
	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 1 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

Manual Técnico para Proyectos de Software

Proyecto FUNED
Fecha: 7 octubre 2025


	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 2 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

HOJA DE CONTROL

Empresa / Organización	Centro de Teleinformática y Producción Industrial – SENA (Equipo FUNED)		
Proyecto	FUNED– App móvil de la academia Funed		
Entregable	Manual Técnico		
Autores	Equipo ADSO – Popayán Camilo Hurtado Santiago Ceron Carlos Imbachi Diego Moncayo		
Versión/Edición	1.0	Fecha Versión	01/09/2025
Aprobado por	Instructores sena	Fecha Aprobación	Registro de Cambios
		Nº Total de Páginas	25

REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
Versión 1.0	Primera versión del manual técnico	Equipo FUNED	01/09/2025

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 3 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

CONTENIDO


Contenido

Contenido

Introducción	3
Objetivo	4
Descripción General del Proyecto	4

Introducción

Este manual técnico describe el aplicativo FUNED para desarrolladores, administradores de sistemas, integradores y usuarios técnicos (incluyendo instructores y evaluadores SENA). Cubre arquitectura, instalación, mantenimiento, extensión del sistema y documentación de componentes.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 4 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

Objetivo


- Establecer lineamientos técnicos y mejores prácticas.
- Documentar arquitectura, componentes, y flujos críticos.
- Proveer instrucciones de instalación, despliegue, operación y soporte.
- Facilitar la integración con sistemas externos y la extensión del aplicativo.

Descripción General del Proyecto

- Propósito y alcance: Gestión académica de FUNED (oferta de cursos, matrículas, pagos, asistencia, calificaciones, certificados) con notificaciones por correo.
- Audiencias: desarrolladores, DevOps/administradores, integradores, instructores/evaluadores SENA y usuarios técnicos.
- Requisitos técnicos principales: Node.js (≥ 18), PostgreSQL (≥ 14), npm, Vite, React, Tailwind, SendGrid, entorno con SSL para BD.
- Contexto: Arquitectura cliente-servidor REST, autenticación por JWT, documentación de API con Swagger.

Arquitectura del sistema

- Capas:
 - Backend: Express + Sequelize sobre PostgreSQL. Rutas REST en `/api` y autenticación en `/auth`.
 - Frontend: React + Vite + Tailwind; cliente SPA que consume la API y gestiona sesión JWT.
 - Notificaciones: Servicio SendGrid para envíos de correo (bienvenida, pago aprobado, recuperación).
 - Opcional: Cliente Flutter (aplicacion-funed) para experiencias móviles.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 5 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

Flujo de datos


- Frontend obtiene token vía `/auth/login` y lo persiste (localStorage). Axios envía `Authorization: Bearer <JWT>`.
- Backend valida JWT (`JWT_SECRET`) y atiende recursos: personas, cursos, oferta, módulos, matrículas, pagos, asistencia, calificaciones y certificados.
- BD recibe operaciones mediante Sequelize. Relaciones y constraints se aplican vía migraciones.
- Correos se envían por SendGrid con plantillas simples HTML.

Decisiones de diseño

- PostgreSQL con SSL por compatibilidad cloud y consistencia ACID.
- Swagger en `/api-docs` con seguridad Bearer para exploración y pruebas.
- Job periódico (configurable por `AUTO_APPROVE_INTERVAL_MS`) para auto-aprobar matrículas de ofertas finalizadas.
- Middleware de compresión opcional para mejorar rendimiento cuando está disponible.

Herramientas utilizadas

- Lenguajes: JavaScript/Node.js, TypeScript parcial en frontend (env.ts), Dart (Flutter opcional).
- Frameworks: Express, React, Vite, TailwindCSS.
- Librerías clave: Sequelize, `swagger-ui-express`, `swagger-jsdoc`, `@sendgrid/mail`, `jsonwebtoken`, `axios`, `@supabase/supabase-js`.
- Construcción: Vite en el frontend; `nodemon` en backend para desarrollo.
- Control de versiones: Git.
- CI/CD: (recomendado) GitHub Actions para lint, build y despliegue.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 6 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

Instalación y configuración

Base de datos

- Requisitos: PostgreSQL 14+ con SSL habilitado en ambientes cloud.
- Conexión: se usa `DATABASE_URL` (en todos los ambientes) y `ssl` requerido (`rejectUnauthorized: false`). Ejemplo:
 - `DATABASE_URL=postgres://usuario:contraseña@host:5432/funed?sslmode=require`
- Usuarios y permisos: rol con privilegios `CONNECT`, `CREATE`, `USAGE` sobre esquema usado por Sequelize.

Backend API

- Ubicación: `proyecto-funed-backend/`
- Dependencias: `npm install`
- Variables requeridas (ejemplo `.env`):

...

PORT=3000

DATABASE_URL=postgres://usuario:contraseña@host:5432/funed?sslmode=require

JWT_SECRET=tu_secreto_seguro

SENDGRID_API_KEY=SG.xxxxxx


MAIL_FROM=no-reply@funed.org

MAIL_FROM_NAME=FUNED

APP_URL=https://proyecto-funed-frontend.example.com

AUTO_APPROVE_INTERVAL_MS=3600000

...

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 7 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

- Inicio local: `npm start` (usa `nodemon`), API en `http://localhost:3000`
- Salud: `GET /api/health` devuelve `{ status: 'ok', uptime, timestamp }`
- Swagger: `GET /api-docs` muestra documentación (servidores: local y producción). Seguridad Bearer JWT.
- Rutas base: `/api` para recursos, `/auth` para autenticación.

Frontend Web

- Ubicación: `funed-fronted/`
- Dependencias: `npm install`
- Variables (archivo `.env`):

...

Local

VITE_API_URL=http://localhost:3000

Producción

VITE_API_URL=https://proyecto-funed-backend.onrender.com


Supabase

VITE_SUPABASE_URL=https://<tu-proyecto>.supabase.co

VITE_SUPABASE_ANON_KEY=<tu_anon_key>

...

- Desarrollo: `npm run dev` (Vite), preview: `npm run preview`, build: `npm run build`.
- Autenticación: `AuthContext.jsx` guarda token y usuario en `localStorage` y configura `axios.defaults.headers.common.Authorization`.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 8 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

Variables de entorno y secretos

- Backend: `PORT`, `DATABASE_URL`, `JWT_SECRET`, `SENDGRID_API_KEY`, `MAIL_FROM`, `MAIL_FROM_NAME`, `APP_URL`, `AUTO_APPROVE_INTERVAL_MS`.
- Frontend: `VITE_API_URL`, `VITE_SUPABASE_URL`, `VITE_SUPABASE_ANON_KEY`.
- Gestión de credenciales: usar almacenes seguros (Secrets de CI/CD, Vault/1Password). Evitar exponer llaves en repositorios.
- Rotación: programar rotación periódica de `JWT_SECRET` y claves de SendGrid; implementar invalidación y reemisión segura.

Perfiles y ambientes

- `development`, `test`, `production` comparten `DATABASE_URL` con SSL requerido (ver `config/config.json`).
- CORS abierto (`origin: "*"`) por defecto; ajustar origen en producción.

Despliegue

- Servidores tradicionales: Node.js con `pm2` detrás de `nginx` o servicios PaaS.
- Cloud: Render (URL de producción incluida en Swagger), Railway, Heroku, Fly.io.
- Docker (opcional):

...

version: '3.8'


services:

backend:

image: node:20

working_dir: /app

volumes:

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 9 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

- ./proyecto-funed-backend:/app

environment:

- PORT=3000
- DATABASE_URL=\${DATABASE_URL}
- JWT_SECRET=\${JWT_SECRET}
- SENDGRID_API_KEY=\${SENDGRID_API_KEY}
- MAIL_FROM=\${MAIL_FROM}
- MAIL_FROM_NAME=\${MAIL_FROM_NAME}
- APP_URL=\${APP_URL}

command: sh -c "npm install && node server.js"

ports:

- "3000:3000"

Verificación integral (Smoke Test)

- API:


- `GET /api/health` debe responder 200 con `status: ok`.
- `GET /api-docs` debe cargar Swagger.
- `POST /auth/login` con credenciales válidas debe devolver JWT.

- Frontend:

- Variables cargadas y llamadas a `VITE_API_URL` funcionan.
- Inicio de sesión persistente y navegación protegida con roles.

Solución de problemas (FAQ)

- Error DB (SSL): asegurar `?sslmode=require` en `DATABASE_URL` y que el proveedor soporte SSL.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 10 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		


- Falta `SENDGRID_API_KEY`: el servicio de correo lanza error; agregar la clave y reiniciar.
- `Token inválido`: verificar `JWT_SECRET` y expiración del JWT.
- CORS bloquea: ajustar `origin` y `allowedHeaders` en `server.js`.
- Migraciones: ejecutar `npx sequelize-cli db:migrate` y revisar modelos/rutas.

Operación diaria (Administradores de sistemas)

- Tareas diarias
 - Verificar `GET /api/health` (latencia < 500ms; `status: ok`).
 - Revisar logs de backend y buscar errores recurrentes (auth, DB, correo).
 - Confirmar ejecución del job de auto-aprobación (mensajes `[AutoAprobación]` en logs).
 - Monitorear entregabilidad de correo (SendGrid): tasa de bounces y claves activas.
 - Revisar conexiones y tiempos de respuesta de BD (pool, timeouts, SSL).
 - Validar estado de respaldos (`pg_dump`) y espacio en disco.
- Rotación de claves
 - JWT (`JWT_SECRET`):
 - Planificar ventana de mantenimiento; actualizar variable y reiniciar servicio.
 - Invalidar sesiones si es necesario; emitir nuevos tokens.
 - SendGrid (`SENDGRID_API_KEY`):
 - Crear nueva clave, actualizar `.env`, probar envío y revocar la anterior.
 - Supabase: rotar `ANON_KEY` si se compromete; actualizar cliente y validar accesos.

Mantenimiento semanal

- Actualizar dependencias críticas, correr smoke tests y revisar compatibilidades.
- Analizar índices y consultas críticas; optimizar si hay regresiones.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 11 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

- Alertas y escalamiento

- Configurar alertas (CPU, memoria, errores 5xx, colas de correo) y escalar a devops/desarrollo ante umbrales.

Diccionario de datos

Principales tablas y campos (extraídos de migraciones):

- `personas`:

- `id` (PK), `nombre`, `apellido`, `numeroIdentificacion` (único), `tipoIdentificacion` (Cédula|Tarjeta de identidad), `fechaNacimiento`, `telefono`, `correo`, `rol` (Administrador|Usuario|Estudiante), `createdAt`, `updatedAt`.

- `usuarios`:

- `id`, `idPersona` (FK personas.id), `email`, `password`.

- `cursos`:

- `id`, `nombreCurso`, `duracion` (horas), `temario` (texto), `tipoCurso` (Técnico|Corto).

- `ofertaCursos`:

- `id`, `codigoCurso`, `idCurso` (FK cursos.id), `fechaInicioCurso`, `fechaFinCurso`, `horario`, `cupos`, `precio`, `foto`.

- `modulos`:

- `id`, `nombre`, `id_curso` (FK cursos.id).


- `moduloDocentes`:

- `id`, `idModulo` (FK modulos.id), `idDocente` (FK docentes.id), `idOfertaCurso` (FK ofertaCursos.id), `resultado` (Aprobado|Reprobado|Pendiente).

- `cursoMatriculados`:

- `id`, `idCursoOferta` (FK ofertaCursos.id), `idPersona` (FK personas.id), `estado` (Aprobado|Reprobado|Cancelado|Pendiente), `resultado`.

- `pagos`:

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 12 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		


- `id`, `idPersona` (FK personas.id), `idCursosMatriculados` (FK cursoMatriculados.id), `formaPago`, `monto`, `estado`, `fechaPago`.
- `asistencia`:
 - `id`, `idCursosMatriculados` (FK cursoMatriculados.id), `asistio` (`Si`|`No`), `fecha`.
- `certificados`:
 - `id`, `idCursoMatriculado` (FK cursoMatriculados.id), `fechaEmision`, `urlCertificado`.
- `documentos`:
 - `id`, `id_persona` (FK personas.id), `documento_identidad`, `foto_3x4`, `recibo_publico`, `acta_grado`, `eps_salud`.
- `contenidoapoyos`:
 - `id`, `id_oferta_curso`, `titulo`, `descripcion`, `url_contenido`, `createdAt`, `updatedAt`.
- `notas_modulos`:
 - `id`, `id_curso_matriculado` (FK), `id_modulo` (FK), `estado` (ENUM), `fecha_registro`.

Relaciones destacadas:

- Persona ↔ Usuario (1:1), Persona ↔ Matriculado (1:N), Curso ↔ Oferta (1:N), Oferta ↔ Módulos (1:N), Matriculado ↔ Pago/Asistencia/Certificado (1:N), Módulo ↔ Docente (N:M por oferta).

Normalización e integridad

- Claves foráneas entre entidades principales; enums para estados y tipos.
- IDs autoincrementales; restricciones de unicidad en identificación personal.
- Recomendación: índices adicionales en FKs (`idPersona`, `idCursoOferta`, `idCursosMatriculados`, `idOfertaCurso`) para mejorar consultas.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 13 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

Seguridad de datos

- Conexión SSL requerida (ver `config/config.json`).
- Recomendación: cifrado en reposo a nivel de servidor/volumen, máscara de datos sensibles para ambientes no productivos.
- Auditoría: registrar eventos críticos (login, creación de pagos, emisión de certificados).

Migraciones y versionado


- ORM: Sequelize.
- Comandos:
 - `npx sequelize-cli db:migrate`
 - `npx sequelize-cli migration:generate --name <nombre>`
 - `npx sequelize-cli db:undo` (usar con precaución)
- Política: migraciones atómicas por cambio; revisiones en PR.

Backups y restauración

- `pg_dump -Fc -d funed -h host -U usuario -f backup.dump`
- `pg_restore -d funed -h host -U usuario backup.dump`
- Programar respaldos diarios; probar restauraciones trimestralmente.

Monitoreo y mantenimiento

- Health checks en `/api/health`.
- Logs: capturar stdout/stderr del proceso; establecer alertas ante errores recurrentes.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 14 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

- Job de auto-aprobación configurable y supervisado.

7. Desarrollo del Software

Estructura de carpetas


- Raíz: `c:\proyecto-funed-frontend-r\`
- Backend: `proyecto-funed-backend/`
 - `server.js` (entrada), `router/` (rutas), `controller/`, `services/`, `models/`, `migrations/`, `middlewares/`, `swagger/`.
- Frontend: `funed-fronted/`
 - `src/` (App.jsx, AuthContext.jsx, pages/, components/), `env.ts`, `supabaseClient.ts`, `vite.config.js`, `tailwind.config.js`.
- Flutter (opcional): `aplicacion-funed/flutter_application_1/`.

Estándares de codificación

- JS/TS: estilo consistente, nombres descriptivos, evitar variables de un carácter.
- Eslint: `npm run lint` en frontend.
- Control de errores: respuestas JSON claras, status codes correctos.

Flujo de trabajo con Git

- Estrategia: branches por feature (`feature/`), correcciones (`fix/`), release (`release/`).
- Commits: mensajes claros, referencia a issue cuando aplique.
- Code reviews: revisar seguridad, rendimiento y cobertura.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 15 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

Arquitectura y estado

- Backend: patrón controlador-servicio-modelo; middleware JWT.
- Frontend: Context de autenticación, protección de rutas y control por roles; React Router para navegación.
- Supabase: cliente creado en `supabaseClient.ts` para servicios auxiliares.

Pruebas y calidad


- Unit/integration (recomendado): Jest en backend, React Testing Library en frontend.
- Lint: obligatorio en CI.
- Cobertura: objetivo mínimo 70% (recomendado).

CI/CD y releases


- Pipeline: instalar, lint, build, pruebas, despliegue.
- Versionado: SemVer para releases.
- Artefactos: `dist/` del frontend; backend sin build (Node directo).

Rutas y endpoints

- Base: `/api` (recursos), `/auth` (autenticación).
- Salud: `GET /api/health`.
- Swagger: `GET /api-docs` (Bearer JWT requerido para rutas protegidas).
- Autenticación (`/auth`):
 - `POST /login` → Inicia sesión, devuelve JWT.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 16 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

- `POST /register` → Registra usuario.
- `GET /user` → Lista usuarios.
- `PATCH /user/:id` → Actualiza usuario.
- `POST /forgot-password` → Envía contraseña temporal y actualiza hash.
- Cursos y oferta (`/api`):
 - `GET /cursos`, `POST /cursos`, `GET /cursos/:id`, `PATCH /cursos/:id`, `DELETE /cursos/:id`.
 - `GET /ofertaCursos`, `POST /ofertaCursos`, `GET /ofertaCursos/docente/:id_persona`.
- Módulos y docentes:
 - `GET /modulos`, `POST /modulos`, `GET /modulos/oferta/:id_oferta_curso`.
 - `GET /modulo-docente`, `POST /modulo-docente`, `GET /modulo-docente/oferta/:id_oferta_curso`.
- Personas y documentos:
 - `GET /personas`, `POST /personas`, `PATCH /personas/:id`, `POST /personas/:id/send-welcome`.
 - `GET /documento`, `POST /documento`.
- Matrículas, pagos, asistencia, calificaciones, certificados:
 - `GET /matriculas`, `POST /matriculas`, `GET /cursosPersonas/:id`.
 - `GET /pago`, `POST /pago`, `GET /pago/:id`, `PATCH /pago/:id`, `DELETE /pago/:id`.
 - `GET /asistencia`, `POST /asistencia`, `GET /asistencia/persona/:id_persona/curso/:id_curso_matriculado`.
 - `POST /notas-modulo`, `GET /notas-modulo/modulo/:id_modulo/oferta/:id_oferta_curso`.
 - `GET /certificado`, `POST /certificado`, `POST /certificado/generar`, `GET /certificado/:id`, `PATCH /certificado/:id`, `DELETE /certificado/:id`.
- Email:

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 17 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

- `POST /email/send-welcome`, `POST /email/send-payment-approved`.

Ejemplo cURL (login):

...

```
curl -X POST https://proyecto-funed-backend.onrender.com/login \
-H "Content-Type: application/json" \
-d '{"email":"user@example.com","password":"secret"}'
```

...

Uso con JWT:

...

```
curl -H "Authorization: Bearer <TOKEN>" https://proyecto-funed-backend.onrender.com/api/cursos
```

...


Accesibilidad e i18n

- WCAG: contraste adecuado, navegación por teclado, etiquetas accesibles.
- Localización: preparar cadenas y formatos; evaluación futura de i18n en frontend.

Rendimiento

- Compresión HTTP opcional (gzip) si la librería está disponible.
- Caching del lado cliente (axios) y memoización en componentes.
- Índices en BD en claves de consulta frecuente.

Seguridad en el ciclo de vida

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 18 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

- OWASP: validar entradas, evitar inyección, gestionar sesiones y errores.
- JWT: expiración y rotación; proteger `JWT_SECRET`.
- Correos: proteger `SENDGRID_API_KEY` y limitar datos sensibles en mensajes.

Artefactos y empaquetado

- Frontend: `vite build` genera `dist/` listo para servir por CDN/web server.
- Backend: Node sin build; desplegar como servicio con logs y restart automático.

Archivos relevantes


- Backend: `server.js`, `router/index.js`, `middlewares/auth.middleware.js`, `services/emailService.js`.
- Frontend: `src/App.jsx`, `src/auth/AuthContext.jsx`, `src/supabaseClient.ts`, `vite.config.js`.

8. Funcionalidades del software

- Gestión de personas y usuarios.
- Gestión de cursos, módulos y oferta de cursos.
- Matrículas, pagos y estados de aprobación.
- Asistencia y calificaciones por módulo.
- Emisión y descarga de certificados (PDF).
- Notificaciones por correo (bienvenida, pago aprobado, recuperación).

9. Interfaces del sistema

- API REST JSON segura con JWT.
- Documentación OpenAPI/Swagger en `/api-docs`.

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 19 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		

- Integración de correo mediante SendGrid.
- Cliente web SPA (React) y posible cliente Flutter.


10. Bibliografía

Url Repositorio:

<https://github.com/oKCam04/proyecto-funed-frontend-r.git>

<https://github.com/oKCam04/proyecto-funed-backend.git>

- Express: <https://expressjs.com/>
- Sequelize: <https://sequelize.org/>
- Swagger UI Express: <https://www.npmjs.com/package/swagger-ui-express>
- SendGrid: <https://docs.sendgrid.com/>
- React: <https://react.dev/>
- Vite: <https://vitejs.dev/>
- TailwindCSS: <https://tailwindcss.com/>
- Supabase JS: <https://supabase.com/docs/reference/javascript>

	Centro de Teleinformática y Producción Industrial - Regional Cauca			Página 20 de 20
	Número de Documento:	FS-DOC Formato Manual Técnico	Fecha de Creación: 01/02/2025	Elaborado por: Grupo instructores área de software
	Nombre del Documento:	Formato para la construcción del Manual Técnico		