Documentación del proyecto Intercambius

Intercambius — El club de confianza
Documentación técnica de lo implementado: Mail, Dashboard Admin, Métricas, Usuarios, Newsletter, mejoras del front y exportación a Excel.

---

1. Mail (correo electrónico)

Configuración

- **Motor**: Nodemailer con SMTP.
- **Variables de entorno** (backend `.env`):
  - `SMTP_HOST` (default: `smtp.gmail.com`)
  - `SMTP_PORT` (default: `587`)
  - `SMTP_SECURE` (`true` para 465)
  - `SMTP_USER` y `SMTP_PASS` (ej. cuenta Gmail; para Gmail puede usarse "Contraseña de aplicación").
  - `SMTP_FROM`: remitente mostrado (opcional; si no está, se usa `SMTP_USER` o `"Intercambius" <noreply@intercambius.com>`).
  - `FRONTEND_URL`: URL base del front (para enlaces en los emails).
  - `LOGO_URL`: URL del logo en los correos (default: `{FRONTEND_URL}/logo-intercambius.png`).

Si `SMTP_USER` no está definido, no se envía correo; solo se registra en consola (útil en desarrollo).

Plantilla unificada

Todos los correos usan la misma plantilla HTML:

- Cabecera con fondo oscuro y logo (imagen desde `LOGO_URL`).
- Cuerpo con contenido específico de cada tipo de email.
- Pie: "Intercambius — El club de confianza" y enlace al sitio.

Incluye versión en texto plano para clientes que no muestran HTML.

Tipos de email implementados

Tipo	Cuándo se envía	Contenido principal
MFA	Tras login con email/contraseña correctos	Código de 6 dígitos, válido 10 minutos.
Restablecer contraseña	Solicitud "Olvidé mi contraseña"	Botón/enlace para restablecer; indica minutos de validez.
Bienvenida	Tras registro exitoso	Mensaje de bienvenida y enlace al market.
Login exitoso	Tras completar MFA correctamente	Aviso de inicio de sesión.
Compra confirmada	Tras confirmar un intercambio (como comprador)	Producto, precio en IX, enlace al chat.
Venta	Tras confirmar un intercambio (como vendedor)	Quién compró, producto, precio, enlace al chat.

Nuevo mensaje	Cuando alguien envía un mensaje en un chat	Vista previa del mensaje y enlace a la conversación.
Newsletter	Envío desde el panel Admin	Asunto y cuerpo (HTML/texto) personalizados.

Los envíos se hacen en "fire-and-forget" (salvo MFA y reset de contraseña, que se esperan) para no bloquear la respuesta al usuario; los errores se registran en consola.

---

2. Dashboard Admin

Acceso

- **Ruta front**: `/admin` (login) y `/admin/dashboard` (panel).
- **Credenciales**: definidas en el backend con `ADMIN_EMAIL` y `ADMIN_PASSWORD`. Si no están configuradas, el login de admin responde 503.
- **Autenticación**: el backend expone `POST /api/auth/admin-login` con `{ email, password }` y devuelve un JWT con flag `admin: true`. El front guarda el token en `localStorage` (`intercambius_admin_token`) y lo envía en `Authorization: Bearer ...` en todas las peticiones a `/api/admin/*`.

Middleware de admin (backend)

- Rutas bajo `/api/admin/*` usan `adminMiddleware`: verifican JWT y que el payload tenga `admin: true`. Si no, responden 401 o 403.

Secciones del dashboard

1. **Métricas** – KPIs y gráficos (ver sección 3).
2. **Usuarios** – Tabla paginada con acciones ban / unban / eliminar.
3. **Productos** – Listado paginado de publicaciones del market.
4. **Intercambios** – Listado paginado de transacciones confirmadas.
5. **Newsletter** – Editor de contenido y envío masivo (ver sección 5).

El dashboard usa **modo claro/oscuro** (next-themes) con variables CSS (`:root` y `.dark`) coherentes con el resto del front.

---

3. Métricas

API

- (requiere token admin).

Devuelve:

- **Usuarios**: total y serie `porMes` (últimos 6 meses) para gráfico.
- **Productos**: total, activos, y `porEstado` (active, sold, etc.) para gráfico de torta.
- **Ventas/Compras**: transacciones totales y `porMes` (cantidad y volumen en IX) para gráficos.
- **Token**: saldo en circulación, volumen de transacciones, token gastado en compras, token recibido en ventas.
- **Contacto**: conversaciones totales, mensajes totales, mensajes por mes (últimos 6 meses).

Las series por mes se calculan en PostgreSQL con `date_trunc('month', ...)` y `to_char(..., 'YYYY-MM')`.

Front (pestaña Métricas)

- **KPIs** en cards: usuarios, productos, transacciones, volumen IX, conversaciones, mensajes, etc.
- **Gráficos** (Recharts):
  - Usuarios por mes (área).
  - Transacciones / volumen por mes (barras).
  - Productos por estado (torta).
  - Mensajes por mes (área o barras).

---

4. Usuarios (gestión admin)

Modelo User (Prisma)

Campos relevantes para admin y auth:

- `mfaCode`, `mfaCodeExpiresAt`: MFA por email (código 6 dígitos).
- `passwordResetToken`, `passwordResetExpiresAt`: flujo "olvidé contraseña".
- `bannedAt`: si no es `null`, el usuario está baneado.

Comportamiento

- **Login normal**: si el usuario está baneado (`bannedAt` no null), el backend rechaza el login (auth middleware o caso de uso correspondiente).
- **Admin**:
  - `/api/admin/users` – Lista usuarios con paginación (`page`, `limit`), incluyendo `bannedAt`, conteo de productos e intercambios.
  - `/api/admin/users/:id/ban` – Escribe `bannedAt: new Date()`.
  - `/api/admin/users/:id/unban` – Pone `bannedAt: null`.
  - `/api/admin/users/:id` – Elimina el usuario (Prisma cascade elimina productos, favoritos, etc.).

En el front, la tabla de usuarios muestra estado de baneo y botones Ban / Desbanear / Eliminar, con `AlertDialog` de confirmación para eliminar.

---

5. Newsletter

Backend

- `/api/admin/newsletter` (body JSON):
- `subject` (obligatorio).
- `bodyHtml` y/o `bodyText` (al menos uno).
- `enviarATodos: true` → se envían a todos los usuarios no baneados.
- O `emails: string[]` → solo a esa lista (se buscan por email en la base).

Para cada destinatario se llama a `emailService.sendNewsletter(email, nombre, subject, html, bodyText)`, usando la misma plantilla con logo y pie. La respuesta indica enviados, total y hasta 20 errores (si los hay).

Front (pestaña Newsletter del admin)

- **Editor rico**: ReactQuill con toolbar (títulos, negrita, listas, enlace, imagen, limpiar).
- Campos: asunto, cuerpo HTML (Quill), opción "Enviar a todos" o lista de emails.
- Al enviar se muestra resultado: "Enviados X de Y" y eventuales errores.

---

6. Mejoras del front

Auth y flujos

- **Login**: paso 1 email/contraseña; paso 2 código MFA (6 dígitos por email). Si no hay SMTP, el código se puede ver en logs del backend (solo desarrollo).
- **Olvidé contraseña**: página que pide email → backend envía enlace por correo.
- **Restablecer contraseña**: página con token en URL; dos campos de contraseña (validación de coincidencia) y envío al backend.
- **Registro**: formulario con validaciones; tras registro se envía email de bienvenida.

Componentes

- **PasswordInput**: input tipo password con botón para mostrar/ocultar (ojito); usado en Login, Registro, Restablecer contraseña y Admin login.
- **AlertDialog** (shadcn/ui): usado para confirmar eliminación de usuario en el dashboard.

Diseño y tema

- **Design system** en `src/index.css`: paleta negro + dorado ámbar, variables para primary, card, chart, sidebar, etc.
- **Modo claro/oscuro**: `next-themes`; en el admin hay toggle Sol/Luna.
- **Variables de gráficos**: `--chart-1` … `--chart-5` para mantener coherencia en métricas.

CORS y permisos

- CORS en el backend configurado para reflejar el origen de la petición (sin forzar barra final), evitando fallos por diferencia origen con/sin trailing slash.
- Permissions policy (p. ej. en Netlify): `geolocation=(self)` si se usa geolocalización.

---

7. Exportación a Excel

Librería

- **xlsx** (SheetJS) en el front; se generan archivos en el navegador y se descargan.

Funciones

Función	Contenido del Excel
`exportMetricsToExcel(metrics)`	Una hoja “Métricas” con métricas globales (usuarios, productos, transacciones, token, contacto).
`exportUsersToExcel(data)`	Hoja “Usuarios”: id, nombre, email, contacto, saldo, límite, ubicación, miembroDesde, verificado, baneado, productos publicados, intercambios.
`exportProductosToExcel(data)`	Hoja “Productos”: id, título, precio, estado, rubro, ubicación, vendedor (id, nombre, email), creado.
`exportIntercambiosToExcel(data)`	Hoja “Transacciones”: id, comprador, vendedor, créditos, estado, producto, fecha.

Los datos para usuarios, productos e intercambios se obtienen desde el admin con `getUsersForExport`, `getProductosForExport`, `getIntercambiosForExport` (paginación amplia, p. ej. limit 10000). El nombre del archivo incluye la fecha (YYYY-MM-DD).

En el dashboard, cada pestaña (Métricas, Usuarios, Productos, Intercambios) tiene botón de exportar que llama a la función correspondiente.

---

8. Deploy (Vercel) y base de datos

- **Build backend**: script `scripts/vercel-build.cjs` ejecuta `prisma generate` y, por defecto en Vercel, **no** ejecuta `prisma migrate deploy` para evitar timeout del advisory lock de Postgres.
- Las migraciones deben ejecutarse manualmente desde un entorno con acceso directo a la base (por ejemplo: `cd backend && npx prisma migrate deploy`).
- Si en el futuro se configura `DIRECT_DATABASE_URL` y se quiere correr migraciones en el build, en el schema se puede descomentar `directUrl` y en Vercel definir `RUN_MIGRATE_IN_VERCEL=1`.

Resumen de archivos clave

Área	Backend	Front
Mail	`backend/src/infrastructure/services/email.service.ts`	—
Auth (MFA, reset)	`backend/src/application/use-cases/auth/*`, `AuthController`	`Login.tsx`, `OlvideContrasena.tsx`, `RestablecerContrasena.tsx`
Admin API	`backend/src/presentation/controllers/AdminController.ts`, `routes/admin.ts`, `middleware/adminAuth.ts`	`src/services/admin.service.ts`
Dashboard	—	`src/pages/AdminDashboard.tsx`, `AdminLogin.tsx`
Métricas	`AdminController.getMetrics`	Pestaña Métricas + Recharts + `index.css` (--chart-*)
Newsletter	`AdminController.sendNewsletter`	Pestaña Newsletter + ReactQuill
Excel	—	`src/lib/exportExcel.ts`, botones en AdminDashboard
Usuarios (ban/delete)	`AdminController.banUser`, `unbanUser`, `deleteUser`	Tabla usuarios + AlertDialog

---