbitro);;
    Navigate(scrGestionArbitros; ScreenTransition.Fade);;
    Notify("Árbitro registrado correctamente";
     NotificationType.Success; 2000)


    Notify("Hay errores en el formulario. Corrige
     los campos indicados."; NotificationType.Error)
)
```

Es importante destacar que hay una serie de validaciones que se deben cumplir para que el formulario se envie. En concreto, los campos de “Nombre”, “Email”, y “Categoría” son obligatorios. A mayores, se utiliza una expresión regular para que el “Email” tenga un formato correcto (como por ejemplo, example@mail.com). Para mostrar los errores concretos, se utiliza una etiqueta de texto en cada campo, y se programa tanto la propiedad *Text* para indicar el mensaje, y la propiedad *Visible* para que se muestre cuando es necesario. A continuación se muestra cómo quedaría para el campo del “Email” (análogo para el “Nombre” y “Categoría”)

Text

```
If(
    IsBlank(DataCardValue2.Text);
    "obligatorio";
    If(
        !IsMatch(DataCardValue2.Text; "^[\w\.-]+@[\w\.-]+\.\w{2,4}$");
        "formato incorrecto";
        ""
    )
)
```

Visible

```
var_intento_envio && (
    IsBlank(DataCardValue2.Text) ||
    !IsMatch(DataCardValue2.Text; "^[\w\.-]+@[\w\.-]+\.\w{2,4}$")
)
```

Por último, hay una foto de perfil precargada en el caso de que el usuario no inserte una foto suya. Para ello, es suficiente con insertar esta foto de perfil predeterminada en el campo de la imagen del formulario.

Image

```
If(
    IsBlank(AddPicture1.Media);
    defaultPerfil;
    AddPicture1.Media
)
```

7.1.4. Ver Árbitro

Esta pantalla está diseñada para mostrar de forma detallada la información de un árbitro específico. Se accede a ella desde la galería que se encuentra en la vista de gestión de árbitros y permite consultar todos los datos registrados del árbitro seleccionado.

Para acceder a los datos del árbitro desde la galería, hay que usar el comando *galArbitros.Selected*, cargándose asi el registro entero. Por ejemplo, si ponemos en una etiqueta de texto el siguiente comando

Text

```
galArbitros.Selected.Email
```

mostraría el correo electrónico del árbitro previamente seleccionado. Para los demás campos se hace de manera análoga.

Además, desde esta vista se puede acceder a acciones complementarias como editar (botón con forma de lápiz):

OnSelect

```
ResetForm(frmEditarArbitro);;
EditForm(frmEditarArbitro);;
Set(var_intento_envio; false);;
Navigate(scrEditarArbitro; ScreenTransition.CoverRight)
```

o eliminar el registro (botón con forma de papelera):

OnSelect

```
Remove([@ARBITROS]; galArbitros.Selected);;
Notify("Árbitro eliminado correctamente";
 NotificationType.Success; 2000);;
Navigate(scrGestionArbitros; ScreenTransition.UnCoverRight)
```

7.1.5. Editar Árbitro

En cuanto a su implementación, esta pantalla comparte la misma estructura y funcionamiento que la utilizada para añadir un nuevo árbitro (para las validaciones es igual). De la misma forma en esa vista, se emplea un formulario conectado a la base de datos para gestionar los datos del árbitro. La principal diferencia radica en que, en este caso, los campos del formulario se cargan automáticamente con la información del árbitro previamente seleccionado. Para ello, basta con combinar la propiedad *Item* que tiene el formulario con el comando que se acaba de ver para seleccionar el árbitro *galArbitros.Selected*:

Item

```
galArbitros.Selected
```

7.1.6. Gestión Partidos

La vista de gestión de partidos tiene un comportamiento similar al que tiene la de gestión de árbitros. Sin embargo, el hecho de que haya filtros, hace que el código para mostrar los elementos de la galería de partidos sea algo más complejo.

Item

```
SortByColumns(
    Filter(
        PARTIDOS;
        (IsBlank(ddwCategoria.Selected.Value) ||
         Categoria.Value = ddwCategoria.Selected.Value);
        (IsBlank(ddwLiga.Selected.Value) ||
         Liga.Value = ddwLiga.Selected.Value);
        (IsBlank(txtJornada.Text) ||
         Jornada = Value(txtJornada.Text))
    );
    "Jornada";
    SortOrder.Ascending
)
```

Para los filtros de categoría y liga, su código en la propiedad *Items* es el siguiente (respectivamente)

Items

```
Table(
    {Value: ""};
    Choices(PARTIDOS.Categoria)
)
```

Items

```
Table(
    {Value: ""};
    Choices(PARTIDOS.Liga)
)
```

En cuanto al filtro de jornada, no es más que un componente de entrada de texto, donde el usuario puede introducir el número de la jornada que desee. Para finalizar esta vista, el botón que limpia los filtros contiene estas instrucciones

OnSelect

```
Reset(ddwCategoria);;
Reset(ddwLiga);;
Reset(txtJornada)
```

7.1.7. Ver Partidos

Esta vista esta implementada de la misma forma que la pantalla utilizada para consultar los detalles de un árbitro, con la diferencia de que, en este caso, se muestra la información correspondiente a un partido.

Para aquellos partidos que no tengan ni horario ni árbitro asignado, se muestra un mensaje de la siguiente forma en la etiqueta de texto correspondiente a cada caso

Text
```
If(IsBlank(galPartidos.Selected.Fecha);
    "(Sin asignar)";
    galPartidos.Selected.Fecha.Value
)
```

Text
```
If(IsBlank(galPartidos.Selected.Arbitro);
    "(Sin asignar)";
    LookUp(ARBITROS; ID = galPartidos.Selected.Arbitro.Value;
    NombreArbitro)
)
```

7.1.8. Asignar Árbitros

Esta pantalla permite asignar los árbitros a los partidos de una jornada. Para ello, se han usado gran cantidad de elementos y distintas de sus propiedades.

En primer lugar, tenemos un desplegable con las opciones disponibles para que el usuario marque la jornada a asignar

Items
```
ForAll(
    Sequence(10);
    Value
)
```

Como sabemos que cada liga tiene 10 jornadas (Cuadro 5.15) bastaría con esto, pero en el caso de que hubiera más jornadas, habría que retocar este componente.

Posteriormente, hay una galería de árbitros simple

Items
```
ARBITROS
```

que muestra la imagen y el nombre del árbitro

Image

```
ThisItem.Foto
```

Text

```
ThisItem.'Nombre (NombreArbitro)'
```

además de tres interruptores a mayores para definir la disponibilidad o no del árbitro en los tres turnos predefinidos donde hay partidos (S1, S2, y D1).

En cuanto al botón de "Asignar" (inicia el flujo lógico para las designaciones arbitrales), se puede observar que tiene el siguiente comportamiento:

OnSelect

```
ForAll(galDisponibilidades.AllItems;
    Patch(
        ARBITROS;
        ThisRecord;
        {
            TurnoS1: togS1.Value;
            TurnoS2: togS2.Value;
            TurnoD1: togD1.Value
        }
    )
);;


Set(var_mensaje_flujo; "");;

If(IsBlank(LookUp(PARTIDOS;Jornada =
 Value(ddwJornada.Selected.Value)).Arbitro);
    Set(var_fin_flujo;true);;
    Set(resultado_flujo; Asignaciones.Run(ddwJornada.Selected.Value));;
    Set(var_mensaje_flujo; resultado_flujo.salida);

    Notify("Esa jornada ya ha sido asignada anteriormente";
     NotificationType.Error;3000);;
)
```

En este fragmento de código podemos distinguir dos funcionalidades específicas.

La primera de ellas es destacable por la función *Patch*, que permite actualizar un registro existente o agregar uno nuevo en una tabla de datos, especificando los campos y valores que se desean guardar. Es especialmente útil cuando se necesita más control que

el que ofrecen los formularios prediseñados. Este en concreto, lo que hace es actualizar las disponibilidades para tenerlas en cuenta antes de ejecutar el flujo de designaciones arbitrales.

Por otro lado, en la segunda parte vemos que se implementa la condición de que la jornada a asignar no debe haber sido asignada anteriormente. El comando *Notify()* muestra una notificación al usuario de distintos tipos (error, advertencia, éxito) y con el mensaje que tenga por parámetro.

Para terminar, si se pulsa el botón de "Asignar" hace que aparezca un pop-up, por lo que se juega con la propiedad vista anteriormente *Visible*:

Visible

```
var_fin_flujo
```

Esta variable booleana marca la visibilidad o no de componentes como el gif de carga mientras se ejecuta el flujo, las etiquetas de texto o el botón que aparece.

Para notificar al usuario de cómo va el flujo, si se está ejecutando, ha tenido algún error, o ha finalizado correctamente, se utiliza la variable *var_ mensaje_ flujo*. Ejemplos de esto se ven en el título:

Text

```
If(var_mensaje_flujo = "";
    "El flujo se está ejecutando";
    "El flujo ha finalizado"
)
```

y en el cuerpo del pop-up:

Text

```
If(
    var_mensaje_flujo = "";
    "Espere a que termine la ejecución del flujo. Solo será un momento.";
    If(
        var_mensaje_flujo = "OK";
        "Flujo finalizado correctamente.";
        "ERROR DURANTE LA EJECUCIÓN: " & resultado_flujo.salida
    )
)
```

Finalmente, el botón que que permite volver al inicio tiene una visibilidad un poco distinta:

Visible

```
var_fin_flujo And Not(var_mensaje_flujo = "")
```

y una vez se pulsa, se ejecuta lo siguiente:

OnSelect

```
If(var_mensaje_flujo = "OK";
    EnviarCorreo.Run(ddwJornada.Selected.Value)
);;

Navigate(scrHome; ScreenTransition.Fade);;
```

Nótese que es aqui donde se ejecuta el flujo lógico para mandar los correos, ya que esta funcionalidad depende de si se ha realizado correctamente el proceso de designaciones arbitrales.

7.2. Flujos Lógicos

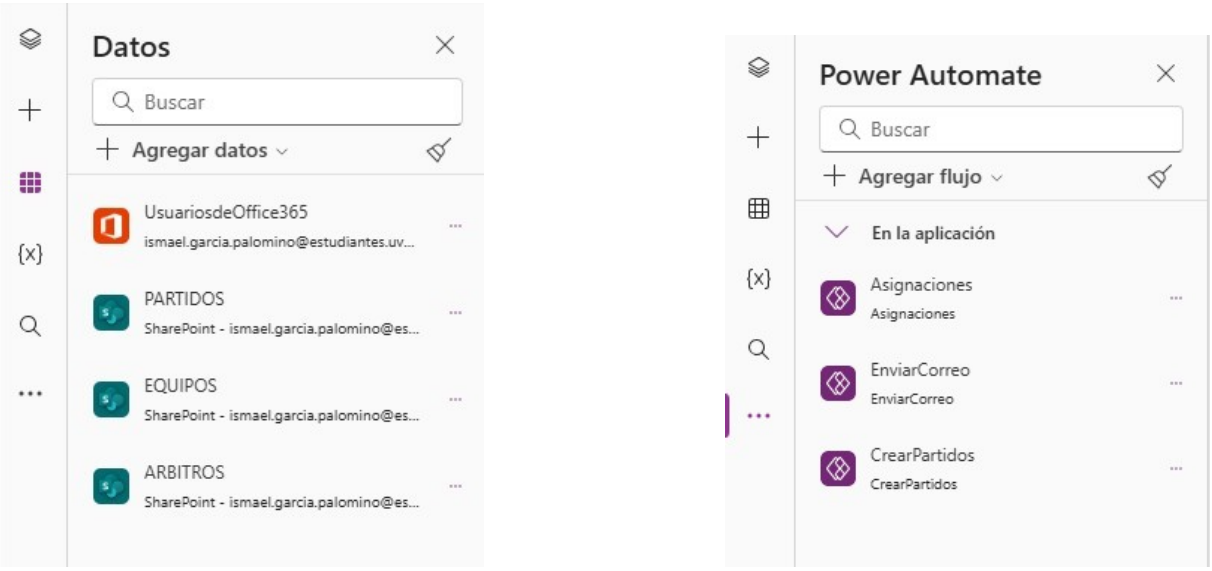
A continuación, se detallan los flujos lógicos que se han usado para la implementación de SegoFutsal, desarrollados en Microsoft Power Automate. Los principales son los siguientes:

- Crear Partidos
- Asignaciones
- Enviar Correo

Antes de ver los flujos es importante recalcar que hay que conectar tanto los datos de Microsoft Lists como los flujos que se van a utilizar de Microsoft Power Automate a la aplicación en Microsoft Power Apps. La Figura 7.1 muestran cómo se hace.

7.2.1. Crear Partidos

Este flujo de Microsoft Power Automate se ha creado para generar automáticamente los partidos a partir de la información contenida en la lista de equipos. El objetivo principal de este flujo es facilitar la creación de los registros necesarios en la lista de partidos sin intervención manual, generando un conjunto de datos iniciales con los que comenzar a trabajar en SegoFutsal. Aunque está conectado al sistema, no se utiliza en el funcionamiento habitual de la aplicación, ya que su uso está limitado a la fase de preparación y carga inicial de datos.



(a) Conectar listas o tablas

(b) Conectar flujos

Figura 7.1: Conexiones en Microsoft Power Apps

En primer lugar, el flujo genera automáticamente todos los partidos posibles asignando a cada equipo el rol de local dentro de su correspondiente liga y categoría. Para cada uno de estos partidos, se crea un registro en la lista de partidos que ya incluye la información relativa a la liga, la categoría y el lugar (campo del equipo local) asociado al equipo que organiza el encuentro.

En segundo lugar, y utilizando la lógica de Microsoft Power Automate, se ha completado cada uno de los registros generados asociándoles un equipo visitante que pertenezca a la misma liga y categoría que el equipo local. De esta forma, se garantiza que todos los partidos queden correctamente definidos con ambos equipos asignados y dentro del contexto competitivo correspondiente.

Finalmente, la asignación de la jornada a cada partido se ha realizado manualmente, una vez completada la generación automática de los registros. Tal como se menciona en la Sección 4.2, una posible mejora futura sería automatizar por completo el proceso de carga de los partidos de una temporada, eliminando así la intervención tanto por etapas como de manera manual.

7.2.2. Asignaciones

Este es el flujo principal de la aplicación, ya que realiza las tareas de determinar las fechas de los partidos, las designaciones de los árbitros siguiendo las restricciones del Cuadro 5.14, y generando la copia de seguridad con los los resultados. A continuación, se va a ir explicando paso a paso el proceso del flujo lógico.

En primer lugar, en la Figura 7.2 podemos ver como se desencadena el flujo, una vez ha sido llamado desde SegoFutsal e inicializa algunas de las variables a usar.

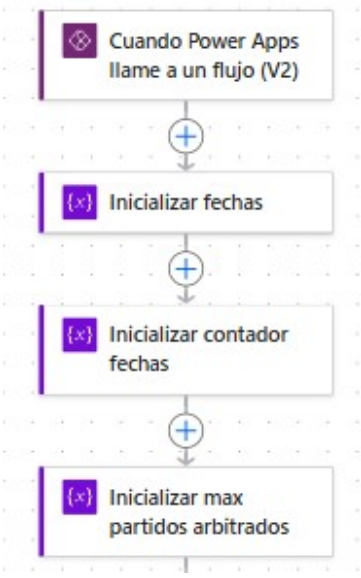


Figura 7.2: Inicio del flujo asignar árbitros

A continuación, lo que se hace en la Figura 7.3 es determinar la fecha a los partidos de una misma jornada. Como se obtienen de manera ascendente según su categoría, y liga, basta con tener un contador con el valor interno de la fecha para actualizar el registro.

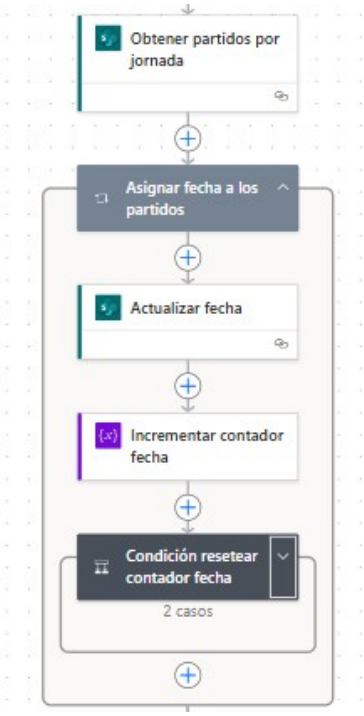


Figura 7.3: Fijar fecha a los partidos

Posteriormente, una de las grandes ventajas que tiene Microsoft Power Automate es

su facilidad para paralelizar tareas. Para que el algoritmo sea más óptimo, se ha utilizado esta opción y se paraleliza la asignación de árbitros para cada turno (S1, S2, y D1), de tal manera que se crean 3 ramas lógicas, como se puede ver en la Figura 7.4. Por contrapartida, cada rama debe tener sus propias variables, ya que una rama lógica no puede usar una variable definida en otra rama.

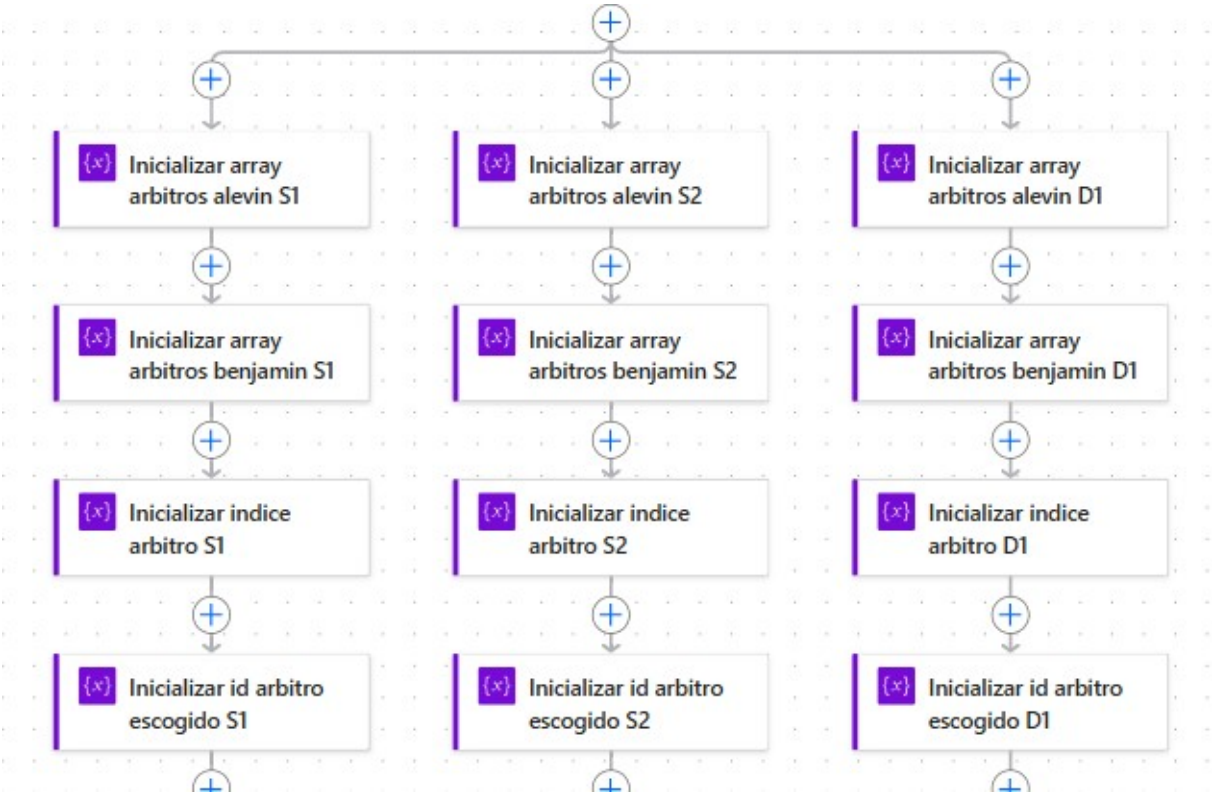


Figura 7.4: Paralelización de los tres horarios de los partidos

Por lo tanto, lo que sigue a continuación no es más que replicarlo para cada horario de partido. Una vez ya se han definido las variables a usar, la Figura 7.5 muestra cómo se genera un array con la información de los árbitros para cada categoría. Además, como consulta de filtro al obtener los árbitros, se aplica la restricción de que no puedan arbitrar más de 30 partidos durante la temporada. Por último, se comprueba si hay un número suficiente de árbitros tanto de alevín como de benjamín, ya que en caso negativo se debe finalizar el flujo y notificarselo al usuario en la aplicación.

A continuación, se realiza la propia designación del árbitro (escogido de manera al azar entre todos los posibles) al partido de categoría alevín. Posteriormente se actualiza su número de partidos arbitrados y se le elimina de la lista de árbitros que pueden pitar un partido. Todo ello se muestra en la Figura 7.6.

Acto seguido, y ya para terminar las designaciones arbitrales, se vuelcan los árbitros sobrantes con categoría de alevín a la lista de árbitros con categoría benjamín, y se realiza el mismo proceso que antes pero para los partidos de categoría benjamín, tal y como se muestra en la Figura 7.7.

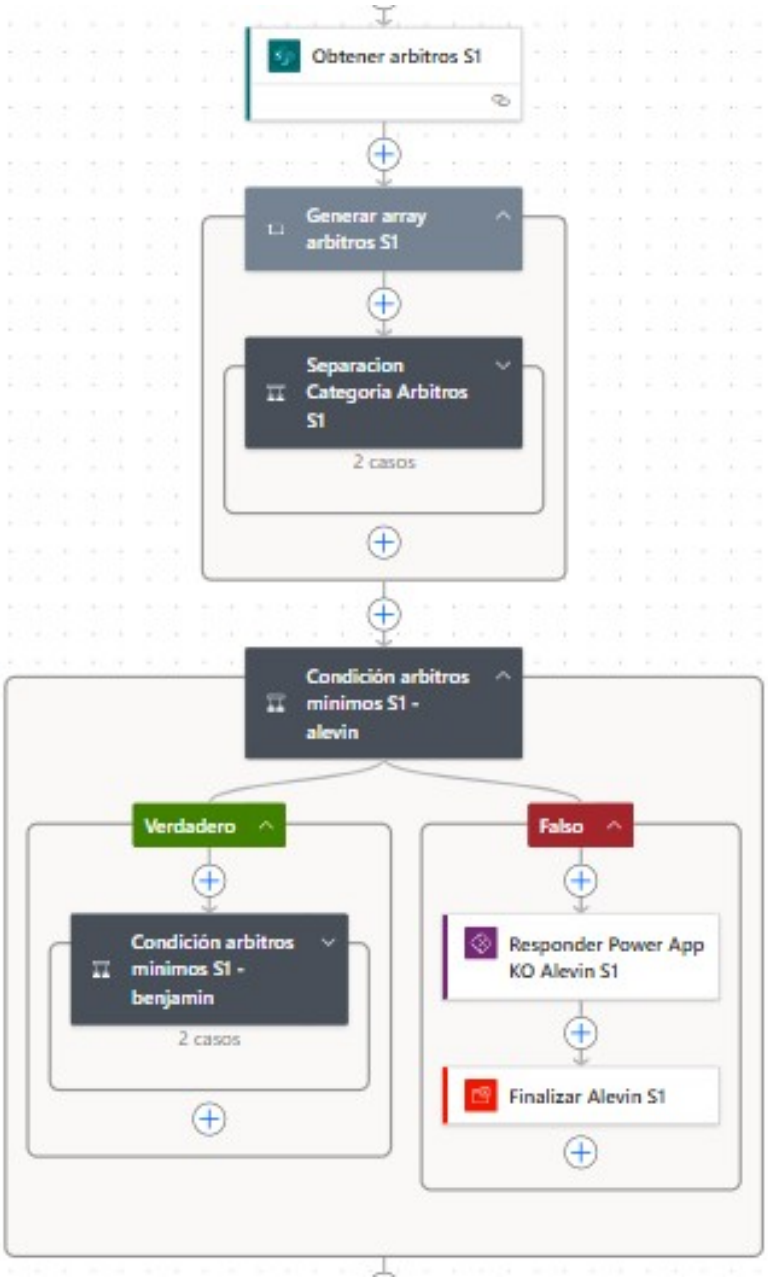
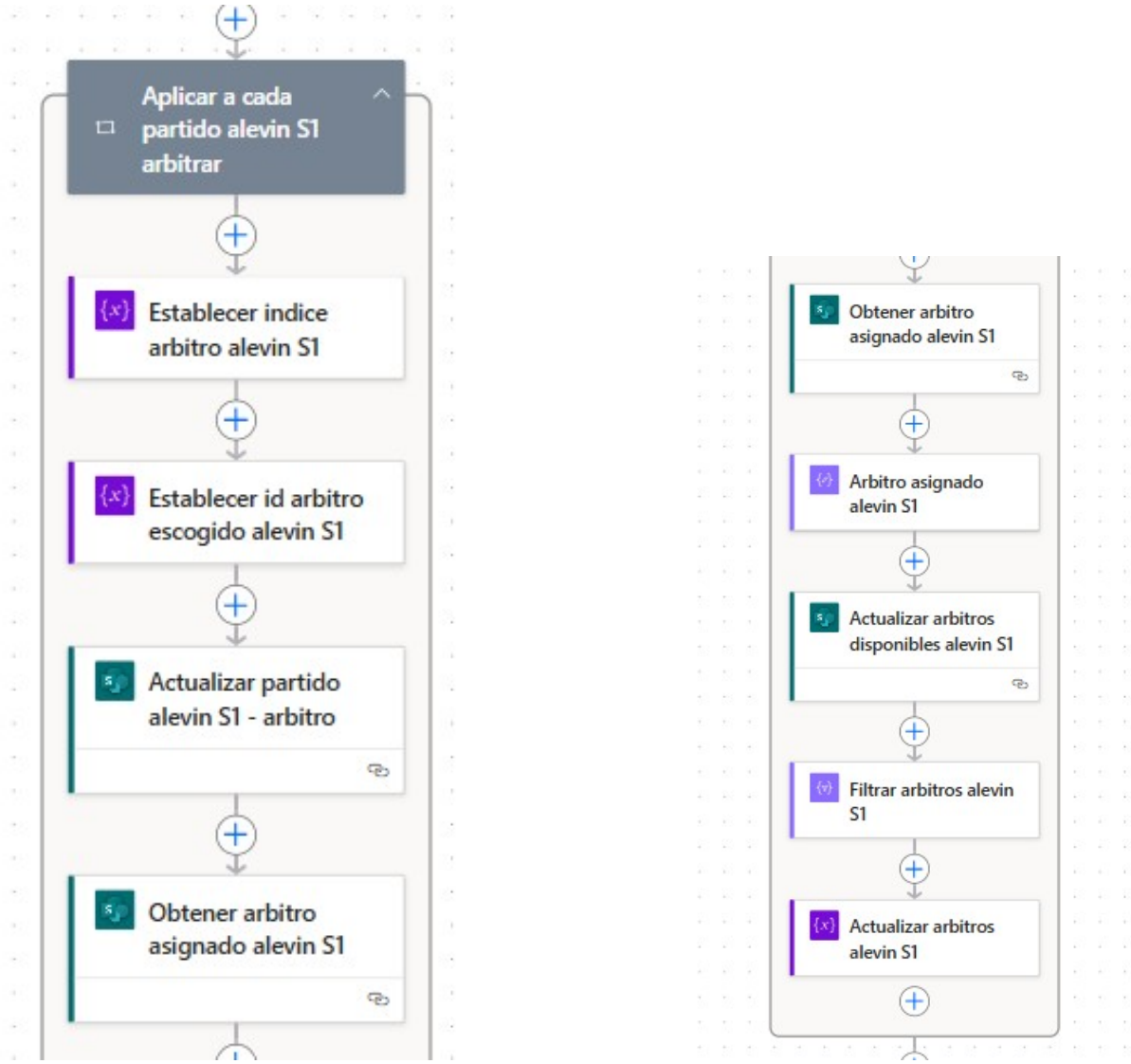


Figura 7.5: División de árbitros por categoría y comprobación de si hay suficientes

Por último, se termina la paralelización del proceso de asignación arbitral y se notifica a SegoFutsal que el proceso ha finalizado correctamente. Es importante, y como se ha marcado en el Cuadro 5.6, que hasta este punto se haya ejecutado el flujo en menos de minuto y medio, ya que de forma nativa Microsoft Power Apps tiene un timeout de minuto y medio para esperar una respuesta desde Microsoft Power Automate, y en caso de que tardara más, saltaría un error, a pesar de que todo haya ido bien. A continuación, se procede a hacer la copia de seguridad de la jornada. Para ello, se toma una plantilla



(a) Designar árbitros

(b) Designar árbitros (continuación)

Figura 7.6: Designación arbitral

guardada en una carpeta del proyecto y se crea un archivo nuevo rellenando dicha plantilla con los datos correspondientes. Todo esto se representa en la Figura 7.8. Una vez se termina este proceso, el flujo ha finalizado. La Figura 7.9 muestra un ejemplo de cómo quedaría una copia de seguridad.

7.2.3. Enviar Correos

El flujo de envío de correos tiene como objetivo notificar a cada árbitro la información correspondiente al partido que debe arbitrar. Para ello, el flujo lee automáticamente los datos de la jornada seleccionada y de los árbitros asignados, y genera un correo indivi-

Figura 7.7: Designación arbitral para los partidos de categoría benjamín

dualizado para cada uno de ellos. En dicho correo se incluye toda la información relevante del partido: equipos, lugar, fecha, hora y demás detalles necesarios. Este proceso permite una comunicación eficiente y automatizada, evitando el envío manual de mensajes. En la Figura 7.10 se expone cómo se ha hecho, mientras que el formato del correo enviado se muestra en la Figura 7.11.

7.3. Conclusión

El uso de plataformas de bajo código como Microsoft Power Apps y MIcrosoft Power Automate ha permitido desarrollar soluciones funcionales de forma rápida y con relativa sencillez, incluso para procesos que, en un entorno tradicional, requerirían mayor complejidad técnica. Esta metodología facilita enormemente la creación de interfaces y automatizaciones, acercando el desarrollo a perfiles no técnicos. Sin embargo, también implica una limitación importante: es necesario adaptarse a las reglas y estructuras que impone la plataforma. En muchas ocasiones, más que diseñar libremente, se trata de buscar cómo encajar las necesidades dentro de las posibilidades que ofrece la herramienta, lo cual puede condicionar ciertas decisiones de diseño o funcionalidad.

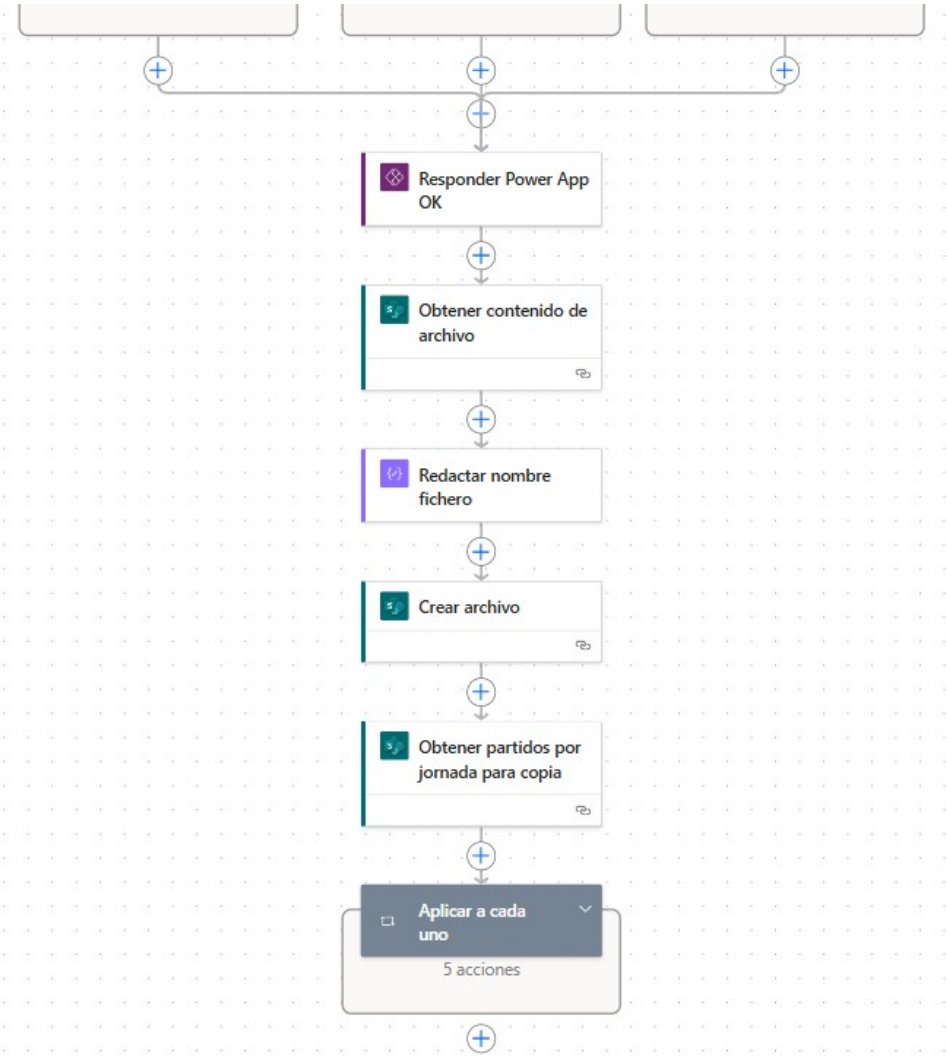
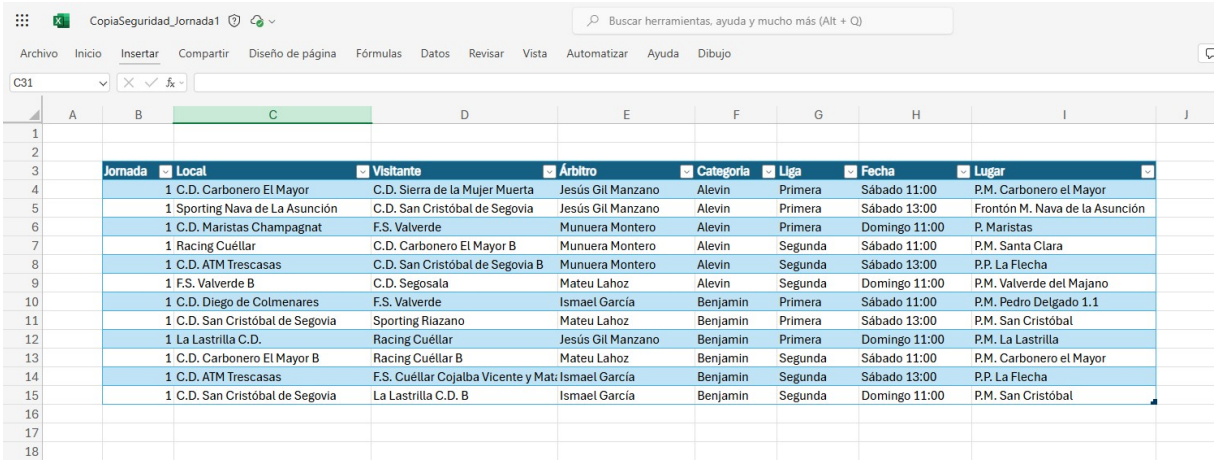


Figura 7.8: Copia de seguridad de la jornada



Jornada	Local	Visitante	Árbitro	Categoría	Liga	Fecha	Lugar
1	C.D. Carbonero El Mayor	C.D. Sierra de la Mujer Muerta	Jesús Gil Manzano	Alevín	Primera	Sábado 11:00	P.M. Carbonero el Mayor
1	Sporting Nava de La Asunción	C.D. San Cristóbal de Segovia	Jesús Gil Manzano	Alevín	Primera	Sábado 13:00	Frontón M. Nava de la Asunción
1	C.D. Maristas Champagnat	F.S. Valverde	Munuera Montero	Alevín	Primera	Domingo 11:00	P. Maristas
1	Racing Cuéllar	C.D. Carbonero El Mayor B	Munuera Montero	Alevín	Segunda	Sábado 11:00	P.M. Santa Clara
1	C.D. ATM Trescasas	C.D. San Cristóbal de Segovia B	Munuera Montero	Alevín	Segunda	Sábado 13:00	P.P. La Flecha
1	F.S. Valverde B	C.D. Segosala	Mateu Lahoz	Alevín	Segunda	Domingo 11:00	P.M. Valverde del Majano
1	C.D. Diego de Colmenares	F.S. Valverde	Ismael García	Benjamín	Primera	Sábado 11:00	P.M. Pedro Delgado 1.1
1	C.D. San Cristóbal de Segovia	Sporting Riazano	Mateu Lahoz	Benjamín	Primera	Sábado 13:00	P.M. San Cristóbal
1	La Lastrilla C.D.	Racing Cuéllar	Jesús Gil Manzano	Benjamín	Primera	Domingo 11:00	P.M. La Lastrilla
1	C.D. Carbonero El Mayor B	Racing Cuéllar B	Mateu Lahoz	Benjamín	Segunda	Sábado 11:00	P.M. Carbonero el Mayor
1	C.D. ATM Trescasas	F.S. Cuéllar Cojalba Vicente y Mat	Ismael García	Benjamín	Segunda	Sábado 13:00	P.P. La Flecha
1	C.D. San Cristóbal de Segovia	La Lastrilla C.D. B	Ismael García	Benjamín	Segunda	Domingo 11:00	P.M. San Cristóbal

Figura 7.9: Ejemplo de una copia de seguridad

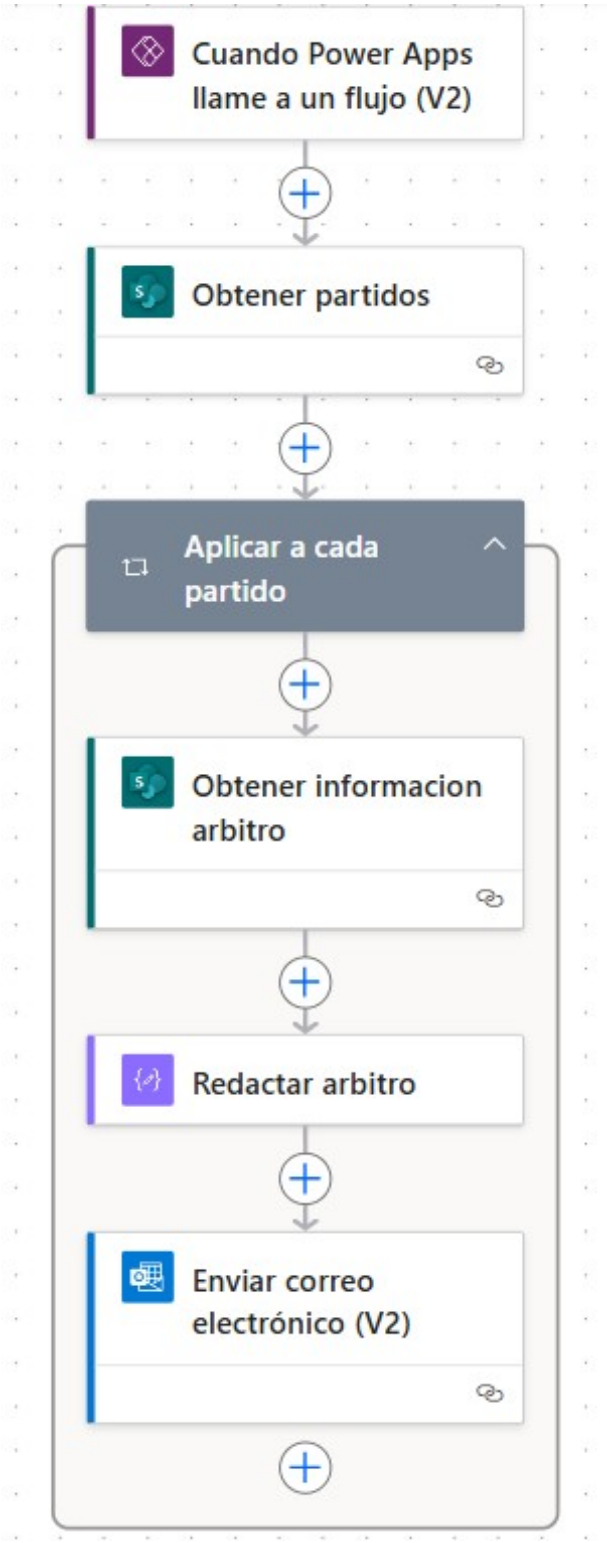


Figura 7.10: Flujo para enviar los correos a los árbitros

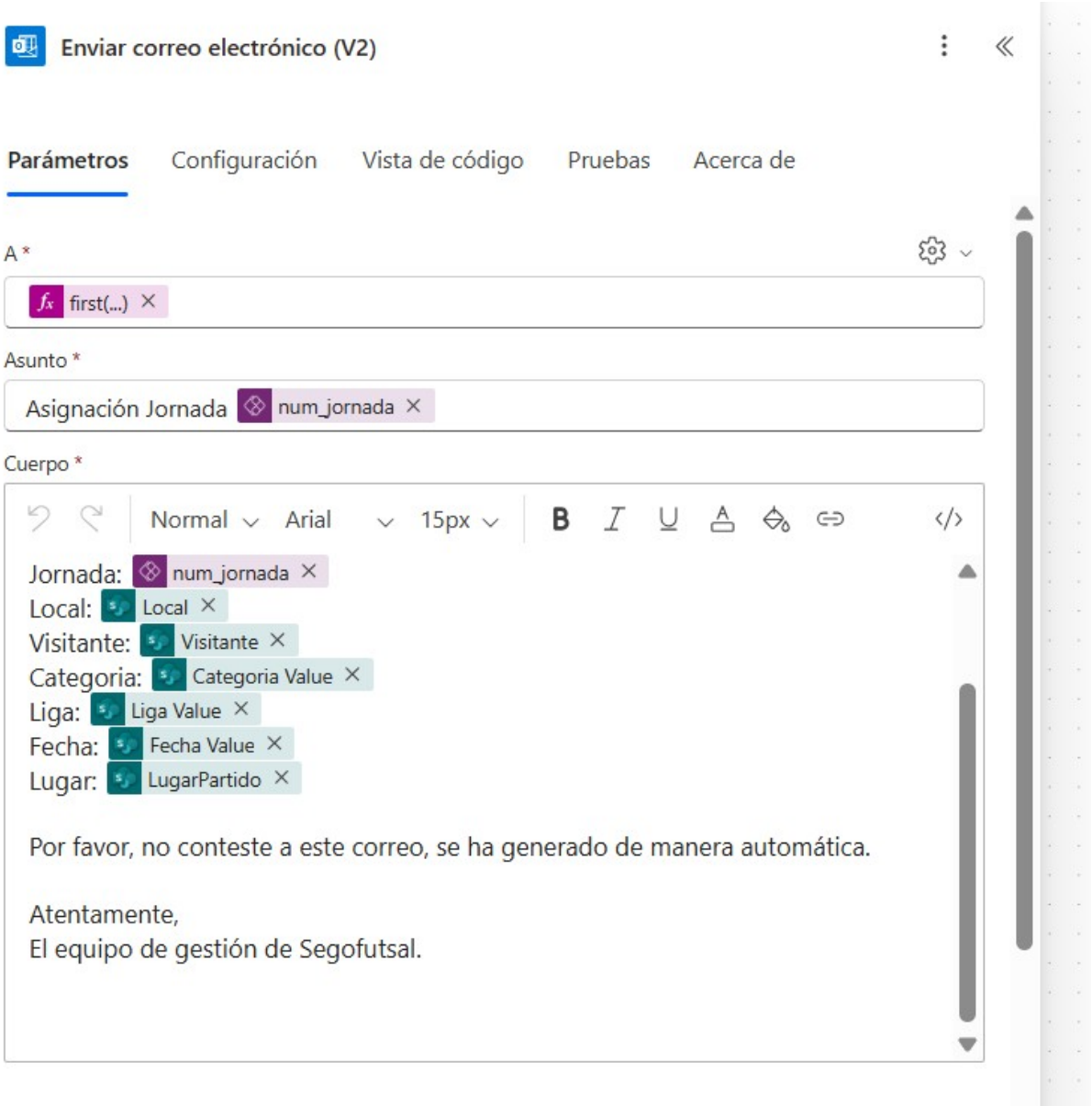


Figura 7.11: Detalle del correo enviado

Capítulo 8

Pruebas

En esta sección se describirá la realización de pruebas de caja negra, caja blanca y de seguridad, fundamentales para garantizar la calidad y robustez de SegoFutsal. Se realizarán pruebas de caja negra, de caja blanca, y de seguridad.

8.1. Pruebas de caja negra

Las pruebas de caja negra se enfocan en evaluar la funcionalidad del sistema desde el punto de vista del usuario, sin involucrarse en los detalles internos del código. Este tipo de prueba se realiza con base en los requisitos y especificaciones del proyecto, verificando que la entrada y la salida de los datos sean correctas. Es ideal para garantizar que el sistema cumpla con las expectativas funcionales y responda adecuadamente a diferentes escenarios.

ID de prueba	PCN-01
Objetivo de la prueba	Crear árbitro correctamente
Precondiciones	Tener acceso a la aplicación
Datos de entrada	▪ Nombre: Jose María Sanchez Martínez ▪ Email: sanchezmartinez@mail.com ▪ Categoría: Alevín
Respuesta esperada	Árbitro creado y listado en el sistema
Resultado final	Correcto

Cuadro 8.1: PCN-01 Crear árbitro

ID de prueba	PCN-02
Objetivo de la prueba	No permitir crear árbitro sin nombre
Precondiciones	Tener acceso a la aplicación
Datos de entrada	▪ Nombre: [vacío] ▪ Email: sanchezmartinez@mail.com ▪ Categoría: Alevín
Respuesta esperada	Mensaje de "obligatorio" en el campo "Nombre"
Resultado final	Correcto

Cuadro 8.2: PCN-02 Crear árbitro sin nombre

ID de prueba	PCN-03
Objetivo de la prueba	No permitir crear árbitro sin email
Precondiciones	Tener acceso a la aplicación
Datos de entrada	▪ Nombre: Jose María Sanchez Martínez ▪ Email: [vacío] ▪ Categoría: Alevín
Respuesta esperada	Mensaje de "obligatorio" en el campo "Email"
Resultado final	Correcto

Cuadro 8.3: PCN-03 Crear árbitro sin email

ID de prueba	PCN-04
Objetivo de la prueba	No permitir crear árbitro sin categoría.
Precondiciones	Tener acceso a la aplicación
Datos de entrada	▪ Nombre: Jose María Sanchez Martínez ▪ Email: sanchezmartinez@mail.com ▪ Categoría: [vacío]
Respuesta esperada	Mensaje de "obligatorio" en el campo "Categoría"
Resultado final	Correcto

Cuadro 8.4: PCN-04 Crear árbitro sin categoría

ID de prueba	PCN-05
Objetivo de la prueba	No permitir crear árbitro con el campo email mal formado
Precondiciones	Tener acceso a la aplicación
Datos de entrada	▪ Nombre: Jose María Sanchez Martínez ▪ Email: josemasanmar ▪ Categoría: Alevín
Respuesta esperada	Mensaje de "formato incorrecto" en el campo "Email"
Resultado final	Correcto

Cuadro 8.5: PCN-05 Crear árbitro con el email mal formado

ID de prueba	PCN-06
Objetivo de la prueba	Eliminar árbitro
Precondiciones	Árbitro creado
Datos de entrada	—
Respuesta esperada	Árbitro eliminado del sistema
Resultado final	Correcto

Cuadro 8.6: PCN-06 Eliminar árbitro

ID de prueba	PCN-07
Objetivo de la prueba	Editar árbitro correctamente
Precondiciones	Árbitro existente en el sistema
Datos de entrada	▪ Sánchez Martínez en vez de Jose María Sánchez Martínez
Respuesta esperada	Árbitro actualizado correctamente
Resultado final	Correcto

Cuadro 8.7: PCN-07 Editar árbitro correctamente

ID de prueba	PCN-08
Objetivo de la prueba	No permitir editar árbitro sin nombre.
Precondiciones	Árbitro existente
Datos de entrada	▪ Eliminar el nombre de un árbitro al editar
Respuesta esperada	Mensaje de “obligatorio” en el campo “Nombre”
Resultado final	Correcto

Cuadro 8.8: PCN-08 Editar árbitro sin nombre

ID de prueba	PCN-09
Objetivo de la prueba	No permitir editar árbitro sin email
Precondiciones	Árbitro existente
Datos de entrada	• Eliminar el email de un árbitro al editar
Respuesta esperada	Mensaje de "obligatorio" en el campo "Email"
Resultado final	Correcto

Cuadro 8.9: PCN-09 Editar árbitro sin email

ID de prueba	PCN-10
Objetivo de la prueba	• No permitir editar árbitro sin categoría
Precondiciones	Árbitro existente
Datos de entrada	Eliminar la categoría de un árbitro al editar
Respuesta esperada	Mensaje de "obligatorio" en el campo "Categoría"
Resultado final	Correcto

Cuadro 8.10: PCN-10 Editar árbitro sin categoría

ID de prueba	PCN-11
Objetivo de la prueba	No permitir editar árbitro con el email mal formado.
Precondiciones	Árbitro existente
Datos de entrada	• Cambiar el email sanchezmartinez@mail.com por sanchezmar
Respuesta esperada	Error: "Formato de email incorrecto".
Resultado final	Correcto

Cuadro 8.11: PCN-11 Editar árbitro con el formato del email incorrecto

ID de prueba	PCN-12
Objetivo de la prueba	Listar partidos sin aplicar filtros
Precondiciones	Existen partidos cargados
Datos de entrada	• Filtros: ninguno
Respuesta esperada	Todos los partidos de la temporada mostrados
Resultado final	Correcto

Cuadro 8.12: PCN-12 Ver partidos sin filtros

ID de prueba	PCN-13
Objetivo de la prueba	Listar partidos filtrando por número de jornada
Precondiciones	Existen partidos cargados
Datos de entrada	• Jornada: 5
Respuesta esperada	Partidos de la jornada 5 mostrados
Resultado final	Correcto

Cuadro 8.13: PCN-13 Ver partidos con filtro de jornada

ID de prueba	PCN-14
Objetivo de la prueba	Listar partidos filtrando por categoría de equipos
Precondiciones	Existen partidos cargados
Datos de entrada	• Categoría: Alevín
Respuesta esperada	Solo partidos de categoría Alevín mostrados
Resultado final	Correcto

Cuadro 8.14: PCN-14 Ver partidos con filtro de categoría

ID de prueba	PCN-15
Objetivo de la prueba	Listar partidos filtrando por liga
Precondiciones	Existen partidos cargados
Datos de entrada	▪ Liga: Primera
Respuesta esperada	Solo partidos de la liga Primera mostrados
Resultado final	Correcto

Cuadro 8.15: PCN-15 Ver partidos con filtro de liga

ID de prueba	PCN-16
Objetivo de la prueba	Listar partidos aplicando todos los filtros juntos
Precondiciones	Existen partidos cargados
Datos de entrada	▪ Jornada: 5 ▪ Categoría: Benjamín ▪ Liga: Segunda
Respuesta esperada	Partidos que cumplan jornada 5, categoría Benjamín y liga Segunda.
Resultado final	Correcto

Cuadro 8.16: PCN-16 Ver partidos con múltiples filtros

ID de prueba	PCN-17
Objetivo de la prueba	Buscar árbitro por nombre y obtener resultados.
Precondiciones	Existen árbitros cuyo nombre coincida
Datos de entrada	▪ Nombre buscado: “Mateu”
Respuesta esperada	Listado de árbitros con "Mateu.en el nombre
Resultado final	Correcto

Cuadro 8.17: PCN-17 Búsqueda de árbitros por nombre correcta

ID de prueba	PCN-18
Objetivo de la prueba	Buscar árbitro por nombre sin resultados
Precondiciones	No existen árbitros con ese nombre
Datos de entrada	▪ Nombre buscado: "Xyz"
Respuesta esperada	Mensaje: “No existen árbitros con ese nombre”
Resultado final	Correcto

Cuadro 8.18: PCN-18 Búsqueda de árbitros por nombre incorrecta

ID de prueba	PCN-19
Objetivo de la prueba	Asignar árbitros cuando hay disponibilidad suficiente
Precondiciones	Existen suficientes árbitros disponibles para la jornada
Datos de entrada	▪ Jornada: 4 ▪ Árbitros seleccionados: todos los disponibles en turnos S1, S2, D1
Respuesta esperada	Árbitros asignados, correos enviados y copia de seguridad creada.
Resultado final	Correcta

Cuadro 8.19: PCN-19 Asignación de árbitros cuando hay disponibilidad suficiente

ID de prueba	PCN-20
Objetivo de la prueba	Error al asignar en turno S1 con árbitros insuficientes
Precondiciones	Árbitros disponibles en S1 insuficientes
Datos de entrada	▪ Jornada: 5 ▪ Árbitros disponibles en S1 para alevín insuficientes
Respuesta esperada	Mensaje de error "No hay arbitros con categoría ALEVIN suficientes para el SÁBADO 11:00"
Resultado final	Correcto

Cuadro 8.20: PCN-20 Asignación de árbitros con falta de ellos para S1

ID de prueba	PCN-21
Objetivo de la prueba	Error al asignar en turno S2 con árbitros insuficientes
Precondiciones	Árbitros disponibles en S2 insuficientes
Datos de entrada	▪ Jornada: 5 ▪ Árbitros disponibles en S2 para alevín insuficientes
Respuesta esperada	Mensaje de error "No hay arbitros con categoría ALEVIN suficientes para el SÁBADO 13:00"
Resultado final	Correcto

Cuadro 8.21: PCN-21 Asignación de árbitros con falta de ellos para S2

ID de prueba	PCN-22
Objetivo de la prueba	Error al asignar en turno D1 con árbitros insuficientes
Precondiciones	Árbitros disponibles en D1 insuficientes
Datos de entrada	▪ Jornada: 5 ▪ Árbitros disponibles en D1 para alevín insuficientes
Respuesta esperada	Mensaje de error "No hay arbitros con categoría ALEVIN suficientes para el DOMINGO 11:00"
Resultado final	Correcto

Cuadro 8.22: PCN-22 Asignación de árbitros con falta de ellos para D1

8.2. Pruebas de caja blanca

En las pruebas de caja blanca se analiza a fondo la estructura y lógica interna del código fuente, asegurando la correcta ejecución de cada componente individual y su interacción con el sistema en general. Estas pruebas permiten identificar posibles errores en el flujo de control, variables no inicializadas, bucles infinitos u otros defectos de implementación. Además, ayudan a garantizar que el sistema funcione de manera óptima y eficiente, cubriendo caminos lógicos específicos del programa.

ID de prueba	PCB-01
Objetivo de la prueba	Verificar el correcto flujo interno en las funciones de creación, edición y búsqueda de árbitros, incluyendo la validación de campos
Precondiciones	—
Datos de entrada	▪ Árbitro con datos válidos e inválidos ▪ Consultas de búsqueda con nombre parcial
Respuesta esperada	Todas las condiciones internas se evalúan correctamente
Resultado final	Correcto

Cuadro 8.23: PCB-01 Lógica interna de la creación, edición y búsqueda de árbitros

ID de prueba	PCB-02
Objetivo de la prueba	Evaluar el recorrido de condiciones lógicas internas al aplicar filtros por categoría, liga y jornada en la visualización de partidos
Precondiciones	Datos de partidos cargados en el sistema
Datos de entrada	▪ Filtro por categoría ▪ Filtro por liga ▪ Filtro por jornada
Respuesta esperada	Todas las condiciones de filtrado se evalúan correctamente
Resultado final	Correcto

Cuadro 8.24: PCB-02 Lógica interna del filtrado de partidos

ID de prueba	PCB-03
Objetivo de la prueba	Verificar que la lógica de asignación recorre correctamente las estructuras de árbitros y partidos, respetando las restricciones
Precondiciones	Partidos y árbitros cargados apropiadamente
Datos de entrada	▪ Árbitros con diferentes disponibilidades ▪ Partidos de una jornada seleccionada
Respuesta esperada	La función de asignación selecciona correctamente árbitros válidos y muestra el error en caso de fallo
Resultado final	Correcto

Cuadro 8.25: PCB-03 Lógica interna de asignación de árbitros

ID de prueba	PCB-04
Objetivo de la prueba	Evaluar que el fragmento de flujo para la creación de la copia de seguridad gestiona correctamente la lectura y escritura de datos.
Precondiciones	Jornada asignada de manera correcta en los instantes previos
Datos de entrada	—
Respuesta esperada	Se crea la copia de seguridad correctamente
Resultado final	Correcto

Cuadro 8.26: PCB-04 Lógica interna de creación de la copia de seguridad

ID de prueba	PCB-05
Objetivo de la prueba	Verificar el correcto recorrido del flujo de envío de correos a los árbitros asignados
Precondiciones	Jornada asignada de manera correcta en los instantes previos
Datos de entrada	—
Respuesta esperada	El flujo recorre correctamente la lista de árbitros y partidos, compone los mensajes, y realiza el envío del correo
Resultado final	Correcto

Cuadro 8.27: PCB-05 Lógica interna del envío de correos a árbitros

8.3. Pruebas de seguridad

Las pruebas de seguridad son esenciales para garantizar que el sistema sea resistente frente a posibles amenazas y ataques malintencionados. Estas pruebas se enfocan en identificar vulnerabilidades que puedan comprometer la confidencialidad, integridad o disponibilidad de los datos y servicios del sistema.

ID de prueba	PS-01
Objetivo de la prueba	Comprobar que antes de iniciar la aplicación, necesitas haber iniciado sesión en Microsoft
Precondiciones	Tener acceso al enlace a la aplicación
Datos de entrada	—
Respuesta esperada	Redirección a la página de Microsoft de inicio de sesión
Resultado final	Correcto

Cuadro 8.28: PS-01 Acceso a SegoFutsal

8.4. Pruebas de rendimiento

Las pruebas de rendimiento son fundamentales para asegurar que el sistema pueda responder de manera eficiente bajo diferentes niveles de carga. Estas pruebas evalúan aspectos como la velocidad, la capacidad de respuesta y la estabilidad del sistema, identificando posibles cuellos de botella que puedan afectar la experiencia del usuario o el funcionamiento general en condiciones exigentes.

ID de prueba	PR-01
Objetivo de la prueba	Comprobar que el flujo lógico de asignaciones tarda menos de minuto y medio (sin contar el proceso de realización de copia de seguridad)
Precondiciones	Tener suficientes árbitros disponibles en cada categoría
Datos de entrada	Turnos S1, S2, y D1 disponibles para todos los árbitros
Respuesta esperada	Designaciones realizadas adecuadamente
Resultado final	Correcto

Cuadro 8.29: PR-01 Tiempo máximo en ejecutar el flujo de asignaciones arbitrales

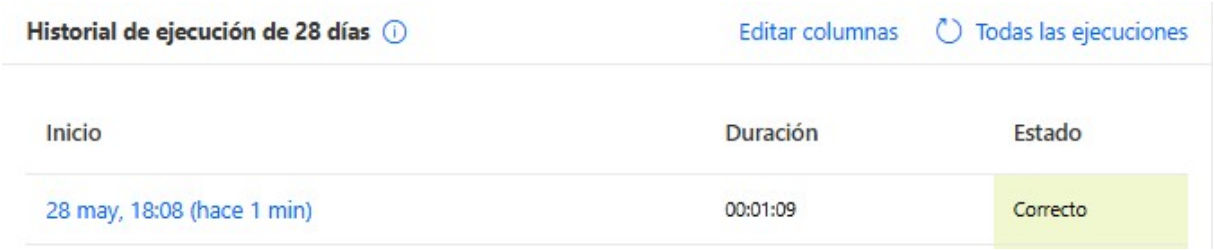


Figura 8.1: Tiempo de ejecución del flujo de asignaciones

Parte III

Manuales de la Aplicación

Capítulo 9

Manual de Instalación

Dado que el desarrollo de la aplicación se ha realizado utilizando herramientas de Microsoft 365, el proceso de instalación como tal es prácticamente inexistente. La puesta en marcha del sistema se basa simplemente en compartir la aplicación con los usuarios finales, otorgándoles los permisos necesarios para su uso o edición de la propia aplicación, según se desee. Del mismo modo, las listas de Microsoft Lists que funcionan como base de datos deben compartirse con los usuarios correspondientes. Para ello eso sí, es imprescindible estar conectado a Internet. Esto permite una implementación rápida, sin necesidad de instalaciones locales ni configuraciones complejas. En las Figuras 9.1 y 9.2 se muestran unas capturas de pantalla de como habría que proceder para compartir la aplicación

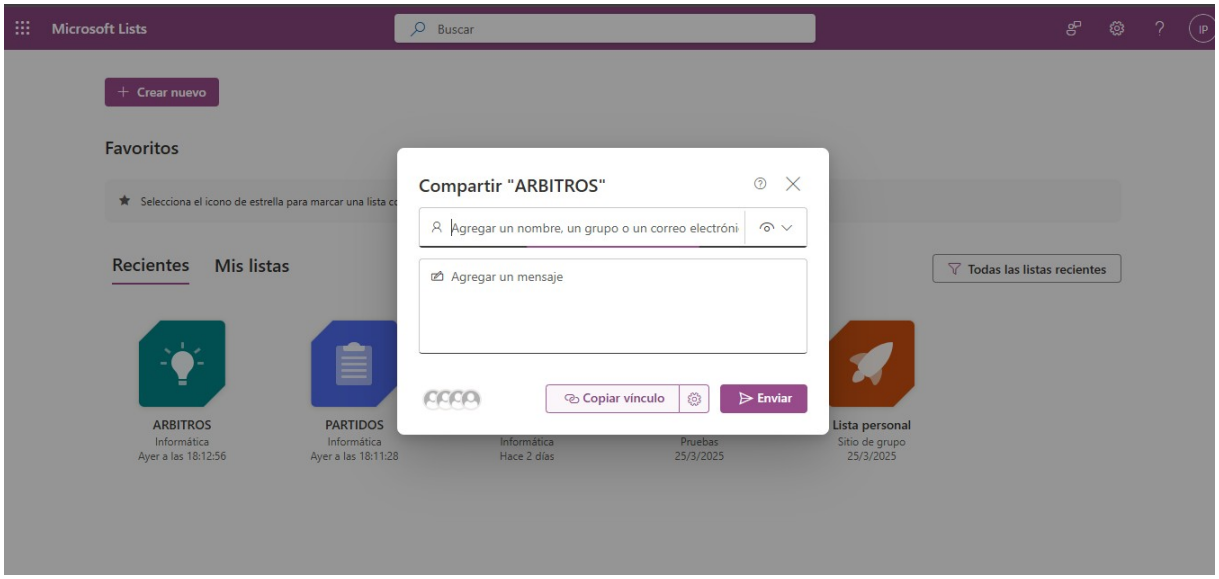


Figura 9.1: Compartir lista con otros usuarios

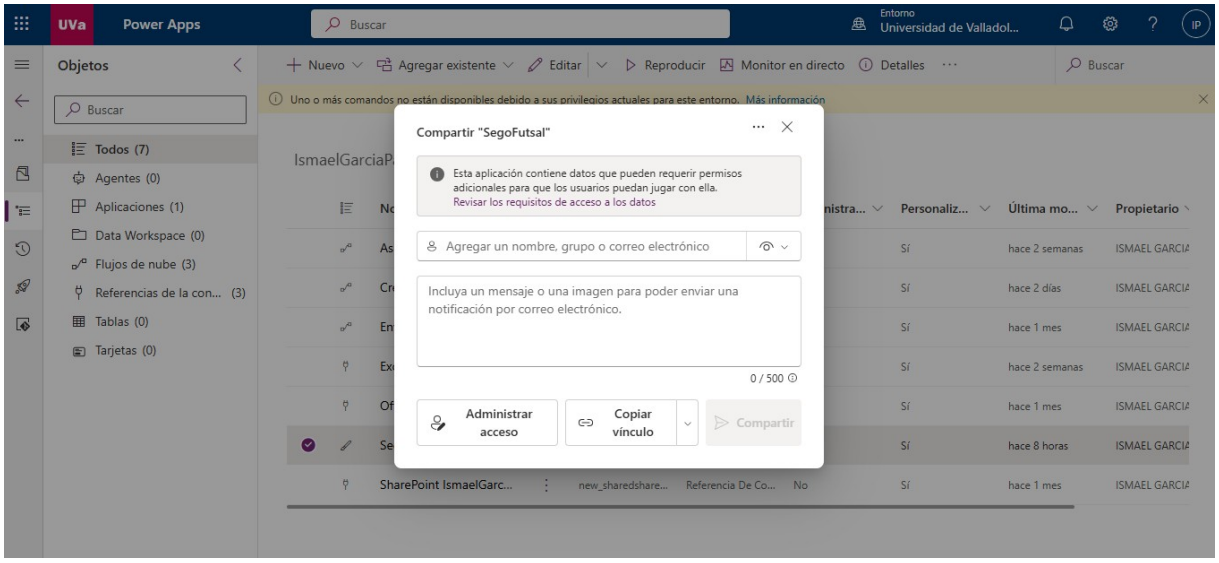


Figura 9.2: Compartir SegoFutsal con otros usuarios

Capítulo 10

Manual de Usuario

Este último capítulo tiene como objetivo guiar al usuario en el uso adecuado de SegoFutsal. Se detallan las principales funciones, junto con la navegación por la interfaz y recomendaciones para un uso óptimo. A lo largo del capítulo se incluyen capturas de pantalla, pasos secuenciales y descripciones claras para facilitar la comprensión del funcionamiento general de la aplicación.

10.1. Inicio

En la Figura 10.1 se muestra la vista de inicio de SegoFutsal. En esta pantalla se presenta un mensaje de bienvenida acompañado de tres botones, cada uno asociado a una funcionalidad específica de la aplicación. Para acceder a cualquiera de ellas, simplemente se debe pulsar el botón correspondiente.

10.2. Gestión de partidos

Si se pincha en el botón "Gestión de Partidos" de la Figura 10.1, iremos a toda la parte que gestiona los partidos ya cargados de las dos categorías y ligas. La Figura 10.2 muestra la vista de esta sección. Aquí, se puede observar los tres filtros disponibles que hay (jornada, liga, y categoría) para los partidos. Si se desea eliminar los filtros, basta con pulsar el botón que aparece llamado "Limpiar Filtros".

Para poder obtener más información de los partidos, se debe presionar la flecha que aparece a la derecha de cada registro de cada partido, ya que se navegará a la vista de detalle del partido (Figura 10.3).

Finalmente, si el usuario desea volver a la pantalla anterior, es suficiente con tocar la flecha que aparece en la esquina superior izquierda. Esta funcionalidad de volver hacia atrás está presente del mismo modo en toda la aplicación, por lo que no se indicará más para futuras ocasiones.

Figura 10.1: Vista home de SegoFutsal

10.3. Gestión de árbitros

Si se pincha en el botón "Gestión de Árbitros" de la Figura 10.1, iremos a toda la parte que gestiona los árbitros. La Figura 10.4 muestra la vista de esta sección. En ella, se puede observar una lista de los árbitros registrados en SegoFutsal. Cabe destacar que en caso de que haya más de cinco árbitros registrados, no todos ellos son visibles en pantalla. Sin embargo, el usuario puede desplazarse hacia abajo deslizando con el dedo para visualizar el resto de árbitros.

En la esquina superior derecha, se encuentran dos botones. El primero de ellos (la ruleta), sirve para recargar la vista y actualizar la información de los árbitros en caso de que no se actualice. El segundo de ellos, definido por "+", dirige a la página de añadir árbitros. La Figura 10.5 enseña cómo es el formulario para agregar árbitros a la aplicación.

Finalmente, si se desea ver más detalles de la información de los árbitros, no hay más que pulsar la flecha que aparece a la derecha de cada árbitros, de la misma forma que en la gestión de partidos. La Figura 10.6 presenta esta funcionalidad. Además, en esta pantalla, aparecen los botones de edición (lápiz) y eliminación (papelera) del árbitro. La vista de la edición del árbitro es similar a la de la Figura 10.5, pero con los campos rellenados con la información del árbitro seleccionado.

Figura 10.2: Vista de la gestión de partidos

10.4. Asignar árbitros a partidos

Si se pincha en el botón "Asignar Árbitros" de la Figura 10.1, iremos a toda la parte que gestiona las designaciones arbitrales. La Figura 10.7 muestra la vista de esta sección. En primer lugar, se encuentra el filtro de jornada, necesario para indicar la jornada a realizar las designaciones arbitrales. Posteriormente se encuentra una lista con los distintos árbitros registrados en la aplicación, teniendo al lado de su respectiva foto tres interruptores cuya finalidad es marcar si están disponibles para los horarios que indican.

Una vez ya está todo listo, el usuario puede presionar el botón que aparece en la parte inferior de la página "Asignar". Este botón ejecuta el flujo explicado en la Sección 7.2.2, mostrándose el pop-up que aparece en la Figura 10.8. Una vez el flujo haya finalizado de manera exitosa, se actualiza este pop-up, como indica la Figura 10.9. En el caso de que hubiera errores durante la ejecución del flujo, es en este mismo pop-up donde se mostrarían dicho errores.

Para acabar, el botón "Volver a inicio" redirige a la pantalla Home de SegoFutsal.

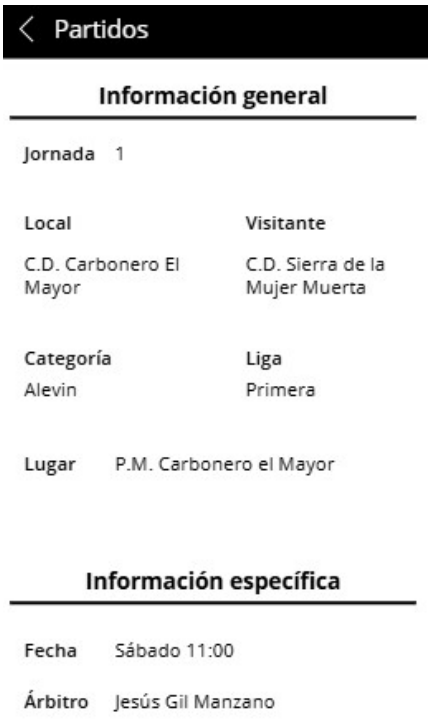


Figura 10.3: Vista de detalle de los partidos

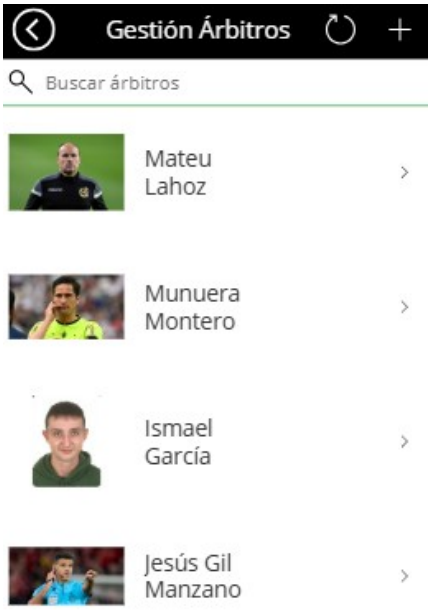


Figura 10.4: Vista de la gestión de los árbitros

Figura 10.5: Vista del formulario para agregar árbitros

Figura 10.6: Vista de detalle de los árbitros

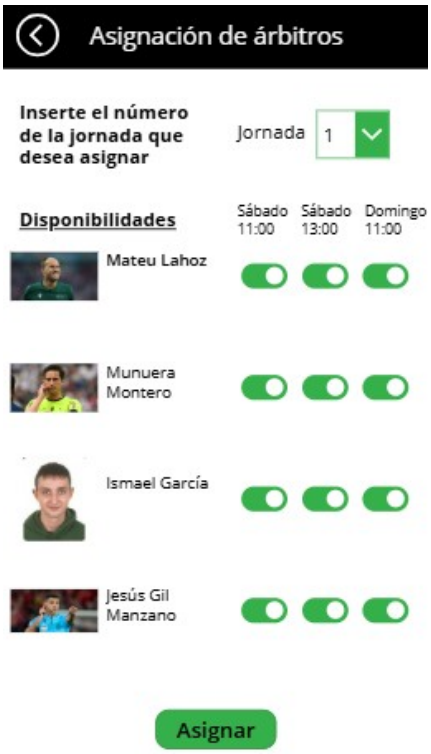


Figura 10.7: Vista de las designaciones arbitrales

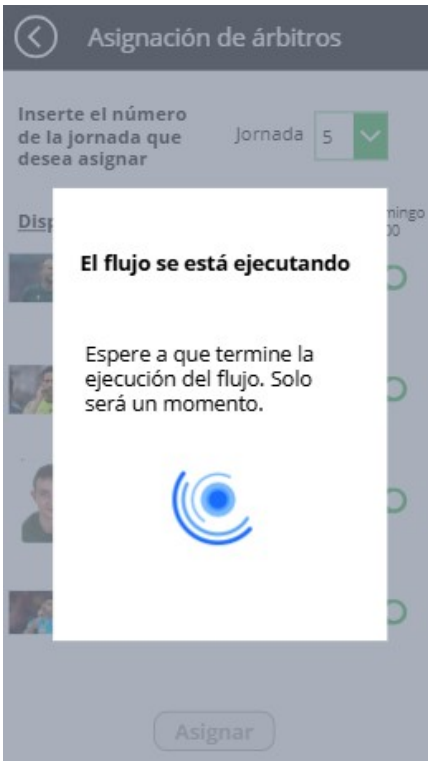


Figura 10.8: Pop-up durante la ejecución del flujo

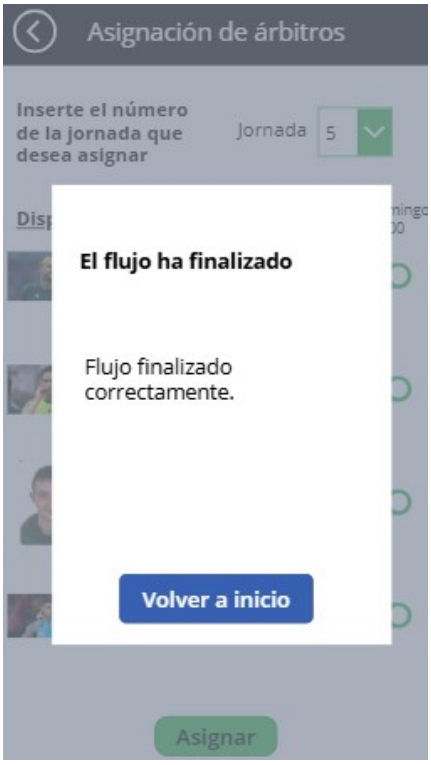


Figura 10.9: Pop-up una vez ha finalizado el flujo

Bibliografía

[1] Alarcón Lorca, F. E. (2009). *Asignación de árbitros para un campeonato de fútbol mediante el uso de programación matemática*, Universidad de Chile.

[2] CepaCode. (2 de junio de 2025). *Power Apps - Curso básico*. Recuperado de https://www.youtube.com/watch?v=hM4qaN_GGJA

[3] González Cabrera, F. J. (2024). *Modelado Software de Sistemas de Información. Bloque II: Etapas del Desarrollo de Requisito*, Universidad de Valladolid.

[4] Martín de Andrés, D. & Serrano Nieto, N. (2024). *GPTI: Gestión de Proyectos Basados en las Tecnologías de la Información. Bloque III. Estimación para proyectos software*, Universidad de Valladolid.

[5] Microsoft. (2 de junio de 2025). *Comparar todos los planes de Microsoft 365*. Recuperado de https://www.microsoft.com/es-es/microsoft-365/business/compare-all-microsoft-365-business-products-monthly

[6] Microsoft. (1 de junio de 2025). *Microsoft Copilot*. Recuperado de https://www.microsoft.com/es-es/microsoft-copilot

[7] Microsoft Lists. (1 de junio de 2025). *Información general de Microsoft Lists*. Recuperado de https://www.microsoft.com/es-es/microsoft-365/microsoft-lists

[8] Microsoft Power Apps. (1 de junio de 2025). *Power Apps*. Recuperado de https://powerapps.microsoft.com/es-es/

[9] Microsoft Power Apps. (1 de junio de 2025). *Información general de creación de aplicaciones en Power Apps*. Recuperado de https://docs.microsoft.com/es-es/power-apps/maker/

[10] Microsoft Power Apps. (1 de junio de 2025). *¿Qué es Power Apps?* Recuperado de https://docs.microsoft.com/es-es/power-apps/powerapps-overview

[11] Microsoft Power Automate. (3 de junio de 2025). *Power Automate*. Recuperado de https://www.microsoft.com/es-es/power-platform/products/power-automate/

[12] Microsoft Power Automate. (1 de junio de 2025). *Documentación de Microsoft Power Automate*. Recuperado de https://docs.microsoft.com/es-es/power-automate/

[13] Microsoft Power Automate. (1 de junio de 2025). *Introducción a Power Automate*. Recuperado de https://docs.microsoft.com/es-es/power-automate/getting-started

[14] Microsoft Power Platform. (1 de junio de 2025). *Información general de Microsoft Power Platform*. Recuperado de https://www.microsoft.com/es-es/power-platform

[15] Microsoft SharePoint. (1 de junio de 2025). *SharePoint*. Recuperado de https://www.microsoft.com/es-es/microsoft-365/sharepoint/collaboration

[16] Microsoft SharePoint. (1 de junio de 2025). *Introducción a SharePoint*. Recuperado de https://docs.microsoft.com/es-es/sharepoint/introduction

[17] OpenAI. (3 de junio de 2025). *ChatGPT*. Recuperado de https://chat.openai.com

[18] Real Federación de Castilla y León de Fútbol. (1 de junio de 2025). *Directorio de Equipos*. Recuperado de https://www.rfcylf.es/pnfg/NPcd/NFG_LstDirectorioEquipos?cod_primaria=1000117

[19] Real Federación de Castilla y León de Fútbol. (1 de junio de 2025). *Página principal*. Recuperado de https://www.rfcylf.es

[20] Real Federación Española de Fútbol. (3 de junio de 2025). *Página principal*. Recuperado de https://rfef.es/es

[21] RefAssist. (1 de junio de 2025). *Official Management Software*. Recuperado de https://refassist.com/en