🎓 Sistema de Gestión de Préstamos Universitarios

Sistema web completo para la gestión y administración de préstamos estudiantiles en instituciones educativas. Desarrollado con **Django REST Framework** (Backend) y **Angular** (Frontend).

Mostrar imagen Mostrar imagen Mostrar imagen

📋 Tabla de Contenidos

✨ Características

Funcionalidades Principales

- 👥 **Gestión de Usuarios**
  - Registro y autenticación de estudiantes
  - Control de roles (Administrador/Estudiante)
  - Perfiles con información académica
- 💰 **Gestión de Préstamos**
  - Solicitud de préstamos en línea
  - Aprobación y seguimiento de solicitudes
  - Cálculo automático de cuotas e intereses
  - Estados: Pendiente, Aprobado, Activo, Finalizado, Rechazado
- 💳 **Control de Pagos**
  - Registro de pagos realizados

    - Historial de transacciones
    - Métodos de pago: Efectivo, Transferencia, Tarjeta
- 🔔 **Sistema de Notificaciones**
    - Alertas de aprobación/rechazo
    - Recordatorios de vencimiento
    - Notificaciones de pagos
- 📊 **Dashboard y Reportes**
    - Estadísticas en tiempo real
    - Visualización de préstamos activos
    - Resumen de pagos realizados

🛠️ Tecnologías

Backend

- **Python 3.13+**
- **Django 5.2.7**
- **Django REST Framework 3.15+**
- **MySQL 9.2** (Base de datos)
- **django-cors-headers** (Manejo de CORS)
- **djangorestframework-simplejwt** (Autenticación JWT)

Frontend

- **Angular 19**
- **TypeScript 5.7+**
- **Bootstrap 5.3+**
- **Bootstrap Icons**
- **RxJS** (Programación reactiva)

📦 Requisitos Previos

Antes de comenzar, asegúrate de tener instalado:

- **Python 3.13 o superior** → Descargar Python
- **Node.js 18+ y npm** → Descargar Node.js
- **MySQL 8.0 o superior** → Descargar MySQL
- **Angular CLI** → `npm install -g @angular/cli`
- **Git** → Descargar Git

🚀 Instalación

1. Clonar el Repositorio

bash
```
git clone https://github.com/tu-usuario/sistema-prestamos-universitarios.git
cd sistema-prestamos-universitarios
```

---

2. Configuración del Backend (Django)

2.1. Crear y activar entorno virtual

Windows:

bash
```
cd backend
python -m venv venv
venv\Scripts\activate
```

Linux/Mac:

bash
```
cd backend
python3 -m venv venv
source venv/bin/activate
```

2.2. Instalar dependencias

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

Si no existe `requirements.txt`, instala manualmente:

bash
```
pip install django djangorestframework django-cors-headers djangorestframework-simplejwt
mysqlclient pillow
```

2.3. Configurar variables de entorno

Crea un archivo `.env` en la carpeta `backend/` con el siguiente contenido:

env
```
# Base de datos
DB_NAME=prestamos_universitarios_db
DB_USER=root
```

```
DB_PASSWORD=root123
DB_HOST=localhost
DB_PORT=3306

# Django
SECRET_KEY=django-insecure-tu-clave-secreta-aqui
DEBUG=True
ALLOWED_HOSTS=localhost,127.0.0.1
```

---

3. Configuración del Frontend (Angular)

3.1. Instalar dependencias

bash

```
cd frontend/prestamos-frontend
npm install
```

3.2. Configurar URL de la API

Edita `src/app/services/api.service.ts` y verifica que la URL sea correcta:

typescript

```
private apiUrl = 'http://127.0.0.1:8000/api';
```

🗄️ Configuración de la Base de Datos

1. Crear la Base de Datos en MySQL

Opción A: Usando MySQL Workbench

1. Abre **MySQL Workbench**
2. Conéctate a tu servidor MySQL
3. Ejecuta el siguiente script:

sql

```
CREATE DATABASE prestamos_universitarios_db
CHARACTER SET utf8mb4
COLLATE utf8mb4_unicode_ci;
```

Opción B: Usando línea de comandos

bash

```
mysql -u root -p
```

Luego ejecuta:

sql
```
CREATE DATABASE prestamos_universitarios_db CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
EXIT;
```

---

2. Configurar Django para MySQL

Edita `backend/prestamos_backend/settings.py`:

python
```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'prestamos_universitarios_db',
        'USER': 'root',
        'PASSWORD': 'root123',  # ↩️ Cambiar por tu contraseña de MySQL
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
```

---

3. Aplicar Migraciones

Desde la carpeta `backend/`:

bash
```
python manage.py makemigrations
python manage.py migrate
```

Deberías ver:
```
Running migrations:
  Applying usuarios.0001_initial... OK
  Applying prestamos.0001_initial... OK
  Applying notificaciones.0001_initial... OK
  ...
```

---

4. Crear Superusuario

bash

```
python manage.py createsuperuser
```

Datos sugeridos:

- Username: `superadmin`
- Email: `admin@universidad.edu.co`
- Password: `superadmin12345`

---

5. Cargar Datos de Prueba

Ejecuta el script de datos de prueba:

bash

```
python crear_datos.py
```

Esto creará:

- ✅ 4 usuarios (1 admin + 3 estudiantes)
- ✅ 3 préstamos (pendiente, aprobado, activo)
- ✅ 2 pagos
- ✅ 2 notificaciones

▶️ Ejecución del Proyecto

Iniciar el Backend (Django)

Abre una terminal en la carpeta `backend/`:

bash

```
# Windows
venv\Scripts\activate

# Linux/Mac
source venv/bin/activate

# Iniciar servidor
python manage.py runserver
```

El backend estará disponible en: **http://127.0.0.1:8000/**

Iniciar el Frontend (Angular)

Abre **otra terminal** en la carpeta `frontend/prestamos-frontend/`:

bash

```
ng serve
```

O para abrir automáticamente en el navegador:

bash

```
ng serve --open
```

```

El frontend estará disponible en: **http://localhost:4200/**

---

## 📁 *Estructura del Proyecto*

```
sistema-prestamos-universitarios/
│
├── backend/   # Backend Django
│   ├── prestamos_backend/   # Configuración del proyecto
│   │   ├── settings.py   # Configuración principal
│   │   ├── urls.py   # Rutas principales
│   │   └── wsgi.py
│   │
│   ├── usuarios/   # App de usuarios
│   │   ├── models.py   # Modelo PerfilEstudiante
│   │   ├── serializers.py   # Serializers para API
│   │   ├── views.py   # ViewSets
│   │   └── admin.py   # Configuración del admin
│   │
│   ├── prestamos/   # App de préstamos
│   │   ├── models.py   # Modelos Prestamo, Pago
│   │   ├── serializers.py
│   │   ├── views.py
│   │   └── admin.py
│   │
│   ├── notificaciones/   # App de notificaciones
│   │   ├── models.py   # Modelo Notificacion
│   │   ├── serializers.py
│   │   ├── views.py
│   │   └── admin.py
│   │
│   ├── manage.py   # CLI de Django
│   ├── crear_datos.py   # Script de datos de prueba
```
```

```
│   └── requirements.txt            # Dependencias Python
│
└── frontend/                       # Frontend Angular
    └── prestamos-frontend/
        ├── src/
        │   ├── app/
        │   │   ├── components/     # Componentes reutilizables
        │   │   │   └── navbar/
        │   │   │
        │   │   ├── pages/          # Páginas principales
        │   │   │   ├── dashboard/
        │   │   │   ├── prestamos/
        │   │   │   ├── solicitar-prestamo/
        │   │   │   └── notificaciones/
        │   │   │
        │   │   ├── services/       # Servicios API
        │   │   │   └── api.service.ts
        │   │   │
        │   │   ├── app.routes.ts   # Rutas de Angular
        │   │   └── app.component.ts
        │   │
        │   ├── index.html
        │   └── main.ts
        │
        ├── angular.json            # Configuración de Angular
        ├── package.json            # Dependencias npm
        └── tsconfig.json
```

🔌 API Endpoints

Base URL

```
http://127.0.0.1:8000/api/
```

Endpoints Disponibles

Usuarios

http
GET   /api/usuarios/   # Listar todos los usuarios
GET   /api/usuarios/{id}/   # Obtener usuario por ID
POST   /api/usuarios/   # Crear nuevo usuario
PUT   /api/usuarios/{id}/   # Actualizar usuario

DELETE /api/usuarios/{id}/   # Eliminar usuario

Perfiles de Estudiantes

http
```
GET    /api/perfiles/          # Listar perfiles
GET    /api/perfiles/{id}/     # Obtener perfil por ID
POST   /api/perfiles/          # Crear perfil
PUT    /api/perfiles/{id}/     # Actualizar perfil
DELETE /api/perfiles/{id}/     # Eliminar perfil
```

Préstamos

http
```
GET    /api/prestamos/                # Listar todos los préstamos
GET    /api/prestamos/{id}/           # Obtener préstamo por ID
GET    /api/prestamos/pendientes/     # Listar préstamos pendientes
GET    /api/prestamos/activos/        # Listar préstamos activos
POST   /api/prestamos/                # Crear préstamo
PUT    /api/prestamos/{id}/           # Actualizar préstamo
DELETE /api/prestamos/{id}/           # Eliminar préstamo
```

Pagos

http
```
GET    /api/pagos/            # Listar todos los pagos
GET    /api/pagos/{id}/       # Obtener pago por ID
POST   /api/pagos/            # Registrar pago
PUT    /api/pagos/{id}/       # Actualizar pago
DELETE /api/pagos/{id}/       # Eliminar pago
```

Notificaciones

http
```
GET    /api/notificaciones/    # Listar notificaciones
GET    /api/notificaciones/{id}/ # Obtener notificación por ID
POST   /api/notificaciones/    # Crear notificación
PUT    /api/notificaciones/{id}/ # Actualizar notificación
DELETE /api/notificaciones/{id}/ # Eliminar notificación
```

Autenticación (JWT)

http
```
POST   /api/token/            # Obtener token de acceso
POST   /api/token/refresh/    # Refrescar token
```

Ejemplo de uso (cURL)

bash
```
# Obtener lista de préstamos
curl -X GET http://127.0.0.1:8000/api/prestamos/
```

```
# Crear un nuevo préstamo
curl -X POST http://127.0.0.1:8000/api/prestamos/ \
  -H "Content-Type: application/json" \
  -d '{
    "estudiante_id": 2,
    "monto_solicitado": 5000000,
    "plazo_meses": 12,
    "tasa_interes": 4.5,
    "motivo": "Préstamo para matrícula"
  }'
```

🔑 Credenciales de Acceso

Django Admin
```
URL: http://127.0.0.1:8000/admin/
Usuario: superadmin
Contraseña: superadmin12345
```

Usuarios de Prueba (Frontend/API)
```
Estudiante 1:
- Username: maria.rodriguez
- Password: pass123
- Rol: Estudiante
- Código: EST-2021-0156

Estudiante 2:
- Username: juan.gomez
- Password: pass123
- Rol: Estudiante
- Código: EST-2022-0287

Estudiante 3:
- Username: laura.silva
- Password: pass123
- Rol: Estudiante
- Código: EST-2020-0423
```

Base de Datos MySQL
```
Host: localhost
Port: 3306
```

Database: prestamos_universitarios_db
User: root

Password: root123

📸 Capturas de Pantalla

Dashboard

Mostrar imagen *Panel principal con estadísticas y resumen de préstamos*

Lista de Préstamos

Mostrar imagen *Vista completa de todos los préstamos con filtros*

Solicitar Préstamo

Mostrar imagen *Formulario para solicitar un nuevo préstamo*

Django Admin

Mostrar imagen *Panel de administración de Django*

🔧 Solución de Problemas Comunes

Error: "Access denied for user 'root'@'localhost'"

Solución: Verifica la contraseña de MySQL en `settings.py`

python

```
DATABASES = {
    'default': {
        # ...
        'PASSWORD': 'tu_contraseña_correcta',  # ⬅️ Cambiar aquí
    }
}
```

Error: "No module named 'MySQLdb'"

Solución: Instala `mysqlclient`

bash

```
pip install mysqlclient
```

Error de CORS en Angular

Solución: Verifica que `django-cors-headers` esté configurado en `settings.py`:

python

```
INSTALLED_APPS = [
    # ...
    'corsheaders',
]

MIDDLEWARE = [
    # ...
    'corsheaders.middleware.CorsMiddleware',
    # ...
]

CORS_ALLOWED_ORIGINS = [
    "http://localhost:4200",
]
```

Angular no muestra datos

Solución:

1. Verifica que Django esté corriendo en http://127.0.0.1:8000/
2. Abre la consola del navegador (F12) y verifica errores
3. Prueba la API directamente: http://127.0.0.1:8000/api/prestamos/

🧪 Pruebas

Verificar que la API funciona

bash

```
# Desde la terminal
curl http://127.0.0.1:8000/api/prestamos/
```

Verificar conexión desde Angular

Abre la consola del navegador (F12) y ejecuta:

```
javascript
fetch('http://127.0.0.1:8000/api/prestamos/')
  .then(res => res.json())
  .then(data => console.log(data));
```

📦 Crear archivo requirements.txt

Si no existe, créalo ejecutando:

```
bash
cd backend
pip freeze > requirements.txt
```

🚢 Despliegue en Producción

Backend (Django)

1. Configurar variables de entorno
2. Cambiar `DEBUG = False` en `settings.py`
3. Configurar `ALLOWED_HOSTS`
4. Usar Gunicorn o uWSGI
5. Configurar Nginx como proxy inverso

Frontend (Angular)

```
bash
ng build --configuration production
```

Los archivos compilados estarán en `dist/`

🤝 Contribuir

Las contribuciones son bienvenidas. Por favor:

1. Fork el proyecto
2. Crea una rama (`git checkout -b feature/nueva-funcionalidad`)
3. Commit tus cambios (`git commit -m 'Agregar nueva funcionalidad'`)

4. Push a la rama (`git push origin feature/nueva-funcionalidad`)
5. Abre un Pull Request

📄 Licencia

Este proyecto está bajo la Licencia MIT. Ver el archivo `LICENSE` para más detalles.

👨‍💻 Autor

Tu Nombre

- GitHub: @tu-usuario
- Email: tu-email@ejemplo.com

🙏 Agradecimientos

- Django Software Foundation
- Angular Team
- Bootstrap Team
- Comunidad de desarrolladores

📞 Soporte

Si tienes preguntas o problemas:

1. Revisa la sección de Solución de Problemas
2. Abre un Issue en GitHub
3. Contacta al autor
