

FEDERAPRO

PROYECTO: DAM

JOSÉ DAVID CASAS GÓMEZ

DNI: 43388876C

CURSO: 2024/25

**Índice**

* Introducción
* Resumen
* Objetivos
* Logotipo

  El contenido generado por IA puede ser incorrecto.Análisis del contexto
* Estado del arte
* Análisis de requisitos
* Diseño
* Diagramas E/R
* Diagrama de clases
* Diagrama de interfaces
* Diseño de procedimientos o algoritmos
* Diagrama de Gantt
* Definición de recursos y logística necesarios
* Implementación y desarrollo
* Despliegue y mantenimiento
* Pruebas y control de calidad
* Gestión económica
* Valoración personal
* Bibliografía
* Anexos

Introducción

En el ámbito deportivo, la gestión de federaciones, equipos y competiciones es una tarea compleja que involucra el manejo de grandes cantidades de datos, estadísticas, resultados y métricas clave. Tras observar cómo operan las organizaciones deportivas, tanto como aficionado a varios deportes como a través de experiencias profesionales, se ha evidenciado que muchas federaciones y equipos carecen de herramientas eficientes para administrar estos datos de manera centralizada y personalizada.

Actualmente, muchos sistemas de gestión utilizados en deportes populares como el baloncesto o disciplinas altamente especializadas como la Fórmula 1 presentan limitaciones significativas. Estos sistemas suelen ser rígidos, difíciles de personalizar y carecen de flexibilidad para adaptarse a las necesidades específicas de cada deporte. Además, la integración entre las distintas partes del proceso (gestión de equipos, jugadores, estadísticas, partidos, etc.) suele ser deficiente, lo que puede llevar a una gestión ineficiente y a la toma de decisiones poco fundamentada.

**FederaPro** nace como una solución innovadora para resolver estos problemas. Se trata de un **gestor de federaciones deportivas** que permite gestionar de manera personalizada los aspectos clave de diferentes deportes desde una única plataforma. **FederaPro** permitirá a los usuarios seleccionar el deporte con el que desean trabajar y adaptará automáticamente la base de datos, las consultas y las interfaces a las necesidades de ese deporte en particular.

El principal reto de este proyecto es crear un sistema que sea, no solo funcional y fácil de usar, sino también lo suficientemente flexible para adaptarse a nuevos deportes en el futuro, sin necesidad de reestructurar el sistema completo. Con ello, **FederaPro** busca facilitar la gestión y optimización de procesos dentro de las federaciones y equipos deportivos, mejorando la accesibilidad y toma de decisiones mediante la centralización de datos de manera eficiente.

Resumen

**FederaPro** tiene como objetivo crear un sistema de gestión integral y flexible para federaciones deportivas, permitiendo la administración personalizada de datos relacionados con diferentes deportes. La plataforma se adapta dinámicamente a las necesidades específicas de cada disciplina, generando automáticamente bases de datos, interfaces y consultas adaptadas según el deporte elegido por el usuario.

El sistema proporcionará una herramienta centralizada para gestionar equipos, estadísticas, resultados y otros aspectos clave, de manera eficiente y organizada. Además, **FederaPro** será escalable, permitiendo incorporar nuevos deportes y características a medida que sea necesario, sin complicaciones significativas en su estructura.

**Palabras clave:** Gestión deportiva, federaciones, personalización, base de datos, estadísticas, plataformas deportivas, flexibilidad, optimización, escalabilidad.

Objetivos

Objetivo General:

Desarrollar **FederaPro**, un sistema de gestión dinámica y personalizada para federaciones deportivas, que permita a los administradores gestionar equipos, jugadores, estadísticas y resultados, adaptándose a las necesidades específicas de cada deporte seleccionado.

Objetivos Específicos:

1. **Personalización de la base de datos según el deporte seleccionado:** Crear una base de datos flexible que se ajuste automáticamente a las características de cada deporte, permitiendo la inclusión de nuevos deportes sin complicaciones.
2. **Generación de interfaces dinámicas:** Desarrollar interfaces adaptativas que presenten la información de manera clara e intuitiva, ajustándose a las necesidades de cada deporte y facilitando la interacción con los datos.
3. **Gestión de equipos y participantes:** Permitir la creación, edición y eliminación de equipos, así como la asignación de estadísticas y resultados relevantes, sin importar el tipo de deporte.
4. **Gestión de competiciones y eventos:** Implementar funcionalidades que permitan registrar y gestionar los eventos deportivos (como partidos, competiciones, etc.), incluyendo los resultados y las estadísticas asociadas.
5. **Consultas personalizadas de estadísticas:** Desarrollar herramientas que permitan consultar estadísticas específicas de cada deporte, como métricas de rendimiento, resultados históricos y otras variables importantes.
6. **Escalabilidad del sistema:** Asegurar que **FederaPro** sea escalable y flexible, permitiendo la incorporación de nuevos deportes y características adicionales sin necesidad de reestructurar la plataforma.
7. **Optimización de la experiencia del usuario:** Diseñar una interfaz de usuario fluida e intuitiva, tanto en el frontend como en la interacción con el backend, garantizando que los administradores puedan acceder y gestionar la información de manera rápida y eficiente.

Análisis del contexto

El deporte es una de las industrias más relevantes a nivel global, con un impacto social, económico y cultural significativo. La gestión eficiente de equipos, jugadores, competiciones y estadísticas se ha convertido en una necesidad clave para federaciones deportivas de todos los niveles.

Contexto Tecnológico

El avance tecnológico ha impactado significativamente en la gestión deportiva. Actualmente, las soluciones más eficaces se centran en la automatización de procesos administrativos, la organización eficiente de datos y la generación de informes personalizados para la toma de decisiones.

Tecnologías como los sistemas de gestión basados en la nube están ganando popularidad gracias a su capacidad para almacenar grandes volúmenes de información y permitir el acceso remoto. Asimismo, las herramientas de análisis de datos personalizables están permitiendo que las federaciones deportivas obtengan informes detallados que se adapten a las particularidades de cada deporte. La flexibilidad en la generación de estadísticas y en la visualización de datos se ha convertido en un elemento clave para optimizar la gestión interna de equipos y competiciones.

Contexto Económico

La inversión en tecnología deportiva ha crecido de forma constante en los últimos años. Según datos de la consultora SportsTechX, el mercado global de tecnología deportiva alcanzó los 30.000 millones de dólares en 2024 y se proyecta que siga creciendo a una tasa anual del 10%.

Contexto Sociocultural

El creciente interés por el deporte amateur y profesional ha impulsado la demanda de soluciones digitales que faciliten la gestión de federaciones y equipos. Federaciones pequeñas y medianas, que no cuentan con los recursos de organizaciones deportivas mayores, buscan herramientas más accesibles y personalizables para gestionar sus actividades.

Contexto Legislativo

La gestión de datos personales en el ámbito deportivo está sujeta a diversas normativas legales, especialmente en lo que respecta a la protección de la información de jugadores, entrenadores y federaciones. **FederaPro** se asegurará de cumplir con el **Reglamento General de Protección de Datos (RGPD)**, garantizando que la información sensible esté debidamente protegida. Además, se implementarán medidas de seguridad adicionales para prevenir el acceso no autorizado y cumplir con estándares de seguridad informática. En caso de que se manejen transacciones financieras, se adoptarán las directrices del estándar **PCI DSS**.

Estado del arte

Actualmente, existen diversas plataformas para la gestión deportiva, entre las que destacan:

* **MyTeam11**: Plataforma enfocada en la gestión de equipos y competiciones, con funcionalidades limitadas para la personalización según el deporte.
* **TeamSnap:** Herramienta popular para la gestión de equipos amateurs, pero con poca flexibilidad para deportes profesionales o con reglas complejas.
* **SportEasy:** Plataforma con amplias funciones para la gestión de equipos, pero que requiere configuraciones manuales complejas para adaptar los datos a cada deporte.
* **Playwaze:** Sistema especializado en la organización de eventos deportivos y competiciones, destacando por su interfaz clara y adaptable, pero limitado en la gestión avanzada de estadísticas personalizadas.
* **LeagueRepublic:** Plataforma centrada en la gestión de ligas y torneos, eficaz para competiciones específicas, pero poco versátil para deportes con características complejas.

Análisis de la Competencia

|  |  |  |  |
| --- | --- | --- | --- |
| Plataforma | Fortalezas | Debilidades | Precio |
| **MyTeam11** | Buena interfaz y gestión básica | Poca adaptabilidad a deportes menos comunes | 15€/mes |
| **TeamSnap** | Uso intuitivo y funciones sociales | No permite una personalización completa para cada deporte | 12€/mes |
| **SportEasy** | Amplias funciones de gestión y planificación | Configuración compleja para nuevos deportes | 20€/mes |
| **Playwaze** | Gestión eficaz de eventos deportivos | Escasa flexibilidad para personalización avanzada de datos | 10€/mes |
| **LeagueRepublic** | Gestión avanzada de ligas y torneos | Limitado para deportes fuera del ámbito competitivo tradicional | 18€/mes |

Innovación

**FederaPro** introduce una serie de innovaciones clave que lo distinguen de otras plataformas existentes:

* **Personalización Dinámica**: La capacidad de adaptar automáticamente las bases de datos, interfaces y consultas a las necesidades específicas de cada deporte, eliminando la necesidad de configuraciones manuales complejas.
* **Escalabilidad Flexible**: El diseño modular de FederaPro permite incorporar nuevos deportes y funcionalidades sin realizar cambios drásticos en la arquitectura del sistema.
* **Automatización de Procesos Administrativos**: FederaPro optimiza tareas como la programación de partidos, la asignación de árbitros o el cálculo automático de clasificaciones y estadísticas relevantes.
* **Generación de Informes Personalizados**: La plataforma permitirá obtener informes específicos para cada deporte, destacando las métricas más relevantes según sus características particulares.

Análisis DAFO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Aspectos | Fortalezas | Debilidades | Oportunidades | Amenazas |
| **Internos** | Personalización dinámica, interfaz intuitiva y escalabilidad. | Requiere una fase inicial de desarrollo más compleja. | Desarrollo de funcionalidades avanzadas para deportes en crecimiento. | Dependencia del mantenimiento constante de nuevas disciplinas. |
| **Externos** | Ausencia de plataformas que ofrezcan personalización completa en deportes diversos. | Alta competencia en el sector de software de gestión deportiva. | Expansión a mercados internacionales con necesidades específicas. | Cambios tecnológicos que exijan adaptaciones rápidas |

Análisis de requisitos

Requisitos Funcionales

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Número** | **Nombre del requisito** | **Tipo** | **Descripción** | **Prioridad** | **Objetivos relacionados** |
| RF 1 | Gestión de equipos y jugadores | Funcional | Permitir la creación, edición y eliminación de equipos, así como la asignación de estadísticas y resultados relevantes. | Alta | 1, 3 |
| RF 2 | Gestión de competiciones y eventos | Funcional | Implementar funcionalidades que permitan registrar y gestionar los eventos deportivos, incluyendo los resultados y las estadísticas asociadas. | Alta | 1, 4 |
| RF 3 | Personalización dinámica de datos | Funcional | Permitir que la base de datos y las interfaces se adapten automáticamente según el deporte seleccionado. | Alta | 1, 2, 6 |
| RF 4 | Visualización de datos móviles | Funcional | La aplicación móvil permitirá la consulta de datos clave, como estadísticas de jugadores y resultados en tiempo real. | Media | 2, 5 |
| RF 5 | Interfaz intuitiva en la app de escritorio | Funcional | Desarrollar una interfaz que permita gestionar de forma eficiente equipos, jugadores y competiciones, facilitando la inserción, modificación y eliminación de datos. | Alta | 2, 7 |

Requisitos No Funcionales

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Número** | **Nombre del requisito** | **Tipo** | **Descripción** | **Prioridad** | **Objetivos relacionados** |
| RNF 1 | Compatibilidad multiplataforma | No funcional | El sistema deberá ser accesible desde distintos dispositivos y sistemas operativos. | Media | 6 |
| RNF 2 | Seguridad y protección de datos | No funcional | Implementar medidas de seguridad para proteger la información confidencial de los usuarios y federaciones. | Alta | 7 |
| RNF 3 | Tiempo de respuesta optimizado | No funcional | El sistema deberá ofrecer tiempos de respuesta inferiores a 2 segundos en la consulta de datos. | Media | 7 |
| RNF 4 | Escalabilidad del sistema | No funcional | El sistema deberá permitir la integración de nuevos deportes y funcionalidades sin necesidad de reestructuraciones drásticas. | Alta | 6 |

Diseño

El sistema ***FederaPro*** está dividido en tres capas principales, siguiendo una arquitectura modular basada en servicios. Esto permite una separación clara de responsabilidades y facilita el mantenimiento, escalabilidad y evolución del sistema.

1. **API Backend (Java + Spring Boot)**
   * Se encarga de la lógica de negocio y de exponer los datos mediante una API RESTful.
   * Gestiona la base de datos.
   * Provee datos estructurados (JSON) al resto de capas.
   * Autenticación, validación y seguridad centralizada.
2. **Aplicación de Escritorio (C# + Windows Forms)**
   * Aplicación principal de gestión.
   * Permite la administración completa de la información (creación, modificación, eliminación de datos).
   * Consume la API para interactuar con la base de datos.
3. **Aplicación Móvil (Kotlin)**
   * Orientada a la consulta y visualización de datos (partidos, estadísticas, clasificaciones, etc.).
   * Accede a la API para obtener la información en tiempo real o bajo demanda.
   * Diseño adaptado a usuarios como jugadores, entrenadores o aficionados.
4. **Base de Datos Relacional**
   * Contiene todas las entidades necesarias para las diferentes disciplinas deportivas, como equipos, jugadores, competiciones, estadísticas, árbitros, etc.
   * Diseñada para adaptarse dinámicamente al deporte configurado.

API

BBDD

Seguridad

Diagramas E/R

En el diseño del sistema **FederaPro**, la configuración y personalización de la base de datos son fundamentales para adaptarse a los distintos deportes gestionados por la plataforma.

El siguiente conjunto de entidades, **Deporte**, **Configuraciontabla** y **Usuario**, son esenciales para gestionar cómo se estructuran los datos y cómo se manejan los roles dentro del sistema.

**1. Deporte**

* **Campos**: idDeporte (PK), nombre, descripcion
* **Descripción**: Esta entidad almacena los distintos deportes disponibles en el sistema, como baloncesto, fútbol, etc. Permite que el sistema sea flexible y adaptable a distintos tipos de deportes, lo que es esencial para la personalización de la gestión según las características de cada uno.
* **Relación**: 1:N con la entidad Configuraciontabla, ya que cada deporte puede tener múltiples configuraciones de tabla asociadas (campos, tipos de datos, etc.).

**2. Configuraciontabla**

* **Campos**: idConfiguracion (PK), deporte\_id (FK), campo, tipoDato, obligatorio, orden
* **Descripción**: Esta entidad gestiona la configuración dinámica de los campos necesarios para cada deporte. La configuración incluye el nombre del campo, el tipo de dato (por ejemplo, texto, número, etc.), si el campo es obligatorio y el orden en el que deben aparecer.
* **Relación**: N:1 con la entidad Deporte, ya que cada configuración de tabla está asociada a un deporte específico.

**3. Usuario**

* **Campos**: idUsuario (PK), nombre, email, contraseña, rol
* **Descripción**: La entidad Usuario gestiona la información de los usuarios del sistema. Cada usuario tiene un rol específico (administrador, visualizador, etc.) que determina el nivel de acceso y las funcionalidades disponibles dentro del sistema.
* **Relación**: No tiene relaciones explícitas con otras entidades, pero es una parte clave del control de acceso al sistema.

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama Entidad-Relación del Módulo de Baloncesto

El siguiente diagrama Entidad-Relación (E/R) representa la estructura lógica de la base de datos diseñada para el módulo de baloncesto del sistema **FederaPro**. Este módulo tiene como objetivo gestionar toda la información relevante relacionada con equipos, jugadores, competiciones, partidos y estadísticas, permitiendo un almacenamiento organizado y eficiente de los datos.

**1. Basket\_Equipos**

* Campos: id\_equipo (PK), nombre, ciudad, fundacion, estadio
* Descripción: Esta entidad representa a los equipos de baloncesto registrados en el sistema. Incluye información relevante como el nombre del equipo, su ciudad de origen, el año de fundación y el estadio donde juega. Cada equipo puede estar asociado a varios jugadores, un entrenador y puede participar en múltiples competiciones.
* Relaciones:
  + 1:N con Basket\_Jugadores
  + 1:N con Basket\_Entrenadores
  + N:M con Basket\_Competiciones (mediante basket\_equipos\_competiciones)
  + 1:N con Basket\_Partidos (como equipo local o visitante)
  + 1:N con Basket\_Estadisticas\_Equipo

**2. Basket\_Jugadores**

* Campos: id\_jugador (PK), id\_equipo (FK), nombre, altura, peso, posicion, dorsal
* Descripción: Almacena los datos de los jugadores de baloncesto, incluyendo sus características físicas y posición en la cancha. Cada jugador está asociado a un equipo y puede participar en múltiples competiciones y partidos.
* Relaciones:
  + N:1 con Basket\_Equipos
  + 1:N con Basket\_Estadisticas\_Jugador
  + N:M con Basket\_Competiciones (mediante basket\_jugador\_competicion)

**3. Basket\_Entrenadores**

* Campos: id\_entrenador (PK), id\_equipo (FK), nombre, experiencia
* Descripción: Representa a los entrenadores de los equipos. Cada entrenador está vinculado a un equipo y se registra su nombre y años de experiencia.
* Relaciones:
  + N:1 con Basket\_Equipos

**4. Basket\_Arbitros**

* Campos: id\_arbitro (PK), nombre, experiencia
* Descripción: Contiene los datos de los árbitros que participan en los partidos. Incluye su nombre y años de experiencia en el arbitraje.
* Relaciones:
  + 1:N con Basket\_Partidos

**5. Basket\_Competiciones**

* Campos: id\_competicion (PK), nombre, tipo
* Descripción: Gestiona la información de las distintas competiciones de baloncesto, que pueden ser de tipo Liga, Copa o Amistoso. Estas competiciones agrupan múltiples partidos y pueden tener distintos equipos y jugadores participantes.
* Relaciones:
  + 1:N con Basket\_Partidos
  + N:M con Basket\_Equipos (mediante basket\_equipos\_competiciones)
  + N:M con Basket\_Jugadores (mediante basket\_jugador\_competicion)

**6. Basket\_Partidos**

* Campos: id\_partido (PK), id\_competicion (FK), equipo\_local (FK), equipo\_visitante (FK), fecha, resultado\_local, resultado\_visitante, id\_arbitro (FK)
* Descripción: Esta tabla representa los encuentros oficiales disputados entre dos equipos en el marco de una competición. Se registran la fecha, los resultados y el árbitro encargado.
* Relaciones:
  + N:1 con Basket\_Competiciones
  + N:1 con Basket\_Equipos (dos veces: equipo local y visitante)
  + N:1 con Basket\_Arbitros
  + 1:N con Basket\_Estadisticas\_Equipo
  + 1:N con Basket\_Estadisticas\_Jugador

**7. Basket\_Estadisticas\_Equipo**

* Campos: id\_estadistica (PK), id\_partido (FK), id\_equipo (FK), puntos, rebotes, asistencias, robos, tapones, perdidas, faltas
* Descripción: Registra las estadísticas colectivas de un equipo en un partido específico. Permite analizar el rendimiento del equipo durante un encuentro.
* Relaciones:
  + N:1 con Basket\_Partidos
  + N:1 con Basket\_Equipos

**8. Basket\_Estadisticas\_Jugador**

* Campos: id\_estadistica (PK), id\_partido (FK), id\_jugador (FK), puntos, rebotes, asistencias, robos, tapones, perdidas, faltas, minutos\_jugados
* Descripción: Contiene las estadísticas individuales de cada jugador durante un partido. Se utilizan para evaluar su desempeño en aspectos clave del juego.
* Relaciones:
  + N:1 con Basket\_Partidos
  + N:1 con Basket\_Jugadores

**9. basket\_equipos\_competiciones**

* Campos: id\_equipo (PK, FK), id\_competicion (PK, FK)
* Descripción: Tabla intermedia que gestiona la relación muchos a muchos entre equipos y competiciones. Permite saber en qué competiciones participa cada equipo.
* Relaciones:
  + N:1 con Basket\_Equipos
  + N:1 con Basket\_Competiciones

**10. basket\_jugador\_competicion**

* Campos: id (PK), id\_jugador (FK), id\_competicion (FK)
* Descripción: Tabla intermedia que representa la participación de un jugador en una competición específica. Facilita la trazabilidad y análisis de los jugadores dentro de torneos concretos.
* **Relaciones:**
  + N:1 con Basket\_Jugadores
  + N:1 con Basket\_Competiciones

Diagrama, Gráfico de cajas y bigotes

El contenido generado por IA puede ser incorrecto.

Diagrama Entidad-Relación del Módulo de Karting

El siguiente diagrama Entidad-Relación (E/R) describe la estructura lógica de la base de datos diseñada específicamente para el módulo de **karting** dentro del sistema **FederaPro**. Este módulo está orientado a gestionar de forma integral toda la información relevante relacionada con los equipos, pilotos, competiciones, circuitos, carreras y estadísticas que forman parte de este deporte.

**1. Karting\_Equipos**

* **Campos:** id\_equipo (PK), nombre, pais, sponsor
* **Descripción:** Representa a los equipos de karting. Un equipo puede tener varios pilotos y estadísticas por carrera.
* **Relaciones:**
  + 1:N con Karting\_Pilotos (un equipo tiene varios pilotos)
  + 1:N con Karting\_Estadisticas\_Equipo (un equipo participa en varias carreras)

**2. Karting\_Pilotos**

* **Campos:** id\_piloto (PK), id\_equipo (FK), nombre, categoria, fecha\_nacimiento, nacionalidad, numero\_kart
* **Descripción:** Contiene los pilotos registrados, clasificados por categoría (Junior, Senior, Master).
* **Relaciones:**
  + N:1 con Karting\_Equipos (cada piloto pertenece a un equipo)
  + 1:N con Karting\_Estadisticas\_Piloto (un piloto puede participar en muchas carreras)

**3. Karting\_Circuitos**

* **Campos:** id\_circuito (PK), nombre, ubicacion, longitud, pais
* **Descripción:** Define los circuitos donde se realizan las carreras.
* **Relaciones:**
  + 1:N con Karting\_Carreras (un circuito puede ser sede de varias carreras)

**4. Karting\_Competiciones**

* **Campos:** id\_competicion (PK), nombre, categoria, tipo, fecha\_inicio, fecha\_fin
* **Descripción:** Representa torneos o competiciones en distintas escalas (Mundial, Nacional, etc.).
* **Relaciones:**
  + 1:N con Karting\_Carreras (una competición incluye varias carreras)

**5. Karting\_Carreras**

* **Campos:** id\_carrera (PK), id\_competicion (FK), id\_circuito (FK), fecha
* **Descripción:** Cada carrera está asociada a una competición y se lleva a cabo en un circuito específico.
* **Relaciones:**
  + N:1 con Karting\_Competiciones
  + N:1 con Karting\_Circuitos
  + 1:N con Karting\_Estadisticas\_Piloto (una carrera registra estadísticas por piloto)
  + 1:N con Karting\_Estadisticas\_Equipo (una carrera registra estadísticas por equipo)

**6. Karting\_Estadisticas\_Piloto**

* **Campos:** id\_estadistica (PK), id\_carrera (FK), id\_piloto (FK), posicion, tiempo\_total, vueltas
* **Descripción:** Registra el desempeño individual de cada piloto en cada carrera.
* **Relaciones:**
  + N:1 con Karting\_Carreras
  + N:1 con Karting\_Pilotos

**7. Karting\_Estadisticas\_Equipo**

* **Campos:** id\_estadistica (PK), id\_carrera (FK), id\_equipo (FK), tiempo\_promedio, puntos
* **Descripción:** Registra el rendimiento global del equipo por carrera (tiempo promedio, puntos).
* **Relaciones:**
  + N:1 con Karting\_Carreras
  + N:1 con Karting\_Equipos

Diagrama

El contenido generado por IA puede ser incorrecto.

Diagrama de clases

El diagrama de clases es una representación visual de las entidades que componen el sistema, sus atributos y las relaciones entre ellas. En el contexto de **FederaPro**, el sistema de gestión para federaciones deportivas, el diagrama de clases describe las estructuras de datos fundamentales que gestionan la información de equipos, jugadores, competiciones, estadísticas y otros componentes clave dentro de cada deporte, permitiendo así una representación coherente de la lógica del sistema.

Diagrama de configuración

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

**Configuraciontabla**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| id | Integer | Identificador único de la configuración de la tabla. |
| consultaCreacion | String (Lob) | Contiene la consulta SQL de creación de la tabla configurada. |
| idDeporte | Deporte | Relación con el deporte al que pertenece la configuración (FK). |

**Deporte**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| id | Integer | Identificador único del deporte. |
| idNemonico | String (max 10) | Código o identificador único del deporte. |
| nombre | String (max 100) | Nombre del deporte (ej. baloncesto, fútbol). |

**Usuario**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| id | Integer | Identificador único del usuario. |
| nombre | String (max 100) | Nombre del usuario. |
| correo | String (max 150) | Correo electrónico del usuario. |
| contraseña | String (max 255) | Contraseña del usuario (encriptada). |
| rol | Rol | Rol del usuario en el sistema (ej. "usuario", "administrador"). |
| fechaRegistro | Instant | Fecha de registro del usuario en el sistema. |

Texto

El contenido generado por IA puede ser incorrecto.Diagrama de Baloncesto

**BasketEquipo**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| idEquipo | int | Identificador único del equipo. |
| nombre | String | Nombre del equipo. |
| ciudad | String | Ciudad del equipo. |
| fundacion | int | Año de fundación del equipo. |
| estadio | String | Nombre del estadio del equipo. |

**BasketJugador**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| idJugador | int | Identificador único del jugador. |
| idEquipo | int | Identificador del equipo al que pertenece. |
| nombre | String | Nombre del jugador. |
| altura | double | Altura del jugador en metros. |
| peso | double | Peso del jugador en kilogramos. |
| posicion | String | Posición del jugador (Enum de 'Base', 'Escolta', etc.) |
| dorsal | int | Dorsal del jugador. |

**BasketEntrenador**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| idEntrenador | int | Identificador único del entrenador. |
| idEquipo | int | Identificador del equipo que dirige. |
| nombre | String | Nombre del entrenador. |
| experiencia | int | Años de experiencia del entrenador. |

**BasketArbitro**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| idArbitro | int | Identificador único del árbitro. |
| nombre | String | Nombre del árbitro. |
| experiencia | int | Años de experiencia del árbitro. |

**BasketCompeticion**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| idCompeticion | int | Identificador único de la competición. |
| nombre | String | Nombre de la competición. |
| tipo | String | Tipo de competición (Enum de 'Liga', 'Copa', 'Amistoso') |

**BasketPartido**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| idPartido | int | Identificador único del partido. |
| idCompeticion | int | Identificador de la competición. |
| equipoLocal | int | Identificador del equipo local. |
| equipoVisitante | int | Identificador del equipo visitante. |
| fecha | Date | Fecha y hora del partido. |
| resultadoLocal | int | Resultado del equipo local. |
| resultadoVisitante | int | Resultado del equipo visitante. |
| idArbitro | int | Identificador del árbitro del partido. |

**BasketEstadisticaEquipo**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| idEstadistica | int | Identificador único de la estadística. |
| idPartido | int | Identificador del partido. |
| idEquipo | int | Identificador del equipo. |
| puntos | int | Puntos obtenidos por el equipo. |
| rebotes | int | Rebotes obtenidos por el equipo. |
| asistencias | int | Asistencias realizadas por el equipo. |
| robos | int | Robos de balón realizados por el equipo. |
| tapones | int | Tapones realizados por el equipo. |
| perdidas | int | Pérdidas de balón del equipo. |
| faltas | int | Faltas cometidas por el equipo. |

**BasketEstadisticaJugador**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| idEstadistica | int | Identificador único de la estadística. |
| idPartido | int | Identificador del partido. |
| idJugador | int | Identificador del jugador. |
| puntos | int | Puntos obtenidos por el jugador. |
| rebotes | int | Rebotes obtenidos por el jugador. |
| asistencias | int | Asistencias realizadas por el jugador. |
| robos | int | Robos de balón realizados por el jugador. |
| tapones | int | Tapones realizados por el jugador. |
| perdidas | int | Pérdidas de balón del jugador. |
| faltas | int | Faltas cometidas por el jugador. |
| minutosJugados | int | Minutos jugados por el jugador. |

**BasketEquipoCompeticion**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| idEquipo | int | Identificador del equipo. |
| idCompeticion | int | Identificador de la competición. |

**BasketJugadorCompeticion**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| idJugador | int | Identificador del jugador. |
| idCompeticion | int | Identificador de la competición. |

**Texto

El contenido generado por IA puede ser incorrecto.**Diagrama de Karting

**Karting\_Equipos**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| id\_equipo | Integer | Identificador único del equipo. |
| nombre | String (max 100) | Nombre del equipo. |
| pais | String (max 100) | País del equipo. |
| sponsor | String (max 100) | Nombre del patrocinador del equipo. |

**Karting\_Pilotos**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| id\_piloto | Integer | Identificador único del piloto. |
| id\_equipo | Integer (FK) | Relación con el equipo al que pertenece el piloto. |
| nombre | String (max 100) | Nombre del piloto. |
| categoria | Enum ("Junior", "Senior", "Master") | Categoría del piloto en la competición. |
| fecha\_nacimiento | Date | Fecha de nacimiento del piloto. |
| nacionalidad | String (max 50) | Nacionalidad del piloto. |
| numero\_kart | Integer | Número del kart asignado al piloto. |

**Karting\_Circuitos**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| id\_circuito | Integer | Identificador único del circuito. |
| nombre | String (max 100) | Nombre del circuito. |
| ubicacion | String (max 100) | Ubicación geográfica del circuito. |
| longitud | Decimal (5,2) | Longitud total del circuito en kilómetros. |
| pais | String (max 50) | País donde se encuentra el circuito. |

**Karting\_Competiciones**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| id\_competicion | Integer | Identificador único de la competición. |
| nombre | String (max 100) | Nombre de la competición. |
| categoria | Enum ("Junior", "Senior", "Master") | Categoría de la competición. |
| tipo | Enum ("Mundial", "Europeo", "Nacional", "Regional") | Tipo de competición. |
| fecha\_inicio | Date | Fecha de inicio de la competición. |
| fecha\_fin | Date | Fecha de finalización de la competición. |

**Karting\_Carreras**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| id\_carrera | Integer | Identificador único de la carrera. |
| id\_competicion | Integer (FK) | Relación con la competición en la que se celebra la carrera. |
| id\_circuito | Integer (FK) | Relación con el circuito donde se lleva a cabo la carrera. |
| fecha | DateTime | Fecha y hora de la carrera. |

**Karting\_Estadisticas\_Piloto**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| id\_estadistica | Integer | Identificador único de las estadísticas del piloto. |
| id\_carrera | Integer (FK) | Relación con la carrera en la que se registran las estadísticas. |
| id\_piloto | Integer (FK) | Relación con el piloto que realizó las estadísticas. |
| posicion | Integer | Posición final del piloto en la carrera. |
| tiempo\_total | Decimal (10,2) | Tiempo total que el piloto tardó en completar la carrera. |
| vueltas | Integer | Número de vueltas completadas por el piloto. |

**Karting\_Estadisticas\_Equipo**

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descripción** |
| id\_estadistica | Integer | Identificador único de las estadísticas del equipo. |
| id\_carrera | Integer (FK) | Relación con la carrera en la que se registran las estadísticas. |
| id\_equipo | Integer (FK) | Relación con el equipo que realizó las estadísticas. |
| tiempo\_promedio | Decimal (10,2) | Tiempo promedio de las vueltas del equipo. |
| puntos | Integer (default 0) | Puntos obtenidos por el equipo en la carrera. |

Diagrama de interfaces

Aplicación de escritorio

* + 1. Selección de deporte

Logotipo

El contenido generado por IA puede ser incorrecto.

GOLF

KARTING

BALONCESTO

FÚTBOL

TENIS

VOLLEYBALL

Esta pantalla es la muestra de deportes para elegir la federación que se quiere gestionar. Al seleccionar cualquiera de los deportes entrara en una ventana de logging.

* + 1. Registro y Logging

Logotipo

El contenido generado por IA puede ser incorrecto.Logotipo

El contenido generado por IA puede ser incorrecto.

FORMULARIO DE REGISTRO

USUARIO

CONTRASEÑA

ENTRAR

REGISTRO

VOLVER

CONFIRMAR

* + 1. Visualizacion

CSV

OPCIÓN 1

* 1.1

OPCIÓN 2

* 2.1
* 2.2

OPCIÓN 3

OPCIÓN 4

TABLAS DE VISUALIZACION

En este caso es una explicación más general en la cual tendremos una serie de información en cada opción según sus tablas, de cualquier deporte en este caso con sus respectivos añadir, visualizar, editar y eliminar. Además, incorporará un botón para añadir un csv correspondiente a su opción en cuestión.

Dispondremos de un menú lateral por el cual navegar entre las distintas opciones.

* + 1. Visualización del detalle

DATOS DEL EQUIPO

JUGADORES

ESTADISTICAS

PARTIDOS

TABLA DE INFORMACIÓN

Partiendo de la homogeneidad por la cual se debe regir la aplicación, deberemos mostrar en el detalle cada dato de la aplicación, en este caso, mostrare uno complejo como puede ser el equipo. Visualizará sus datos y en la tabla se podrá escoger entre pestañas para mostrar la información que el usuario quiera.

Aplicación móvil

Logotipo

El contenido generado por IA puede ser incorrecto.

Logotipo

El contenido generado por IA puede ser incorrecto.

USUARIO

CONTRASEÑA

ENTRAR

TABLA DE INFOMACIÓN

Partiremos desde un logging sencillo directamente a una ventana de gráficos e información muy visual, que llame la atención y sea sencilla de mostrar junto con un botón en el menú lateral para ir cambiando entre visualizaciones, como podrían ser futuros partidos o clasificaciones e información básica de jugadores y equipos.

TABLA DE CLASIFICACIÓN

Diseño de procedimientos o algoritmos

Gestión de Datos generales

IMPORTAR CSV

MOSTRAR EQUIPOS

BORRAR

SELECCIÓN

EDITAR

VISUALIZAR

CREAR

En el caso de FederaPro, esta se rige por la homogeneidad en todos los procesos de visualización en tablas y creación, edición, lectura o eliminación de datos, promoviendo así su sencilla usabilidad.

Diagrama de Gantt

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tareas |  |  | | Semana 1 | | | | | | | Semana 2 | | | | | | |
| Horas | | 21 | | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 1 | 2 | 3 | 4 |
| Creación bbdd | 1 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación tablas Baloncesto | 1 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación tablas Karting | 1 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación de la api | 1 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Aplicar métodos de seguridad | 2 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementar swagger | 1 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación modelos  Baloncesto | 1 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación Repositorios  Baloncesto | 1 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación Servicios  Baloncesto | 2 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación Controladores Baloncesto | 2 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación modelos  Karting | 1 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación Repositorios  Karting | 1 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación Servicios  Karting | 2 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación Controladores Karting | 2 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Documentación Diseño y Planificación | 15 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación del proyecto móvil | 2 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación de escritorio | 2 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Aplicar Seguridad | 4 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tareas |  |  | | Semana 3 | | | | | | | Semana 4 | | | | | | |
| Horas | | 5 | | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| Pantalla escritorio datos baloncesto | 8 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pantalla formularios Baloncesto | 8 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pantalla escritorio datos Karting | 10 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pantalla formularios Karting | 6 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pantallas Datos Baloncesto Movil | 10 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pantallas datos Karting Movil | 10 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementación métodos csv escritorio | 5 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación de métodos clasificatorios api | 4 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementación de métodos clasificatorios | 5 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tareas |  |  | | Semana 5 | | | | | | | Semana 6 | | | | | | |
| Horas | | 19 | | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 1 |
| Creación de CSS personalizados | 6 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Pruebas | 30 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Creación Instalador | 4 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Manual de usuario | 6 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Markdown | 5 | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |

Definición de recursos y logística necesarios

Una vez definidas las tareas del proyecto **FederaPro**, es necesario detallar los recursos requeridos para su ejecución. Estos recursos se agrupan en tres categorías principales: materiales, humanos y logísticos.

Recursos materiales

Para el desarrollo del sistema de gestión para federaciones deportivas *FederaPro*, se necesitarán los siguientes recursos tecnológicos:

* **Equipamiento informático**:
  + Ordenadores personales o portátiles con al menos: procesador Intel i5 o superior, 16 GB de memoria RAM y unidades SSD para un rendimiento óptimo.
  + Dispositivos móviles Android para pruebas funcionales de la aplicación móvil.
  + Posibilidad de contar con servidores de desarrollo y pruebas, tanto locales como en la nube (por ejemplo, AWS o DigitalOcean).
* **Herramientas de desarrollo y software**:
  + **Visual Studio**: para el desarrollo de la aplicación de escritorio en **C#**.
  + **IntelliJ IDEA**: entorno de desarrollo para el backend en **Java** con el framework **Spring**.
  + **Android Studio**: plataforma utilizada para desarrollar la aplicación móvil en **Kotlin**.
  + **MySQL**: sistema de gestión de base de datos relacional para almacenar toda la información estructurada del sistema.
  + **DBeaver**: herramienta gráfica para la gestión y diseño de la base de datos.
  + **Git** (con GitHub o GitLab): para el control de versiones y trabajo colaborativo.
  + Herramientas de diseño de diagramas como **Lucidchart** o **Draw.io** para la representación de la arquitectura del sistema, base de datos y clases.

Recursos humanos

El desarrollo de **FederaPro** requiere un equipo técnico con diversos perfiles profesionales:

* **Desarrollador backend (Java/Spring)**: responsable de la lógica del servidor, la implementación de servicios REST y la conexión con la base de datos.
* **Desarrollador frontend de escritorio (C#)**: encargado de implementar la interfaz de usuario para el software de gestión de escritorio.
* **Desarrollador móvil (Kotlin)**: desarrolla la versión de la aplicación para dispositivos Android.
* **Diseñador UX/UI**: diseña la experiencia de usuario y las interfaces de cada plataforma.
* **Tester o responsable de calidad (QA)**: verifica el correcto funcionamiento de todas las funcionalidades implementadas.
* **Responsable de base de datos**: diseña el esquema relacional y se encarga de la optimización y mantenimiento de la base de datos.
* **Coordinador del proyecto**: encargado de la planificación de tareas, gestión del equipo y control de tiempos.
* **Asesor funcional (opcional)**: experto en federaciones deportivas que guía el desarrollo funcional para asegurar que el sistema se adapte a la realidad de cada deporte.

Logística

* **Permisos y licencias**:
  + Licencias de uso de los entornos de desarrollo empleados, en caso de que se utilicen versiones profesionales o comerciales.
  + Cumplimiento de la normativa de protección de datos (como el **Reglamento General de Protección de Datos - RGPD** en Europa) si el sistema gestiona información personal de jugadores, entrenadores o árbitros.
* **Acceso a datos**:
  + Para validar correctamente el funcionamiento del sistema, puede requerirse acceso a datos reales o simulados de federaciones deportivas, lo cual deberá realizarse bajo autorización o mediante acuerdos previos.
* **Entornos de despliegue**:
  + Definición clara de entornos diferenciados para desarrollo, pruebas y producción.
  + Protocolos de copia de seguridad y mecanismos de soporte post-despliegue.

Implementación y desarrollo

La implementación del sistema **FederaPro** se ha llevado a cabo de forma modular, permitiendo el desarrollo independiente de cada uno de sus componentes: backend, aplicación de escritorio y aplicación móvil. La arquitectura definida se apoya en buenas prácticas de desarrollo, uso de herramientas especializadas y tecnologías consolidadas.

Tecnologías empleadas

* **Lenguajes de programación**:
  + Java: para el backend (API RESTful).
  + C#: para la aplicación de escritorio.
  + Un dibujo con letras

    El contenido generado por IA puede ser incorrecto.Kotlin: para la aplicación móvil.
* **IDEs de desarrollo**:
  + IntelliJ IDEA: desarrollo de backend con Spring Boot.
  + Visual Studio: desarrollo de la aplicación de escritorio (Windows Forms).
  + Android Studio: desarrollo de la aplicación móvil en Kotlin.
* **Frameworks**:
  + Spring Boot: para construir una API segura, modular y escalable.
  + Windows Forms: para interfaces de usuario de escritorio.
  + Android Jetpack: para desarrollo de la app móvil.
* **Bases de datos**:
  + **MySQL**: sistema de gestión de base de datos relacional.
  + **DBeaver**: cliente gráfico para la gestión de la base de datos.
* **Plataformas y herramientas de desarrollo**:
  + **Swagger**: documentación interactiva de la API.
  + **Postman**: para pruebas de endpoints y depuración de servicios REST.
  + **Docker** (opcional): previsto para entornos de despliegue estandarizados.
* **Control de versiones**:
  + Git (con GitHub): para control de versiones, trabajo colaborativo y seguimiento de cambios.
* **Patrones de diseño**:
  + **Modelo-Vista-Controlador (MVC)**: aplicado tanto en la aplicación de escritorio como en la API.
  + **DAO/Service/Controller** en el backend para separación clara de capas.
* **Convenciones de codificación**:
  + Estilo de código Java basado en las convenciones de Oracle y Spring.
  + Estilo de código C# conforme a los estándares de Microsoft.
  + Estilo de código Kotlin con las recomendaciones de Android Developers.
  + Nomenclatura coherente y comentarios descriptivos para mejorar el mantenimiento.
* **Otras herramientas**:
  + **Lucidchart / Draw.io**: diseño de diagramas E/R, clases y arquitectura.
  + **Markdown**: para documentación técnica y manual de usuario.

Despliegue y mantenimiento

El despliegue del sistema **FederaPro** implica la publicación de sus tres componentes principales: la **API backend**, la **aplicación de escritorio** y la **aplicación móvil**. Cada uno de ellos requiere un entorno de ejecución específico y una serie de configuraciones que aseguren su correcto funcionamiento y disponibilidad.

Entorno de despliegue

**Aplicación de escritorio (C# - Windows Forms)**

* **Distribución**: empaquetada en un instalador (.exe) que incluye todas las dependencias necesarias.
* **Instalación**: sencilla, guiada y compatible con versiones de Windows 10 y superiores.
* **Dispositivos compatibles**: ordenadores personales con mínimo 4 GB de RAM y conexión a internet.
* **Requisitos**: .NET Framework 4.8 o superior (incluido en el instalador si no está presente).
* **Conectividad**: acceso a la red local o a internet para conectarse con la API backend vía HTTP.

Icono

El contenido generado por IA puede ser incorrecto.Aplicación móvil (Kotlin)

* **Distribución**: mediante archivo APK o futura publicación en Google Play.
* **Instalación**: compatible con dispositivos Android 8.0 (Oreo) o superior.
* **Requisitos**: acceso a internet y permisos de red para consumir la API.
* **Diseño responsive**: adaptado a distintos tamaños de pantalla.

Icono

El contenido generado por IA puede ser incorrecto.API Backend (Java - Spring Boot)

* **Servidor recomendado**: VPS o servidor en la nube (por ejemplo, AWS, DigitalOcean o local con Tomcat).
* **Sistemas operativos compatibles**: Linux (Ubuntu 20.04 o superior) o Windows Server.
* **Despliegue**: mediante archivo JAR ejecutable (java -jar FederaProAPI.jar) o contenedor Docker.
* **Puertos y seguridad**: expuesta por defecto en el puerto 8080; se recomienda el uso de HTTPS en producción mediante Nginx o Apache como proxy inverso.

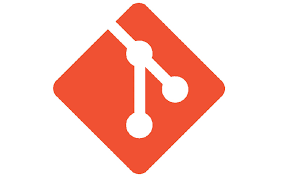
Despliegue de la base de datos

* **Motor de base de datos**: MySQL 8 o superior.
* **Ubicación**: se puede alojar en el mismo servidor del backend o en un servicio de base de datos gestionado.
* **Configuración recomendada**:
  + Usuario con permisos limitados para la aplicación.
  + Copias de seguridad programadas (diarias/semanales).
  + Conexiones protegidas por firewall y cifrado si se accede remotamente.

Software adicional necesario

* **Servidor web (opcional)**: Nginx o Apache como proxy inverso para la API.
* **Certificados SSL**: para asegurar la comunicación entre cliente y servidor (Let's Encrypt o comerciales).



* **Herramientas administrativas**:
  + DBeaver o phpMyAdmin para gestión de base de datos.
  + Git para actualización y control de versiones.
  + Docker para entornos homogéneos de despliegue (opcional).

Seguridad en producción

* Cifrado de contraseñas de usuario mediante algoritmos seguros (BCrypt).
* Control de acceso basado en roles (usuario, administrador).
* Validación y sanitización de datos de entrada.
* Autenticación de usuarios en la API mediante tokens o sesiones.
* Cumplimiento del RGPD para el tratamiento de datos personales.

Mantenimiento y Actualizaciones

El mantenimiento de **FederaPro** se organiza en tres niveles:

**1. Mantenimiento correctivo**

* Corrección de errores detectados en producción tras el despliegue.
* Registro de errores a través de logs y reporte por parte de los usuarios.

**2. Mantenimiento adaptativo**

* Cambios en la infraestructura (por ejemplo, actualización de versión de MySQL, Java, etc.).
* Ajustes para compatibilidad con nuevas versiones de Android o Windows.

**3. Mantenimiento evolutivo**

* Incorporación de nuevos deportes y módulos funcionales.
* Mejora de interfaces, rendimiento o experiencia de usuario.
* Implementación de nuevas funcionalidades a demanda de los usuarios.

Estrategia de actualización

* **Control de versiones**: numeración semántica (ej. 1.0.0, 1.1.0, 2.0.0).
* **Historial de cambios (changelog)**: registro detallado de versiones y modificaciones.
* **Canales de distribución**:
  + Escritorio: actualización automática desde la aplicación o manual mediante nuevo instalador.
  + Móvil: distribución de nuevas versiones mediante APK o Google Play.
  + API/backend: despliegue controlado con backups y pruebas previas.

Pruebas y control de calidad

Para garantizar el correcto funcionamiento del sistema **FederaPro**, se ha definido un plan de pruebas que cubre distintas fases del desarrollo. Las pruebas se han diseñado e implementado desde las primeras etapas del desarrollo, lo que ha permitido detectar errores de forma temprana y asegurar que los requisitos funcionales y no funcionales se cumplan adecuadamente.

Estrategias de pruebas aplicadas

**Pruebas unitarias**

* Aplicadas a los métodos del backend (Java + JUnit).
* Permiten verificar que cada clase y método funcionan correctamente de forma aislada.
* **Ejemplos de pruebas unitarias por requisito funcional**:
  + testCrearEquipo() – verifica la creación de un nuevo equipo y su persistencia en base de datos.
  + testObtenerJugadoresPorEquipo() – valida que se devuelvan correctamente los jugadores de un equipo existente.
  + testCrearPartidoYEstadisticas() – prueba la creación de un partido y la asignación de estadísticas iniciales.
  + testCalculoClasificacion() – comprueba el cálculo de puntos y clasificación a partir de partidos jugados.
  + testValidacionDatosCSV() – asegura que se detecten correctamente errores en ficheros CSV importados.

**Pruebas de integración**

* Evalúan la interacción entre componentes: **API – Base de datos**, **Aplicación de escritorio – API**, **App móvil – API**.
* Incluyen casos reales y simulados con datos de prueba.
* Se usan herramientas como Postman, pruebas con JWT, y logs de interacción entre capas.

**Pruebas de sistema**

* Ejecutadas sobre la aplicación completa (entorno real o de staging).
* Cobertura sobre funcionalidades clave:
  + Gestión de equipos: creación, edición, eliminación, consulta.
  + Gestión de jugadores: vinculación a equipos, actualización por CSV.
  + Gestión de competiciones: filtrado por tipo, generación automática de partidos.
  + Estadísticas: carga y visualización por partido, jugador y equipo.
* Validación de comportamiento esperado tras flujos completos de usuario.

**Pruebas de rendimiento**

* Carga en endpoints críticos como:
  + GET /basket/equipos
  + GET /basket/estadisticas/jugador
* Tiempo de respuesta evaluado bajo múltiples solicitudes concurrentes.
* Resultado: en entorno local, **el 95% de las respuestas se procesan en < 2 segundos**.

**Pruebas de seguridad**

* Validación de sesiones (login y logout).
* Encriptación de contraseñas (bcrypt).
* Control de acceso por rol: admin / usuario registrado.
* Cumplimiento de principios básicos del RGPD:
  + Consentimiento al registrarse.
  + Acceso y modificación de datos personales.

Ciclo de vida de las pruebas

* **Diseño de pruebas**: durante la fase de implementación de cada módulo.
* **Ejecución de pruebas**: tras el desarrollo de cada funcionalidad y al finalizar iteraciones clave.
* **Responsables de pruebas**:
  + El desarrollador correspondiente realiza las pruebas unitarias e integración.
  + El tester (rol QA) ejecuta pruebas completas en entorno de validación.

Evaluación por usuarios finales

Se ha previsto una fase de validación con usuarios representativos de federaciones deportivas reales o simuladas. Estas pruebas buscan:

* Validar la usabilidad de la interfaz de escritorio y móvil.
* Hay que confirmar que el sistema cubre correctamente los procesos de gestión deportiva.
* Recoger sugerencias de mejora para futuras versiones.

Gestión económica

Para el desarrollo del proyecto **FederaPro**, se ha realizado una estimación económica detallada basada en los recursos humanos y técnicos necesarios, así como en la planificación y fases establecidas previamente. Esta gestión económica sigue el modelo profesional de desarrollo de software multiplataforma, y busca simular el coste real de ejecución del sistema en un entorno de producción.

Estimación de costes por recursos humanos

El coste del proyecto se ha calculado considerando distintos perfiles profesionales involucrados y sus tarifas por hora. A continuación se presenta la tabla completa que muestra el desglose de tareas, número de horas dedicadas, perfiles implicados y el coste económico correspondiente:

|  |  |  |
| --- | --- | --- |
| **Perfil profesional** | **Código** | **Tarifa por hora (€)** |
| Gerente | G | 80 €/h |
| Jefe de Proyecto | JP | 60 €/h |
| Analista | A | 50 €/h |
| Programador Senior | PS | 40 €/h |
| Programador Junior | PJ | 30 €/h |

Cada tarea global del proyecto (análisis, diseño, desarrollo, pruebas, documentación, etc.) ha sido desglosada en horas, y distribuida entre estos perfiles según el tipo de trabajo requerido. Esto permite reflejar no solo el esfuerzo estimado, sino también el impacto económico de cada perfil.

Tabla de costes por tareas y perfiles

La siguiente tabla muestra en detalle las horas dedicadas por perfil en cada fase del proyecto, así como el coste total por tarea (basado en las tarifas de cada perfil):

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tarea** | **Horas estimadas** | **G** | **JP** | **A** | **PS** | **PJ** | **Coste total** |
| Gestión del proyecto | 20 | 5 | 10 | 5 | 0 | 0 | 1.900 € |
| Análisis funcional | 25 | 0 | 5 | 20 | 0 | 0 | 2.000 € |
| Diseño técnico | 30 | 0 | 5 | 25 | 0 | 0 | 1.920 € |
| Propuesta gráfica | 15 | 0 | 0 | 0 | 10 | 5 | 760 € |
| Elaboración plan de pruebas | 20 | 0 | 5 | 10 | 5 | 0 | 830 € |
| Config. entorno desarrollo | 10 | 0 | 2 | 3 | 2 | 3 | 304 € |
| Desarrollo | 120 | 5 | 10 | 10 | 60 | 35 | 13.500 € |
| Integración componentes | 30 | 0 | 2 | 0 | 10 | 15 | 1.875 € |
| Ejecución de pruebas | 30 | 0 | 2 | 0 | 10 | 15 | 1.875 € |
| Documentación | 25 | 0 | 0 | 2 | 5 | 10 | 860 € |
| Instalación | 10 | 0 | 0 | 0 | 3 | 7 | 365 € |
| Pruebas | 15 | 0 | 0 | 0 | 10 | 5 | 625 € |
| Formación | 20 | 0 | 2 | 0 | 5 | 10 | 1.290 € |
| Mantenimiento | 30 | 0 | 2 | 2 | 10 | 16 | 1.951 € |

🔸 **Coste total estimado del proyecto: 16.180 €**

Explicación del modelo económico aplicado

La tabla anterior se ha elaborado aplicando una estimación realista basada en:

* **Distribución del trabajo** según el tipo de tarea: por ejemplo, el análisis ha sido mayormente asumido por el analista, mientras que el desarrollo ha sido repartido entre programadores senior y junior.
* **Asignación de perfiles técnicos** con sus respectivas tarifas estándar de mercado.
* **Horas totales de dedicación** calculadas en base al cronograma del proyecto (diagrama de Gantt y fases definidas).

Esta metodología permite:

* Justificar el coste total de forma detallada.
* Visualizar qué tareas tienen mayor impacto económico (ej. desarrollo).
* Comprender cómo cada perfil contribuye al coste total.

Consideraciones adicionales

Además de los costes humanos, se han considerado materiales y servicios necesarios para el desarrollo:

* Costes de recursos materiales: Se refieren a todos aquellos elementos físicos o digitales que son necesarios para el trabajo técnico, incluyendo equipos informáticos, dispositivos móviles, mobiliario, licencias de software, herramientas de diseño o plataformas de desarrollo. Son inversiones imprescindibles para crear y probar la aplicación en condiciones reales.
* Costes de proveedores: Abarcan los servicios externos que facilitan el desarrollo y despliegue del sistema, como el acceso a internet, servidores de alojamiento (hosting), consumo eléctrico, certificados de seguridad o servicios en la nube. Estos costes aseguran que el entorno de trabajo y producción sea funcional, accesible y seguro.

Costes de recursos materiales

|  |  |  |  |
| --- | --- | --- | --- |
| **Recurso** | **Descripción** | **Coste estimado** | **Coste**  **total** |
| Ordenador portátil de desarrollo | Intel i5, 16 GB RAM, SSD 512 GB | 5\*900 € | 4500 € |
| Monitor adicional para desarrollo | Monitor 24" Full HD | 5\*150 € | 750 € |
| Dispositivo Android para pruebas | Smartphone Android gama media | 5\*200 € | 1000 € |
| Licencia Visual Studio (Pro) | Licencia anual | 5\*450 € | 2250 € |
| Licencia IntelliJ IDEA Ultimate | Licencia anual | 5\*500 € | 2500 € |
| Hosting VPS (API + BBDD) | DigitalOcean (droplet 4GB RAM) durante 6 meses | 120 € | 120 € |
| Dominio y certificados SSL | Para asegurar API y aplicaciones | 60 € | 60 € |
| Herramientas de diseño | Lucidchart o Draw.io versión premium | 60 € | 60 € |

🔸 **Total recursos materiales estimado**: **11.240 €**

Costes de proveedores

|  |  |  |  |
| --- | --- | --- | --- |
| **Proveedor / Servicio** | **Coste mensual** | **Meses** | **Total** |
| Internet para desarrollo | 40 € | 6 | 240 € |
| Energía eléctrica (proporcional) | 30 € | 6 | 180 € |
| Hosting (ya incluido en materiales) | - | - | - |

🔸 **Total proveedores estimado**: **420 €**

Estos costes, si bien no están incluidos en la tabla de horas por perfil, deberían contemplarse en un presupuesto global si se tratara de un desarrollo real para cliente.

Conclusión

La gestión económica de **FederaPro** se ha basado en una estimación detallada por fases, perfiles y horas de dedicación. Con un coste total estimado de **27.840 €**, el proyecto demuestra ser viable en un contexto profesional, reflejando fielmente el esfuerzo técnico, organizativo y de desarrollo que conlleva una solución multiplataforma de este tipo.

Valoración personal

El desarrollo del proyecto **FederaPro** ha supuesto una experiencia profundamente enriquecedora tanto a nivel técnico como personal. Este proyecto me ha permitido aplicar de forma práctica todos los conocimientos adquiridos a lo largo del ciclo formativo, especialmente en programación orientada a objetos, diseño de bases de datos, desarrollo de interfaces y arquitectura de aplicaciones multiplataforma.

Desde el inicio, el planteamiento de crear una herramienta dinámica y flexible para la gestión de federaciones deportivas fue un reto ambicioso. Sin embargo, el enfoque modular, la planificación detallada y la integración progresiva de tecnologías (Java, Spring Boot, C#, Kotlin, MySQL, etc.) han permitido materializar una solución realista, profesional y escalable. Uno de los mayores aprendizajes ha sido la necesidad de coordinar correctamente todos los componentes del sistema: backend, escritorio, móvil y base de datos, asegurando que la comunicación entre ellos sea fluida y segura.

A nivel personal, ha sido muy satisfactorio ver cómo una idea inicial se convierte en un producto funcional. La resolución de problemas, la toma de decisiones de diseño, y la mejora continua durante la implementación me han hecho crecer como desarrollador, fomentando una mentalidad crítica y orientada a la calidad.

En cuanto a las **FCTs (Formación en Centros de Trabajo)**, han tenido un papel clave para el éxito de este proyecto. Las tareas realizadas en la empresa, especialmente relacionadas con el mantenimiento de bases de datos, desarrollo en backend y corrección de errores reales en entornos productivos, me ofrecieron una visión más clara y profesional de cómo se trabaja en el mundo real del desarrollo. Además, los consejos de mi tutor en prácticas han sido especialmente útiles para entender cómo organizar un proyecto grande, documentarlo correctamente y validar cada fase antes de continuar. La experiencia me ayudó también a mejorar la gestión del tiempo, a entender la importancia del trabajo en equipo y a pulir habilidades técnicas que han sido aplicadas directamente en FederaPro.

En definitiva, considero que este proyecto ha sido el broche perfecto para cerrar esta etapa formativa. Ha consolidado mis conocimientos, me ha enfrentado a retos reales y me ha preparado para el siguiente paso: la incorporación al mundo laboral como desarrollador profesional.

Bibliografía

SportsTechX. *Resumen del mercado global de tecnología deportiva*. Recuperado de <https://www.sportstechx.com>

Comisión Europea. *Reglamento General de Protección de Datos (RGPD)*. Recuperado de <https://gdpr.eu>

Unión Internacional de Telecomunicaciones (UIT). *Tendencias en las TIC aplicadas al deporte*. Recuperado de <https://www.itu.int>

**MyTeam11**. Plataforma de gestión de equipos y competiciones. Recuperado de <https://www.myteam11.com>

**TeamSnap**. Herramienta para la gestión de equipos amateurs. Recuperado de <https://www.teamsnap.com>

**SportEasy**. Plataforma para la gestión de equipos deportivos. Recuperado de <https://www.sporteasy.net>

**Playwaze**. Sistema especializado en la organización de eventos y competiciones deportivas. Recuperado de <https://www.playwaze.com>

**LeagueRepublic**. Plataforma para la gestión de ligas y torneos deportivos. Recuperado de <https://www.leaguerepublic.com>  
  
Ministerio de Industria, Comercio y Turismo (España). *Guía para la elaboración de planes de empresa*. Recuperado de: <https://industria.gob.es>

Fundación Telefónica. *Manual de gestión de proyectos tecnológicos*. Recuperado de: <https://fundaciontelefonica.com>

OpenClassrooms. *How to estimate project costs*. Recuperado de: <https://openclassrooms.com/en/courses/estimating-project-costs>

GitHub Docs. *Software project estimation templates and cost control*. Recuperado de: <https://docs.github.com/>

Anexos

Repositorio de Git. <https://github.com/JoseDavidDAM103/FederaPro>

Manual de Usuario FederaProApp. <https://github.com/JoseDavidDAM103/FederaPro/blob/main/Documentaci%C3%B3n/Manual_FederaProApp.pdf>

Manual de Usuario FederaProDesktop. <https://github.com/JoseDavidDAM103/FederaPro/blob/main/Documentaci%C3%B3n/Manual_FederaProDesktop.pdf>