SISTEMA DE DIGITALIZACIÓN Y RECONOCIMIENTO AUTOMÁTICO DE ACTAS DEPORTIVAS

DIGITIZATION AND AUTOMATIC RECOGNITION SYSTEM FOR SPORTS RECORDS

TRABAJO FIN DE GRADO
CURSO 2022-2023

AUTOR
GABRIEL CASADO VALCÁRCEL

DIRECTOR
JOSE IGNACIO HIDALGO PÉREZ

GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

SISTEMA DE DIGITALIZACIÓN Y RECONOCIMIENTO AUTOMÁTICO DE ACTAS DEPORTIVAS

DIGITIZATION AND AUTOMATIC RECOGNITION SYSTEM FOR SPORTS RECORDS

TRABAJO DE FIN DE GRADO EN INGENIERÍA INFORMÁTICA

AUTOR
GABRIEL CASADO VALCÁRCEL

DIRECTOR
JOSE IGNACIO HIDALGO PÉREZ

CONVOCATORIA: SEPTIEMBRE 2026

GRADO EN INGENIERÍA INFORMÁTICA
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

4 SEPTIEMBRE 2026

DEDICATORIA

Al equipo de Rugby Físicas, por haberme regalado una familia y una infinidad de experiencias inolvidables.

AGRADECIMIENTOS

A Jose Ignacio Hidalgo Pérez, por toda la paciencia y la confianza que ha depositado y sigue depositando en mí, sin duda, este trabajo no se podría haber finalizado sin él. A toda mi familia que siempre me ha estado acompañando y apoyando, amigos de la carrera y de la universidad, a los que estuvieron, los que están y los que seguirán estando.

Muchas gracias.

RESUMEN

Sistema de digitalización y reconocimiento automático de actas deportivas

Este Trabajo de Fin de Grado presenta el diseño, la implementación y el despliegue de un sistema para digitalizar, verificar y publicar el histórico de las clasificaciones de las competiciones deportivas internas de la Universidad Complutense de Madrid. Los datos de partida son los cuadernillos con los que la Oficina de Deportes de la UCM ha documentado, año a año desde el curso 1990-91, los resultados del Trofeo Rector y del Trofeo Alfonso XIII: documentos en papel, posteriormente escaneados, que hasta ahora solo eran consultables de forma física y que no estaban recogidos en ningún soporte digital accesible para la comunidad universitaria.
El trabajo se ha dividido en dos grandes bloques. El primero aborda la extracción y verificación de los datos: se exploraron varias vías de digitalización totalmente automática, desde el análisis de texto plano hasta el uso de servicios de reconocimiento de documentos e inteligencia artificial, para terminar optando por un proceso de transcripción y verificación asistido por un asistente de programación basado en inteligencia artificial pero dirigido y supervisado en todo momento por el propio estudiante, contrastando sistemáticamente cada año contra su documento original, dada la heterogeneidad del formato de los cuadernillos a lo largo de más de tres décadas. El resultado es una base de datos con más de 7.400 registros correspondientes a 26 temporadas, 18 deportes distintos y más de un centenar de equipos. El segundo bloque cubre el diseño, la implementación y el despliegue de una aplicación web de solo lectura que permite consultar ese histórico por año, deporte, género y categoría, ver la trayectoria de un equipo o de un deporte a lo largo de los años, y explorar un conjunto de estadísticas predefinidas, todo ello construido con SQLPage y MySQL/MariaDB sobre contenedores Docker, sin backend intermedio propio.
El objetivo final es que este histórico, hasta ahora guardado en un cajón, pase a estar disponible en línea para cualquier miembro de la comunidad complutense.

Palabras clave

Digitalización de documentos, bases de datos históricas, SQLPage, Docker, verificación de datos, deporte universitario, Complutense.

ABSTRACT

Digitization and automatic recognition system for sports records

This Bachelor's Thesis presents the design, implementation and deployment of a system to digitize, verify and publish the historical results of Universidad Complutense de Madrid's internal sports competitions. The source material consists of booklets in which the University's Sports Office has documented, year after year since the 1990-91 season, the results of the Rector Trophy and the Alfonso XIII Trophy: paper documents, later scanned, that until now could only be consulted physically and were not available in any accessible digital format for the university community.

The work is divided into two main blocks. The first one addresses data extraction and verification: several fully automatic digitization approaches were explored, from plain-text parsing to document-recognition and artificial-intelligence services, before settling on a transcription and verification process assisted by an AI-based programming assistant but directed and supervised at every step by the student himself, systematically checking every year against its original document, given how inconsistent the booklets' format is across more than three decades. The result is a database with more than 7,400 records spanning 26 seasons, 18 different sports and over a hundred teams. The second block covers the design, implementation and deployment of a read-only web application that allows browsing this history by year, sport, gender and category, following a team's or a sport's history through the years, and exploring a set of predefined statistics, all of it built with SQLPage and MySQL/MariaDB on top of Docker containers, without a custom intermediate backend.

The ultimate goal is for this history, until now kept in a drawer, to become available online to any member of the Complutense community.

Keywords

Document digitization, historical databases, SQLPage, Docker, data verification, university sports, Complutense.

ÍNDICE DE CONTENIDOS

Capítulo 1 - Introducción 1

1.1 Motivación 1

1.2 Objetivos 1

1.3 Plan de trabajo 2

1.4 Organización de la memoria 3

Capítulo 2 - Estado de la cuestión 5

2.1 Extracción automática de información de documentos escaneados 5

2.2 Arquitecturas para aplicaciones web de consulta de datos 5

Capítulo 3 - Análisis y tratamiento de los datos de partida 7

3.1 Los cuadernillos de la Oficina de Deportes 7

3.2 Primeras aproximaciones a la digitalización automática 7

3.2.1 Extracción por reglas sobre texto plano 8

3.2.2 Reconocimiento de documentos con Azure AI Document Intelligence 8

3.2.3 Extracción asistida por modelos de lenguaje 9

3.2.4 Por qué se descartó la extracción totalmente automática 9

3.3 Diseño del modelo de datos 9

3.4 Proceso de verificación de los datos 10

3.5 Normalización de nomenclatura 11

Capítulo 4 - Diseño e implementación de la aplicación web 14

4.1 Requisitos y restricciones de diseño 15

4.2 Arquitectura: SQLPage sin backend intermedio 15

4.3 El contrato de datos 16

4.4 Páginas y funcionalidades 16
4.4.1 Página de inicio 16
4.4.2 Clasificación de un año, deporte, género y categoría 17
4.4.3 Histórico de un equipo 18
4.4.4 Histórico de un deporte 21
4.4.5 Explorador de estadísticas 22
4.5 Explorador de estadísticas 23
4.6 Diseño visual 23
Capítulo 5 - Despliegue 24
5.1 Contenerización con Docker 24
5.2 Dos imágenes independientes, una red compartida 24
5.3 Despliegue en un servidor real 24
5.4 Incidencia de compatibilidad de CPU y migración a MariaDB 25
Capítulo 6 - Tecnologías empleadas 26
6.1 Lenguajes y bases de datos 26
6.2 SQLPage 26
6.3 Docker y Docker Compose 26
6.4 Claude Code 27
6.5 Herramientas exploradas para la digitalización automática y sin supervisión 27
6.6 Otras herramientas 27
Capítulo 7 - Pruebas y resultados 28
7.1 Verificación de los datos 28
7.2 Validación de índices con EXPLAIN 28
7.3 Pruebas funcionales de la aplicación 29
Capítulo 8 - Conclusiones y trabajo futuro 31

Introduction 33

Conclusions and future work 35

Bibliografía 37

Apéndice A - Estructura del repositorio y guía de despliegue 39

Apéndice B - Resumen de incidencias de normalización de datos 41

ÍNDICE DE FIGURAS

Figura 3-1. Cuadernillo original de una temporada 13

Figura 4-1. Arquitectura de la aplicación (navegador – SQLPage – base de datos) **¡Error! Marcador no definido.**

Figura 4-2. Página de inicio

Figura 4-3. Clasificación de un año, deporte, género y categoría

Figura 4-4. Listado de todos los equipos

Figura 4-5. Histórico de un equipo

Figura 4-6. Listado de deportes

Figura 4-7. Histórico de un deporte

Figura 4-8. Explorador de estadísticas

Figura 5-1. Contenedores de datos y de aplicación sobre la red compartida

ÍNDICE DE TABLAS

Tabla 3-1. Ejemplos de normalización de nombres de equipo 14

Capítulo 1 - Introducción

1.1 Motivación

Desde el curso 1990-91, la Oficina de Deportes de la Universidad Complutense de Madrid organiza cada año una competición interna entre facultades, colegios mayores y otros centros adscritos a la universidad: la liga del Trofeo Rector, junto con una clasificación adicional, el Trofeo Alfonso XIII, que premia la participación conjunta de un mismo centro en varios deportes. Los resultados de cada temporada, clasificaciones, ascensos y descensos, descalificaciones, se han recogido tradicionalmente en un cuadernillo impreso por curso, uno por cada uno de los deportes y categorías disputados ese año.

Con el paso de las décadas se han acumulado más de treinta cuadernillos, que en los últimos años se conservan también en formato PDF escaneado, pero que en ningún caso están volcados en una base de datos ni son consultables por Internet. Toda esa información, quién quedó campeón de qué deporte en qué año, qué facultades ascendieron o descendieron de categoría, qué colegio mayor ha sido el más laureado. Existe únicamente sobre el papel, guardada en un cajón de la Oficina de Deportes.

La motivación de este trabajo nace precisamente de ahí: de la idea de recuperar esa memoria del deporte universitario complutense, verificarla con el mismo rigor con el que se trataría cualquier fuente histórica, y ponerla a disposición de toda la comunidad universitaria a través de una aplicación web sencilla de consultar, en vez de dejarla reservada a quien tenga acceso físico a esos cuadernillos.

1.2 Objetivos

Los objetivos de este Trabajo de Fin de Grado son los siguientes:

- Recuperar y verificar el histórico completo de clasificaciones del Trofeo Rector y del Trofeo Alfonso XIII de la UCM del que se dispone de documentación, trasladándolo a un modelo de datos único, coherente y consultable.

- Estudiar la viabilidad de automatizar total o parcialmente la extracción de estos datos a partir de los cuadernillos escaneados, mediante herramientas de reconocimiento de documentos e inteligencia artificial.
- Diseñar un proceso de verificación que garantice la fidelidad de los datos digitalizados frente al documento original, documentando cualquier incidencia o duda encontrada en el propio proceso.
- Diseñar e implementar una aplicación web de consulta, reutilizable frente a cualquier conjunto de datos con el mismo contrato, que permita explorar el histórico por año, deporte, género, categoría, equipo y estadísticas agregadas.
- Desplegar la aplicación mediante contenedores Docker, separando el ciclo de vida de los datos del de la aplicación, de forma que ambos puedan actualizarse o sustituirse de manera independiente.
- Dejar el sistema preparado para que, en una fase posterior, pueda adaptarse visualmente a la identidad gráfica de la UCM o de la Oficina de Deportes y publicarse de forma accesible para toda la comunidad universitaria.

1.3 Plan de trabajo

El trabajo se organizó en dos fases claramente diferenciadas, ejecutadas de forma mayoritariamente secuencial aunque con realimentación entre ambas.

La primera fase se centró en los datos: se dedicó un tiempo considerable a explorar distintas formas de extraer automáticamente y sin supervisión la información de los cuadernillos escaneados (análisis de texto plano, reconocimiento de documentos con servicios en la nube, extracción asistida por modelos de lenguaje), evaluando en cada caso su fiabilidad frente a la enorme heterogeneidad de formato entre años y deportes. Al comprobar que ninguna de estas vías ofrecía garantías suficientes para un histórico que debía ser correcto, no solo plausible, se optó por un proceso de transcripción y verificación distinto: con la asistencia de Claude Code para redactar cada clasificación y ejecutar comprobaciones repetitivas, pero con el estudiante contrastando personalmente cada dato contra el documento original, año por año, y con herramientas propias para detectar automáticamente inconsistencias (duplicados,

valores fuera de rango, violaciones de la clave primaria) una vez aceptados los datos de cada año.

La segunda fase abordó el diseño, la implementación y el despliegue de la aplicación web de consulta: definición del contrato de datos, elección de tecnología, construcción de las distintas páginas y funcionalidades, validación de rendimiento, contenerización y despliegue en un servidor.

1.4 Organización de la memoria

El resto de esta memoria se organiza como sigue. El capítulo 2 repasa el estado de la cuestión: qué herramientas existen para digitalizar documentos de este tipo y qué opciones de arquitectura se barajaron para la aplicación web. El capítulo 3 describe en detalle el tratamiento de los datos de partida, incluyendo las aproximaciones de extracción automática que se probaron y por qué se descartaron, el diseño del modelo de datos y el proceso de verificación seguido. El capítulo 4 presenta el diseño y la implementación de la aplicación web. El capítulo 5 describe el proceso de despliegue en contenedores Docker y en un servidor real. El capítulo 6 detalla las tecnologías empleadas a lo largo de todo el proyecto. El capítulo 7 recoge las pruebas realizadas y los resultados obtenidos. Por último, el capítulo 8 presenta las conclusiones y las líneas de trabajo futuro.

Capítulo 2 - Estado de la cuestión

No existe, hasta donde se ha podido comprobar, ningún precedente de digitalización del histórico de competiciones internas de la UCM ni de otra universidad española con características similares: la información relevante para este trabajo son los propios cuadernillos en papel y PDF de la Oficina de Deportes, sin ningún sistema digital previo con el que compararse. El estado de la cuestión se plantea, por tanto, en dos frentes independientes: por un lado, qué herramientas existen para extraer información estructurada de documentos escaneados; por otro, qué arquitecturas son razonables para una aplicación web de consulta de datos históricos como la que se necesitaba construir.

2.1 Extracción automática de información de documentos escaneados

Los cuadernillos de años más recientes están disponibles como PDF generado a partir del escaneado de las páginas originales, es decir, como imágenes sin una capa de texto fiable. Para este tipo de documento existen, a grandes rasgos, tres familias de herramientas: el reconocimiento óptico de caracteres (OCR) clásico sobre texto plano ya extraído; los servicios de reconocimiento de documentos e inteligencia de formularios en la nube, que además de reconocer el texto identifican su estructura (tablas, campos, encabezados); y, más recientemente, los modelos de lenguaje multimodales, capaces de recibir directamente una imagen o un PDF y devolver una respuesta en un formato estructurado como JSON. En este trabajo se evaluaron representantes de las tres familias —analizando texto extraído directamente del PDF, empleando Azure AI Document Intelligence (conocido anteriormente como Azure Form Recognizer) para el reconocimiento de tablas, y empleando la API de OpenAI para pedir directamente una extracción en JSON— antes de decidir el enfoque final para este proyecto, tal y como se detalla en el capítulo 3.

2.2 Arquitecturas para aplicaciones web de consulta de datos

Para una aplicación cuyo único cometido es consultar datos ya existentes, sin edición ni lógica de negocio compleja, la arquitectura habitual de tres capas (frontend

en JavaScript, backend con un framework como Django, Flask, Spring o Express, y una base de datos relacional) resulta, en buena medida, sobredimensionada: implica mantener un lenguaje y un framework de backend, una capa de API propia, y un framework de frontend aparte, únicamente para trasladar filas de una tabla a una página HTML. Frente a esto, en los últimos años han aparecido herramientas pensadas específicamente para construir aplicaciones de este tipo escribiendo solo SQL, sin backend ni frontend independientes; SQLPage, la herramienta finalmente elegida para este trabajo, es una de ellas. Se explica con más detalle, junto con el resto de alternativas consideradas, en el apartado 4.2.

Capítulo 3 - Análisis y tratamiento de los datos de partida

Este capítulo describe el origen de los datos, las distintas aproximaciones de extracción automática que se probaron antes de decidir el enfoque definitivo, el diseño del modelo de datos y el proceso de verificación seguido para garantizar su fiabilidad.

3.1 Los cuadernillos de la Oficina de Deportes

La Oficina de Deportes de la UCM organiza, desde el curso 1990-91, una competición interna entre facultades, colegios mayores y otros centros universitarios, estructurada en dos clasificaciones: la liga del Trofeo Rector, específica de cada deporte, género y categoría, y el Trofeo Alfonso XIII, una clasificación adicional que premia la participación conjunta de un mismo centro en el conjunto de deportes. Cada temporada se documentó en un cuadernillo impreso con las clasificaciones de cada deporte, que en los años más recientes se conserva también como PDF escaneado.

El formato de estos cuadernillos dista mucho de ser uniforme a lo largo de más de treinta años. Cambia la disposición de las tablas, el número de páginas por deporte, si un deporte se organiza por grupos con una fase final o directamente por liga, si existe una clasificación de ascenso independiente, y si el Trofeo Alfonso XIII se presenta en una sección propia o mezclado con las clasificaciones de liga. Además, el propio Trofeo Rector como clasificación general no existió como tal hasta la temporada 2005-06: en las temporadas anteriores cada deporte se resuelve de forma independiente. Estas diferencias se fueron documentando de forma sistemática a medida que se revisaba cada año, y resultaron determinantes tanto para descartar la extracción totalmente automática y sin supervisión como para diseñar el proceso de verificación asistida descrito más adelante en este capítulo.

3.2 Primeras aproximaciones a la digitalización automática

Antes de decidir el enfoque definitivo, se exploraron varias vías para automatizar, total o parcialmente y sin intervención humana en cada caso concreto, la extracción de las clasificaciones a partir de los cuadernillos escaneados. Se describen a

continuación por orden cronológico, junto con los motivos por los que cada una se fue descartando o quedándose corta.

3.2.1 Extracción por reglas sobre texto plano

La primera aproximación consistió en convertir cada página del cuadernillo a texto plano y analizarlo línea a línea con un conjunto de reglas: detectar la palabra «Categoría» para identificar el deporte y el género, «Clasificación» para marcar el inicio de una tabla de resultados, «Descalificado» para las bajas, etc. Este enfoque, implementado en varias iteraciones sucesivas a medida que se iban descubriendo nuevos casos no contemplados, resultó extremadamente frágil: cada deporte, cada año e incluso cada categoría dentro de un mismo deporte podían presentar el resultado con un formato de línea ligeramente distinto (columnas en otro orden, presencia o ausencia de una jornada de descalificación, tabuladores en vez de espacios), lo que obligaba a añadir una excepción nueva por cada caso encontrado sin llegar nunca a una cobertura fiable.

3.2.2 Reconocimiento de documentos con Azure AI Document Intelligence

La segunda aproximación sustituyó el análisis de texto plano por un servicio de reconocimiento de documentos en la nube, Azure AI Document Intelligence (denominado Azure Form Recognizer en el momento de realizar estas pruebas), capaz de identificar no solo el texto de cada página sino también su estructura en tablas. El PDF de cada año se dividía en documentos de dos páginas con la biblioteca pypdf, se enviaba cada uno al servicio, y el resultado (líneas de texto y tablas reconocidas) se procesaba con una lógica similar a la del enfoque anterior para generar las sentencias SQL de inserción correspondientes. Para facilitar las pruebas sobre distintos documentos se construyó además una pequeña interfaz gráfica en Tkinter que permitía seleccionar un PDF y lanzar el proceso completo. Este enfoque mejoró sensiblemente el reconocimiento de las tablas de clasificación, pero seguía dependiendo de la misma lógica basada en reglas para interpretar el contexto alrededor de cada tabla (deporte, género, categoría, si se trataba del Trofeo Alfonso), y por tanto heredaba buena parte de la fragilidad del enfoque anterior frente a los cambios de formato entre años.

3.2.3 Extracción asistida por modelos de lenguaje

La tercera vía explorada consistió en delegar tanto el reconocimiento como la interpretación en un modelo de lenguaje multimodal, a través de la API de OpenAI: se enviaba directamente el PDF de un bloque de páginas junto con una serie de instrucciones pidiendo como respuesta un JSON con los registros reconocidos, para posteriormente convertir ese JSON en sentencias SQL. Este enfoque simplificaba mucho el código propio, al trasladar al modelo la tarea de interpretar el contexto de cada tabla, pero introducía un problema distinto: no hay garantía de que un modelo de lenguaje no complete o corrija silenciosamente un dato mal escaneado o parcialmente ilegible con un valor plausible pero incorrecto, lo cual es especialmente delicado al construir lo que se pretende que sea un histórico oficial de resultados deportivos.

3.2.4 Por qué se descartó la extracción totalmente automática

Ninguna de las tres aproximaciones anteriores llegó a ofrecer garantías suficientes de fidelidad frente al documento original en el conjunto completo de años y deportes. La causa de fondo es siempre la misma: más de treinta años de cuadernillos, elaborados por personas distintas en épocas distintas, no siguen un formato lo bastante regular como para que un proceso sin ningún tipo de supervisión humana generalice de un año a otro con garantías. Dado que el objetivo del trabajo era producir un histórico correcto, no simplemente plausible, lo que se descartó no fue el uso de herramientas de inteligencia artificial en sí, sino la idea de dejar que una de ellas propusiera la transcripción completa de un año sin que cada dato pasara después por una revisión humana contra el documento original. El apartado 3.4 describe el proceso que se siguió en su lugar, en el que sí se recurrió a un asistente de programación basado en inteligencia artificial, pero integrado en un flujo de trabajo dirigido y verificado en todo momento por el propio estudiante.

3.3 Diseño del modelo de datos

Todas las clasificaciones, con independencia del año, el deporte, el género o la categoría, se representan con una única tabla,

Clasificaciones, con una fila por cada equipo y clasificación en la que ha participado. Cada fila queda identificada por la combinación de año, deporte, equipo, género y categoría, que forma la clave primaria de la tabla, y contiene además el puesto obtenido, los puntos, si hubo ascenso o descenso, y la posición en el Trofeo Alfonso XIII si la hubiera. Los valores -1 se usan de forma consistente para representar una descalificación, tanto en el puesto de liga como en el del Trofeo Alfonso, mientras que los valores nulos indican que ese dato no aplica a la fila en cuestión (por ejemplo, un equipo que solo aparece en la clasificación del Trofeo Alfonso no tiene puesto de liga).

Este diseño, deliberadamente sencillo, permite responder a cualquier consulta sobre el histórico —la clasificación de un año y deporte concretos, el historial completo de un equipo, el palmarés de un deporte, comparativas entre equipos— con una única tabla y sin necesidad de normalizar en varias tablas relacionadas información que, en la práctica, rara vez cambia de forma independiente.

3.4 Proceso de verificación de los datos

Descartada la idea de una extracción totalmente automática y sin supervisión, la transcripción de los 26 años se llevó a cabo con la asistencia de Claude Code, un asistente de programación de Anthropic basado en modelos de lenguaje, usado aquí como herramienta de trabajo y en ningún momento como sustituto de la revisión del propio estudiante: para cada año se le indicaba trabajar sobre el documento original, el PDF escaneado del cuadernillo correspondiente y redactar, deporte a deporte, las sentencias SQL de inserción de la clasificación siguiendo el contrato de datos descrito en el apartado anterior. A partir de ahí, cada fichero se sometió a una revisión sistemática, dirigida en todo momento por el estudiante, contrastando cada dato con el documento original antes de darlo por válido.

Se eligió esta forma de trabajar, y no una transcripción manual completa sin ningún apoyo, porque permitía dedicar el esfuerzo del estudiante íntegramente a la parte que ninguna herramienta puede hacer por él, decidir si un dato transcrito corresponde realmente a lo que dice el documento original y resolver los casos ambiguos con criterio propio, en lugar de a la mecánica de teclear cada fila. El asistente aceleraba la primera redacción de cada clasificación y la ejecución repetida de las

comprobaciones descritas más abajo, pero ninguna fila se dio por buena sin que el estudiante la hubiera cotejado personalmente con el cuadernillo; cuando una transcripción no coincidía con el original, o cuando un caso resultaba ambiguo, la decisión final, incluida la de dejar constancia escrita de la duda cuando no podía resolverse con total certeza, fue siempre del estudiante, nunca del asistente.

Además de esa revisión página a página, se automatizaron dos comprobaciones objetivas para detectar deslices que una revisión visual pudiera pasar por alto: cada fichero, una vez aceptado, se cargaba en una base de datos SQLite para comprobar que no violara la clave primaria de la tabla (lo que habría delatado una fila duplicada o un dato de identificación mal transcrito) y que todos los valores estuvieran dentro de rango. Al cierre de la revisión, los 26 años con documento disponible cargan sin errores de sintaxis ni violaciones de clave primaria, con un total de 7.479 filas.

Todas las incidencias encontradas durante este proceso, desde erratas puntuales en el cuadernillo original hasta patrones de error sistemáticos que se repetían en varios años, se documentaron por escrito a medida que se detectaban, junto con el criterio que el estudiante decidió aplicar para resolver cada una, de forma que cualquier decisión tomada durante la digitalización quedara trazada y fuera reproducible.

3.5 Normalización de nomenclatura

Al tratarse de más de treinta cuadernillos redactados en épocas distintas, un mismo equipo o un mismo deporte no siempre se nombraba igual de un año a otro: abreviaturas distintas para un mismo colegio mayor, uso indistinto de mayúsculas o solo la inicial en mayúscula, o distinto número de jugadores indicado para una misma modalidad de fútbol. Sin una normalización posterior, la aplicación mostraría como equipos o deportes distintos entradas que en realidad se refieren siempre a lo mismo.

Antes de fusionar dos nombres candidatos a ser el mismo equipo, se comprobó contra los propios datos que nunca coincidieran como entradas distintas dentro de una misma clasificación (mismo año, deporte, género y categoría): si coincidían, se trataba en realidad de dos equipos distintos con nombres parecidos, y se dejaban sin fusionar. Este criterio, aplicado de forma sistemática, permitió reducir alrededor de 180 formas distintas de escribir un nombre de equipo a 110 equipos reales, y unificar bajo un único

nombre varios deportes que en unos años se registraban con una palabra y en otros con el número de jugadores (por ejemplo, fútbol registrado unas veces como «Fútbol» y otras como «Fútbol 11»). Este análisis, de nuevo con la ayuda de Claude Code para cruzar de forma sistemática cada pareja de nombres candidata contra los más de 7.000 registros, se revisó caso por caso: los cruces sin ambigüedad se unificaron directamente, y los que quedaban abiertos a interpretación —por ejemplo, si dos nombres muy parecidos correspondían de verdad al mismo equipo o a dos equipos distintos— se plantearon explícitamente al estudiante, que fue quien decidió en cada caso el nombre final o si debían mantenerse separados.

13.

COMPETICION INTERNA OFICIAL - FACULTADES - CURSO 1.991/92

1ª CATEGORIA - RUGBY MASCULINO

CLASIFICACION FINAL

1º	-	FISICAS	15 puntos.
2º	-	SAN PABLO CEU	14 puntos.
3º	-	GEOGRAFIA E HISTORIA	14 puntos.
4º	-	VETERINARIA	13 puntos.
5º	-	GEOLOGICAS	12 puntos.
6º	-	QUIMICAS	8 puntos.
7º	-	DERECHO	6 puntos.
8º	-	MATEMATICAS	4 puntos.
9º	-	ECONOMICAS	2 puntos.
10º	-	POLITICAS	0 puntos.

DESCALIFICADOS.:

- C. Información.
- Farmacia.

SE CLASIFICAN PARA EL TROFEO ALFONSO XIII

1º - FISICAS.

2º - SAN PABLO CEU.

Figura 3-1. Cuadernillo original de una temporada

La Tabla 3-1 recoge, a modo de ejemplo, algunos de los casos de normalización aplicados.

Forma(s) originales	Nombre final	Motivo
CMU BARBERAN /C.M. BARBERAN / Barberan	C.M. BARBERAN	Mismo colegio mayor con distinta abreviatura y distintas mayúsculas
Fútbol	Fútbol 11	Mismo deporte, distinto nombre según el año
Fútbol Siete	Fútbol 7	Mismo deporte, unificado bajo la forma numérica

Tabla 3-1. Ejemplos de normalización de nombres de equipo

Un efecto colateral de este proceso fue el descubrimiento de un defecto de configuración en la imagen Docker de la base de datos: al cargar los ficheros SQL, correctamente codificados en UTF-8, sin forzar ese mismo juego de caracteres en el cliente de la base de datos, algunos acentos se almacenaban mal codificados, generando de forma artificial nombres de equipo adicionales que en realidad eran el mismo con un carácter corrupto. Corregido el juego de caracteres del cliente, el problema desapareció por completo.

Capítulo 4 - Diseño e implementación de la aplicación web

Una vez verificados los datos de las 26 temporadas disponibles, el segundo bloque del trabajo consistió en construir una aplicación web que permitiera consultarlos sin necesidad de acceder directamente a la base de datos. Este capítulo describe los requisitos que se fijaron para esa aplicación, la arquitectura elegida y el motivo, y las distintas páginas y funcionalidades implementadas.

4.1 Requisitos y restricciones de diseño

Antes de elegir tecnología se fijaron una serie de restricciones de diseño que debía cumplir la aplicación:

- Ser de solo lectura: la aplicación consulta el histórico, no lo edita. No es necesario ni deseable exponer ninguna operación de escritura sobre los datos.
- No depender de ningún dato concreto escrito en el código: ningún año, deporte, equipo o categoría debía aparecer escrito a mano en ninguna página. Todas las opciones de los filtros y todas las cifras debían obtenerse en tiempo real a partir de la base de datos, de forma que la misma aplicación sirviera para el dataset actual y para cualquier otro que respete el mismo contrato de datos.
- Minimizar las piezas móviles: evitar un backend a medida, una API propia y un framework de frontend independiente si no aportan nada que no resuelva ya el propio motor de base de datos.
- Ser desplegable con Docker de forma reproducible, separando el ciclo de vida de los datos del de la aplicación.

4.2 Arquitectura: SQLPage sin backend intermedio

Para cumplir estas restricciones se eligió SQLPage, una herramienta de código abierto que permite construir aplicaciones web completas escribiendo únicamente consultas SQL. Cada página de la aplicación es un fichero .sql que SQLPage ejecuta contra la base de datos; el resultado de cada consulta se interpreta como un componente visual (una tabla, un formulario, una tarjeta con una cifra…) que SQLPage renderiza como HTML, sin que el desarrollador tenga que escribir plantillas ni JavaScript. La arquitectura resultante es tan simple como se muestra en la Figura 4-1: el navegador habla directamente con SQLPage, y SQLPage con la base de datos, sin ningún backend propio ni API REST intermedia.

Figura 4-1. Arquitectura de la aplicación (navegador – SQLPage – base de datos)

Se valoraron alternativas más convencionales —un backend en Flask o Django con una API propia y un frontend independiente— pero se descartaron por resultar desproporcionadas para una aplicación cuyo único cometido es consultar datos ya existentes: habrían añadido un lenguaje, un framework y una capa de API completos

únicamente para trasladar filas de una tabla a una página HTML, sin ningún beneficio adicional dado que no hay lógica de negocio ni operaciones de escritura que justifiquen esa complejidad.

Todas las páginas comparten una cabecera y un menú comunes, definidos en un único fichero de configuración (shell.json) que SQLPage aplica a cada página, de forma que cambiar el título, el menú o el pie de página de toda la aplicación no requiere tocar cada fichero por separado.

4.3 El contrato de datos

La aplicación no asume nada sobre el contenido de la base de datos: se limita a exigir que exista una tabla Clasificaciones con las columnas descritas en el apartado 3.3, documentadas en un fichero schema.sql que actúa como contrato entre los datos y la aplicación. Cualquier base de datos MySQL o MariaDB que exponga esa misma tabla puede conectarse a la aplicación sin modificar ni una línea de las páginas: ni el número de deportes, ni los equipos, ni el rango de años están fijados en ningún sitio del código, sino que se obtienen siempre mediante consultas DISTINCT o agregaciones sobre la tabla en el momento de renderizar cada página. Esto permite, en principio, reutilizar la misma aplicación para digitalizar el histórico de otra competición universitaria distinta sin más que cargar sus datos con ese mismo formato.

4.4 Páginas y funcionalidades

La aplicación se organiza en seis páginas, descritas a continuación.

4.4.1 Página de inicio

Muestra un resumen del dataset completo (número de temporadas, de deportes, y de deportes en los que ha habido competición masculina y femenina) y un formulario con cuatro desplegables; año, deporte, género y categoría, que se muestran siempre desde el principio, cada uno con todas las opciones que existen en los datos. Al pulsar «Buscar», si los cuatro campos están rellenos se pasa directamente a la clasificación solicitada; si falta alguno, se muestra un aviso indicando exactamente qué campo falta por elegir, en lugar de dejar el botón sin reaccionar.

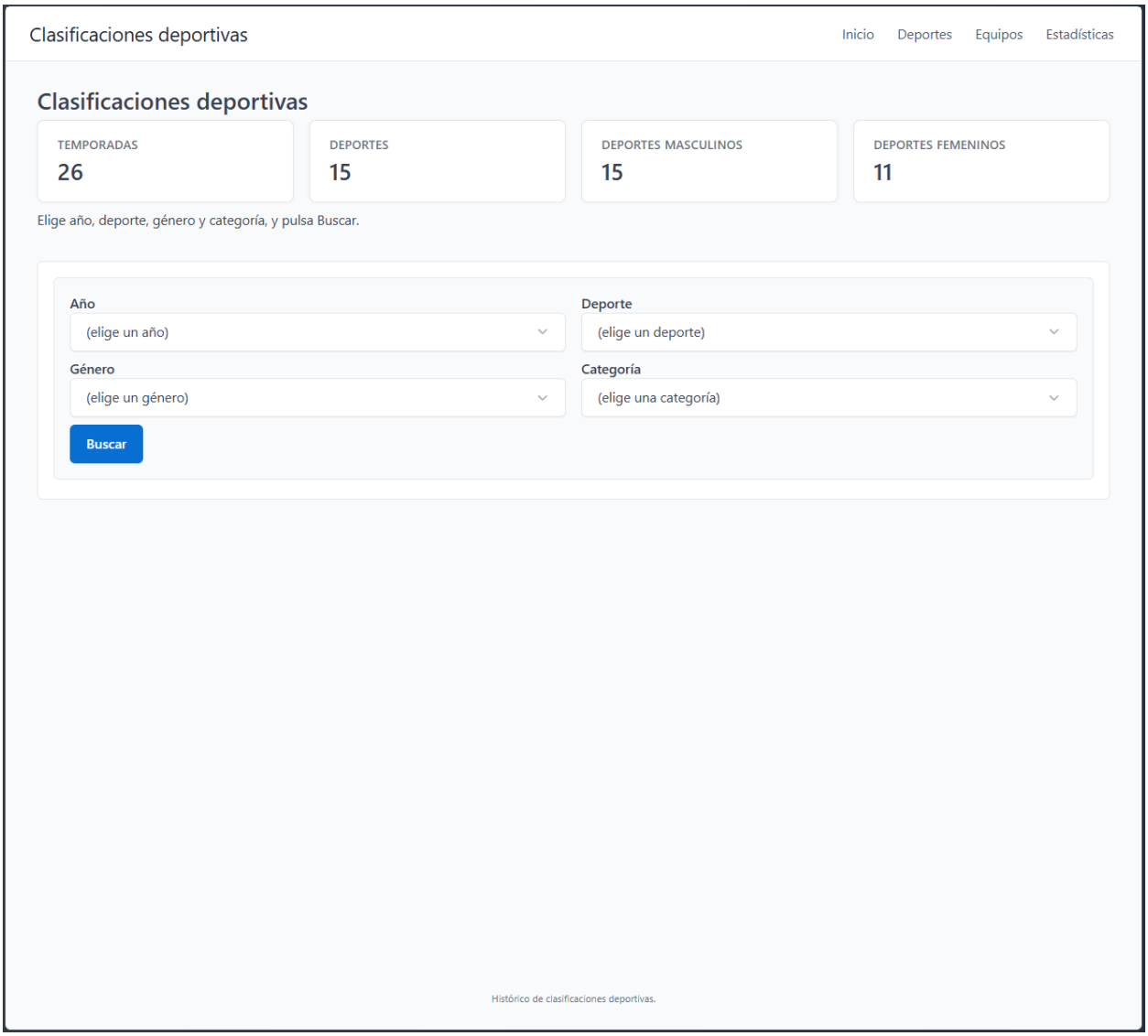


Figura 4-2. Página de inicio

4.4.2 Clasificación de un año, deporte, género y categoría

Muestra la tabla de posiciones (posición, equipo y puntos) de la clasificación solicitada, excluyendo los equipos sin puesto de liga, y debajo, si corresponde, una segunda tabla con la clasificación del Trofeo Alfonso XIII de esa misma combinación. Si la combinación de año, deporte, género y categoría solicitada no existe en los datos, se muestra un aviso claro en vez de una tabla vacía.

Clasificaciones deportivas

Inicio Deportes Equipos Estadísticas

Inicio / RUGBY / 1991-92 · Masculino · 1ª categoría

RUGBY

POSICIÓN	EQUIPO	PUNTOS
1	FISICAS	15
2	SAN PABLO CEU	14
3	GEOGRAFIA E HISTORIA	14
4	VETERINARIA	13
5	GEOLOGICAS	12
6	QUIMICAS	8
7	DERECHO	6
8	MATEMATICAS	4
9	ECONOMICAS	2
10	POLITICAS	0
DSQ	C. INFORMACION	—
DSQ	FARMACIA	—

Trofeo Alfonso

POSICIÓN	EQUIPO
1	FISICAS
2	C.M. XIMENEZ DE CISNEROS
3	C.M. LA SALLE
4	SAN PABLO CEU

Histórico de clasificaciones deportivas.

Figura 4-3. Clasificación de un año, deporte, género y categoría

4.4.3 Histórico de un equipo

Al entrar desde la opción «Equipos» del menú superior sin haber elegido todavía ningún equipo, la página muestra un listado con todos los equipos que aparecen en el dataset, cada uno con el número de participaciones y de deportes distintos en los que ha competido (Figura 4-4).

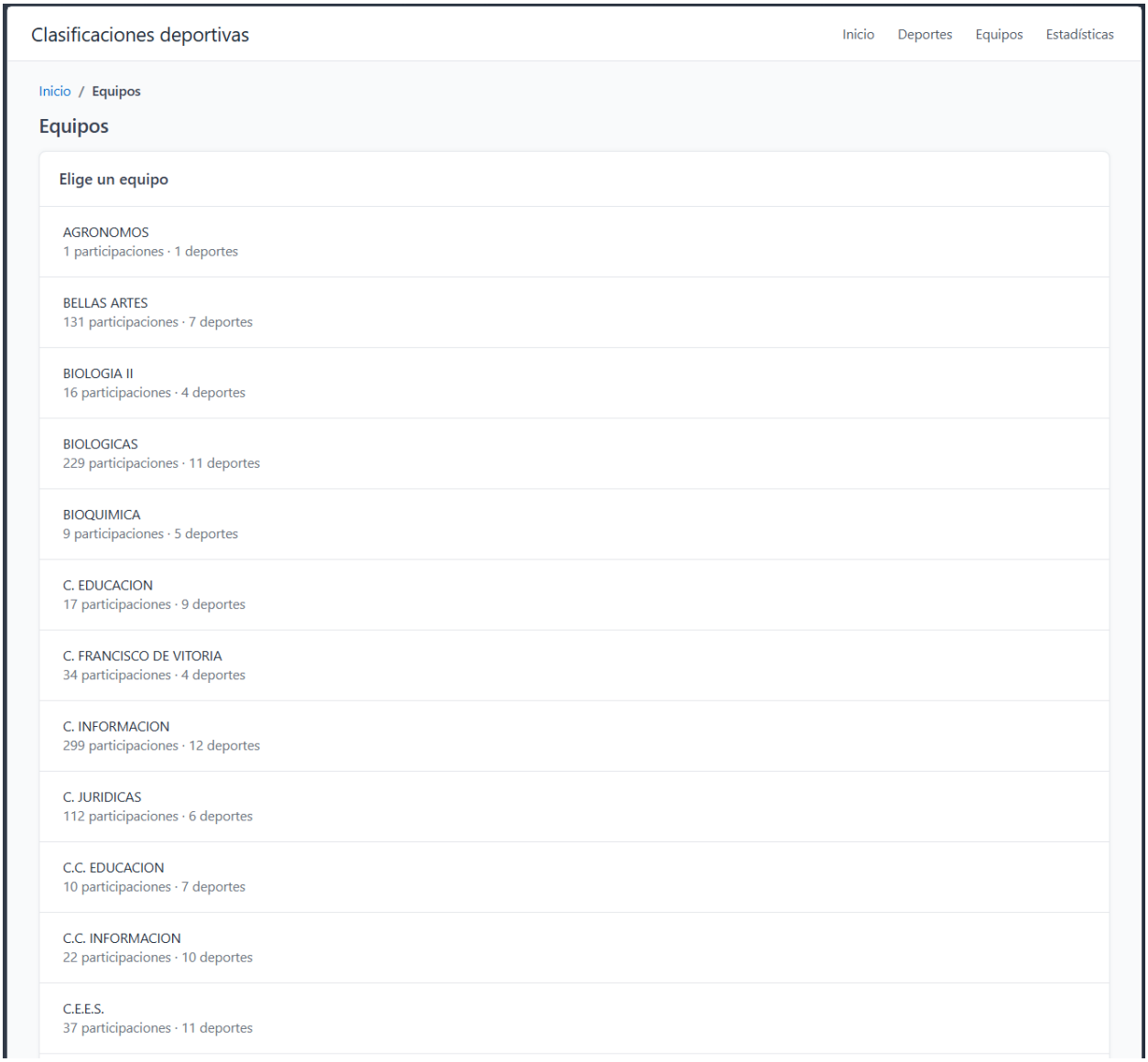
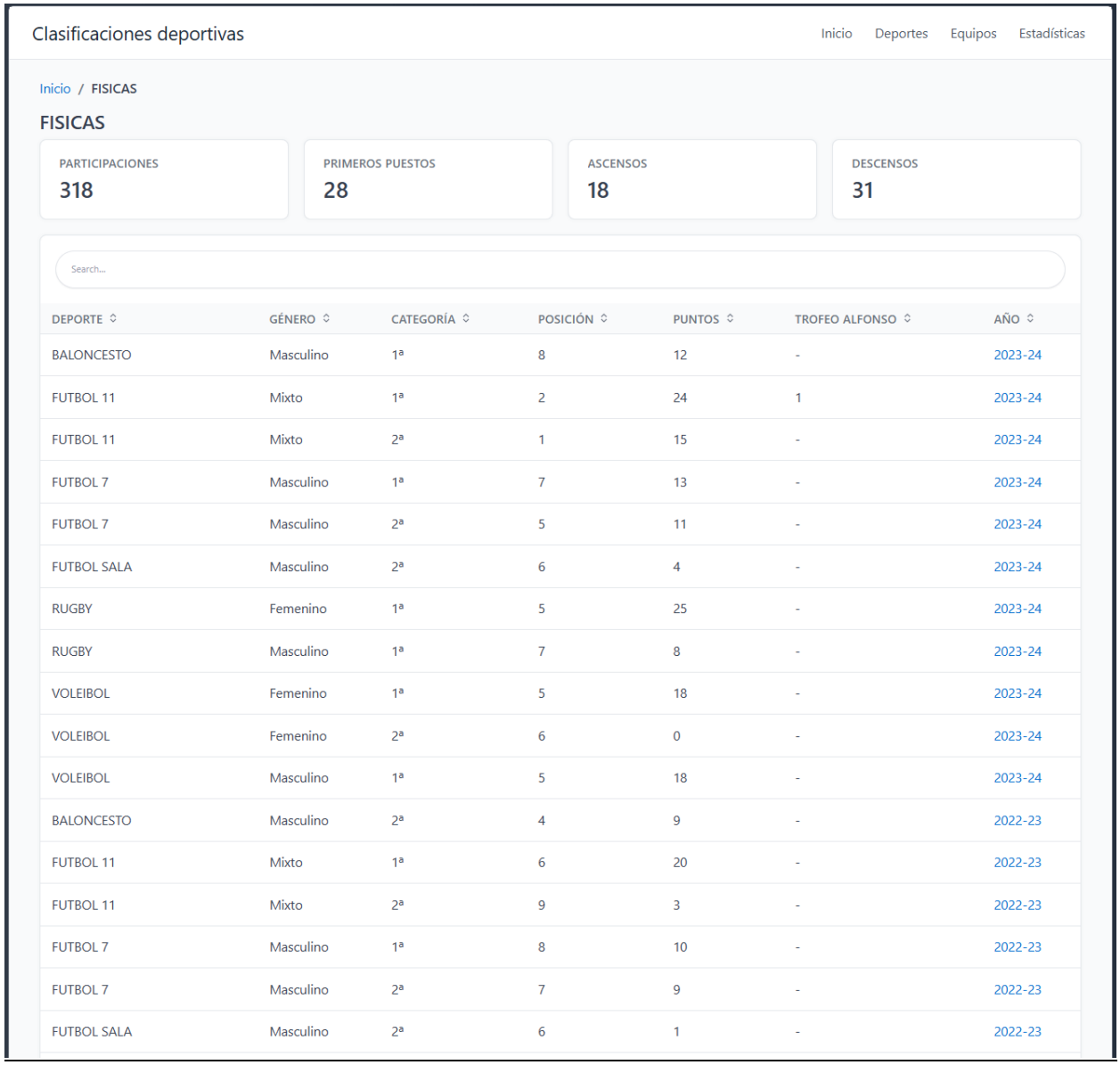


Figura 4-4. Listado de todos los equipos



DEPORTE	GÉNERO	CATEGORÍA	POSICIÓN	PUNTOS	TROFEO ALFONSO	AÑO
BALONCESTO	Masculino	1ª	8	12	-	2023-24
FUTBOL 11	Mixto	1ª	2	24	1	2023-24
FUTBOL 11	Mixto	2ª	1	15	-	2023-24
FUTBOL 7	Masculino	1ª	7	13	-	2023-24
FUTBOL 7	Masculino	2ª	5	11	-	2023-24
FUTBOL SALA	Masculino	2ª	6	4	-	2023-24
RUGBY	Femenino	1ª	5	25	-	2023-24
RUGBY	Masculino	1ª	7	8	-	2023-24
VOLEIBOL	Femenino	1ª	5	18	-	2023-24
VOLEIBOL	Femenino	2ª	6	0	-	2023-24
VOLEIBOL	Masculino	1ª	5	18	-	2023-24
BALONCESTO	Masculino	2ª	4	9	-	2022-23
FUTBOL 11	Mixto	1ª	6	20	-	2022-23
FUTBOL 11	Mixto	2ª	9	3	-	2022-23
FUTBOL 7	Masculino	1ª	8	10	-	2022-23
FUTBOL 7	Masculino	2ª	7	9	-	2022-23
FUTBOL SALA	Masculino	2ª	6	1	-	2022-23

Figura 4-5. Histórico de un equipo

Al elegir uno de ellos, la página recoge toda la trayectoria de ese equipo a lo largo de los años: deporte, género, categoría, posición de liga (con una flecha que indica ascenso o descenso, cuando lo hubo), puntos y posición en el Trofeo Alfonso XIII de cada temporada en la que ha participado, además de un resumen con el número de participaciones, primeros puestos, ascensos y descensos totales. El año de cada fila enlaza directamente a la clasificación completa de esa combinación (Figura 4-5).

4.4.4 Histórico de un deporte

Al entrar desde la opción «Deportes» del menú superior sin haber elegido todavía ningún deporte, la página muestra un listado con todos los deportes disputados a lo largo del histórico, cada uno con el número de participaciones y de temporadas registradas (Figura 4-6). Al elegir uno, se muestran dos pestañas: todas las combinaciones de año, género y categoría disputadas en ese deporte, cada una con enlace a su clasificación completa, o bien la relación completa de sus campeones a lo largo de los años, con ese mismo enlace en cada fila (Figura 4-7).

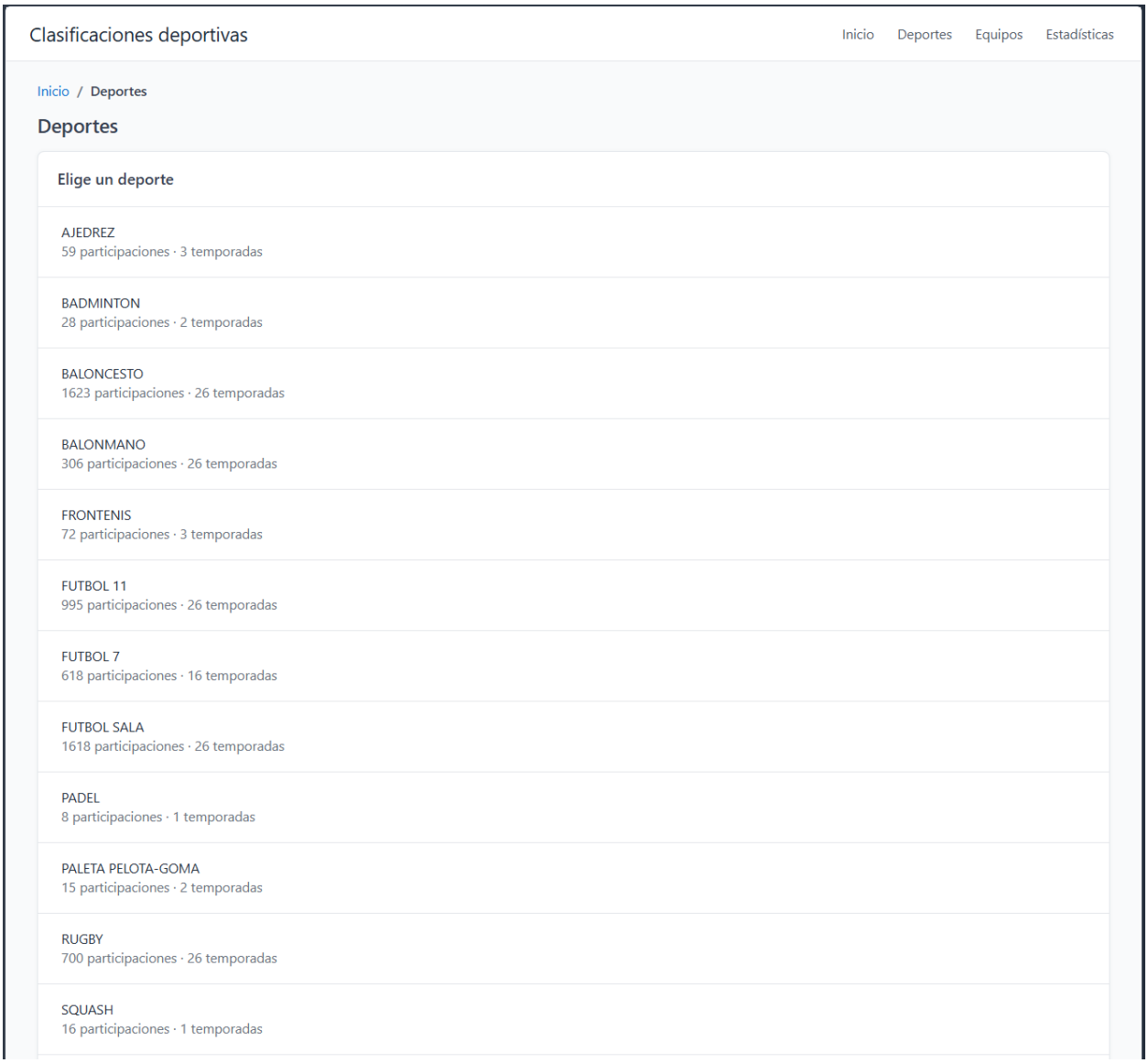


Figura 4-6. Listado de deportes

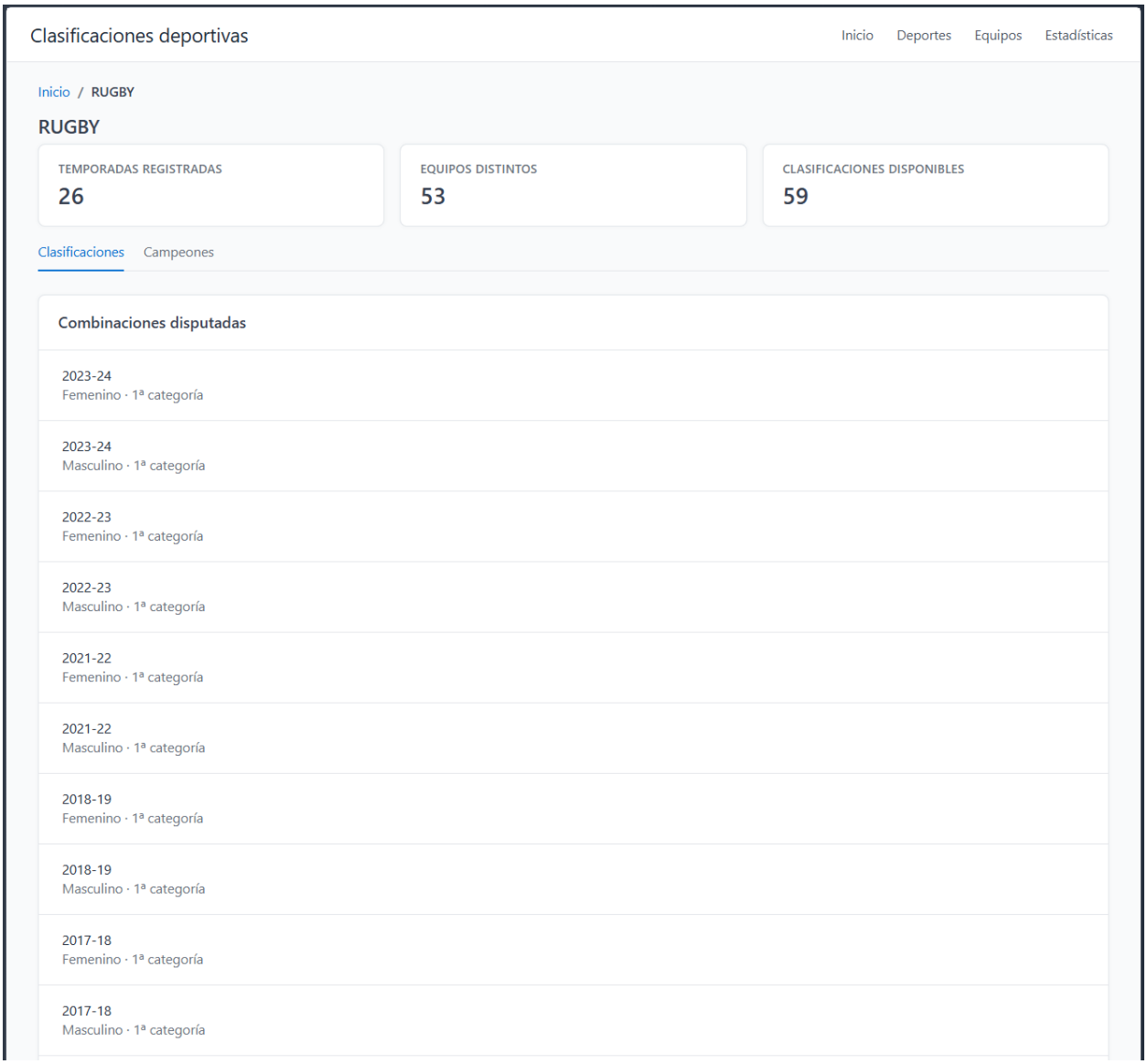


Figura 4-7. Histórico de un deporte

4.4.5 Explorador de estadísticas

Se describe en detalle en el apartado 4.5.

Figura 4-8. Explorador de estadísticas

4.5 Explorador de estadísticas

Además de la consulta de clasificaciones concretas, la aplicación ofrece un catálogo cerrado de doce estadísticas predefinidas: número de títulos de un equipo, mejor y peor puesto histórico, ascensos y descensos, comparativa entre dos equipos, equipos con más títulos o con más apariciones en el Trofeo Alfonso, campeones distintos en una temporada, entre otras. Cada una con sus propios filtros (equipo, deporte, género, año), que se muestran u ocultan según lo que necesite la estadística elegida. Cuando un filtro admite no aplicarse, se ofrece explícitamente una opción «(Todos)» en vez de dejar el campo vacío de forma ambigua.

Se valoró inicialmente permitir preguntas en lenguaje natural resueltas por un modelo de lenguaje que generase la consulta SQL correspondiente, pero se descartó por el riesgo que implica ejecutar SQL generado dinámicamente sin una validación estricta previa; se optó en su lugar por un catálogo cerrado de preguntas ya definidas, cada una resuelta con una consulta SQL fija y parametrizada de forma segura a través de los mecanismos propios de SQLPage, sin generar SQL en tiempo de ejecución.

4.6 Diseño visual

El aspecto visual de la aplicación es el que ofrece SQLPage por defecto, basado en el framework Tabler, con un CSS propio mínimo, limitado a los pocos ajustes que los componentes nativos no resuelven por sí solos (por ejemplo, agrupar visualmente un formulario de filtros con su botón de búsqueda). Se ha optado deliberadamente por no invertir esfuerzo en un diseño gráfico propio en esta fase del proyecto: el objetivo prioritario ha sido garantizar que los datos fueran correctos y que la aplicación fuera funcional, dejando preparado el terreno. Una estructura HTML y CSS limpia y estándar, sin elementos gráficos ni colores propios. Para que, en una fase posterior, el aspecto visual se pueda adaptar a la identidad gráfica de la Universidad Complutense o, más concretamente, a la de la propia web de la Oficina de Deportes de la UCM, de cara a una eventual publicación oficial dentro de ese entorno.

Capítulo 5 - Despliegue

Este capítulo describe cómo se ha empaquetado y desplegado la aplicación, separando de forma deliberada el ciclo de vida de los datos del de la aplicación web, y una incidencia real de compatibilidad encontrada al desplegar en un servidor concreto.

5.1 Contenerización con Docker

Tanto la base de datos como la aplicación web se distribuyen como imágenes Docker, de forma que todo el entorno necesario para ejecutarlas, motor de base de datos, SQLPage, y en el caso de los datos, el propio dataset ya cargado. Queda fijado y es reproducible en cualquier máquina que tenga Docker instalado, sin depender de lo que haya instalado el sistema anfitrión.

5.2 Dos imágenes independientes, una red compartida

Se decidió separar los datos de la aplicación en dos imágenes Docker independientes, cada una con su propio docker-compose: clasificaciones-data, que contiene el motor de base de datos con el dataset ya cargado a partir de los ficheros SQL verificados, y clasificaciones-web, que contiene únicamente SQLPage y las páginas de la aplicación. Ambas se comunican a través de una red Docker compartida, de forma que pueden desplegarse en el mismo host o en distintos, y sobre todo, pueden actualizarse de forma independiente: cargar un nuevo año de datos no implica reconstruir la aplicación web, y viceversa.

```
PS C:\Users\gabri\Desktop\UNIVERSIDAD\6º\TFG\Cuadernillos> docker ps --format "table {{.Names}}\t{{.Status}}\t{{.Ports}}"
NAMES                        STATUS                  PORTS
clasificaciones-web-sqlpage-1   Up 6 minutes            0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp
clasificaciones-data         Up 41 minutes (healthy)   0.0.0.0:3306->3306/tcp, [::]:3306->3306/tcp
PS C:\Users\gabri\Desktop\UNIVERSIDAD\6º\TFG\Cuadernillos> docker network inspect clasificaciones-net --format "Red: {{.Name}}    Contenedores: {{range .Containers}}{{.Name}} {{end}}"
Red: clasificaciones-net  →  Contenedores: clasificaciones-web-sqlpage-1 clasificaciones-data
```

Figura 5-1. Contenedores de datos y de aplicación sobre la red compartida

5.3 Despliegue en un servidor real

Además de ejecutarse en local para desarrollo, la aplicación se ha desplegado en una máquina virtual del Departamento de Arquitectura de Computadores y Automática de la UCM, accesible únicamente a través de la red del campus o de su VPN. El despliegue se realizó transfiriendo el proyecto por SFTP, generando credenciales

de base de datos nuevas y robustas específicas para ese entorno (nunca las mismas que se usan en desarrollo local) y configurando SQLPage en modo de producción, en el que se cachean las páginas y se ocultan los detalles internos de cualquier error frente al usuario final.

Al no estar el puerto de la aplicación expuesto públicamente en esa máquina, el acceso durante las pruebas se realizó mediante un túnel SSH sobre el mismo puerto utilizado para la conexión remota, quedando pendiente, para una publicación definitiva, solicitar la apertura de un puerto público o su integración detrás de un servidor web ya existente.

5.4 Incidencia de compatibilidad de CPU y migración a MariaDB

Al desplegar por primera vez la imagen de base de datos en el servidor del departamento, el contenedor entraba en un bucle de reinicio con el error «Fatal glibc error: CPU does not support x86-64-v2». La causa resultó ser que esa máquina virtual expone a sus contenedores una CPU virtual genérica («Common KVM processor»), pensada para ser compatible con distintos servidores físicos subyacentes, que carece de las instrucciones (SSE4.2, POPCNT) que las imágenes oficiales recientes de MySQL exigen desde que su sistema base se compila con un nivel de arquitectura x86-64-v2.

Descartada la opción de modificar la configuración de la máquina virtual, fuera del alcance de este trabajo, se comprobó qué alternativas eran compatibles con esa CPU. MariaDB, un motor de base de datos derivado de MySQL con el que comparte protocolo de red y dialecto SQL, sí resultó compatible, por lo que se sustituyó MySQL por mariadb:10.11 en la imagen de datos. Al compartir MariaDB el mismo protocolo de red que MySQL, SQLPage se conecta exactamente igual, y ninguna de las consultas SQL de la aplicación, incluidas las funciones de agregación en JSON empleadas en varios formularios, tuvo que modificarse.

Esta incidencia, aunque anecdótica, ilustra un principio general de cualquier despliegue en un entorno que no se controla por completo: conviene comprobar la compatibilidad real del software en el hardware (o hardware virtualizado) de destino, en lugar de asumir que el mismo comportamiento observado en desarrollo se sostendrá siempre en producción.

Capítulo 6 - Tecnologías empleadas

Este capítulo recoge, a modo de referencia, las principales tecnologías, lenguajes, bibliotecas y herramientas empleadas a lo largo de todo el trabajo, tanto en la fase de tratamiento de los datos como en la de la aplicación web.

6.1 Lenguajes y bases de datos

- SQL, como lenguaje principal tanto de la aplicación web (todas sus páginas están escritas en SQL) como del propio proceso de digitalización de los datos (las clasificaciones se transcriben directamente como sentencias INSERT).
- SQLite, empleado como motor auxiliar y desechable durante la fase de verificación de los datos, para comprobar de forma automática que cada fichero anual cargaba sin errores de sintaxis ni violaciones de la clave primaria.
- MySQL 8 y, tras la incidencia descrita en el apartado 5.4, MariaDB 10.11, como motor de base de datos de la aplicación en producción.
- Python, empleado en la fase inicial de exploración de la extracción automática de los cuadernillos (lectura y división de PDF, llamadas a servicios de reconocimiento de documentos y a modelos de lenguaje, generación de SQL a partir de los resultados) y en distintos scripts de apoyo durante el despliegue.

6.2 SQLPage

Motor sobre el que se construye toda la aplicación web: interpreta ficheros .sql como páginas web, traduciendo el resultado de cada consulta en componentes visuales predefinidos (tablas, formularios, tarjetas de cifras, alertas, migas de pan, pestañas) sin necesidad de escribir HTML, JavaScript ni backend propio.

6.3 Docker y Docker Compose

Empleados para empaquetar y desplegar tanto la base de datos como la aplicación web de forma reproducible, como se describe en el capítulo 5.

6.4 Claude Code

Asistente de programación de Anthropic basado en modelos de lenguaje, usado como herramienta de trabajo a lo largo de todo el proyecto, en ambos bloques descritos en el plan de trabajo (apartado 1.3): en el primero, para transcribir cada clasificación a sentencias SQL a partir del cuadernillo original y para ejecutar de forma repetida las comprobaciones de consistencia descritas en el apartado 3.4; en el segundo, como apoyo para escribir, depurar y probar las páginas SQLPage de la aplicación y los ficheros de despliegue con Docker. En ningún caso se empleó de forma autónoma: cada sentencia SQL, cada página de la aplicación y cada decisión de diseño fue revisada, probada y aprobada por el propio estudiante antes de darla por buena, tal y como se detalla en los capítulos 3 y 7.

6.5 Herramientas exploradas para la digitalización automática y sin supervisión

- Azure AI Document Intelligence (anteriormente Azure Form Recognizer), servicio de reconocimiento de documentos e identificación de tablas en la nube, empleado junto con la biblioteca de Python pypdf para dividir los PDF en bloques de páginas antes de enviarlos a analizar.
- API de OpenAI (modelos de la familia GPT), empleada para explorar la extracción directa de información estructurada en JSON a partir del PDF escaneado.
- Poppler (a través de la utilidad pdftoppm), empleada para convertir las páginas de un PDF escaneado en imágenes JPEG de alta resolución.
- Tkinter, biblioteca estándar de Python para interfaces gráficas de escritorio, empleada para construir una pequeña aplicación de apoyo con la que lanzar el reconocimiento de un documento sin tener que ejecutar el script por línea de comandos.

6.6 Otras herramientas

- Tabler, framework de componentes visuales sobre el que SQLPage construye su interfaz por defecto.

- Git, para el control de versiones tanto del proyecto de digitalización de datos como del código de la aplicación web.
- OpenSSH, para el acceso y despliegue remoto en el servidor del departamento descrito en el apartado 5.3.

Capítulo 7 - Pruebas y resultados

A lo largo del desarrollo se siguió una metodología de comprobación sistemática, tanto de los datos como de la aplicación, en lugar de dar por buena una funcionalidad por el simple hecho de no producir un error visible. Este capítulo resume esa metodología y los resultados obtenidos.

7.1 Verificación de los datos

Como se ha descrito en el apartado 3.4, cada uno de los 26 años con documento disponible se transcribió con la asistencia de Claude Code y se contrastó personalmente, dato a dato, contra su cuadernillo original, además de comprobarse de forma automática que cargara sin errores de sintaxis ni violaciones de la clave primaria. El resultado final es una base de datos de 7.479 filas, correspondientes a 26 temporadas (desde 1990-91 hasta 2023-24), 18 deportes distintos y 110 equipos reales tras el proceso de normalización de nomenclatura.

7.2 Validación de índices con EXPLAIN

Sobre esa base de datos ya cargada se analizaron con el comando EXPLAIN los planes de ejecución de las consultas más frecuentes de la aplicación, la búsqueda de una clasificación concreta, el histórico de un equipo y el histórico de un deporte, detectando en los tres casos un recorrido completo de la tabla o una ordenación adicional en memoria. A partir de ese análisis se diseñaron tres índices secundarios, y se volvió a comprobar con EXPLAIN que cada uno eliminaba efectivamente el problema detectado en su consulta correspondiente, en lugar de asumir que un índice cualquiera mejoraría el rendimiento sin comprobarlo.

7.3 Pruebas funcionales de la aplicación

Cada página y cada combinación relevante de filtros se probó de extremo a extremo contra la aplicación en ejecución, comparando además los valores mostrados con el resultado de la consulta SQL equivalente ejecutada directamente sobre la base de datos, para descartar tanto errores de sintaxis como errores puramente lógicos que no producen ningún mensaje de error pero sí un dato incorrecto en pantalla. Este segundo tipo de comprobación resultó especialmente necesario en el explorador de estadísticas: al combinar en una misma página numerosos bloques de resultado que se muestran u ocultan según la estadística elegida, una comprobación superficial («la página carga sin errores») no habría detectado, por ejemplo, que un resultado igual a cero desaparecía de la pantalla en lugar de mostrarse como «0», o que el resultado de una estadística se filtraba incorrectamente hacia otra.

También se comprobó el comportamiento de la aplicación ante combinaciones incompletas o inexistentes: una clasificación que no existe, un equipo o un deporte que no aparece en los datos, un formulario enviado sin todos los campos obligatorios. Verificando en cada caso que se mostrara un mensaje explicativo en lugar de una página en blanco o un error interno.

Capítulo 8 - Conclusiones y trabajo futuro

Este trabajo ha permitido recuperar y verificar el histórico de 26 temporadas de las competiciones deportivas internas de la UCM, hasta ahora disponible únicamente en papel, y ponerlo a disposición de la comunidad universitaria a través de una aplicación web funcional. El proceso seguido demuestra, en primer lugar, que la digitalización totalmente automática y sin supervisión de un archivo histórico heterogéneo no siempre es la opción más adecuada cuando lo que se busca es exactitud: las tres aproximaciones automáticas exploradas: análisis de texto plano, reconocimiento de documentos con Azure AI Document Intelligence y extracción asistida por modelos de lenguaje sin revisión posterior. Resultaron todas insuficientes frente a la variabilidad de formato de más de treinta años de cuadernillos. Fue finalmente un proceso de transcripción asistida por Claude Code, dirigido y verificado dato a dato por el propio estudiante contra cada documento original, y apoyado además en comprobaciones automáticas de consistencia, el que garantizó la fiabilidad del resultado: la inteligencia artificial resultó útil como herramienta que agilizaba la redacción y la repetición de comprobaciones, pero la fiabilidad final dependió, en última instancia, de que ninguna fila se aceptara sin la verificación explícita del estudiante.

En segundo lugar, el trabajo demuestra que es posible construir una aplicación web de consulta completa, funcional y razonablemente sofisticada, con filtros dependientes, un catálogo de estadísticas parametrizadas y varias vistas de los mismos datos. Sin necesidad de un backend a medida ni de una API propia, apoyándose por completo en SQL y en una herramienta como SQLPage pensada específicamente para este tipo de aplicaciones.

Como trabajo futuro, quedan abiertas varias líneas:

- Adaptar el diseño visual de la aplicación a la identidad gráfica de la Universidad Complutense o de la Oficina de Deportes, de cara a una publicación oficial dentro de ese entorno, tal y como se ha dejado preparado en el apartado 4.6.
- Exponer públicamente la aplicación con un puerto o dominio propio, en lugar de depender de un túnel SSH para su acceso, una vez definido su alojamiento definitivo.

- Ampliar el catálogo de estadísticas del explorador con nuevas preguntas que puedan resultar de interés para la comunidad universitaria.

Introduction

Since the 1990-91 season, Universidad Complutense de Madrid's Sports Office has organised an internal competition between faculties, halls of residence and other university centres: the Rector Trophy league, together with an additional ranking, the Alfonso XIII Trophy, which rewards a centre's joint participation across several sports. Each season's results have traditionally been recorded in a printed booklet, and in recent years also as a scanned PDF, but never loaded into a database or made available online.

Motivation

Over the decades, more than thirty of these booklets have accumulated, containing information that exists only on paper, kept in a drawer at the Sports Office. This thesis is motivated precisely by the idea of recovering that history of Complutense university sport, verifying it with the same rigour one would apply to any historical source, and making it available to the whole university community through a simple web application, instead of leaving it accessible only to whoever has physical access to those booklets.

Goals

The goals of this Bachelor's Thesis are the following:

- Recover and verify the complete historical record of the Rector Trophy and the Alfonso XIII Trophy for which documentation is available, translating it into a single, consistent and queryable data model.
- Study the feasibility of fully or partially automating the extraction of this data from the scanned booklets, using document-recognition and artificial intelligence tools.
- Design a verification process that guarantees the digitized data faithfully matches the original document, documenting any issue or doubt found along the way.
- Design and implement a query web application, reusable against any dataset sharing the same contract, allowing the history to be explored by year, sport, gender, category, team and aggregated statistics.

- Deploy the application using Docker containers, separating the data's lifecycle from the applications, so that either can be updated or replaced independently.
- Leave the system ready so that, at a later stage, its visual design can be adapted to Universidad Complutense's or the Sports Office's visual identity and published in a way that is accessible to the whole university community.

Work plan

The work was organized in two clearly differentiated phases, executed mostly sequentially, with some feedback between them. The first phase focused on the data: considerable time was spent exploring different fully automatic, unsupervised ways of extracting information from the scanned booklets (plain-text parsing, cloud document-recognition services, extraction assisted by language models), evaluating in each case its reliability against the huge variability in format between years and sports. Once none of these paths proved reliable enough for a historical record that had to be correct, not merely plausible, a different transcription and verification process was designed instead: Claude Code, an AI-based programming assistant, was used to draft each season's classifications and to run repetitive checks, while the student personally checked every single value against the original document, year by year, before accepting it, supported by tools built to automatically detect inconsistencies once the data had been entered. The second phase covered the design, implementation and deployment of the query web application: defining the data contract, choosing the technology, building the different pages and features, validating performance, containerizing and deploying to a server.

Conclusions and future work

This thesis has made it possible to recover and verify the historical record of 26 seasons of Universidad Complutense de Madrid's internal sports competitions, until now available only on paper, and to make it available to the university community through a working web application. The process followed shows, first, that fully automatic, unsupervised digitization of a heterogeneous historical archive is not always the best option when accuracy is the goal: the three automatic approaches explored: plain-text parsing, document recognition with Azure AI Document Intelligence, and unsupervised language-model-assisted extraction. All proved insufficient given the format variability across more than thirty years of booklets. It was ultimately a transcription process assisted by Claude Code, directed and verified value by value by the student against every original document, and further supported by automatic consistency checks, that guaranteed the reliability of the result: artificial intelligence proved useful as a tool that sped up drafting and repetitive checking, but the final reliability rested on the fact that no row was ever accepted without the student's explicit verification.

Second, this thesis shows that it is possible to build a complete, functional and reasonably sophisticated query web application, with dependent filters, a catalogue of parameterized statistics and several views over the same data, without a custom backend or a dedicated API, relying entirely on SQL and on a tool such as SQLPage, designed specifically for this kind of application.

Future work

Several lines of future work remain open:

- Adapting the application's visual design to Universidad Complutense's or the Sports Office's visual identity ahead of an official publication;
- Exposing the application publicly with its own port or domain instead of relying on an SSH tunnel; and expanding the statistics explorer's catalogue with new questions that may be of interest to the university community.

BIBLIOGRAFÍA

[1] SQLPage — documentación oficial. https://sql-page.com

[2] Oracle Corporation. MySQL 8.0 Reference Manual. https://dev.mysql.com/doc/

[3] MariaDB Foundation. MariaDB Server Documentation. https://mariadb.com/kb/

[4] Docker Inc. Docker documentation. https://docs.docker.com/

[5] Docker Inc. Docker Compose documentation. https://docs.docker.com/compose/

[6] Microsoft. Azure AI Document Intelligence documentation. https://learn.microsoft.com/azure/ai-services/document-intelligence/

[7] OpenAI. OpenAI API Reference. https://platform.openai.com/docs/

[8] Tabler — free and open-source HTML dashboard template. https://tabler.io/

[9] The pypdf community. pypdf documentation. https://pypdf.readthedocs.io/

[10] Freedesktop.org. Poppler PDF rendering library. https://poppler.freedesktop.org/

APÉNDICES

Apéndice A - Estructura del repositorio y guía de despliegue

Este apéndice resume la organización de los ficheros del proyecto y los pasos necesarios para ejecutar la aplicación, ampliando lo descrito en el capítulo 5.

El código completo del proyecto: los ficheros SQL con los datos verificados, la aplicación web y las imágenes Docker. Está disponible en el siguiente repositorio en línea: https://github.com/GaCasado/TFG_GabrielCasado2026

El proyecto se organiza en tres partes: el conjunto de ficheros SQL con los datos verificados de cada temporada (sql/clasificaciones_AAAA_AA.sql, uno por año), la imagen Docker de datos (clasificaciones-data), que construye una base de datos MariaDB a partir de esos ficheros y del contrato de datos (schema.sql), y la imagen Docker de la aplicación web (clasificaciones-web), con las páginas SQLPage.

Ejecución en local

Basta con tener Docker y Docker Compose instalados. Se levanta primero la imagen de datos (docker compose up -d --build, dentro de clasificaciones-data), y a continuación la aplicación web (mismo comando, dentro de clasificaciones-web), configurando previamente en sendos ficheros .env las credenciales de la base de datos y el puerto en el que se quiere exponer la aplicación.

Despliegue en servidor

El procedimiento es el mismo que en local, sustituyendo las credenciales por unas robustas y generadas específicamente para ese entorno, y activando el modo de producción de SQLPage. Si el motor de base de datos de destino no es compatible con las imágenes oficiales de MySQL, como ocurrió en el despliegue descrito en el apartado 5.4, basta con sustituir la imagen base por una versión de MariaDB compatible, sin tocar el resto de la configuración.

Apéndice B - Resumen de incidencias de normalización de datos

Durante la revisión de los 26 años de datos y la posterior normalización de nomenclatura se documentaron, entre otras, las siguientes incidencias: dos ficheros con nombre heredado que no sigue el patrón estándar del resto; dos casos de duplicación de datos detectados y corregidos; cinco patrones de error sistemáticos, repetidos en varios años, identificados y corregidos de forma homogénea en todos los ficheros afectados; y once casos dudosos puntuales que no se pudieron confirmar al cien por cien contra el documento original, resueltos siempre con el criterio más conservador y documentados tanto en un fichero de incidencias como en la cabecera del propio fichero SQL afectado, para que cualquier decisión tomada durante la digitalización quede trazada.