

# MiFutbolC - Sistema Integral de Gestión de Fútbol

version **4.2** language **C** database **SQLite3** platform **Windows | Linux | macOS** license **Open Source**

















 quality gate **failed**  codefactor **A**

**Sistema completo de gestión, análisis y simulación de datos futbolísticos desarrollado en C**

[Características](#) • [Instalación](#) • [Uso](#) • [Documentación](#) • [Arquitectura](#) • [DeepWiki](#)

## Descripción

**MiFutbolC** es un sistema robusto y completo de gestión de fútbol desarrollado íntegramente en lenguaje C. Este proyecto permite administrar todos los aspectos relacionados con el fútbol amateur y profesional, incluyendo:

-  **Gestión de equipamiento** (camisetas con seguimiento de uso)
-  **Infraestructura deportiva** (canchas)
-  **Registro de partidos** (con estadísticas detalladas)
-  **Administración de equipos** (jugadores, formaciones, posiciones)
-  **Organización de torneos** (múltiples formatos y estructuras)
-  **Análisis avanzado** (rendimiento, tendencias, química entre jugadores, meta-análisis)
-  **Gestión de salud** (lesiones y recuperación)
-  **Bienestar integral** (hábitos, planificación, salud y reportes personales)
-  **Gestión de recordatorios** (agenda personal, filtros y seguimiento)
-  **Colecciones e inventario** (items, colecciones y sincronización con camisetas)
-  **Control financiero** (ingresos, gastos, balances)
-  **Sistema gamificado** (logros y badges)
-  **Entrenador IA** (recomendaciones inteligentes)
-  **Exportación multiformato** (CSV, JSON, HTML, TXT)
-  **Informe PDF mejorado** (portada, secciones y más datos)
-  **Importación de datos** (restauración desde JSON/TXT/CSV/HTML)

El sistema utiliza **SQLite3** como base de datos para almacenamiento persistente y eficiente, **cJSON** para manejo de datos JSON, y ofrece una interfaz de consola intuitiva con menús jerárquicos y arte ASCII.

- **Rutas de base de datos más seguras:** refactor de construcción de rutas con **snprintf** para mejorar robustez y legibilidad.
- **Sistema de logs mejorado:** formateo de mensajes de log más consistente y seguro, reduciendo riesgos de errores en cadenas largas.
- **Manejo de imágenes reforzado:** mejoras en selección de imágenes de camisetas y utilidades para actualizar rutas/imágenes en base de datos.
- **Flujo de herramientas de imagen más claro:** nuevos mensajes de éxito y advertencia durante instalación/configuración de herramientas opcionales.
- **Resolución de rutas de imagen desde BD:** soporte ampliado para recuperar y resolver rutas guardadas en base de datos de forma más confiable.

- **Camisetas y canchas con ciclo activo/inactivo:** retiro y reactivación para conservar historial sin perder datos.
- **Eliminación protegida con historial:** opciones para reasignar partidos, eliminar junto con partidos o retirar entidad conservando historial.
- **Instalación por consola en Windows:** nuevo script `install.ps1` para compilar/instalar sin Inno Setup.

## ✨ Características Principales

### 📦 Gestión de Equipamiento y Recursos

- **Camisetas Inteligentes:** Crear, editar, eliminar y listar camisetas con seguimiento automático de uso y rendimiento
- **Ciclo Activa/Inactiva en Camisetas:** retirar y reactivar camisetas para preservar historial
- **Imágenes de Camisetas:** carga, apertura y ajustes de visor (incluye prueba de visor y previsualización en consola)
- **Gestión de Canchas:** administrar infraestructura deportiva (crear, listar, modificar, retirar/reactivar y eliminar con opciones seguras)
- **Análisis de Uso:** Estadísticas de rendimiento por camiseta y cancha

### 👥 Administración de Equipos y Jugadores

- **Equipos Personalizables:** Crear equipos fijos o momentáneos con información completa
- **Modalidades Múltiples:** Soporte para Fútbol 5, 7, 8 y 11
- **Posiciones Definidas:** Arqueros, Defensores, Mediocampistas, Delanteros
- **Formaciones Tácticas:** Configuración de formaciones y designación de capitanes
- **Gestión de Plantillas:** Hasta 11 jugadores por equipo con datos completos

### 🌐 Registro y Gestión de Partidos

- **Partidos Detallados:** Registro completo con fecha, hora, cancha, equipos y resultados
- **Métricas Avanzadas:** Goles, asistencias, rendimiento, estado físico y mental
- **Simulación de Partidos:** Simulador con cancha ASCII animada y eventos aleatorios
- **Historial Completo:** Consulta de partidos pasados con filtros avanzados

### 🏆 Sistema de Torneos

- **Gestión Base de Torneos:** Crear, listar, modificar y eliminar torneos
- **Estructura de Datos:** Modelo preparado para fases, historial y estadísticas de torneos
- **Integración con Temporadas:** Relación con módulos de temporada y reportes

### 📊 Análisis y Estadísticas Avanzadas

#### Estadísticas Generales

- Rendimiento agregado del sistema
- Análisis de estados físicos y mentales
- Estadísticas por camiseta, cancha y equipo

## Análisis Temporal

- **Por Año:** Tendencias longitudinales y comparación interanual
- **Por Mes:** Desglose mensual y análisis de estacionalidad
- **Por Día de Semana:** Rendimiento según día de la semana
- **Por Clima:** Análisis según condiciones climáticas

## Meta-Análisis Avanzado

- **Consistencia de Rendimiento:** Evaluación de variabilidad y estabilidad
- **Detección de Outliers:** Identificación de partidos atípicos
- **Dependencia del Contexto:** Análisis de factores externos
- **Impacto del Cansancio:** Evaluación real de la fatiga en el desempeño
- **Impacto del Estado de Ánimo:** Influencia emocional en resultados
- **Análisis de Eficiencia:** Relación entre esfuerzo y resultados

## Récords y Rankings

- Récords históricos de partidos (máximo de goles, asistencias, rendimiento)
- Mejores y peores rachas (victorias, derrotas, goles)
- Combinaciones óptimas (cancha + camiseta)
- Temporadas destacadas
- Partidos especiales y extremos
- Top 5 mejores partidos por rendimiento
- Ranking de camisetas (partidos, victorias/empates/derrotas, winrate)

## Química entre Jugadores

- **Mejor combinación de jugadores** con cálculo de winrate
- **Modo híbrido:** combina duplas automáticas (equipos/partidos) y registros manuales
- **CRUD de química:** alta, listado, edición y eliminación de estadísticas por jugador-partido
- **Métricas manuales:** goles, asistencias, asistencias al usuario, posición y comentario

## Gestión de Salud y Lesiones

- **Registro de Lesiones:** Crear, editar, eliminar y listar lesiones de jugadores
- **Seguimiento Detallado:** Tipo de lesión, fecha, duración y jugador afectado
- **Estadísticas de Lesiones:** Análisis de frecuencia, tipos más comunes, distribución mensual
- **Impacto en Rendimiento:** Evaluación del efecto de lesiones en el desempeño
- **Análisis por Camiseta:** Identificación de patrones de lesiones por equipamiento

## Gestión Financiera

- **Control de Ingresos:** Registro de cuotas, sponsors y otras entradas
- **Gestión de Gastos:** Categorización detallada (transporte, equipamiento, torneos, arbitraje, canchas, medicina, otros)
- **Balances y Reportes:** Visualización de estado financiero actual
- **Resúmenes por Categoría:** Análisis de gastos por tipo

- **Exportación Financiera:** Reportes en múltiples formatos

## Sistema Gamificado

- **Logros por Categorías:** Goles, asistencias, partidos, victorias, rendimiento, etc.
- **Niveles Progresivos:** Novato, Promedio, Experto, Maestro, Leyenda
- **Seguimiento de Progreso:** Visualización de logros completados y en progreso
- **Badges Especiales:** Hat-tricks, poker de asistencias, rendimiento perfecto
- **Motivación Continua:** Sistema de recompensas para incentivar el uso

## Bienestar Integral

- **Planificación Personal:** Objetivos, rutinas y seguimiento de hábitos
- **Entrenamiento y Alimentación:** Registro y control de prácticas saludables
- **Mental y Salud:** Evaluación de bienestar mental, controles médicos y reportes
- **Informe Personal Mensual (PDF):** Resumen automático con métricas clave

## Recordatorios

- **Gestión de Recordatorios:** Crear, editar, eliminar y listar recordatorios personales
- **Agenda Integrada:** Vista de eventos próximos y pasados
- **Filtros por Temática:** Búsqueda de recordatorios por categoría
- **Portabilidad:** Exportar e importar recordatorios

## Colecciones e Inventario

- **Inventario de Ítems:** Crear y listar items asociados a tu actividad
- **Sincronización con Camisetas:** Vincular camisetas existentes al inventario
- **Colecciones Personalizadas:** Crear colecciones y agregar/quitar items
- **Respaldo del Módulo:** Exportación e importación de backup en JSON

## Reproductor de Música

- **Formatos de audio:** `.mp3`, `.wav`, `.flac`, `.ogg` desde la carpeta `Musica/`
- **Controles completos:** Reproducir, pausar, detener, pista anterior/siguiente, seleccionar pista de la lista
- **Barra de progreso:** tiempo actual y duración total en tiempo real (`mm:ss / mm:ss`)
- **Tiempo restante:** visualización del tiempo que queda en la pista
- **Control de volumen configurable:** paso dinámico ajustable (1 %, 5 %, 10 %, 20 %) con barra visual; persiste entre sesiones
- **Modos de repetición:** Sin repetición, Repetir pista, Repetir lista, Aleatorio (shuffle)
- **Historial shuffle:** al retroceder en modo aleatorio, vuelve a la pista anterior del historial
- **Ecualizador 3 Bandas:** graves (200 Hz), medios (1 kHz) y agudos (8 kHz) de  $\pm 12$  dB con paso de 3 dB
- **Playlists:** crear, cargar y eliminar listas de reproducción personalizadas
- **Exportar catálogo como playlist:** guarda todas las pistas del catálogo (o las filtradas) como un archivo `.txt`
- **Gestión de archivos:** agregar y eliminar canciones directamente desde el menú
- **Búsqueda y filtro:** buscar pistas por nombre con filtro activo persistente; limpiar filtro con `[18]`
- **Retroceder / Avanzar:** salto de  $\pm 10$  segundos dentro de la pista (`[16]/[17]`)
- **Saltar a tiempo exacto:** ir a un momento específico en formato `MM:SS` (`[20]`)

- **Sleep timer:** apagado automático tras N minutos configurables ([19])
- **Información de pista:** muestra nombre, duración, formato, sample rate y canales de la pista activa ([21])
- **Renombrar pista:** cambia el nombre de cualquier pista del catálogo en disco y en memoria ([22])
- **Inicio automático:** reproducción al arrancar la aplicación, configurable desde el menú o desde Ajustes
- **Reanudar posición:** al salir y volver al reproductor, retoma la pista y el instante exacto donde se dejó
- **Fade In/Out:** transiciones suaves entre pistas (400 ms entrada, 250 ms salida)
- **Multipataforma:** Windows y Linux/macOS vía miniaudio (header-only, sin dependencias externas)

## Entrenador IA

- **Recomendaciones Inteligentes:** Sugerencias basadas en análisis de datos históricos
- **Optimización de Formaciones:** Consejos para mejorar el rendimiento del equipo
- **Análisis Predictivo:** Predicciones basadas en tendencias y patrones

Accesible desde **Análisis** → **Entrenador IA**.

## Exportación e Importación de Datos

### Exportación Multiformato

- **Formatos Disponibles:** CSV, JSON, HTML, TXT
- **Acceso desde Ajustes:** Ruta de navegación Ajustes -> Exportar
- **Exportación por Módulo:** Camisetas, partidos, lesiones, estadísticas, análisis y base de datos
- **Exportación Completa:** Backup total del sistema
- **Exportación Mejorada:** Con análisis integrado y métricas adicionales
- **Exportación Selectiva:** Partidos con características específicas
- **TXT adicionales:** finanzas por mes/año, ranking de canchas, partidos por clima, lesiones por tipo/estado, historial de rachas y distribución de estado de ánimo/cansancio

### Importación de Datos

- **Restauración multiformato:** Importación completa de datos desde archivos JSON, TXT, CSV o HTML
- **Acceso desde Ajustes:** Ruta de navegación Ajustes -> Importar
- **Importación rápida:** Opciones Todo JSON rápido y Todo CSV rápido
- **Importación de base de datos:** Restauración directa de archivo SQLite
- **Validación de Datos:** Verificación de estructura antes de importar
- **Manejo de Errores:** Feedback detallado en caso de problemas

## Configuración y Personalización

- **Temas de Interfaz:** Claro, Oscuro, Azul, Verde, Rojo, Púrpura, Clásico, Alto Contraste
- **Multiidioma:** Español e Inglés
- **Multiusuario Local:** Inicio de sesión por perfil local con datos separados por usuario
- **Seguridad por Usuario:** Contraseña opcional por perfil, con gestión desde Ajustes -> Usuario
- **Nombres de Usuario:** Personalización del nombre visible por perfil
- **Configuración Persistente:** Preferencias guardadas en base de datos

## Interfaz y Experiencia de Usuario

- **Menús Jerárquicos:** Navegación intuitiva con estructura organizada
- **Arte ASCII:** Pantallas de bienvenida, simulaciones animadas, decoraciones
- **Interfaz Textual:** No requiere GUI, compatible con cualquier terminal

## Tecnologías Utilizadas

### Lenguaje y Estándares

- **Lenguaje C:** Desarrollo completo en C estándar
- **Estándar:** Compatible con C99/C11
- **Compiladores:** GCC, MinGW, Clang

### Bibliotecas y Dependencias

#### SQLite3 (Incluida)

- **Versión:** 3.x
- **Archivos:** `sqlite3.c`, `sqlite3.h`
- **Propósito:** Base de datos embebida para almacenamiento persistente
- **Características:** Transacciones ACID, consultas SQL completas, sin servidor

#### cJSON (Incluida - Licencia MIT)

- **Archivos:** `cJSON.c`, `cJSON.h`
- **Propósito:** Parsing y generación de JSON para importación/exportación
- **Características:** Ligera, rápida, sin dependencias externas

#### Motor PDF Interno (pdfgen)

- **Archivos:** `pdfgen.c`, `pdfgen.h`
- **Propósito:** Generación de informes PDF del sistema
- **Dependencias externas:** No requiere dependencias adicionales para PDF

#### miniaudio (Incluida - Dominio Público / MIT-0)

- **Archivo:** `miniaudio.h`
- **Propósito:** Motor de audio multiplataforma para reproducción de archivos MP3
- **Características:** Header-only, sin dependencias externas, soporta Windows y Linux/macOS
- **Uso:** Streaming de audio, grafo de nodos DSP para ecualizador, fade in/out y control de volumen

### Herramientas de Desarrollo

- **CodeBlocks:** IDE recomendado para desarrollo (incluye proyecto `.cbp`)
- **Doxygen:** Generación de documentación técnica (opcional)
- **Git:** Control de versiones
- **Inno Setup:** Creación de instalador para Windows

### Dependencias Incluidas

- ✓ **SQLite3**: Biblioteca completa incluida en el proyecto
- ✓ **cJSON**: Biblioteca completa incluida en el proyecto
- ✓ **pdfgen**: Motor PDF incluido en el proyecto
- ✓ **miniaudio**: Motor de audio incluido en el proyecto
- ✓ **Runtime sin DLLs**: El proyecto no requiere DLLs adicionales para ejecutarse

## Instalación y Compilación

### Método 1: Instalador Automático (Windows - Recomendado para Usuarios Finales)

El método más sencillo para usuarios que solo desean usar el programa:

1. Navega a la carpeta `installer/`
2. Ejecuta el archivo `MiFutbolC_Setup.exe`
3. Sigue las instrucciones del instalador
4. El programa se instalará automáticamente con todos los archivos necesarios
5. Busca "MiFutbolC" en el menú de inicio de Windows

### Método 2: Usando CodeBlocks (Recomendado para Desarrolladores)

Ideal para desarrollo, depuración y modificación del código:

#### 1. Instalar CodeBlocks:

- Descarga desde [codeblocks.org](https://codeblocks.org)
- Asegúrate de instalar la versión con MinGW incluido

#### 2. Abrir el Proyecto:

- Abre CodeBlocks
- Ve a `File` → `Open` → Selecciona `MiFutbolC.cbp`

#### 3. Compilar:

- Selecciona `Build` → `Build` (o presiona `Ctrl+F9`)
- El ejecutable se generará en `bin/Debug/MiFutbolC.exe`

#### 4. Ejecutar:

- Presiona `F9` o selecciona `Build` → `Build and Run`

### Método 3: Compilación Manual con GCC (Linux/macOS/Windows)

Para usuarios avanzados que prefieren control total sobre la compilación:

#### Linux / macOS

```
# 1. Clonar o descargar el repositorio
cd /ruta/al/proyecto/MiFutbolC

# 2. Compilar todos los archivos fuente
```

```
gcc -o MiFutbolC \  
  main.c db.c menu.c camiseta.c partido.c equipo.c torneo.c \  
  estadisticas.c estadisticas_generales.c estadisticas_anio.c \  
  estadisticas_mes.c estadisticas_meta.c estadisticas_lesiones.c \  
  analisis.c cancha.c logros.c lesion.c temporada.c \  
  financiamiento.c settings.c entrenador_ia.c \  
  records_rankings.c export.c export_all.c export_all_mejorado.c \  
  export_camisetas.c export_camisetas_mejorado.c \  
  export_lesiones.c export_lesiones_mejorado.c \  
  export_partidos.c export_estadisticas.c \  
  export_estadisticas_generales.c export_records_rankings.c \  
  import.c utils.c sqlite3.c cJSON.c \  
  -I. -lm -lpthread -ldl  
  
# 3. Ejecutar el programa  
./MiFutbolC
```

## Windows (con MinGW)

```
# 1. Abrir terminal (PowerShell o CMD)  
cd D:\ANIME\Libros\Lenguaje C\Proyectos\MiFutbolC  
  
# 2. Definir librerías de Windows (MinGW/Git Bash)  
WIN_LIBS="-lcomdlg32 -lshell32 -lbcrypt"  
  
# 3. Compilar  
gcc -o MiFutbolC.exe main.c db.c menu.c camiseta.c partido.c equipo.c torneo.c  
estadisticas.c estadisticas_generales.c estadisticas_anio.c estadisticas_mes.c  
estadisticas_meta.c estadisticas_lesiones.c analisis.c cancha.c logros.c lesion.c  
temporada.c financiamiento.c settings.c entrenador_ia.c records_rankings.c  
export.c export_all.c export_all_mejorado.c export_camisetas.c  
export_camisetas_mejorado.c export_lesiones.c export_lesiones_mejorado.c  
export_partidos.c export_estadisticas.c export_estadisticas_generales.c  
export_records_rankings.c import.c utils.c sqlite3.c cJSON.c -I. $WIN_LIBS  
  
# 4. Ejecutar  
.\MiFutbolC.exe
```

## Método 4: Script del Proyecto para Linux/macOS

El proyecto incluye un instalador/compilador para terminal:

```
chmod +x Instalador-Linux.sh  
./Instalador-Linux.sh  
  
# Debug  
./Instalador-Linux.sh --debug
```

```
# Compilar y ejecutar
./Instalador-Linux.sh run
```

`Instalador-Linux.sh` puede verificar dependencias, compilar y opcionalmente agregar launcher a `PATH`.

## Método 5: Instalación por Consola en Windows (PowerShell)

```
# Instalación estándar (compila + instala)
powershell -ExecutionPolicy Bypass -File .\install.ps1

# Instalación Debug
powershell -ExecutionPolicy Bypass -File .\install.ps1 -BuildType Debug

# Usar un .exe ya compilado
powershell -ExecutionPolicy Bypass -File .\install.ps1 -SkipBuild
```

El script `install.ps1` instala en `%LOCALAPPDATA%\Programs\MiFutbolC`, crea launcher de consola (`MiFutbolC.cmd`) y puede agregar el directorio al `PATH` de usuario.

### Notas Importantes

- **Base de Datos:** Se crea automáticamente en la primera ejecución (ver sección [Base de Datos](#) para más detalles)
- **Codificación:** El programa usa UTF-8 para soportar caracteres especiales
  - En Windows, se configura automáticamente la consola
- **Permisos:** En Linux/macOS, asegúrate de tener permisos de escritura en el directorio

### Cambios recientes UI/Consola (Windows)

- **Limpieza de consola completa:** evita que al hacer scroll aparezcan menús anteriores.
- **Listados con fecha + día:** Partidos, Lesiones y Transacciones muestran fecha con día de semana.
- **Compilación MinGW:** usa `WIN_LIBS="-lcomdlg32 -lshell32 -lbcrypt"` para prevenir errores de linkeo.

### Solución de Problemas Comunes

#### Error: "sqlite3.h: No such file or directory"

- **Solución:** SQLite3 está incluido en el proyecto. Asegúrate de compilar desde el directorio raíz.

#### Error: "undefined reference to..."

- **Solución:** Verifica que todos los archivos `.c` estén incluidos en el comando de compilación.
- En Windows (MinGW), usa las librerías del bloque `WIN_LIBS` de la sección de compilación.

### Caracteres extraños en la consola (Windows)

- **Solución:** El programa configura automáticamente UTF-8. Si persiste, ejecuta:

```
chcp 65001
```

## No se crea la base de datos

- **Solución:** Verifica permisos de escritura en el directorio `data/` o `%LOCALAPPDATA%\MiFutbolC\data\`

## Uso

### Inicio del Programa

Al ejecutar `MiFutbolC`, el sistema:

1. **Abre el inicio de sesión multiusuario local**
2. **Selecciona o crea un perfil local** (opcionalmente con contraseña)
3. **Inicializa la base de datos del perfil activo**
4. **Carga la configuración** del usuario (tema, idioma)
5. **Muestra el menú principal**

### Menú Principal

---

---

MIFUTBOLC - MENÚ PRINCIPAL

---

---

1. Dashboard
  2. Calendario
  3. Camisetas
  4. Canchas
  5. Equipos
  6. Partidos
  7. Lesiones
  8. Estadísticas
  9. Logros
  10. Financiamiento
  11. Torneos
  12. Temporada
  13. Análisis
  14. Bienestar
  15. Carrera Futbolística
  16. Recordatorios
  17. Colecciones
  18. Ajustes
  19. Música
  0. Salir
- >

## Flujo de Trabajo Típico

### 1 Configuración Inicial

```
Menú Principal → Ajustes (18)
├── Cambiar tema de interfaz
├── Cambiar idioma
└── Usuario (nombre visible, contraseña y cuentas locales)
```

### 2 Crear Recursos Básicos

```
Menú Principal → Camisetas (3) → Crear camiseta
Menú Principal → Canchas (4) → Crear cancha
Menú Principal → Equipos (5) → Crear equipo
```

### 3 Registrar un Partido

```
Menú Principal → Partidos (6) → Crear partido
├── Seleccionar cancha
├── Seleccionar camiseta
├── Ingresar goles y asistencias
├── Evaluar rendimiento (1-10)
├── Indicar cansancio (1-10)
└── Indicar estado de ánimo (1-10)
```

### 4 Organizar un Torneo

```
Menú Principal → Torneos (11)
├── Crear torneo
├── Listar torneos
├── Modificar torneo
└── Eliminar torneo
```

### 5 Analizar Rendimiento

```
Menú Principal → Estadísticas (8)
├── Ver estadísticas generales
├── Estadísticas por año
├── Estadísticas por mes
└── Meta-análisis
```

```
Menú Principal → Análisis (13)
```

- |— Análisis Básico
- |— Comparador Avanzado
- |— Análisis Táctico (Diagramas)
- |— Entrenador IA
- |— Química Entre Jugadores

Análisis → Química Entre Jugadores

- |— Mejor Combinación de Jugadores
- |— Agregar Estadística de Jugador
- |— Listar Estadísticas de Jugadores
- |— Editar Estadística de Jugador
- |— Eliminar Estadística de Jugador

Menú Principal → Bienestar (14)

- |— Planificación Personal
- |— Mentalidad y Hábitos
- |— Entrenamiento
- |— Alimentación
- |— Salud

## 6 Gestionar Finanzas

Menú Principal → Financiamiento (10)

- |— Registrar ingreso
- |— Registrar gasto
- |— Ver balance
- |— Presupuestos mensuales

## 7 Gestionar Recordatorios

Menú Principal → Recordatorios (16)

- |— Listar recordatorios
- |— Agregar/editar/eliminar recordatorio
- |— Filtrar por temática
- |— Agenda (eventos próximos y pasados)

## 8 Gestionar Inventario y Colecciones

Menú Principal → Colecciones (17)

- |— Crear y listar inventario
- |— Sincronizar camisetas al inventario
- |— Crear y listar colecciones
- |— Agregar/quitar ítems a colecciones
- |— Exportar/Importar backup JSON

## 9 Exportar Datos

```
Menú Principal → Ajustes (18) → Exportar
├─ Seleccionar módulo (camisetas, partidos, etc.)
├─ Elegir formato (CSV, JSON, HTML, TXT)
└─ Archivos guardados en el directorio de exportaciones (ver sección [Base de
Datos](#-base-de-datos))
```

El informe PDF total incluye secciones adicionales con resúmenes financieros, ranking de canchas, partidos por clima, lesiones por tipo/estado, historial de rachas y distribución de estado de ánimo/cansancio.

### Ejemplos de Uso Práctico

#### Ejemplo 1: Registrar tu Primera Camiseta

1. Ejecuta el programa
2. Selecciona **3** (Camisetas)
3. Selecciona **1** (Crear camiseta)
4. Ingresa el nombre: "**Camiseta Roja**"
5. La camiseta se guarda automáticamente con ID único

#### Ejemplo 2: Crear un Torneo de Fútbol 5

1. Crea al menos 4 equipos (Menú → Equipos → Crear)
2. Ve a Torneos (opción 11)
3. Selecciona "Crear torneo"
4. Ingresa nombre: "**Copa Primavera 2026**"
5. Selecciona formato: **Round Robin**
6. Agrega los equipos participantes
7. El sistema genera automáticamente el fixture

#### Ejemplo 3: Analizar tu Rendimiento

1. Ve a Análisis (opción 13)
2. El sistema muestra:
  - Comparación últimos 5 partidos vs promedio general
  - Mejor racha de victorias
  - Peor racha de derrotas
  - Mensajes motivacionales personalizados

### Navegación

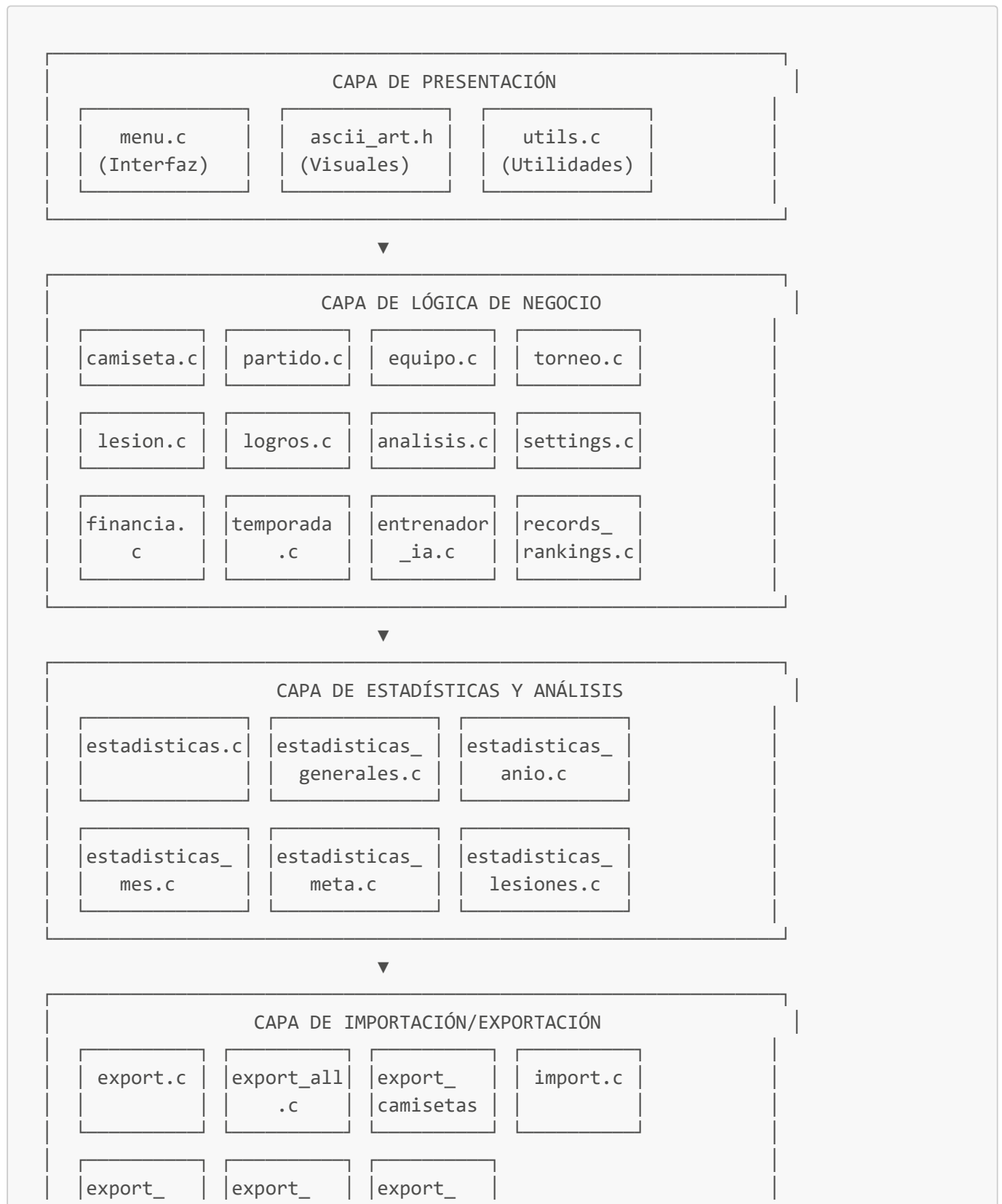
- **Números:** Selecciona opciones ingresando el número correspondiente
- **0:** Volver al menú anterior o salir
- **Enter:** Confirmar selección
- **Ctrl+C:** Salir del programa (en cualquier momento)

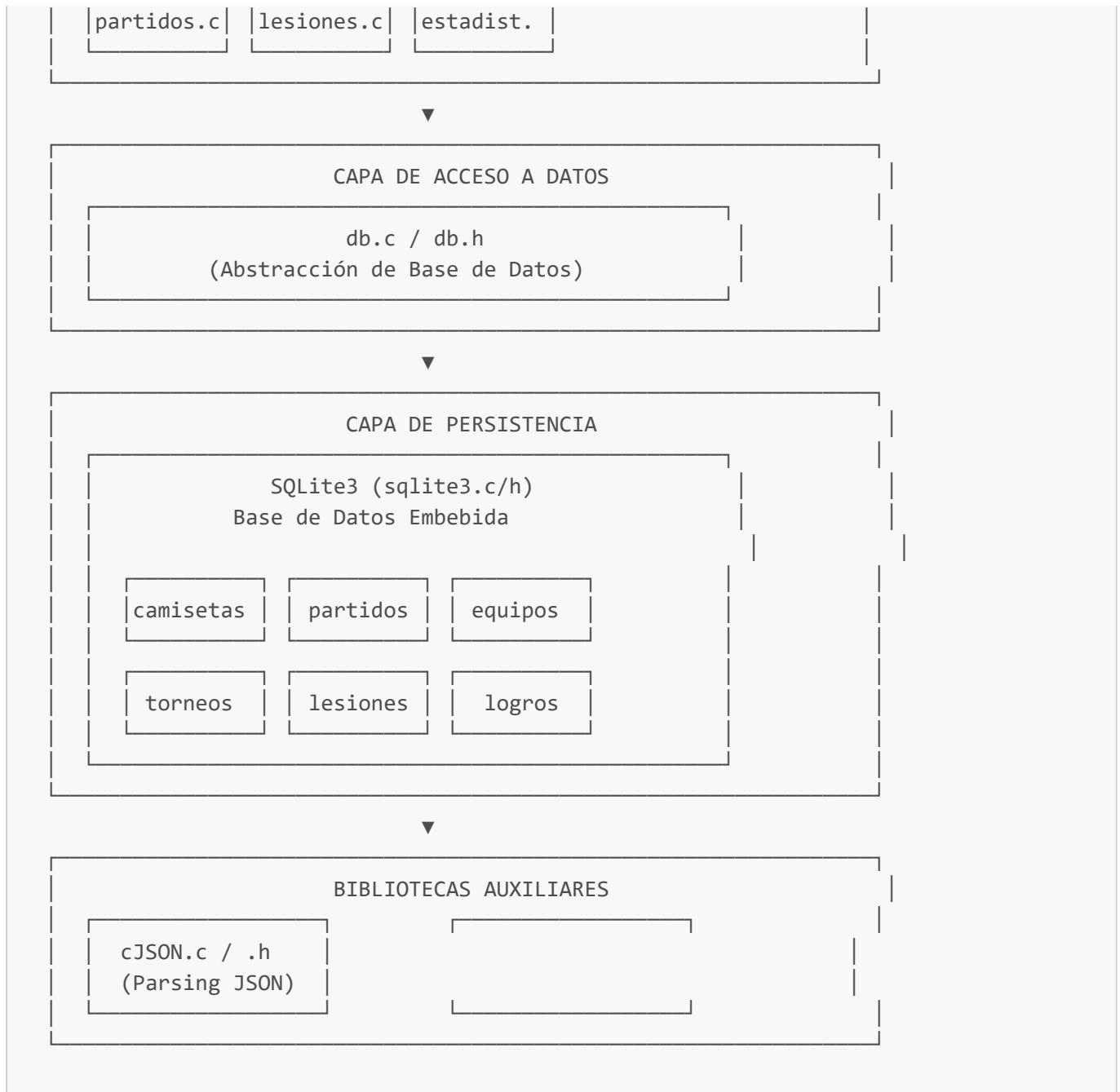
## Atajos de Teclado (en desarrollo)

- **Ctrl+E**: Exportar datos rápidamente
- **Ctrl+S**: Ver estadísticas rápidas
- **Ctrl+H**: Ayuda contextual

## Arquitectura y Diseño

### Diagrama de Arquitectura





## Patrones de Diseño Implementados

### 1. Patrón Modular

- **Descripción:** Cada módulo tiene responsabilidad única y bien definida
- **Implementación:** Archivos `.h` (interfaces) y `.c` (implementaciones) separados
- **Beneficios:** Bajo acoplamiento, alta cohesión, fácil mantenimiento

### 2. Patrón de Comando (Simplificado)

- **Descripción:** Sistema de menús con estructura `MenuItem`
- **Implementación:**

```
typedef struct {
    int opcion;
```

```

const char* texto;
void (*accion)(void);
} MenuItem;

```

- **Beneficios:** Fácil agregar nuevas opciones, navegación flexible

### 3. Patrón Repository (Capa de Acceso a Datos)

- **Descripción:** `db.c/db.h` centraliza todas las operaciones de base de datos
- **Implementación:** Funciones como `db_init()`, `db_close()`, consultas SQL encapsuladas
- **Beneficios:** Abstracción de la persistencia, cambio de BD sin afectar lógica

### 4. Patrón Strategy (Exportación)

- **Descripción:** Múltiples estrategias de exportación (CSV, JSON, HTML, TXT)
- **Implementación:** Funciones especializadas por formato en módulos `export_*.c`
- **Beneficios:** Fácil agregar nuevos formatos, código reutilizable

### 5. Patrón Singleton (Conexión a BD)

- **Descripción:** Una única conexión a la base de datos durante toda la ejecución
- **Implementación:** Variable global `sqlite3 *db` en `db.c`
- **Beneficios:** Eficiencia, consistencia de datos

## Principios de Diseño

### SOLID Principles (Adaptados a C)

- **S - Single Responsibility:** Cada módulo tiene una responsabilidad única
  - `camiseta.c` → Solo gestión de camisetas
  - `partido.c` → Solo gestión de partidos
- **O - Open/Closed:** Abierto a extensión, cerrado a modificación
  - Nuevos formatos de exportación sin modificar código existente
- **L - Liskov Substitution:** Funciones intercambiables con misma interfaz
  - Todas las funciones de exportación tienen firma similar
- **I - Interface Segregation:** Interfaces específicas y pequeñas
  - Headers `.h` con solo declaraciones necesarias
- **D - Dependency Inversion:** Dependencia de abstracciones
  - Módulos dependen de `db.h`, no de SQLite directamente

## Flujo de Datos










```

Usuario → Menu → Módulo de Negocio → db.c → SQLite → Archivo .db
      ↓
      Estadísticas
      ↓
      Exportación → Archivos (CSV/JSON/HTML/TXT)
  
```

## 📁 Estructura del Proyecto

```

MiFutbolC/
├── main.c # Punto de entrada del programa
├── CAPA DE DATOS
│   ├── db.c / db.h # Gestión de base de datos SQLite
│   ├── sqlite3.c / sqlite3.h # Biblioteca SQLite embebida
│   └── models.h # Definiciones de estructuras
├── comunes
│   ├── CAPA DE PRESENTACIÓN
│   │   ├── menu.c / menu.h # Sistema de menús interactivos
│   │   ├── ascii_art.h # Arte ASCII para interfaz
│   │   └── utils.c / utils.h # Utilidades auxiliares
│   ├── CAPA DE LÓGICA DE NEGOCIO
│   │   ├── Gestión de Recursos
│   │   │   ├── camiseta.c / camiseta.h # Gestión de camisetas
│   │   │   ├── cancha.c / cancha.h # Gestión de canchas
│   │   │   └── equipo.c / equipo.h # Gestión de equipos
│   │   ├── Gestión de Partidos y Torneos
│   │   │   ├── partido.c / partido.h # Gestión de partidos
│   │   │   ├── torneo.c / torneo.h # Gestión de torneos
│   │   │   └── temporada.c / temporada.h # Gestión de temporadas
│   │   ├── Gestión de Salud
│   │   │   └── lesion.c / lesion.h # Gestión de lesiones
│   │   ├── Gestión Financiera
│   │   │   └── financiamiento.c / financiamiento.h # Control de ingresos/gastos
│   │   ├── Gamificación
│   │   │   └── logros.c / logros.h # Sistema de logros y badges
│   │   ├── Inteligencia Artificial
│   │   │   └── entrenador_ia.c / entrenador_ia.h # Entrenador IA
│   │   ├── Bienestar y Productividad
│   │   │   ├── bienestar.c / bienestar.h # Módulo de bienestar integral
│   │   │   ├── carrera.c / carrera.h # Carrera futbolística
│   │   │   └── recordatorios.c / recordatorios.h # Agenda y recordatorios
  
```

└─	colecciones.c / colecciones.h	# Inventario y colecciones
└─	 Interfaz de Soporte	
└─	└─ dashboard.c / dashboard.h	# Dashboard del sistema
└─	└─ calendario.c / calendario.h	# Calendario y eventos
└─	 Audio	
└─	└─ musica.c / musica.h	# Reproductor de música
└─	└─ musica_helpers.c / musica_helpers.h	# Helpers de audio
└─	 Configuración	
└─	└─ settings.c / settings.h	# Sistema de configuración
└─	 CAPA DE ESTADÍSTICAS Y ANÁLISIS	
└─	└─ estadisticas.c / estadisticas.h	# Estadísticas generales
└─	└─ estadisticas_generales.c / .h	# Estadísticas detalladas
└─	└─ estadisticas_anio.c / .h	# Estadísticas por año
└─	└─ estadisticas_mes.c / .h	# Estadísticas por mes
└─	└─ estadisticas_meta.c / .h	# Meta-análisis avanzado
└─	└─ estadisticas_lesiones.c / .h	# Estadísticas de lesiones
└─	└─ analisis.c / analisis.h	# Análisis de rendimiento
└─	└─ records_rankings.c / .h	# Récords y rankings
└─	 CAPA DE IMPORTACIÓN/EXPORTACIÓN	
└─	└─  Importación	
└─	└─ import.c / import.h	# Importación multiformato
└─	└─  Exportación	
└─	└─ export.c / export.h	# Exportación individual
└─	└─ export_all.c / .h	# Exportación completa
└─	└─ export_all_mejorado.c / .h	# Exportación mejorada completa
└─	└─ export_camisetas.c / .h	# Exportación de camisetas
└─	└─ export_camisetas_mejorado.c / .h	# Exportación mejorada camisetas
└─	└─ export_lesiones.c / .h	# Exportación de lesiones
└─	└─ export_lesiones_mejorado.c / .h	# Exportación mejorada lesiones
└─	└─ export_partidos.c / .h	# Exportación de partidos
└─	└─ export_partidos_helpers.h	# Helpers para exportación
└─	└─ export_estadisticas.c / .h	# Exportación de estadísticas
└─	└─ export_estadisticas_generales.c / .h	# Exportación estadísticas
└─	generales	
└─	└─ export_records_rankings.c / .h	# Exportación récords/rankings
└─	 BIBLIOTECAS INCLUIDAS	
└─	└─ sqlite3.c / sqlite3.h	# SQLite embebido
└─	└─ cJSON.c / cJSON.h	# Biblioteca cJSON (MIT License)
└─	└─ pdfgen.c / pdfgen.h	# Motor PDF interno
└─	└─ miniaudio.h	# Motor de audio (header-only)
└─	 ARCHIVOS DE CONFIGURACIÓN Y BUILD	
└─	└─ MiFutbolC.cbp	# Proyecto CodeBlocks
└─	└─ MiFutbolC.layout	# Layout de CodeBlocks
└─	└─ MiFutbolC.depend	# Dependencias del proyecto
└─	└─ MiFutbolC.cscope_file_list	# Lista de archivos cscope

```

├── MiFutbolC.iss # Script instalador Inno Setup
├── Instalador-Linux.sh # Script instalación/compilación
Linux/macOS
├── install.ps1 # Script instalación por consola
(Windows)
├── recurso.rc # Recursos de Windows
├── sonar-project.properties # Configuración SonarQube
├── 🎨 RECURSOS
├── MiFutbolC.ico # Icono del programa
├── images/ # Imágenes y capturas
│   └── *.png
├── 📄 DOCUMENTACIÓN
├── README.md # Este archivo
├── README.pdf # Versión PDF del README
├── LICENSE # Licencia del proyecto
├── manual_usuario.md # Manual de usuario
├── Manual_Usuario_MiFutbolC.pdf # Manual de usuario PDF
├── 🏠 DIRECTORIOS DE COMPILACIÓN
├── bin/ # Binarios compilados
│   ├── Debug/
│   │   ├── MiFutbolC.exe # Ejecutable de depuración
│   │   └── data/ # Datos del ejecutable
│   └── obj/ # Archivos objeto (.o)
│       └── Debug/
├── bw-output/ # Salida de análisis
├── 💾 DATOS Y EXPORTACIONES (runtime)
├── data/ # Se crea en ejecución
│   ├── users.db # Registro de usuarios locales
│   ├── mifutbol_<usuario>.db # Base de datos por perfil
│   ├── mifutbol_<usuario>.log # Log por perfil
│   ├── *.csv # Archivos exportados CSV
│   ├── *.txt # Archivos exportados TXT
│   ├── *.json # Archivos exportados JSON
│   └── *.html # Archivos exportados HTML
├── 📦 INSTALADOR
├── installer/
├── MiFutbolC_Setup.exe # Instalador para Windows

```

## Resumen de Archivos

Conteo referencial (puede variar según cambios y scripts de build).

Categoría	Archivos .c	Archivos .h	Total
Lógica de Negocio	12	12	24

Categoría	Archivos .c	Archivos .h	Total
Estadísticas	7	7	14
Exportación	11	11	22
Importación	1	1	2
Interfaz	2	3	5
Base de Datos	2	2	4
Bibliotecas	2	2	4
<b>TOTAL</b>	<b>37</b>	<b>38</b>	<b>75</b>

## Módulos Principales

Módulo	Archivos	Descripción
<b>Core</b>	<code>main.c</code> , <code>db.c</code> , <code>menu.c</code> , <code>utils.c</code> , <code>dashboard.c</code> , <code>calendario.c</code>	Núcleo del sistema e interfaz base
<b>Gestión</b>	<code>camiseta.c</code> , <code>cancha.c</code> , <code>equipo.c</code> , <code>partido.c</code> , <code>torneo.c</code> , <code>temporada.c</code> , <code>lesion.c</code>	Gestión deportiva principal
<b>Análisis</b>	<code>estadisticas*.c</code> , <code>analisis.c</code> , <code>records_rankings.c</code>	Análisis y estadísticas
<b>I/O</b>	<code>export*.c</code> , <code>import.c</code>	Importación/Exportación
<b>Bienestar y Productividad</b>	<code>bienestar.c</code> , <code>carrera.c</code> , <code>recordatorios.c</code> , <code>colecciones.c</code>	Bienestar, agenda y colecciones
<b>Audio</b>	<code>musica.c</code> , <code>musica_helpers.c</code> , <code>miniaudio.h</code>	Reproductor y procesamiento de audio
<b>Extras</b>	<code>logros.c</code> , <code>financiamiento.c</code> , <code>entrenador_ia.c</code> , <code>settings.c</code>	Funcionalidades transversales

## Base de Datos

### Información General

- **Motor:** SQLite3 (embebido)
- **Archivos de datos por perfil:** `mifutbol_<usuario>.db`
- **Registro de usuarios locales:** `users.db`
- **Ubicación:**
  - Windows: `%LOCALAPPDATA%\MiFutbolC\data\mifutbol_<usuario>.db`
  - Linux/macOS: `./data/mifutbol_<usuario>.db`
- **Archivo de usuarios:**
  - Windows: `%LOCALAPPDATA%\MiFutbolC\data\users.db`
  - Linux/macOS: `./data/users.db`
- **Archivo de log:**

- Windows: %LOCALAPPDATA%\MiFutbolC\data\mifutbol\_<usuario>.log
- Linux/macOS: ./data/mifutbol\_<usuario>.log
- **Características:** Transacciones ACID, sin servidor, archivo único

## Directorios de importación/exportación

- **Exportaciones (Windows):** %USERPROFILE%\Documents\MiFutbolC\Exportaciones
- **Importaciones (Windows):** %USERPROFILE%\Documents\MiFutbolC\Importaciones
- **Exportaciones (Linux/macOS):** ./exportaciones
- **Importaciones (Linux/macOS):** ./importaciones

## Esquema (resumen)

El esquema se crea en `db.c` y puede evolucionar con `ALTER TABLE` automáticos. Tablas principales (resumen):

- **Base:** camiseta, cancha, partido, lesion, usuario, quimica\_jugador\_estadistica
- **Equipos/Torneos:** equipo, jugador, torneo, equipo\_torneo, partido\_torneo, equipo\_torneo\_estadisticas, jugador\_estadisticas, equipo\_historial, torneo\_fases, equipo\_fase
- **Temporadas:** temporada, temporada\_fase, torneo\_temporada, equipo\_temporada\_fatiga, jugador\_temporada\_fatiga, equipo\_temporada\_evolucion, temporada\_resumen, mensual\_resumen
- **Configuración/Finanzas:** settings, financiamiento, presupuesto\_mensual, comparacion\_historial

Campos clave (selección, no exhaustivo):

- camiseta: id, nombre, sorteada
- cancha: id, nombre
- partido: id, cancha\_id, fecha\_hora, goles, asistencias, camiseta\_id, resultado, rendimiento\_general, cansancio, estado\_animo, comentario\_personal, clima, dia, precio
- lesion: id, jugador, tipo, descripcion, fecha, camiseta\_id, estado, partido\_id
- equipo: id, nombre, tipo, tipo\_futbol, num\_jugadores, partido\_id
- torneo: id, nombre, tiene\_equipo\_fijo, equipo\_fijo\_id, cantidadequipos, tipo\_torneo, formato\_torneo, fase\_actual

Para el detalle completo, ver la creación del esquema en [db.c](#).

## Inicialización de la Base de Datos

La función `db_init()` en `db.c` se encarga de:

1. **Crear el directorio de datos** si no existe
2. **Abrir/crear la base de datos** SQLite
3. **Crear todas las tablas** con sus respectivos esquemas
4. **Aplicar migraciones simples** (`ALTER TABLE`) si faltan columnas
5. **Inicializar directorios de importación y exportación**

```
// Ejemplo simplificado
bool db_init() {
    if (!setup_database_paths()) return false;
    if (!create_database_connection()) return false;
    if (!create_database_schema()) return false;
    add_missing_columns();
    get_import_dir();
    get_export_dir();
    return true;
}
```

## Consultas Comunes

### Estadísticas por Camiseta

```
SELECT
    c.nombre,
    COUNT(p.id) as partidos,
    SUM(p.goles) as goles_totales,
    SUM(p.asistencias) as asistencias_totales,
    AVG(p.rendimiento_general) as rendimiento_promedio
FROM camiseta c
LEFT JOIN partido p ON c.id = p.camiseta_id
GROUP BY c.id
ORDER BY goles_totales DESC;
```

### Récords Históricos

```
SELECT
    MAX(goles) as max_goles,
    MAX(asistencias) as max_asistencias,
    MAX(rendimiento_general) as mejor_rendimiento
FROM partido;
```

### Análisis de Lesiones

```
SELECT
    tipo,
    COUNT(*) as cantidad
FROM lesion
GROUP BY tipo
ORDER BY cantidad DESC;
```

## Módulo de Estadísticas

El módulo de estadísticas proporciona información agregada sobre el rendimiento de las camisetas en los partidos:

- **Camiseta con más Goles:** Muestra la camiseta que ha acumulado el mayor número de goles en todos los partidos.
- **Camiseta con más Asistencias:** Identifica la camiseta con el mayor número de asistencias registradas.
- **Camiseta con más Partidos:** Lista la camiseta que ha sido utilizada en el mayor número de partidos.
- **Camiseta con más Goles + Asistencias:** Combina goles y asistencias para determinar la camiseta con mejor rendimiento global.

Las estadísticas se calculan en tiempo real mediante consultas SQL que unen las tablas de partidos y camisetas.

## Módulo de Análisis de Rendimiento

El módulo de análisis de rendimiento (`analisis.c`) ofrece una evaluación detallada del desempeño futbolístico mediante la comparación de los últimos 5 partidos con los promedios generales del sistema:

- **Comparación Últimos 5 vs Promedio General:** Analiza métricas como goles, asistencias, rendimiento general, cansancio y estado de ánimo, mostrando diferencias numéricas entre el rendimiento reciente y el histórico.
- **Cálculo de Rachas:** Determina la mejor racha de victorias consecutivas y la peor racha de derrotas consecutivas registradas.
- **Análisis Motivacional:** Proporciona mensajes personalizados basados en el rendimiento comparativo, ofreciendo motivación o consejos constructivos para mejorar.
- **Visualización de Últimos Partidos:** Muestra un resumen de los 5 partidos más recientes con detalles clave como fecha, goles, asistencias, rendimiento y resultado.
- **Química Entre Jugadores (CRUD):** Permite registrar y gestionar estadísticas por jugador-partido (goles, asistencias, asistencias al usuario, posición y comentario).
- **Mejor Combinación Híbrida:** Calcula la mejor dupla usando datos automáticos de equipos/partidos y datos manuales del módulo de química.

Este módulo utiliza consultas SQL avanzadas para calcular promedios y rachas, proporcionando insights valiosos para el seguimiento y mejora del rendimiento futbolístico.

## Sistema de Logros y Badges

El sistema de logros y badges (`logros.c`) implementa un sistema de recompensas basado en estadísticas conseguidas por las camisetas en partidos de fútbol, incentivando el progreso y el logro de metas:

- **Categorías de Logros:** Incluye logros por goles, asistencias, partidos jugados, contribuciones totales (goles + asistencias), victorias, empates, derrotas, rendimiento general, estado de ánimo, canchas distintas, hat-tricks, poker de asistencias, rendimiento perfecto, ánimo perfecto, y logros específicos en victorias, derrotas y empates.
- **Niveles de Dificultad:** Cada categoría tiene múltiples niveles (Novato, Promedio, Experto, Maestro, Leyenda) con objetivos progresivos.

- **Seguimiento de Progreso:** Muestra el progreso actual hacia cada logro, indicando si está no iniciado, en progreso o completado.
- **Visualización por Camiseta:** Permite ver todos los logros, solo los completados o solo los en progreso para una camiseta específica.
- **Interfaz de Menú:** Navegación intuitiva para explorar los logros disponibles.

Este sistema utiliza consultas SQL para calcular estadísticas acumuladas y determinar el estado de cada logro, proporcionando una experiencia gamificada para motivar el uso continuo del sistema.

## Módulo de Gestión de Equipos

El módulo de gestión de equipos ([equipo.c](#) / [equipo.h](#)) permite crear, gestionar y administrar equipos de fútbol con diferentes configuraciones:

- **Tipos de Equipos:** Soporte para equipos fijos (guardados en base de datos) y momentáneos (temporales).
- **Modalidades de Fútbol:** Compatible con Fútbol 5, Fútbol 7, Fútbol 8 y Fútbol 11.
- **Posiciones de Jugadores:** Definición de posiciones estándar (Arquero, Defensor, Mediocampista, Delantero) con posibilidad de designar capitanes.
- **Gestión de Plantillas:** Cada equipo puede tener hasta 11 jugadores con información completa (nombre, número, posición, capitán).
- **Operaciones CRUD:** Crear, listar, modificar y eliminar equipos con validación de datos.

## Módulo de Gestión de Torneos

El módulo de gestión de torneos ([torneo.c](#) / [torneo.h](#)) ofrece en el menú actual operaciones base de administración:

- **Crear torneo.**
- **Listar torneos.**
- **Modificar torneo.**
- **Eliminar torneo.**

Además, el esquema de base de datos incluye tablas de soporte para fases, equipos y estadísticas de torneo.

## Módulo de Gestión Financiera

El módulo de gestión financiera ([financiamiento.c](#) / [financiamiento.h](#)) permite llevar un control detallado de los ingresos y gastos del equipo de fútbol:

- **Tipos de Transacciones:** Soporte para ingresos (cuotas, sponsors, etc.) y gastos (transporte, equipamiento, etc.).
- **Categorías Específicas:** Clasificación detallada incluyendo transporte, equipamiento, cuotas, torneos, arbitraje, canchas, medicina y otros.
- **Gestión Completa:** Crear, listar, modificar y eliminar transacciones financieras.
- **Resúmenes Financieros:** Visualización de balances generales, gastos por categoría y estado financiero actual.
- **Exportación:** Funciones para exportar datos financieros en múltiples formatos.

Este módulo proporciona herramientas esenciales para la administración financiera del equipo, permitiendo un seguimiento preciso de los recursos económicos.

## Módulo de Récorde y Rankings

El módulo de récords y rankings ([records\\_rankings.c](#) / [records\\_rankings.h](#)) ofrece un análisis exhaustivo de los logros históricos y estadísticas destacadas del sistema:

- **Récords de Partidos:** Máximo de goles y asistencias en un partido, mejor y peor rendimiento general.
- **Top 5 Mejores Partidos:** Ranking de partidos por rendimiento general (con goles y asistencias).
- **Combinaciones Óptimas:** Mejor y peor combinación de cancha + camiseta para rendimiento.
- **Ranking de Camisetas:** Tabla comparativa por camiseta con partidos, victorias, empates, derrotas y winrate.
- **Temporadas Destacadas:** Mejor y peor temporada basada en estadísticas acumuladas.
- **Rachas:** Mejor racha goleadora, peor racha, partidos consecutivos anotando.
- **Partidos Especiales:** Partidos sin goles, sin asistencias, mejor combinación de goles + asistencias.
- **Análisis Comparativo:** Funciones para identificar patrones y tendencias históricas.

Este módulo utiliza consultas SQL avanzadas para extraer insights valiosos sobre el rendimiento histórico y ayudar en la toma de decisiones estratégicas.

## Módulo de Estadísticas por Año

El módulo de estadísticas por año ([estadisticas\\_anio.c](#) / [estadisticas\\_anio.h](#)) proporciona análisis longitudinal del rendimiento futbolístico agrupado por año:

- **Agregación Anual:** Estadísticas totales y promedios por camiseta para cada año.
- **Tendencias Temporales:** Seguimiento de la evolución del rendimiento a lo largo del tiempo.
- **Comparación Interanual:** Análisis comparativo entre diferentes años para identificar mejoras o declives.

Este módulo es fundamental para el análisis de largo plazo y la planificación estratégica del equipo.

## Módulo de Estadísticas por Mes

El módulo de estadísticas por mes ([estadisticas\\_mes.c](#) / [estadisticas\\_mes.h](#)) ofrece un desglose detallado del rendimiento mensual:

- **Desglose Mensual:** Estadísticas individuales por camiseta agrupadas por mes.
- **Métricas por Mes:** Partidos jugados, goles, asistencias y promedios mensuales.
- **Análisis de Estacionalidad:** Identificación de patrones estacionales en el rendimiento.

Este módulo permite un seguimiento granular del progreso del equipo y la identificación de tendencias a corto plazo.

## Módulo de Meta-Análisis

El módulo de meta-análisis ([estadisticas\\_meta.c](#) / [estadisticas\\_meta.h](#)) realiza análisis estadísticos avanzados para profundizar en el rendimiento futbolístico:

- **Consistencia de Rendimiento:** Evaluación de la variabilidad y estabilidad del desempeño.

- **Partidos Atípicos:** Identificación de partidos excepcionalmente buenos o malos (outliers).
- **Dependencia del Contexto:** Análisis de cómo factores externos afectan el rendimiento.
- **Impacto del Cansancio:** Evaluación del efecto real de la fatiga en el desempeño.
- **Impacto del Estado de Ánimo:** Análisis de la influencia emocional en los resultados.
- **Eficiencia:** Comparación entre rendimiento y producción (goles vs rendimiento, asistencias vs cansancio).
- **Rendimiento por Esfuerzo:** Evaluación de la relación entre el esfuerzo invertido y los resultados obtenidos.
- **Análisis de Situaciones:** Partidos exigentes bien jugados y partidos fáciles mal jugados.

Este módulo proporciona insights profundos para la optimización del rendimiento y la toma de decisiones tácticas.

## Utilidades y Funciones Auxiliares

El proyecto incluye un módulo de utilidades (`utils.c` / `utils.h`) que proporciona funciones comunes para:

- **Entrada de Datos:** `input_int()` y `input_string()` para leer enteros y cadenas del usuario con validación básica.
- **Manejo de Fecha/Hora:** `get_datetime()` para obtener fecha y hora actual en formato legible, y `get_timestamp()` para nombres de archivos.
- **Interfaz de Consola:** `clear_screen()`, `print_header()`, y `pause_console()` para mejorar la experiencia del usuario en terminal.
- **Validación de Datos:** `existe_id()` para verificar si un ID existe en una tabla de la base de datos.
- **Confirmaciones:** `confirmar()` para solicitar confirmación del usuario antes de operaciones destructivas.
- **Gestión de Exportaciones:** `get_export_dir()` para determinar y crear el directorio de exportación (Documents en Windows, `./exportaciones` en Unix/Linux).

Estas utilidades promueven la reutilización de código y mantienen una interfaz consistente en todo el programa.

## Sistema de Menús

El proyecto implementa un sistema de menús jerárquico y modular mediante las funciones en `menu.c` / `menu.h`:

- **Menú Principal:** Gestionado en `menu.c` (invocado desde `main.c`), presenta las opciones principales del sistema (Dashboard, Calendario, Camisetas, Canchas, Equipos, Partidos, Lesiones, Estadísticas, Logros, Financiamiento, Torneos, Temporada, Análisis, Bienestar, Carrera Futbolística, Recordatorios, Colecciones, Ajustes, Música, Salir).
- **Accesos internos:** El **Entrenador IA** se abre desde Análisis y **Exportar/Importar** desde Ajustes.
- **Submenús:** Cada módulo principal tiene su propio menú (ej. `menu_camisetas()`, `menu_canchas()`, `menu_partidos()`, `menu_logros()`, `menu_lesiones()`, `menu_financiamiento()`).
- **Estructura de Menú:** Utiliza la estructura `MenuItem` definida en `menu.h` para asociar opciones numéricas con textos descriptivos y funciones a ejecutar.
- **Navegación:** La función `ejecutar_menu()` maneja la lógica de mostrar opciones, leer selección del usuario y ejecutar la acción correspondiente.

- **Consistencia:** Todos los menús siguen el mismo patrón, facilitando la adición de nuevas funcionalidades.

## Exportación Completa

Además de las exportaciones individuales, el módulo `export_all.c` / `export_all.h` proporciona la función `exportar_todo()` que:

- Ejecuta automáticamente todas las funciones de exportación disponibles.
- Genera archivos en formatos CSV, TXT, JSON y HTML para camisetas, partidos, estadísticas y lesiones.
- Facilita la copia de seguridad completa de todos los datos del sistema.
- Es accesible desde Ajustes -> Exportar -> Todo.

## Exportación Mejorada

Los módulos de exportación mejorada (`export_all_mejorado.c`, `export_camisetas_mejorado.c`, `export_lesiones_mejorado.c`) proporcionan funcionalidades avanzadas de exportación con análisis integrado:

- **Análisis Avanzado para Camisetas:** Incluye eficiencia de goles/asistencias, porcentaje de victorias, métricas de rendimiento, evaluación de lesiones.
- **Análisis de Impacto para Lesiones:** Evaluación de gravedad, comparación de rendimiento antes/después, identificación de patrones.
- **Exportación Integral:** Genera archivos mejorados en todos los formatos con estadísticas adicionales.

## Importación Completa

El módulo `import.c` / `import.h` permite restaurar datos desde archivos JSON, TXT, CSV o HTML exportados previamente. También incluye opciones rápidas (Todo JSON/Todo CSV) e importación de base de datos. Ver sección [Características Principales](#) para detalles completos sobre la funcionalidad de importación.

## Documentación

### Documentación Disponible

El repositorio incluye documentación técnica y de usuario en formato Markdown y PDF:

- [README.md](#): Documentación técnica general del proyecto.
- [README.pdf](#): Versión PDF del README.
- [manual\\_usuario.md](#): Manual de usuario con flujos y pantallas.
- [Manual\\_Usuario\\_MiFutbolC.pdf](#): Manual de usuario en PDF.

Nota: Los PDF se generan a partir de los `.md` (por ejemplo con Pandoc) si necesitas regenerarlos.

## Desarrollo y Contribución

### Convenciones de Código

#### Estilo de Código

- **Estándar C:** C99/C11 compatible
- **Indentación:** 4 espacios (no tabs)
- **Nombres de Variables:** `snake_case` para variables y funciones
- **Nombres de Constantes:** `UPPER_CASE` para macros y constantes
- **Comentarios:** En español para consistencia con el proyecto

## Estructura de Archivos

```
// Ejemplo de estructura de archivo .h
#ifndef MODULO_H
#define MODULO_H

// Includes necesarios
#include <stdio.h>
#include "db.h"

// Constantes
#define MAX_NOMBRE 100

// Estructuras
typedef struct {
    int id;
    char nombre[MAX_NOMBRE];
} MiEstructura;

// Declaraciones de funciones
void crear_elemento(void);
void listar_elementos(void);
void modificar_elemento(int id);
void eliminar_elemento(int id);

#endif // MODULO_H
```

## Agregar Nuevas Funcionalidades

### Paso 1: Planificación

1. Define claramente la funcionalidad
2. Identifica las tablas de BD necesarias
3. Diseña la interfaz de usuario

### Paso 2: Implementación

1. **Crear archivos** `.h` y `.c` para el nuevo módulo
2. **Definir estructuras** en el header
3. **Implementar funciones** en el archivo `.c`
4. **Actualizar** `db.c` si requiere nuevas tablas
5. **Agregar al menú** en `menu.c`

### Paso 3: Integración

1. Actualizar `MiFutbolC.cbp` (CodeBlocks)
2. Actualizar flujo de build/instalación (`Makefile`, `Instalador-Linux.sh`, `install.ps1`)
3. Agregar funciones de exportación si aplica

### Paso 4: Documentación

1. Documentar funciones con comentarios Doxygen
2. Actualizar README.md
3. Agregar ejemplos de uso

## Testing

### Pruebas Manuales

- Compilar sin warnings
- Ejecutar todas las opciones del menú
- Verificar integridad de la base de datos
- Probar casos límite (valores nulos, IDs inválidos)
- Verificar exportación/importación

### Pruebas de Integración

- Crear datos de prueba
- Verificar relaciones entre tablas
- Probar flujos completos (crear torneo → agregar equipos → jugar partidos)

## Herramientas de Desarrollo Recomendadas

- **IDE:** CodeBlocks, VS Code, CLion
- **Debugger:** GDB, LLDB
- **Análisis Estático:** Cppcheck, SonarQube
- **Profiling:** Valgrind (Linux), Dr. Memory (Windows)
- **Documentación:** Doxygen

## Contribuir al Proyecto

Si deseas contribuir:

1. **Fork** el repositorio
2. **Crea una rama** para tu funcionalidad (`git checkout -b feature/nueva-funcionalidad`)
3. **Commit** tus cambios (`git commit -am 'Agregar nueva funcionalidad'`)
4. **Push** a la rama (`git push origin feature/nueva-funcionalidad`)
5. **Crea un Pull Request**

### Lineamientos para Pull Requests

- Describe claramente los cambios realizados

- Incluye ejemplos de uso si aplica
- Asegúrate de que el código compile sin errores
- Sigue las convenciones de código del proyecto

## Licencia

Este proyecto es de **código abierto**. Consulta el archivo [LICENSE](#) para más detalles.



## Bibliotecas de Terceros

Ver sección [Tecnologías Utilizadas](#) para detalles completos sobre SQLite3 y cJSON.

## Autor

### Thomas Hamer

Proyecto desarrollado como ejemplo educativo y de uso personal de programación en C con SQLite.







-  Email: [thomashamerutn@gmail.com]
-  GitHub: [thomashamer3/MiFutbolC](#)

## Agradecimientos

- **SQLite Team**: Por la excelente base de datos embebida
- **cJSON**: Por la biblioteca ligera de parsing JSON
- **Comunidad de C**: Por los recursos y documentación disponibles
- **CodeBlocks**: Por el excelente IDE multiplataforma

## Notas Adicionales

### Características Técnicas

-  **Persistencia de Datos**: Los datos se guardan automáticamente en SQLite entre ejecuciones
-  **Manejo de Errores**: El programa valida entradas y solicita confirmaciones para operaciones destructivas
-  **Compatibilidad**: Probado en Windows 11, Ubuntu 22.04, macOS 12+
-  **Interfaz Textual**: No requiere GUI, funciona en cualquier terminal con soporte UTF-8
-  **Dependencias incluidas**: SQLite3, cJSON, pdfgen y miniaudio están incluidos en el repositorio
-  **Sin DLLs**: No utiliza DLLs externas para ejecución

### Limitaciones Conocidas

- La interfaz es solo de consola (no hay GUI gráfica)
- Multiusuario local orientado a un mismo equipo/dispositivo (no multiusuario concurrente en red)
- Base de datos local (no hay sincronización en la nube)

## Enlaces Útiles

-  [Manual de Usuario](#)
-  [README \(técnico\)](#)

- [🐛 Reportar un Bug](#)
- [💡 Solicitar una Funcionalidad](#)
- [📖 SQLite Documentation](#)
- [🔑 CodeBlocks](#)

## Estadísticas del Proyecto

Referencial y sujeto a cambios:

- **Tamaño del código:** proyecto C modular (ver árbol del proyecto).
- **Tablas de BD:** esquema definido en [db.c](#).
- **Formatos de exportación:** CSV, JSON, HTML, TXT.
- **Estado:** proyecto educativo en evolución.

---

### **MiFutbolC - Gestión Profesional de Fútbol en C**

Desarrollado por Thomas Hamer.

[↑ Volver arriba](#)