

DeRevés - Plataforma Social y de Gestión Deportiva

"Jugá, conectá y competí. Todo desde una sola red."

DeRevés es una plataforma web completa para la gestión de complejos deportivos y organización de actividades deportivas sociales. Permite a jugadores reservar canchas, participar en torneos, organizar partidos sociales y conectar con la comunidad deportiva.

Resumen

DeRevés es una aplicación web pensada para gestionar complejos deportivos (cancha, reservas, torneos) y al mismo tiempo crear una red social deportiva para jugadores, organizadores y dueños de clubes. El proyecto incluye: modelo de datos, especificaciones UX/UI y reglas de moderación.

Este repositorio contiene documentación y artefactos de arquitectura y diseño para arrancar el desarrollo. No incluye (aún) la implementación completa del backend/frontend.

Características principales

- **Registro y perfiles de usuarios** (jugadores, dueños, organizadores) con **foto de perfil**.
 - Gestión de complejos y canchas (subdominios, fichas públicas).
 - Reservas y calendario de turnos.
 - Partidos sociales (crear/unirse a partidos).
 - Torneos (fixtures, inscripciones, resultados).
 - Red social: publicaciones, seguidores, logros.
 - Valoraciones y reviews de complejos y canchas.
 - Sistema de moderación y control parental para menores.
 - Soporte para login con cuentas Google (Gmail / OAuth2).
-

PROF

Tech stack sugerido

- Backend: Django + Django REST Framework
 - Frontend: HTML5, Bootstrap 5 (prototipo), JS; posible migración a Vue/React
 - DB: PostgreSQL / MySQL
 - Deploy: Docker, Nginx, S3 (media)
-

Arranque rápido (guía para desarrolladores)

Estos pasos son una guía general. Adapta según tus preferencias (pyenv/poetry/pipenv, Docker, etc.).

1. Crear un entorno virtual y activar:

```
python -m venv .venv
source .venv/bin/activate
```

2. Instalar dependencias (si existe `requirements.txt`):

```
pip install -r requirements.txt
```

3. Configurar variables de entorno (ej.: `SECRET_KEY`, `DATABASE_URL`, `GOOGLE_OAUTH_CLIENT_ID`, `GOOGLE_OAUTH_CLIENT_SECRET`).

4. Ejecutar migraciones y crear superusuario:

```
python manage.py migrate
python manage.py createsuperuser
```

5. Levantar el servidor de desarrollo:

```
python manage.py runserver
```

Nota: si prefieres Docker, monta servicios para DB y Redis según convenga.

Documentación (archivos en `/proyecto/docs`)

- `proyecto/derevesV1.md` — Visión general, MVP y roadmap.
- `proyecto/docs/diagrama_datos.md` — Diagrama de datos y decisiones de modelado (incluye campos de usuario y complejo).
- `proyecto/docs/seguridad_y_moderacion.md` — Reglas de moderación, flujo para menores y tutores.
- `proyecto/docs/patron_ui.md` — Patrón de diseño UI y sistema de diseño.
- `proyecto/docs/deiseñoux-ui.md` — Filosofía UX y flujos por tipo de usuario.
- `proyecto/docs/wireframes_base.md` — Wireframes base y pantallas principales.
- `proyecto/docs/models_moderacion_y_tutores.md` — (archivo actualmente vacío, reservado para modelos específicos de moderación/tutores).

Revisa esos MD para detalles de diseño y reglas de negocio antes de implementar los `models.py` y vistas.

Decisiones de modelado y notas importantes

- Usuarios:

- Se sugiere extender `AbstractUser` para añadir campos:
 - `dni` (string): identificación real — manejar como dato sensible (cifrado/retención/consentimiento).
 - `nombre_real` (string): nombre completo para facturación y confianza.
 - `username` puede ser opcional como alias público; el login puede apoyarse en email/Google.
- Integración con Google OAuth2 (recomendada mediante `django-allauth` o `social-auth-app-django`).
- Complejos:
 - Guardar `latitud` / `longitud` (WGS84) para búsquedas por proximidad.
 - `google_place_id`, `direccion_formateada`, `google_maps_url` son campos útiles si usas Google Places API para autocomplete.
- Moderación / Menores:
 - Flujos para cuentas menores: estado `PENDIENTE_TUTOR`, `SUPERVISADA`, `ADULTO_VALIDADO`.
 - Tabla `ReporteContenido` y panel de moderación para revisiones manuales.

Integración con Google (Gmail login)

- Recomendación: usar `django-allauth` o `social-auth-app-django`.
- Configurar scopes mínimos (`openid email profile`) y guardar `google_oauth2_id` y `email_verified`.
- Flujo: vinculación de cuentas por email, evitar sobrescribir datos sin consentimiento.

Diseño, UX y recursos

Consulta los MD en `proyecto/docs` para los patrones de diseño, paleta de colores, tipografía y wireframes. Estos documentos sirven como base para el Design System.

—
PROF

Contribuir

1. Crear un fork / rama feature/issue.
2. Abrir PR contra `main` con descripción clara y tickets referenciados.
3. Incluir tests para cambios de negocio crítico (reservas, pagos, moderación).

Sugiero abrir issues en GitHub para cada feature grande: autenticación, reservas, torneos, social, pago.

Gestión de Complejos y Canchas

Sistema completo para que los dueños gestionen sus complejos deportivos:

 Gestionar Complejo (`/complejos/<slug>/gestionar/`)

Logo del Complejo

- ✓ **Subir logo:** Upload de imagen para el complejo
- ✓ **Preview:** Visualización del logo actual
- ✓ **Formatos:** JPG, PNG, GIF (máximo 5MB)

Servicios y Comodidades

- ✓ **Selección múltiple** con checkboxes:
 - Bufet
 - Parrilla
 - Agua Caliente
 - Wi-Fi
 - Salón
 - Estacionamiento
 - Vestuarios
 - Quincho
- ✓ **Guardado automático:** Los servicios se actualizan al hacer clic en "Guardar Servicios"

Tabla de Canchas

- ✓ **Vista completa** de todas las canchas
- ✓ **Información mostrada:**
 - Foto miniatura (60x60px)
 - Nombre y deporte
 - Tipo de superficie
 - Características (Techada, Iluminada)
 - Precio por hora
 - Horario de funcionamiento
 - Estado (Activa/Inactiva)
- ✓ **Acciones:**
 - Editar cancha
 - Activar/Desactivar

—
PROF

+ Agregar Cancha (</complejos/<slug>/canchas/agregar/>)

Formulario completo para crear nuevas canchas:

Información Básica

- **Nombre:** Identificador de la cancha
- **Deporte:** Selector con opciones (Pádel, Fútbol 5, Fútbol 7, Fútbol 11, Tenis, Básquet, Voley)
- **Foto:** Upload opcional de imagen de la cancha

Características

- **Tipo de superficie:** Césped sintético, cemento, etc.

- **Precio por hora:** Campo numérico con decimales
- **Techada:** Checkbox
- **Iluminación:** Checkbox

Configuración de Horarios

- **Horario apertura:** Time picker (default 08:00)
- **Horario cierre:** Time picker (default 23:00)
- **Duración del turno:** Selector (30, 60, 90, 120 minutos)

Editar Cancha (/complejos/<slug>/canchas/<id>/editar/)

- ✓ **Formulario prellenado** con datos actuales
- ✓ **Preview de foto** si existe
- ✓ **Cambiar foto:** Upload de nueva imagen
- ✓ **Modificar todos los campos**
- ✓ **Validaciones:** Campos requeridos marcados

Activar/Desactivar Cancha

- ✓ **Toggle rápido:** Un clic para cambiar estado
- ✓ **Confirmación:** Dialog de confirmación
- ✓ **Feedback visual:** Badge de estado actualizado
- ✓ **Sin eliminación:** Las canchas se desactivan, no se borran

Gestión de Imágenes

Logo del Complejo

- **Ubicación:** `media/complejos/logos/`
- **Mostrado en:**
 - Card del complejo en dashboard
 - Vista de gestión
 - Vista pública del complejo
 - Estadísticas

Fotos de Canchas

- **Ubicación:** `media/canchas/`
- **Mostrado en:**
 - Tabla de gestión (thumbnail 60x60)
 - Formulario de edición (preview)
 - Vista pública del complejo (200px altura)
 - Calendario de reservas
- **Fallback:** Icono de deporte si no hay foto

Flujo de Trabajo

1. **Crear Complejo** → </complejos/crear/>
2. **Gestionar** → </complejos/<slug>/gestionar/>
3. **Subir Logo** → Formulario en gestión
4. **Configurar Servicios** → Checkboxes en gestión
5. **Agregar Canchas** → Botón "Agregar Cancha"
6. **Configurar cada cancha:** Nombre, deporte, foto, precios, horarios
7. **Activar/Desactivar** según disponibilidad

Seguridad

- ✓ Verificación de propiedad del complejo
- ✓ Solo el dueño puede gestionar
- ✓ Validaciones de formularios
- ✓ Manejo seguro de uploads
- ✓ Confirmación en acciones críticas

Dashboard para Dueños de Complejos

El sistema incluye un **dashboard completo** para que los dueños de complejos puedan gestionar su negocio:

Funcionalidades

1. Dashboard Principal (</complejos/dashboard/>)

- **Estadísticas generales:**
 - Total de complejos y canchas
 - Reservas del mes (confirmadas, pendientes, canceladas)
 - Ingresos del mes
 - Tasa de ocupación
- **Gráfico de reservas:** Visualización de reservas de los últimos 7 días (Chart.js)
- **Canchas más populares:** Top 5 canchas más reservadas
- **Próximas reservas:** Tabla con reservas de hoy y mañana
- **Accesos rápidos:** Cards para navegar a otras secciones

2. Mis Complejos (</complejos/dashboard/complejos/>)

- Vista en cards de todos los complejos del dueño
- Información resumida: número de canchas, reservas del mes
- Acceso rápido a:
 - Ver complejo público
 - Estadísticas detalladas
 - Editar complejo
 - Configuración

3. Gestionar Reservas (</complejos/dashboard/reservas/>)

- **Tabla completa** de todas las reservas
- **Filtros avanzados:**
 - Por estado (pendiente, confirmada, cancelada)
 - Por complejo
 - Por rango de fechas
 - Por estado de pago
- **Estadísticas en tiempo real** según filtros aplicados
- **Acciones rápidas:** Ver detalle, confirmar reservas pendientes
- Información del jugador con foto de perfil

4. Estadísticas por Complejo (/complejos/dashboard/<slug>/estadisticas/)

- Estadísticas detalladas de un complejo específico
- Información general del complejo
- **Tabla de rendimiento por cancha:**
 - Total de reservas
 - Reservas confirmadas
 - Ingresos por cancha
 - Estado activo/inactivo
- Total de ingresos del complejo

Características de Diseño

- **Responsive:** Adaptado a móviles, tablets y escritorio
- **Cards interactivas** con efectos hover
- **Badges de estado** con colores semánticos
- **Iconos Material Symbols** para mejor UX
- **Gráficos con Chart.js** para visualización de datos
- **Breadcrumbs** para navegación clara

Seguridad

- Todas las vistas protegidas con `@login_required`
- Verificación de tipo de usuario (**DUENIO**)
- Solo se muestran datos de complejos propios
- Validación de permisos en cada acción

Tecnologías Utilizadas

- **Backend:** Django views con queries optimizadas (select_related, prefetch_related)
- **Frontend:** Bootstrap 5, Chart.js 4.4.0
- **Base de datos:** Queries con agregaciones (Count, Sum, Avg)
- **Filtros:** QuerySets dinámicos basados en GET parameters

Acceso

Los dueños pueden acceder al dashboard desde:

1. Navbar → Dropdown de usuario → "Dashboard"
 2. Mi Perfil → Sidebar → "Dashboard"
 3. URL directa: `/complejos/dashboard/`
-

Fotos de Perfil

El sistema soporta **fotos de perfil** para todos los tipos de usuario (jugadores y dueños de complejo):

Características

- ✓ Subida de imágenes JPG, PNG o GIF (máximo 5MB)
- ✓ Almacenamiento en `/media/perfiles/`
- ✓ Visualización en:
 - Navbar (avatar circular 32x32px)
 - Mi Perfil (120x120px)
 - Perfil Público (150x150px)
 - Formulario de edición con preview
- ✓ Fallback a avatar con inicial si no hay foto

Uso

1. Ir a "Mi Perfil" → "Editar Perfil"
2. En la sección "Foto de Perfil", seleccionar archivo
3. Guardar cambios
4. La foto aparecerá automáticamente en navbar y perfiles

Configuración técnica

- Campo: `Usuario.foto_perfil` (ImageField)
- URL de medios: `/media/` (servido por Django en desarrollo)
- Directorio: `backend/media/perfiles/`
- Librería: Pillow 12.0.0

PROF

Estado actual y próximos pasos sugeridos

- Documentación de alto nivel completa (MDs).
- Próximo paso: convertir los diagramas y decisiones en `models.py` por app y prototipar endpoints REST.

Prioridades técnicas:

1. Base de usuarios (registro, Google OAuth2, perfil con `dni` opcional).
 2. Módulo `complejos` con geolocalización y autocompletado Places.
 3. Sistema de reservas y calendario.
 4. Moderación y flujo de tutores para menores.
-

Licencia

Por defecto no se ha añadido una licencia. Si quieres que el repositorio sea open-source, indica la licencia (MIT, Apache-2.0, etc.) y la añadiré.

Si quieres, genero también un `CONTRIBUTING.md` y un `requirements.txt` mínimo con dependencias sugeridas (`Django`, `django-rest-framework`, `django-allauth`). ¿Lo genero ahora?

Hecho por Mantis — documentación base para arrancar DeRevés.

dereves
