

## # Planning.md – Arquitectura & Stack para FINTECH ESCROW

### ## Tipo de arquitectura del software

- Web App SPA (Single Page Application).
- Backend basado en microservicios escalables.
- Integración vía API RESTful para módulos de ESCROW, autenticación, pagos y notificaciones[attached\_file:14].

### ## Tecnologías frontend

- Angular como framework principal.
- TailwindCSS para estilos y maquetación responsive.
- Librerías para manejo de formularios, validación dinámica e internacionalización.
- Sistema de notificaciones en tiempo real (ej. Websockets o librería push)[attached\_file:14][attached\_file:15].

### ## Tecnologías backend

- Python con FastAPI para servicios y endpoints RESTful.
- Microservicios independientes para ESCROW, usuarios, pagos/transferencias, seguimiento y cálculo de comisiones.
- Integración de librerías de procesamiento de archivos (CSV para cargas masivas).
- Scripting para verificación documental y automatización IA[attached\_file:14].

### ## Datos/ORMs/Bases de datos

- PostgreSQL como base de datos relacional central.
- ORM sugerido: SQLAlchemy (Python).
- Estructura preparada para integración con servicios externos (bancos, consultas KYC/AML).
- Manejo de documentos y metadatos (imágenes, PDFs validados) asociables a usuarios y operaciones de ESCROW[attached\_file:14].

### ## Modelo de autenticación/autorización esperado

- Doble factor: registro vía correo electrónico con validación y autenticación adicional vía proveedor externo (ej. Truora).
- Posible uso de datos biométricos o QR para acceso condicional.
- Implementación de roles jerárquicos: admin, asesor, vendedor, comprador, broker.
- Validación documental estricta (INE, CURP, RFC, actas, comprobantes) durante onboarding[attached\_file:14][attached\_file:15].

### ## Plataformas de hosting/deploy

- Nube pública (AWS/GCP/Azure) idealmente usando servicios administrados (Elastic Beanstalk/Azure App Service/GKE).
- Deploy automatizado vía CI/CD, con versiones beta/staging accesibles solo para pruebas internas y demo pública[attached\_file:14].
- Escalable para soportar picos de usuarios y pruebas de carga.

### ## Convenciones de trabajo (estilo de commits, ramas, CI/CD, testing)

- Control de versiones en Git siguiendo branching model de Git flow (rama main, develop, feature/bug/hotfix).

- Commits semánticos (Convencional Commits) para trazabilidad y automatización.
- Integración continua (CI) con tests unitarios/funcionales automáticos (PyTest/Jest).
- Despliegue continuo (CD) configurado en pipelines.
- Revisiones de código via pull request.
- Linting y formateo automático en cada push para mantener estándares de calidad[attached\_file:14].

## ## Dependencias y herramientas clave de desarrollo

- FastAPI, SQLAlchemy, Angular, TailwindCSS, Python, PostgreSQL.
- Librerías para autenticación externa (ej. truora/2FA).
- Integración con servicios de correo, SMS y notificaciones push (Twilio, SendGrid, Firebase).
- Herramientas para edición colaborativa Markdown y visualización de flujos (ej. Mermaid, Draw.io).
- Librerías para generación de QR, envío de archivos y carga masiva por CSV.
- Monitorización y logging centralizado (Datadog/Sentry).

---

## ## (Sugerido) Elementos adicionales para VIBE CODING

- Plantillas de flujo para cada rol y operación.
- Sistema de prompts para generación automatizada de contratos y checklist de onboarding.
- Reglas IA para evaluar documentación subida y detectar posibles errores o fraudes.
- Documentar endpoints y reglas internas en formato consumible por bots/externalización IA.

---

¿Deseas ajustar alguna tecnología, arquitectura, modelo de roles, o incluir integración con IA/bots de soporte en el planning? ¿Necesitas ejemplos de endpoints o flujos típicos para CODE? Indica cualquier preferencia específica antes de pasar a la siguiente etapa. Aquí tienes un borrador de **Planning.md** para el proyecto FINTECH, basado en información de tus documentos y lineamientos modernos de desarrollo para una aplicación con lógica ESCROW, e incluyendo buenas prácticas relevantes para escenarios de VIBE CODING. Ajusta o confirma cada sección antes de continuar:[attached\_file:14][attached\_file:15][attached\_file:16]

---