



Guía de Preparación para GitHub - EscalaFin MVP

Instrucciones completas para subir EscalaFin a GitHub de forma segura y profesional



Verificación Pre-Upload



Archivos Preparados

- [x] **.gitignore** actualizado con todas las exclusiones necesarias
- [x] **README.md** completo con documentación actualizada
- [x] **.env.example** con todas las variables de entorno requeridas
- [x] **Documentación completa** (guías, manuales, PDFs)
- [x] **Licencia MIT** incluida
- [x] **Archivos de seguridad** configurados



Pasos para Subir a GitHub

Paso 1: Crear Repositorio en GitHub

1. **Accede a GitHub** → <https://github.com>
2. **Crea nuevo repositorio:**

Nombre: `escalafin-mvp`

Descripción: Sistema integral de gestión de créditos y préstamos

☒ Public (o Private según prefieras)

☒ NO inicializar con README (ya tenemos el nuestro)

☒ NO agregar .gitignore (ya tenemos el nuestro)

☒ Licencia: MIT License

Paso 2: Configurar Git Local

```
# Navegar al directorio del proyecto
cd /home/ubuntu/escalafin_mvp

# Inicializar repositorio Git (si no existe)
git init

# Configurar información del usuario
git config user.name "Tu Nombre"
git config user.email "tu-email@ejemplo.com"

# Agregar archivos al staging
git add .

# Verificar que no se suban archivos sensibles
git status

# Crear primer commit
git commit -m "🚀 Initial commit: EscalaFin MVP - Sistema completo de gestión de préstamos"

Features:
✓ PWA completa con navegación optimizada
✓ Multi-rol (Admin, Asesor, Cliente)
✓ Gestión integral de préstamos y pagos
✓ Integración Openpay + WhatsApp + AWS S3
✓ Dashboard con KPIs y reportes
✓ Landing page y sistema de autenticación
✓ Sidebar responsivo con tema dark/light
✓ Documentación completa incluida"
```

Paso 3: Conectar con GitHub

```
# Conectar con el repositorio remoto (cambiar 'tu-usuario' por tu username)
git remote add origin https://github.com/tu-usuario/escalafin-mvp.git

# Subir código al repositorio
git branch -M main
git push -u origin main
```

Verificación de Seguridad

Archivos EXCLUIDOS (verificar que NO se suban):

- ❌ .env (variables de entorno reales)
- ❌ node_modules/ (dependencias)
- ❌ .next/ (build de Next.js)
- ❌ uploads/ (archivos subidos)
- ❌ cookies.txt
- ❌ dev-server.log
- ❌ test_login.js
- ❌ *.log files
- ❌ Cualquier archivo con credenciales reales

Archivos INCLUIDOS (verificar que SÍ se suban):

- ✓ README.md
- ✓ .env.example (plantilla sin datos reales)
- ✓ .gitignore
- ✓ LICENSE
- ✓ SECURITY.md
- ✓ CONTRIBUTING.md
- ✓ app/ (código fuente completo)
- ✓ Todas las guías .md
- ✓ Archivos de configuración (**package.json**, tailwind, etc.)
- ✓ Documentación PDF
- ✓ Schema de **base** de datos



Configuración de Repository Settings

1. Información General

Repository name: escalafin-mvp
 Description: Sistema integral de gestión de créditos y préstamos con navegación optimizada
 Website: [tu-dominio-si-tienes]
 Topics: fintech, loans, nextjs, typescript, prisma, pwa, tailwindcss

2. Features a Habilitar

- ✓ Issues
- ✓ Projects
- ✓ Wiki
- ✓ Discussions (opcional)
- ✓ Sponsorships (opcional)

3. Security

- ✓ Enable private vulnerability reporting
- ✓ Enable Dependabot alerts
- ✓ Enable Dependabot security updates



Crear Release Inicial

Después del primer push:

1. Ir a tu repositorio → Releases → Create a new release

2. Configurar release:

...

Tag version: v1.0.0

Release title: 🚀 EscalaFin MVP v1.0.0 - Lanzamiento Inicial

Descripción:

🎉 Primera versión estable de EscalaFin MVP

⭐ Características principales:

- ✅ PWA completa con navegación optimizada
- ✅ Sistema multi-rol (Admin, Asesor, Cliente)
- ✅ Gestión integral de préstamos y pagos
- ✅ Dashboard con KPIs y reportes en tiempo real
- ✅ Integración Openpay + WhatsApp + AWS S3
- ✅ Landing page profesional
- ✅ Sidebar responsivo con tema dark/light

🛠 Stack Tecnológico:

- Next.js 14 + TypeScript
- PostgreSQL + Prisma ORM
- Tailwind CSS + Shadcn/ui
- NextAuth.js para autenticación
- AWS S3 para almacenamiento
- EvolutionAPI para WhatsApp

📦 Incluye:

- Documentación completa de instalación
- Guías de migración a DeepAgent
- Manual de usuario
- Esquemas de base de datos
- Variables de entorno de ejemplo

⚡ **Listo para producción**

...

🌟 Configuración Adicional Recomendada

1. Branch Protection Rules

En Settings → Branches → Add rule

Branch name pattern: main

- ✅ Require pull request reviews before merging
- ✅ Require status checks to pass before merging
- ✅ Require branches to be up to date before merging
- ✅ Include administrators

2. Labels del Repositorio

- 🐛 bug - Errores y problemas
- 💡 enhancement - Nuevas características
- 📖 documentation - Mejoras de documentación
- 🔧 maintenance - Tareas de mantenimiento
- 🚀 feature - Nueva funcionalidad
- 💡 idea - Ideas para implementar
- ❓ question - Preguntas y dudas

3. Templates

El repositorio ya incluye:

- **Issue templates** en `.github/ISSUE_TEMPLATE/`
- **Pull request template** en `.github/PULL_REQUEST_TEMPLATE.md`
- **Contributing guidelines** en `CONTRIBUTING.md`
- **Security policy** en `SECURITY.md`



Métricas y Badges Recomendados

Agregar al README.md:

```

[[License: MIT]](https://upload.wikimedia.org/wikipedia/commons/thumb/c/c3/License_icon-mit.svg/2048px-License_icon-mit.svg.png)
[[Next.js]](https://i.ytimg.com/vi/f53RvUpUA8w/hqdefault.jpg)
[[TypeScript]](https://i.ytimg.com/vi/4cgpu9L2AE8/maxresdefault.jpg)
[[PostgreSQL]](https://i.ytimg.com/vi/XdCV1WxG1Ug/sddefault.jpg)
[[Prisma]](https://repository-images.githubusercontent.com/728428101/43bf5fdd-c155-4aa7-b86c-91fe176c131e)
[[Tailwind CSS]](https://i.ytimg.com/vi/cY0XJY98d3w/maxresdefault.jpg)
[[PWA]](https://i.ytimg.com/vi/M_bRwImmImk/mqdefault.jpg)

```



Checklist Final

Antes de hacer el push final, verificar:

- ☐ `.env` eliminado o en `.gitignore`
- ☐ `node_modules/` excluido
- ☐ Logs y archivos temporales excluidos
- ☐ README.md actualizado
- ☐ `.env.example` con todas las variables
- ☐ Licencia MIT incluida
- ☐ Documentación completa
- ☐ Información de contacto actualizada
- ☐ URLs del repositorio actualizadas
- ☐ No hay credenciales hardcodeadas en el código
- ☐ Todas las dependencias están en `package.json`
- ☐ Scripts de build funcionan correctamente

Comando Final

```
# Verificación final antes del push
git status
git log --oneline -5


# Push definitivo
git push origin main

# Verificar en GitHub que todo se subió correctamente
```

Soporte Post-Upload

Una vez subido el proyecto, puedes:

1. **Configurar GitHub Pages** (si quieres demo público)
2. **Habilitar GitHub Actions** para CI/CD automático
3. **Configurar Dependabot** para actualizaciones de seguridad
4. **Agregar colaboradores** si trabajas en equipo
5. **Configurar integración con Vercel/Netlify** para deploys automáticos

 **Resultado:** Repositorio GitHub profesional, seguro y listo para colaboración, con documentación completa y todas las mejores prácticas implementadas.