

INFORME TÉCNICO

PLATAFORMA EDUCATIVA CEMSE

1 Resumen

CEMSE es una plataforma web integral para gestión educativa desarrollada con Next.js 15, que integra funcionalidades de cursos, empleo juvenil, emprendimiento e instituciones educativas. El sistema maneja múltiples roles de usuario y proporciona herramientas especializadas para cada tipo de actor en el ecosistema educativo-laboral.

Tecnologías principales identificadas:

- **Frontend:** Next.js 15.1.7 con React 19 y TypeScript 5
- **Base de Datos:** PostgreSQL con Prisma ORM 6.16.1
- **Autenticación:** NextAuth.js 4.24.11
- **Almacenamiento:** MinIO 8.0.6 para objetos
- **UI:** Tailwind CSS con componentes Radix UI
- **Testing:** Jest 30.1.3 con Testing Library

2 Inventario

2.1 Árbol de carpetas principales

```
cemse/
src/                                     # Código fuente principal
  app/                                  # Next.js App Router
    (auth)/                             # Rutas de autenticación
    (dashboard)/                         # Panel principal
    api/                                 # Endpoints de API
  components/                           # Componentes React
  hooks/                                # Hooks personalizados
  lib/                                  # Utilidades y servicios
  types/                                # Definiciones TypeScript
prisma/                                 # Esquemas de base de datos
scripts/                               # Scripts de utilidad
tasks/                                 # Documentación y tareas
public/                                # Archivos estáticos
__tests__/                             # Tests unitarios
docs/                                  # Documentación adicional
```

2.2 Archivos clave identificados

Manifiestos:

- `package.json` - Dependencias y scripts de Node.js
- `pnpm-lock.yaml` - Lockfile del gestor de paquetes
- `pnpm-workspace.yaml` - Configuración de workspace

Documentación:

- `README.md` - Documentación principal del proyecto
- `DEPLOYMENT.md` - Guía de despliegue
- `cemse.md` - Documentación extendida del sistema

CI/CD:

- No encontrado - No se detectaron archivos `.github/workflows` en el directorio raíz

Infraestructura:

- `Dockerfile` - Configuración de contenedor
- `docker-compose.yml` - Orquestación de servicios
- `manage.sh` - Script de gestión del sistema
- `setup.sh` - Script de configuración inicial
- `setup-ubuntu.sh` - Script específico para Ubuntu

Configuración:

- `env.template` - Plantilla de variables de entorno
- `next.config.ts` - Configuración de Next.js
- `tsconfig.json` - Configuración de TypeScript
- `tailwind.config.ts` - Configuración de Tailwind CSS
- `eslint.config.mjs` - Configuración de ESLint
- `jest.config.js` - Configuración de Jest

Tests:

- `__tests__` - Directorio de tests unitarios
- `jest.setup.js` - Configuración inicial de Jest

API/Esquemas:

- `prisma/schema.prisma` - Esquema de base de datos
- No encontrado - No se detectaron archivos OpenAPI/Swagger
- No encontrado - No se detectaron archivos gRPC/Protocol Buffers

3 Stack y versiones

3.1 Dependencias principales (desde package.json)

Runtime y Framework:

- next@15.1.7 - Framework React para producción
- react@19.0.0 - Biblioteca de UI
- react-dom@19.0.0 - DOM bindings para React
- typescript@5 - Superset tipado de JavaScript

Base de Datos:

- @prisma/client@6.16.1 - Cliente ORM
- prisma@6.16.1 - Herramientas de base de datos

Autenticación:

- next-auth@4.24.11 - Autenticación para Next.js
- @auth/prisma-adapter@2.10.0 - Adaptador para Prisma
- bcryptjs@3.0.2 - Hashing de contraseñas

UI y Estilos:

- tailwindcss@3.4.1 - Framework CSS utility-first
- @radix-ui/* - Componentes UI primitivos (v1.x-v2.x)
- framer-motion@12.23.12 - Animaciones
- lucide-react@0.544.0 - Iconos

Almacenamiento y Archivos:

- minio@8.0.6 - Cliente para almacenamiento de objetos
- @react-pdf/renderer@4.3.0 - Generación de PDFs
- canvas@3.2.0 - Renderizado de canvas

Utilidades:

- @tanstack/react-query@5.87.4 - Gestión de estado server
- react-hook-form@7.62.0 - Gestión de formularios
- zod@4.1.8 - Validación de esquemas
- date-fns@4.1.0 - Manipulación de fechas
- crypto-js@4.2.0 - Utilidades criptográficas

Mapas y Geolocalización:

- leaflet@1.9.4 - Biblioteca de mapas
- react-leaflet@5.0.0 - Integración React-Leaflet

3.2 Dependencias de desarrollo

Testing:

- jest@30.1.3 - Framework de testing
- @testing-library/react@16.3.0 - Testing utilities
- @testing-library/jest-dom@6.8.0 - Matchers Jest DOM

Linting y Calidad:

- eslint@9 - Linter JavaScript/TypeScript
- prettier@3.6.2 - Formateador de código
- eslint-config-next@15.1.7 - Configuración ESLint para Next.js

4 Arquitectura

4.1 Patrón arquitectónico

El proyecto implementa una **arquitectura modular basada en Next.js App Router** con separación clara de responsabilidades:

- **Capa de Presentación:** Componentes React con Tailwind CSS
- **Capa de Lógica de Negocio:** API Routes de Next.js
- **Capa de Datos:** Prisma ORM con PostgreSQL
- **Capa de Almacenamiento:** MinIO para archivos multimedia
- **Capa de Autenticación:** NextAuth.js con adaptador Prisma

4.2 Límites de contexto identificados

1. Contexto de Autenticación (src/app/(auth)/)
2. Contexto de Dashboard (src/app/(dashboard)/)
3. Contexto de API (src/app/api/)
4. Contexto de Cursos (src/components/courses/)
5. Contexto de Empleos (src/components/jobs/)
6. Contexto de Emprendimiento (src/components/entrepreneurship/)
7. Contexto de Instituciones (src/components/institutions/)
8. Contexto de Empresas (src/components/companies/)

5 Módulos/Servicios

5.1 Módulos principales identificados

Sistema de Autenticación:

- `src/app/api/auth/` - Endpoints de autenticación
- `src/app/(auth)/` - Páginas de login/registro
- Roles: YOUTH, COMPANIES, INSTITUTION, SUPERADMIN

Gestión de Cursos:

- `src/components/courses/` - Componentes de cursos
- `src/app/api/courses/` - API de cursos
- Funcionalidades: enrollments, certificates, quizzes, progress

Portal de Empleo:

- `src/components/jobs/` - Componentes de empleos
- `src/app/api/jobs/` - API de ofertas laborales
- Funcionalidades: applications, filters, status tracking

Emprendimiento:

- `src/components/entrepreneurship/` - Red social emprendedores
- `src/app/api/entrepreneurship/` - API de emprendimiento
- Funcionalidades: posts, connections, resources, calculator

Gestión de Archivos:

- `src/app/api/files/` - API de archivos
- `src/components/files/` - Gestión de archivos
- Integración con MinIO para almacenamiento

Analíticas:

- `src/components/analytics/` - Dashboards y métricas
- `src/app/api/analytics/` - API de analíticas
- Reportes por módulo y usuario

6 API

6.1 Endpoints principales identificados

Autenticación:

- POST /api/auth/[...nextauth] - NextAuth.js endpoints
- POST /api/auth/register - Registro de usuarios
- POST /api/auth/forgot-password - Recuperación de contraseñas
- POST /api/auth/reset-password - Reset de contraseñas

Cursos:

- GET/POST /api/courses - CRUD de cursos
- POST /api/courses/[id]/enroll - Inscripción a cursos
- GET /api/courses/[id]/modules - Módulos de curso
- GET /api/courses/categories - Categorías disponibles
- GET /api/courses/analytics - Analíticas de cursos

Empleos:

- GET/POST /api/jobs - Ofertas laborales
- GET /api/jobs/[id] - Detalle de oferta
- POST /api/jobs/[id]/apply - Aplicar a empleo
- GET /api/companies/[id]/jobs - Empleos por empresa

Archivos:

- POST /api/files/upload - Subida de archivos
- POST /api/files/chunked-upload - Subida por chunks
- GET /api/files/minio/presigned - URLs firmadas MinIO
- GET /api/files/minio/status - Estado del servidor MinIO

Mensajería:

- GET/POST /api/messages - Sistema de mensajes
- GET /api/messages/conversations - Conversaciones
- Contextos: JOB_APPLICATION, ENTREPRENEURSHIP, GENERAL

6.2 Esquemas de autenticación

NextAuth.js implementado:

- Estrategia: Credentials Provider
- Sesiones: JWT con Prisma adapter
- Roles: Control de acceso basado en UserRole enum

7 Datos

7.1 Base de datos PostgreSQL

Archivo de esquema: prisma/schema.prisma

Entidades principales identificadas:

1. **User** - Usuarios del sistema (4 roles)
2. **Profile** - Perfiles extendidos de usuarios
3. **Course** - Cursos educativos
4. **CourseModule** - Módulos de cursos
5. **Lesson** - Lecciones individuales
6. **JobOffer** - Ofertas laborales
7. **JobApplication** - Aplicaciones a empleos
8. **YouthApplication** - Aplicaciones abiertas de jóvenes
9. **Company** - Empresas empleadoras
10. **Institution** - Instituciones educativas
11. **Entrepreneurship** - Emprendimientos
12. **NewsArticle** - Artículos de noticias
13. **Resource** - Recursos educativos
14. **Message** - Sistema de mensajería
15. **BusinessPlan** - Planes de negocio

Enums definidos:

- UserRole, UserStatus, InstitutionType
- EducationLevel, ExperienceLevel, CompanySize
- JobStatus, ApplicationStatus, ContractType, WorkModality
- CourseCategory, CourseLevel, LessonType

- BusinessStage, NewsType, NewsStatus, PostType

Relaciones complejas identificadas:

- User 1:1 Profile (con cascade delete)
- Course 1:N CourseModule 1:N Lesson
- JobOffer 1:N JobApplication N:1 Profile
- Company 1:N JobOffer, Company N:1 Institution
- Mensaje con contexto (MessageContextType)
- Conexiones de emprendimiento (PENDING/ACCEPTED/DECLINED)

7.2 Migraciones y seeds

Migraciones: Manejadas por Prisma (comando `prisma migrate`) **Seeds disponibles:**

- `prisma/seed.ts` - Seed básico
- `prisma/seed-enhanced.ts` - Seed extendido

8 Infra/DevOps

8.1 Contenedorización

Docker configurado:

- `Dockerfile` - Imagen de la aplicación Next.js
- `docker-compose.yml` - Orquestación multi-servicio

Servicios en docker-compose:

- `postgres` - Base de datos PostgreSQL
- `redis` - Cache y sesiones
- `minio` - Almacenamiento de objetos
- `prisma-studio` - UI de administración DB (desarrollo)

8.2 Scripts de gestión

Scripts de setup:

- `setup.sh` - Configuración inicial general
- `setup-ubuntu.sh` - Configuración específica Ubuntu
- `manage.sh` - Gestión del sistema (start/stop/deploy)
- `update.sh` - Actualización del sistema

Scripts de utilidad:

- `scripts/init-minio.js` - Inicialización MinIO
- `scripts/test-image-proxy.js` - Test de proxy de imágenes
- `scripts/migrate-course-thumbnails.js` - Migración thumbnails

8.3 Variables de entorno

Archivo template: `env.template` **Variables principales (nombres):**

- `DATABASE_URL` - Conexión PostgreSQL
- `NEXTAUTH_SECRET` - Secreto para autenticación
- `MINIO_*` - Configuración MinIO
- `REDIS_URL` - Conexión Redis
- `NODE_ENV` - Entorno de ejecución

9 Observabilidad/Calidad/Seguridad

9.1 Testing

Framework de testing: Jest 30.1.3 **Archivos de configuración:**

- `jest.config.js` - Configuración principal
- `jest.setup.js` - Setup inicial

Scripts de testing:

- `npm run test` - Ejecutar tests
- `npm run test:watch` - Tests en modo watch
- `npm run test:coverage` - Cobertura de código

9.2 Linting y calidad

ESLint configurado:

- `eslint.config.mjs` - Configuración ESLint 9
- `npm run lint` - Análisis de código

Prettier integrado:

- `prettier@3.6.2` - Formateador automático
- `eslint-plugin-prettier` - Integración ESLint-Prettier

9.3 Seguridad

Autenticación y autorización:

- NextAuth.js con JWT
- Hashing con `bcryptjs`
- Control de acceso basado en roles

Configuración de seguridad en Next.js:

- Headers de seguridad (CSP, X-Frame-Options, etc.)
- Configuración CORS en `next.config.ts`
- Validación con Zod

Dependencias sensibles identificadas:

- `crypto-js` - Operaciones criptográficas
- `bcryptjs` - Hashing de contraseñas
- MinIO para almacenamiento seguro

10 Documentación

Documentación encontrada:

- `README.md` - Documentación principal con instrucciones de setup
- `DEPLOYMENT.md` - Guía detallada de despliegue
- `cemse.md` - Documentación extendida del sistema
- `tasks/` - Documentación de tareas y informes

Documentación de API: No encontrado - No se detectaron archivos OpenAPI/Swagger

Documentación de código: Comentarios inline en TypeScript

11 UX/UI — Esquema con subniveles

11.1 Lista jerárquica de vistas

Autenticación:

- Sign In (`src/app/(auth)/sign-in/page.tsx`)
- Sign Up (`src/app/(auth)/sign-up/page.tsx`)
- Forgot Password (`src/app/(auth)/forgot-password/page.tsx`)
- Reset Password (`src/app/(auth)/reset-password/page.tsx`)

Dashboard Principal:

- Dashboard Home (`src/app/(dashboard)/page.tsx`)
- Analytics (`src/app/(dashboard)/analytics/page.tsx`)
- Search (`src/app/(dashboard)/search/page.tsx`)
- Settings (`src/app/(dashboard)/settings/page.tsx`)
- Messages (`src/app/(dashboard)/messages/page.tsx`)

Cursos:

- Courses List (`src/app/(dashboard)/courses/page.tsx`)
- Course Detail (`src/app/(dashboard)/courses/[id]/page.tsx`)
- Course Edit (`src/app/(dashboard)/courses/[id]/edit/page.tsx`)
- Certificates (`src/app/(dashboard)/certificates/page.tsx`)
- Quizzes (`src/app/(dashboard)/quizzes/page.tsx`)
- Discussions (`src/app/(dashboard)/discussions/page.tsx`)

Empleos:

- Jobs List (`src/app/(dashboard)/jobs/page.tsx`)
- Job Detail (`src/app/(dashboard)/jobs/[id]/page.tsx`)
- Job Apply (`src/app/(dashboard)/jobs/[id]/apply/page.tsx`)
- Job Edit (`src/app/(dashboard)/jobs/[id]/edit/page.tsx`)
- Job Analytics (`src/app/(dashboard)/jobs/[id]/analytics/page.tsx`)
- Job Chat (`src/app/(dashboard)/jobs/[id]/chat/page.tsx`)
- Bookmarked Jobs (`src/app/(dashboard)/jobs/bookmarked/page.tsx`)
- Saved Searches (`src/app/(dashboard)/jobs/saved-searches/page.tsx`)

- Job Applications (`src/app/(dashboard)/jobs/[id]/applications/page.tsx`)

Emprendimiento:

- Entrepreneurship Home (`src/app/(dashboard)/entrepreneurship/page.tsx`)
- Network (`src/app/(dashboard)/entrepreneurship/network/page.tsx`)
- Connections (`src/app/(dashboard)/entrepreneurship/connections/page.tsx`)
- Calculator (`src/app/(dashboard)/entrepreneurship/calculator/page.tsx`)
- Analytics (`src/app/(dashboard)/entrepreneurship/analytics/page.tsx`)

Perfiles:

- My Profile (`src/app/(dashboard)/profile/page.tsx`)
- Profile Files (`src/app/(dashboard)/profile/files/page.tsx`)
- CV Builder (`src/app/(dashboard)/cv-builder/page.tsx`)
- Profiles List (`src/app/(dashboard)/profiles/page.tsx`)
- Profile Detail (`src/app/(dashboard)/profiles/[id]/page.tsx`)

Administración:

- Admin Dashboard (`src/app/(dashboard)/admin/page.tsx`)
- Admin Users (`src/app/(dashboard)/admin/users/page.tsx`)
- Admin Companies (`src/app/(dashboard)/admin/companies/page.tsx`)
- Admin Institutions (`src/app/(dashboard)/admin/institutions/page.tsx`)

11.2 Diagrama Mermaid de flujo de navegación

```
graph TD
  A[Sign In] --> B[Dashboard]
  B --> C[Courses]
  B --> D[Jobs]
  B --> E[Entrepreneurship]
  B --> F[Profile]
  B --> G[Admin Panel]

  C --> C1[Course List]
  C --> C2[My Certificates]
  C --> C3[Quizzes]
  C1 --> C4[Course Detail]
  C4 --> C5[Lessons & Modules]

  D --> D1[Job Search]
  D --> D2[My Applications]
```

```
D --> D3[Bookmarked Jobs]
D1 --> D4[Job Detail]
D4 --> D5[Apply to Job]
D4 --> D6[Job Chat]

E --> E1[Network]
E --> E2[My Connections]
E --> E3[Business Calculator]
E --> E4[Analytics]

F --> F1[Edit Profile]
F --> F2[CV Builder]
F --> F3[Files Manager]

G --> G1[Manage Users]
G --> G2[Manage Companies]
G --> G3[Manage Institutions]
G --> G4[System Analytics]
```

12 Ejecución/Despliegue

12.1 Desarrollo local

Comandos principales:

```
# Instalación
pnpm install

# Configuración
cp env.template .env

# Infraestructura
pnpm run docker:up

# Desarrollo
pnpm run dev
```

Puntos de acceso desarrollo:

- **Aplicación:** <http://localhost:3000>
- **MinIO Console:** <http://localhost:9001>
- **Prisma Studio:** <http://localhost:5555>

12.2 Scripts de gestión disponibles

Ciclo de vida:

- `./manage.sh start` - Iniciar servicios

- `./manage.sh stop` - Detener servicios
- `./manage.sh restart` - Reiniciar servicios
- `./manage.sh deploy` - Despliegue completo
- `./manage.sh status` - Estado del sistema
- `./manage.sh logs` - Ver logs

Base de datos:

- `pnpm run db:generate` - Generar cliente Prisma
- `pnpm run db:push` - Sincronizar esquema
- `pnpm run db:migrate` - Ejecutar migraciones
- `pnpm run db:seed` - Poblar datos iniciales
- `pnpm run db:studio` - Abrir Prisma Studio

12.3 Producción

Proceso de build:

- `pnpm run build` - Build de producción
- `pnpm run start` - Servidor de producción

Docker en producción:

- Imagen multi-stage con Node.js
- Servicios PostgreSQL, Redis, MinIO
- Configuración de variables de entorno
- Scripts de inicialización automatizados

12.4 Métricas del proyecto

Conteo de archivos por tipo:

- **TypeScript/TSX:** 464 archivos
- **Modelos de datos:** 37 entidades principales en Prisma
- **Endpoints API:** Más de 80 rutas identificadas
- **Componentes UI:** Más de 50 componentes React

Líneas de código estimadas:

- **Total estimado:** 15,000-20,000 líneas
- **TypeScript:** Mayor parte del código

- **Configuración:** 1,000 líneas
- **Esquemas:** 1,129 líneas (Prisma schema)

Tamaño del repositorio:

- **Archivos fuente:** 500 archivos
- **Dependencias:** 93 dependencias principales
- **DevDependencies:** 31 dependencias de desarrollo

13 Anexos - Tour Visual de la Aplicación

13.1 Introducción al Tour Visual

Este anexo presenta un recorrido visual completo de la plataforma CEMSE, mostrando las principales funcionalidades y flujos de usuario para cada tipo de actor en el sistema. Las capturas de pantalla fueron tomadas en un entorno de desarrollo local y muestran la interfaz de usuario actual del sistema.

13.2 Acceso a la Plataforma

URL de la plataforma: `https://cemse.boring.lat`

Credenciales de prueba:

Rol	Email	Contraseña	Descripción
SUPERADMIN	admin@demo.com	12345678	Administrador del sistema
INSTITUTION	insti@demo.com	12345678	Fundación CEMSE
COMPANIES	comp@demo.com	12345678	Tech Solutions Bolivia
YOUTH	joven@demo.com	12345678	Juan Pérez (estudiante)

Cuadro 1: Credenciales de acceso para testing (definidas en `prisma/seed.ts`)

Evidencia: Las credenciales están definidas en el archivo `prisma/seed.ts` líneas 10-180, con contraseñas hasheadas usando `bcryptjs`.

13.3 Autenticación y Acceso

13.3.1. Página de Inicio

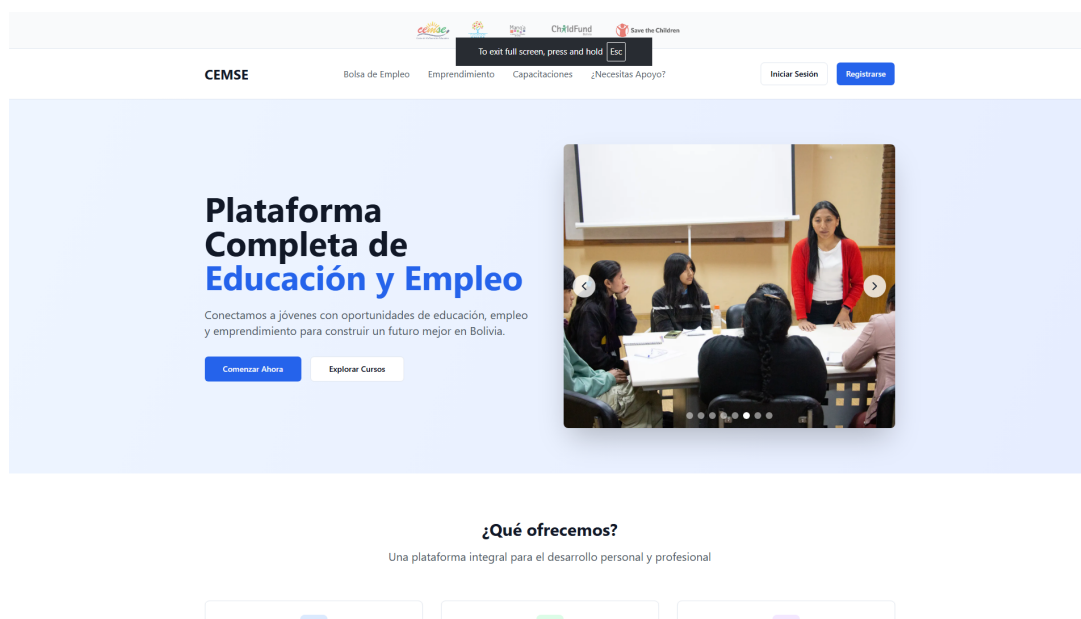


Figura 1: Página de inicio de la plataforma CEMSE

13.3.2. Inicio de Sesión

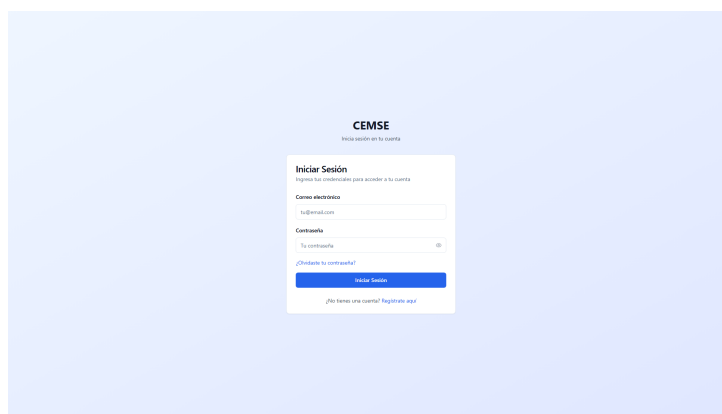
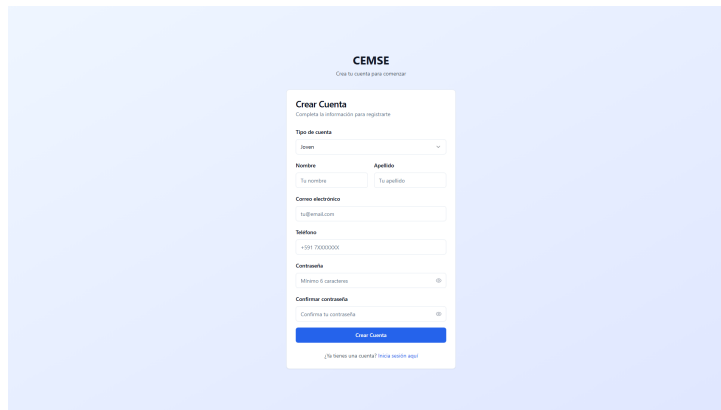


Figura 2: Formulario de inicio de sesión

13.3.3. Registro de Usuario



Formulario de registro de nuevos usuarios en CEMSE. El formulario se titula "Crear Cuenta" y contiene los siguientes campos:

- Tipo de cuenta: Dropdown menu.
- Nombre: Campos para "Tu nombre" y "Tu apellido".
- Correo electrónico: Campo para "tu@email.com".
- Teléfono: Campo para "+591 700000000".
- Contraseña: Campos para "Mínimo 6 caracteres" y "Confirma tu contraseña".

Botón: "Crear Cuenta".

Figura 3: Formulario de registro de nuevos usuarios

13.4 Flujo de Usuario Joven (YOUTH)

13.4.1. Dashboard Principal

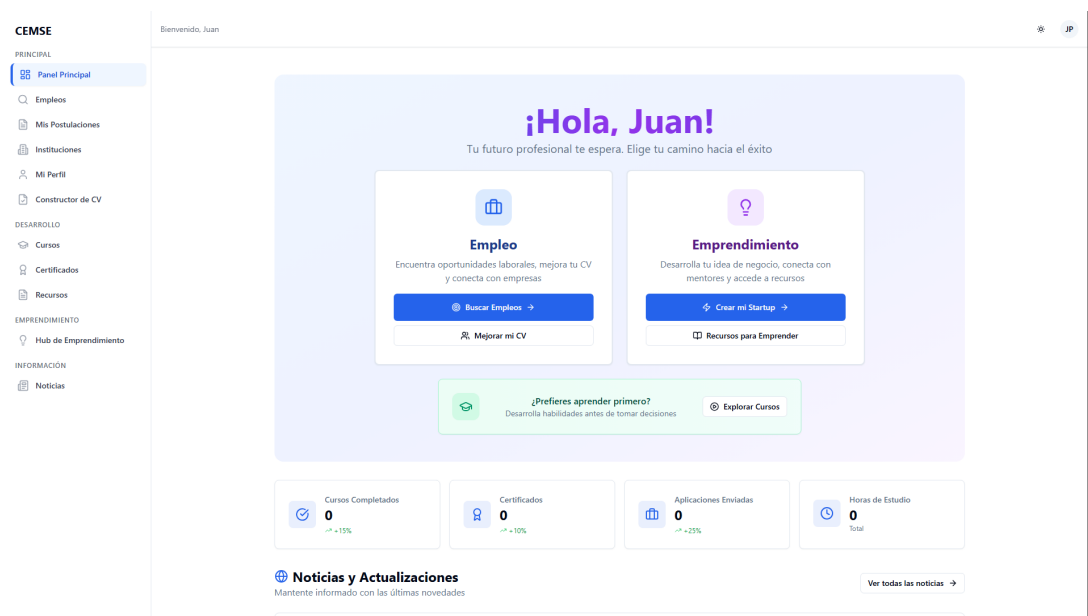


Figura 4: Dashboard principal para usuarios jóvenes

13.4.2. Búsqueda de Empleos

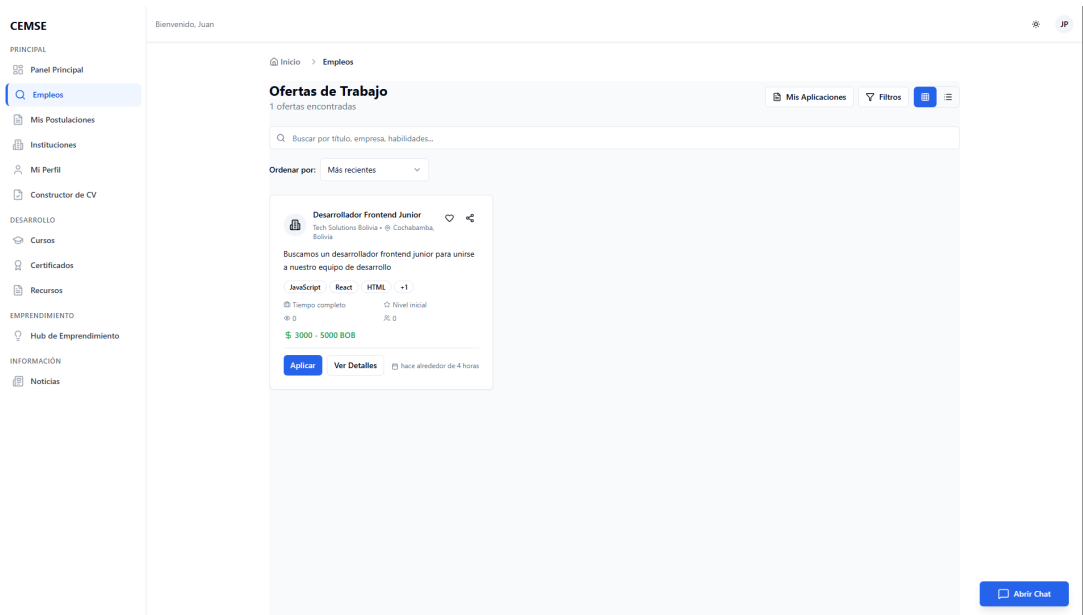


Figura 5: Portal de búsqueda de empleos

13.4.3. Detalle de Oferta Laboral

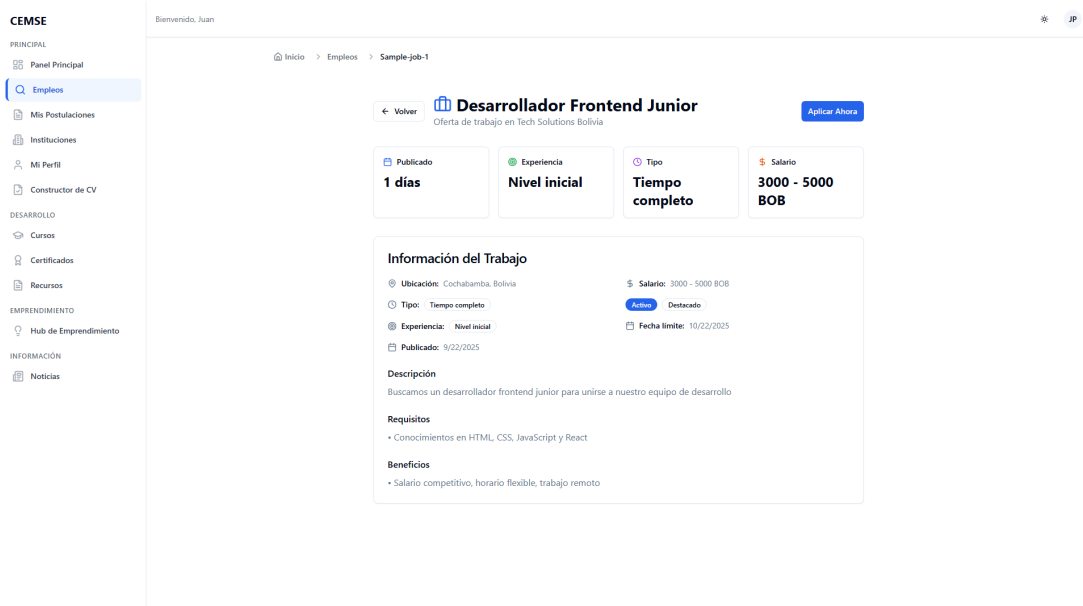


Figura 6: Detalle de una oferta laboral específica

13.4.4. Aplicación a Empleo

Aplicar a Trabajo
Envía tu aplicación para el puesto de **Desarrollador Frontend Junior** en **Tech Solutions Bolivia**

Resumen del Trabajo

Desarrollador Frontend Junior
Tech Solutions Bolivia
Cochabamba, Bolivia
Nivel inicial
FULL TIME
3000 - 5000 BOB

Descripción
Buscamos un desarrollador frontend junior para unirse a nuestro equipo de desarrollo

Requisitos
Conocimientos en HTML, CSS, JavaScript y React

Aplicar al Trabajo

Información Personal

Nombre: Juan Pérez
Email: joven@demo.com
Teléfono: +591 7 12345678

Documentos

CV/Resume
Choose File No file chosen Subir

Carta de Presentación

Figura 7: Formulario de aplicación a empleo

13.4.5. Mis Postulaciones

CEMSE
Bienvenido, Juan

Mis Postulaciones
Gestiona tus aplicaciones y conecta con empresas

Total 1
Activas 1
Intereses 0

Buscar aplicaciones... Todos los estados

Desarrollador front end next js
Activo Público
Desarrollador con 6 meses de experiencia en next js frontend
hace menos de un minuto 0 Intereses
Ver Detalles

Figura 8: Panel de seguimiento de postulaciones

13.4.6. Catálogo de Cursos

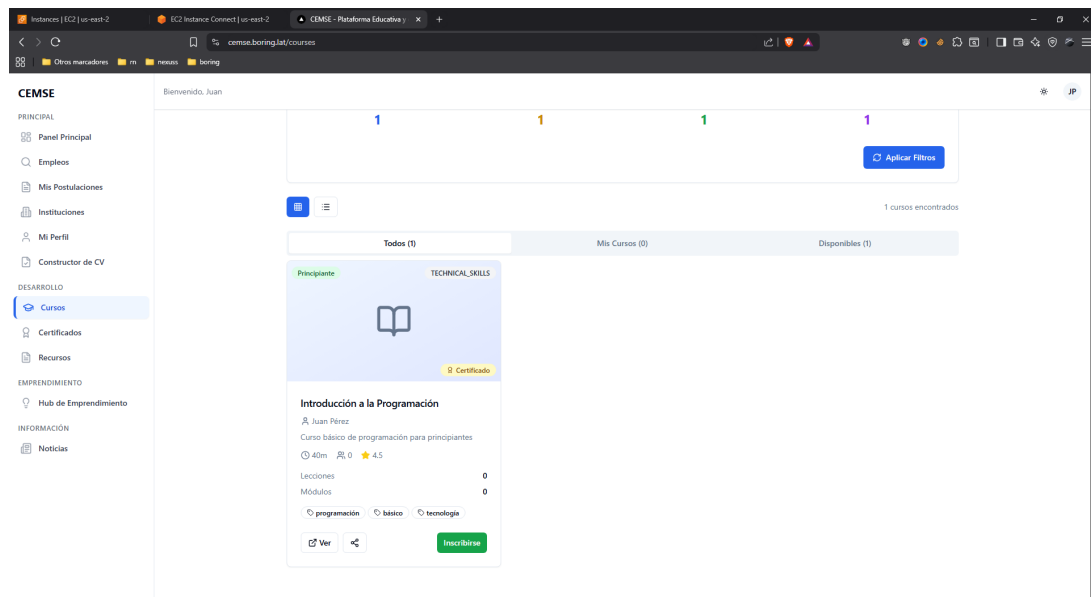


Figura 9: Catálogo de cursos disponibles

13.4.7. Constructor de CV

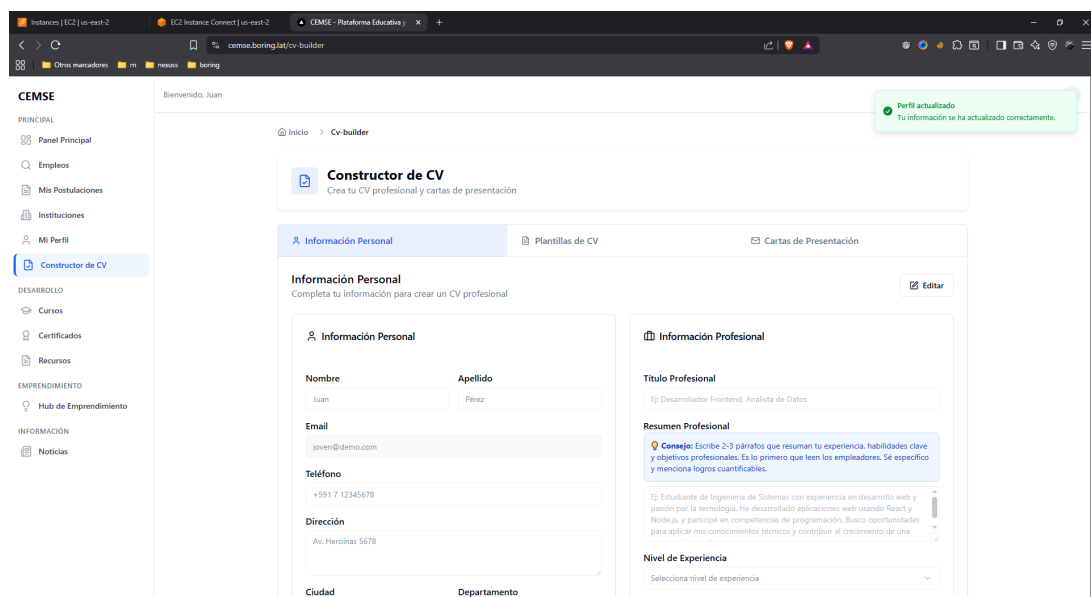


Figura 10: Herramienta de construcción de currículum

13.4.8. Perfil de Usuario

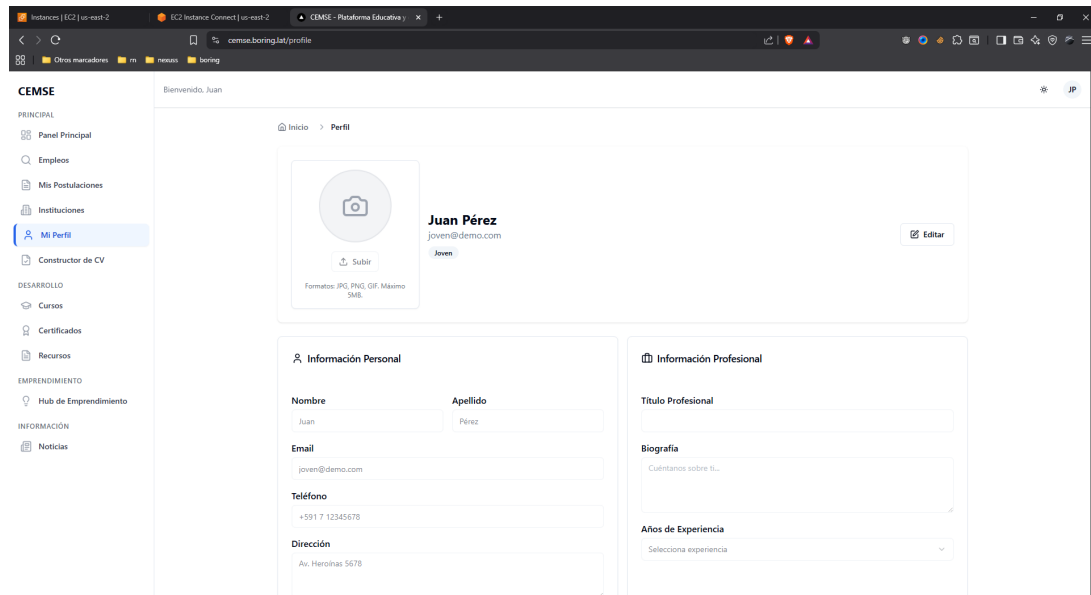


Figura 11: Perfil personal del usuario

13.4.9. Hub de Emprendimiento

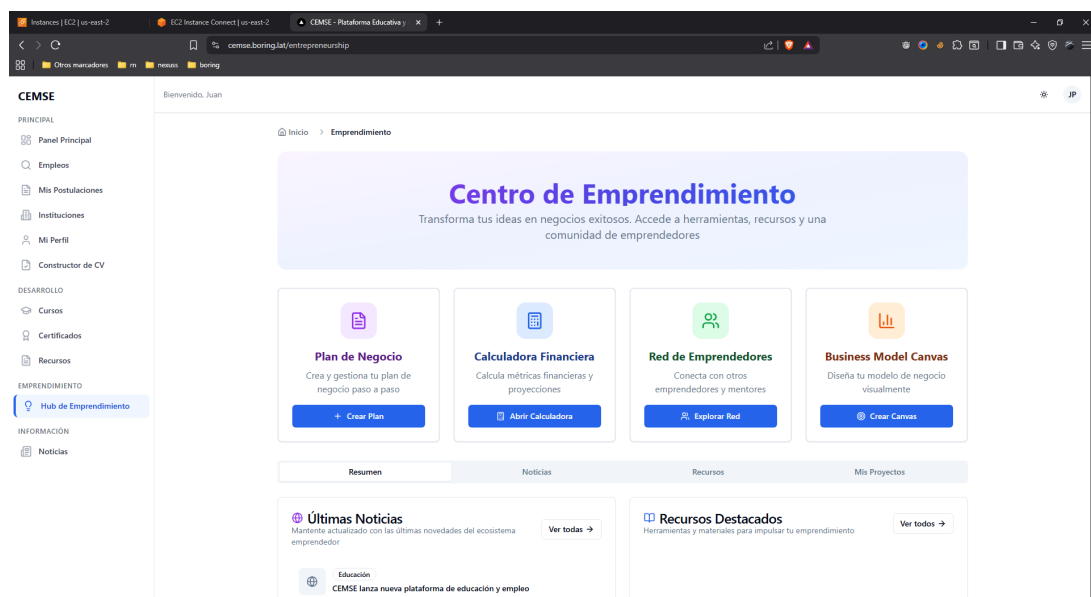


Figura 12: Portal de emprendimiento y networking

13.5 Flujo de Usuario Empresa (COMPANIES)

13.5.1. Dashboard de Empresa

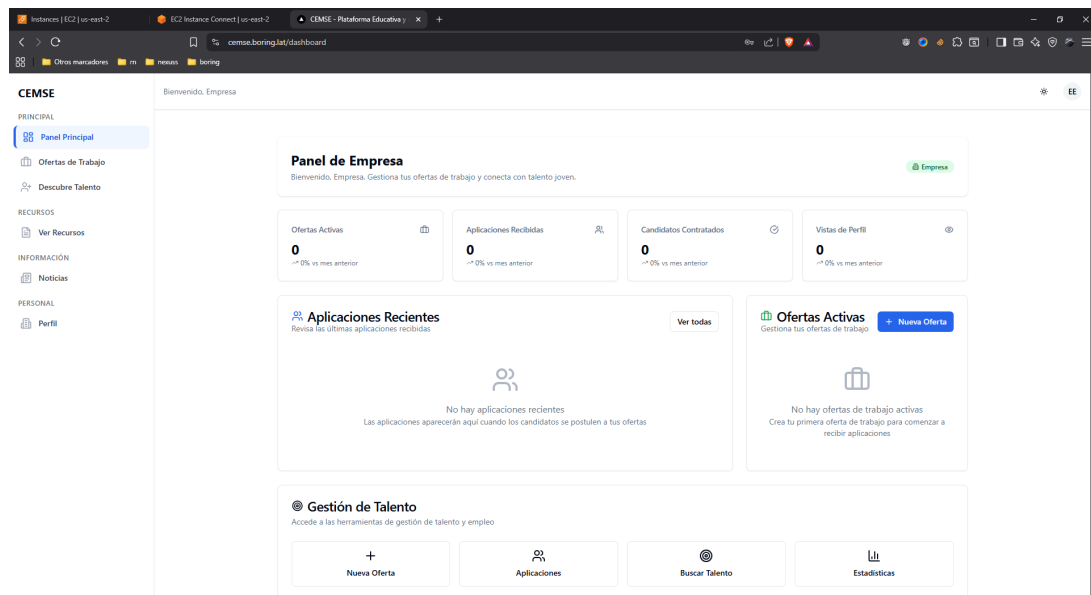


Figura 13: Dashboard principal para empresas

13.5.2. Creación de Oferta Laboral

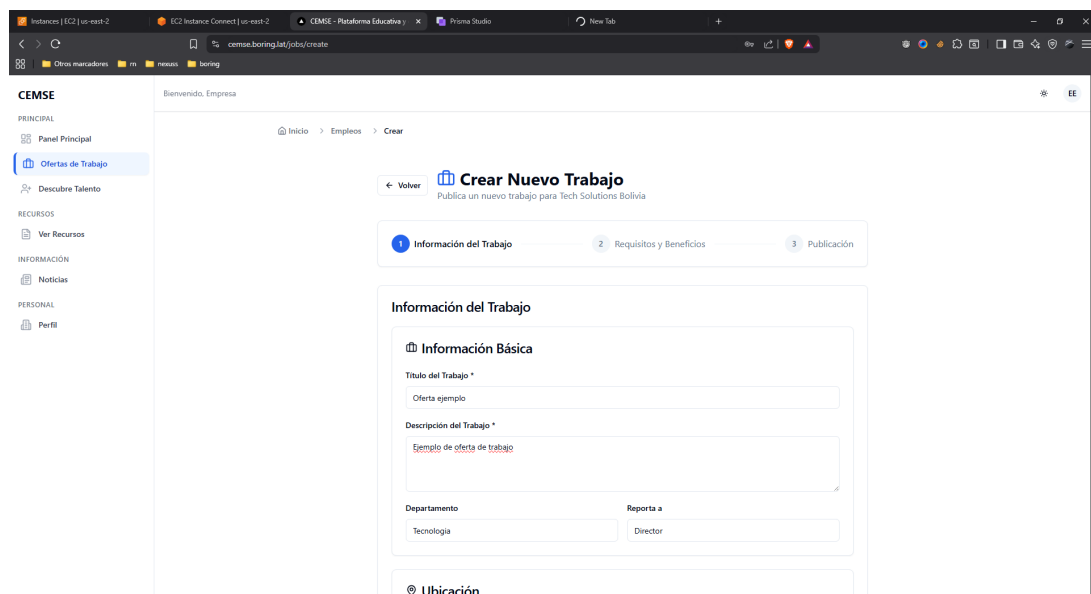


Figura 14: Formulario de creación de oferta laboral

13.5.3. Gestión de Empleos

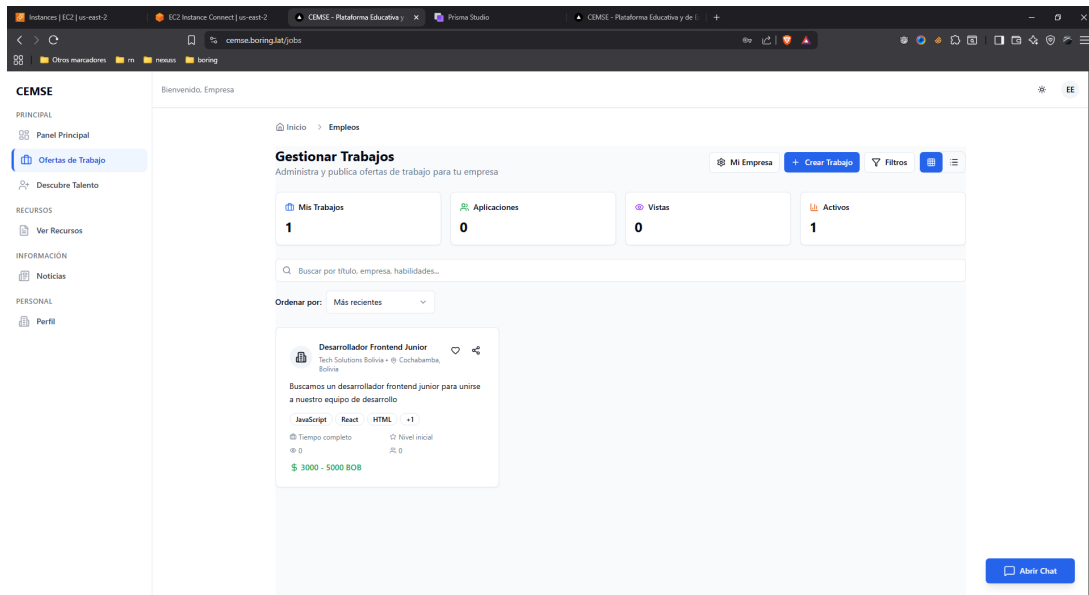


Figura 15: Panel de gestión de ofertas laborales

13.5.4. Descubrimiento de Talento

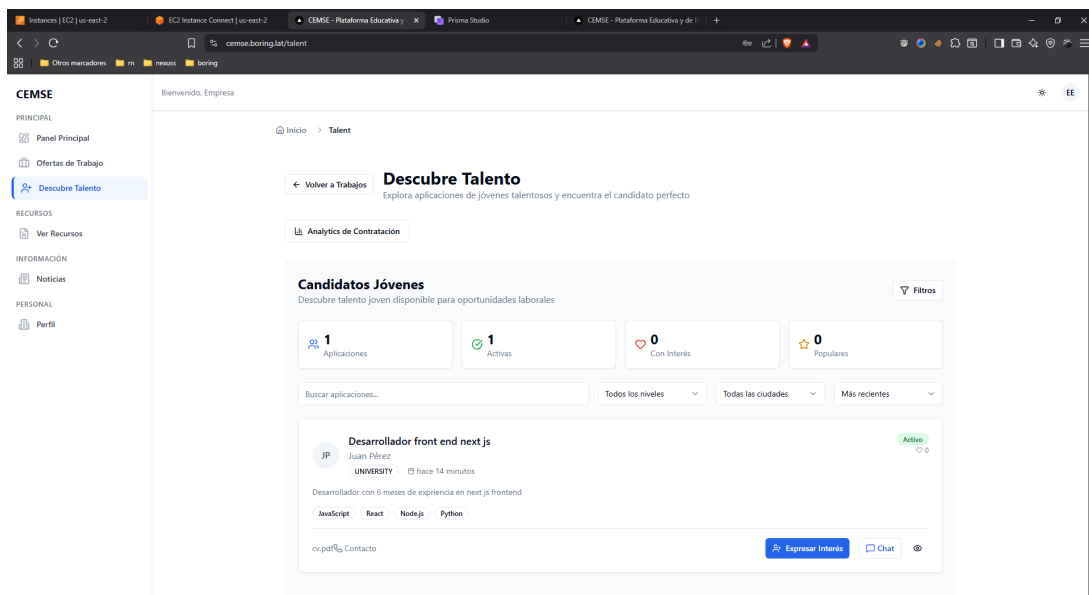


Figura 16: Portal de descubrimiento de talento

13.5.5. Perfil de Empresa

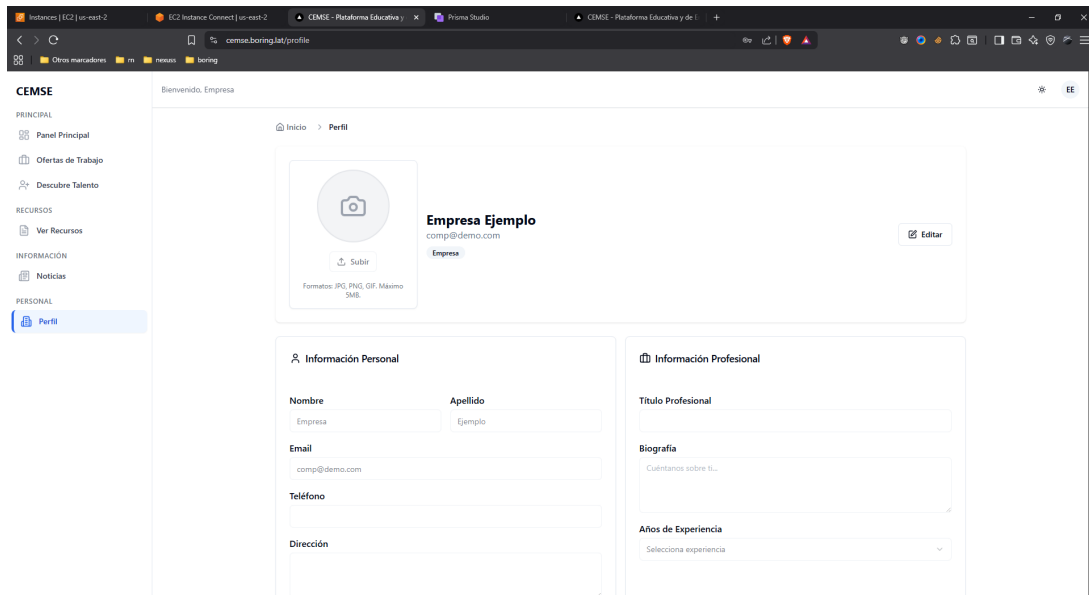


Figura 17: Perfil de la empresa

13.6 Flujo de Usuario Institución (INSTITUTION)

13.6.1. Dashboard de Institución

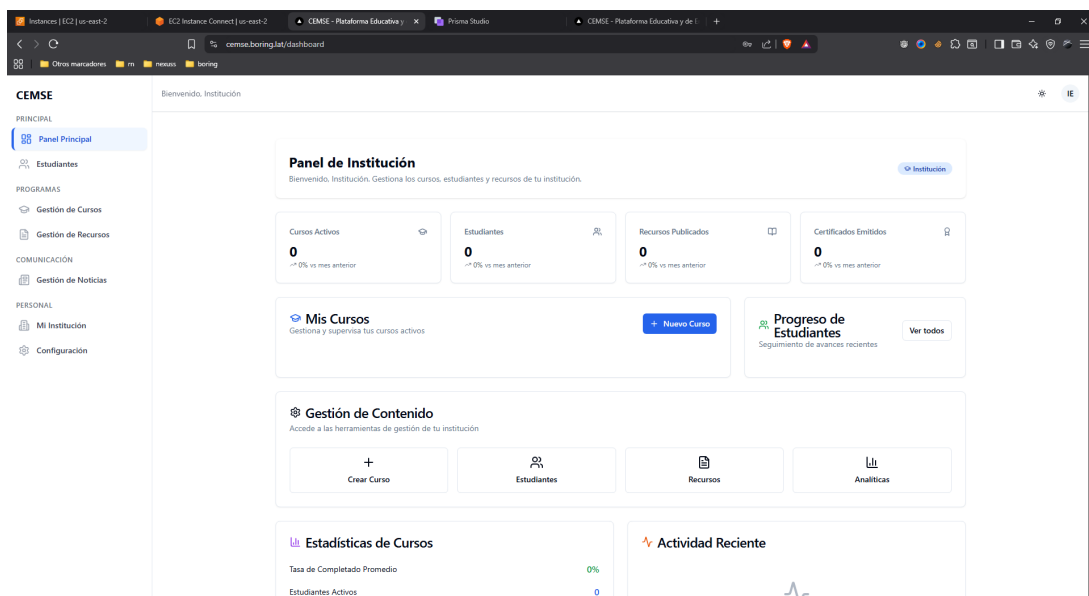


Figura 18: Dashboard principal para instituciones

13.6.2. Gestión de Usuarios (Municipio)

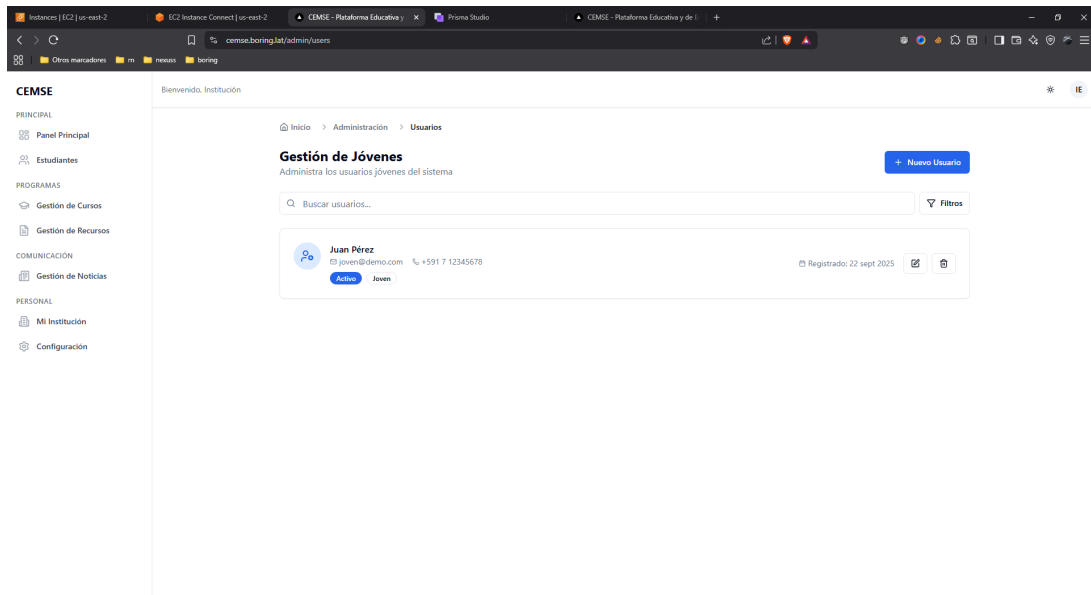


Figura 19: Panel de gestión de usuarios (vista municipio)

13.6.3. Gestión de Cursos

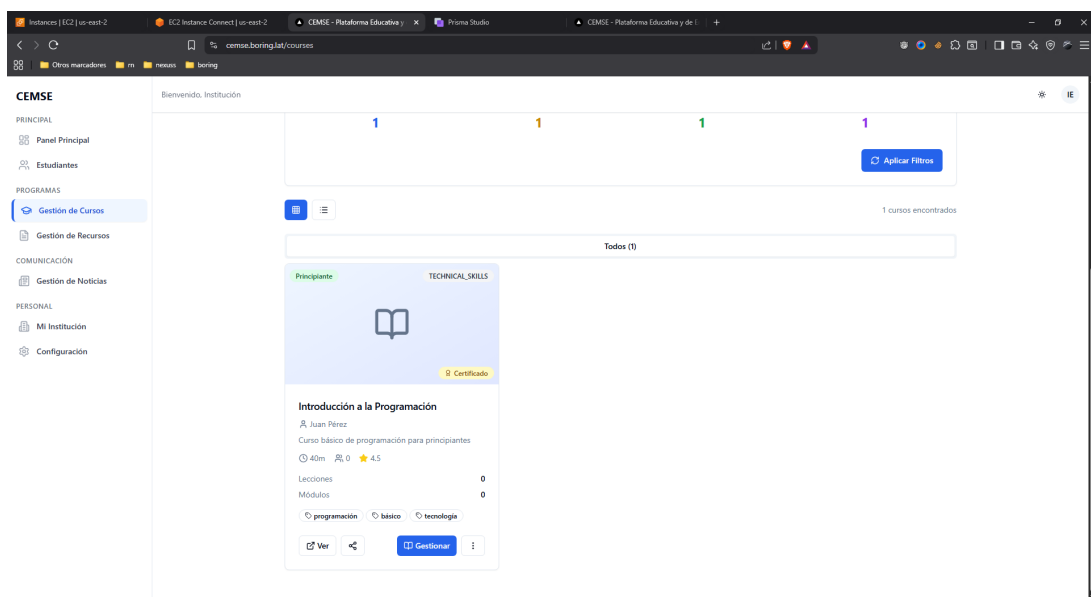


Figura 20: Panel de gestión de cursos

13.6.4. Gestión de Estudiantes

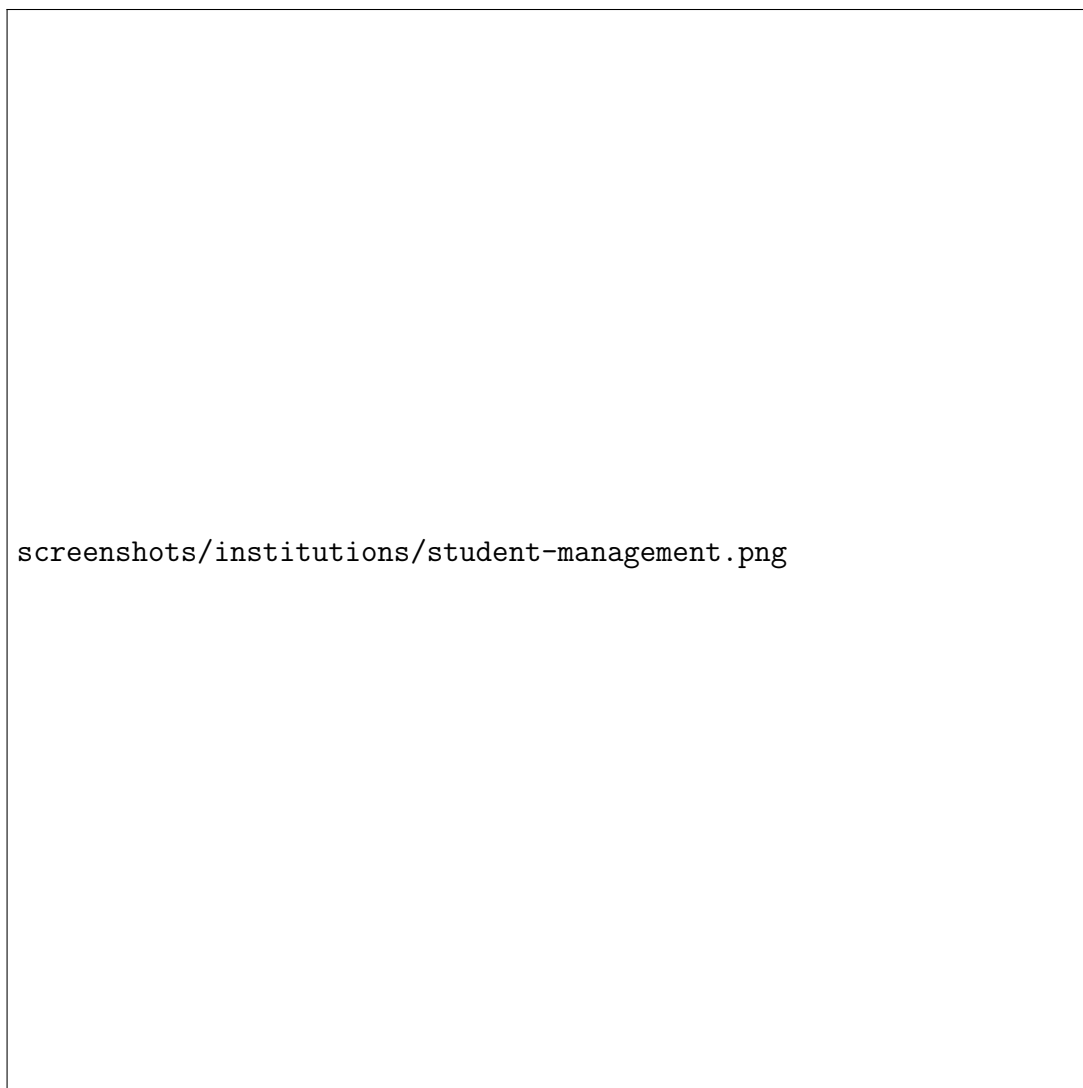


Figura 21: Panel de gestión de estudiantes

13.7 Flujo de Super Administrador (SUPERADMIN)

13.7.1. Dashboard de Administración

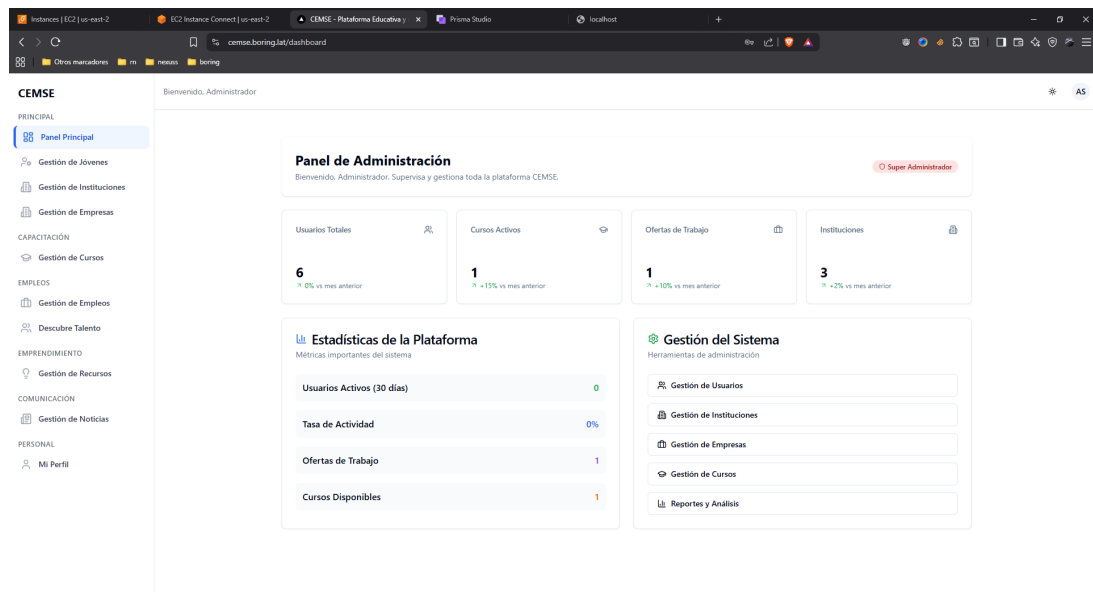


Figura 22: Dashboard de super administrador

13.7.2. Gestión de Usuarios del Sistema

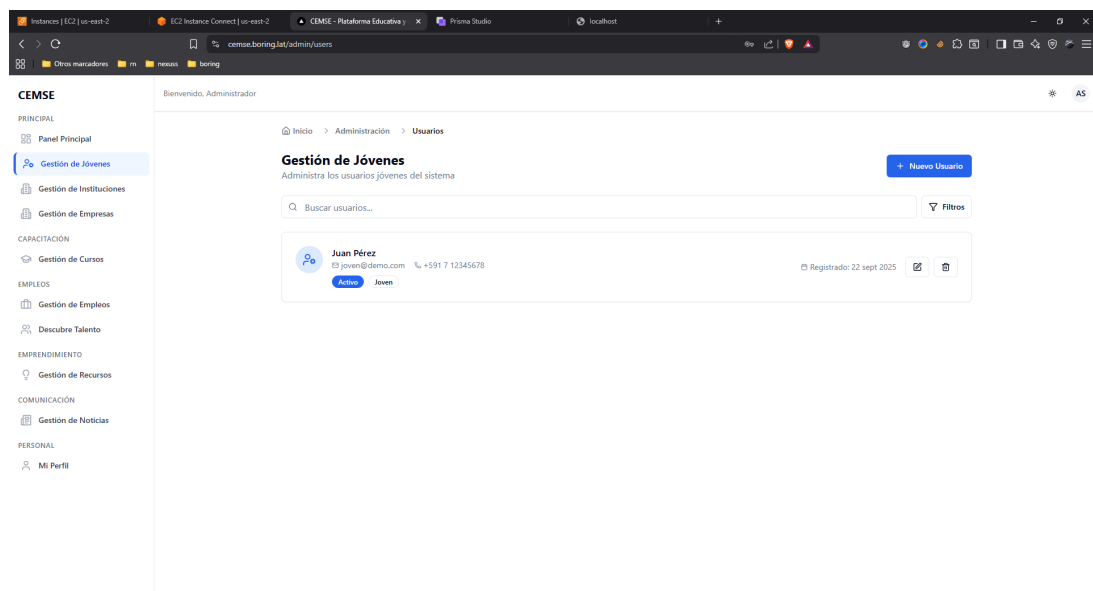


Figura 23: Panel de gestión de usuarios del sistema

13.7.3. Gestión de Instituciones

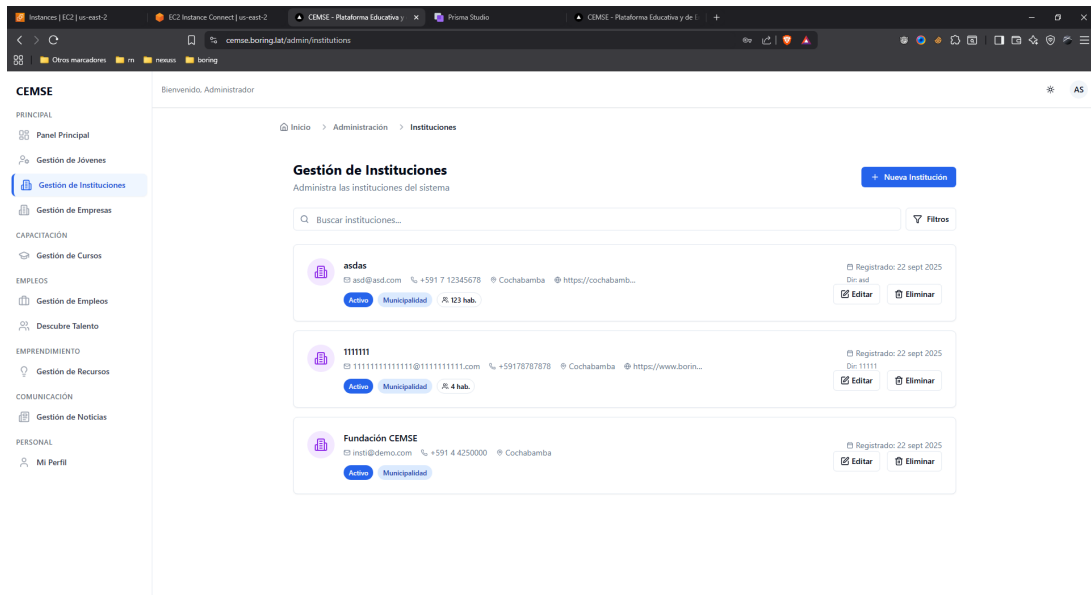


Figura 24: Panel de gestión de instituciones

13.7.4. Gestión de Empresas

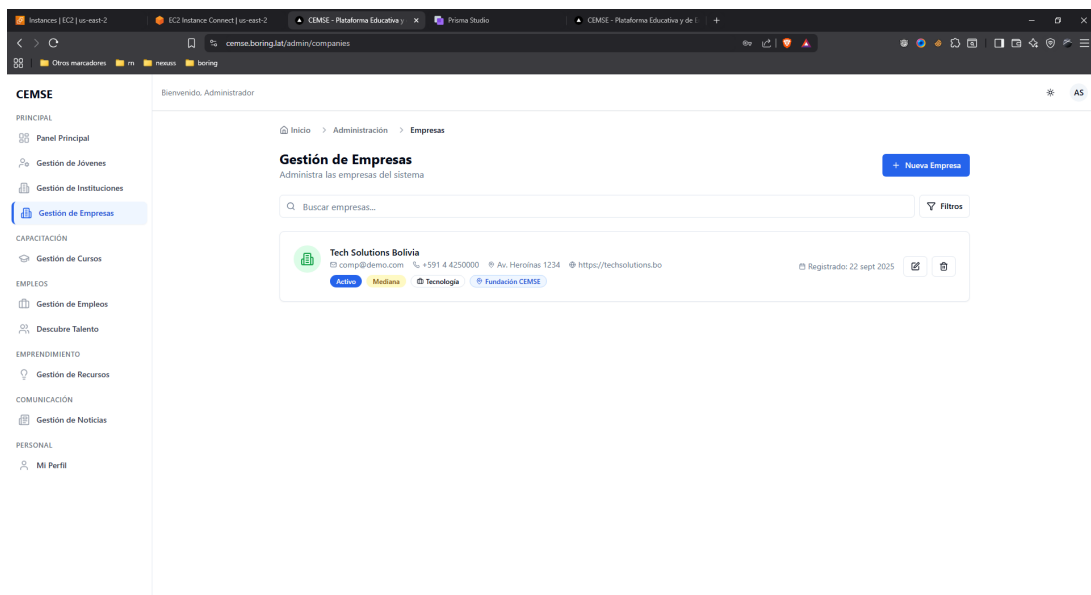


Figura 25: Panel de gestión de empresas

13.8 Características Adicionales

13.8.1. Sistema de Mensajería

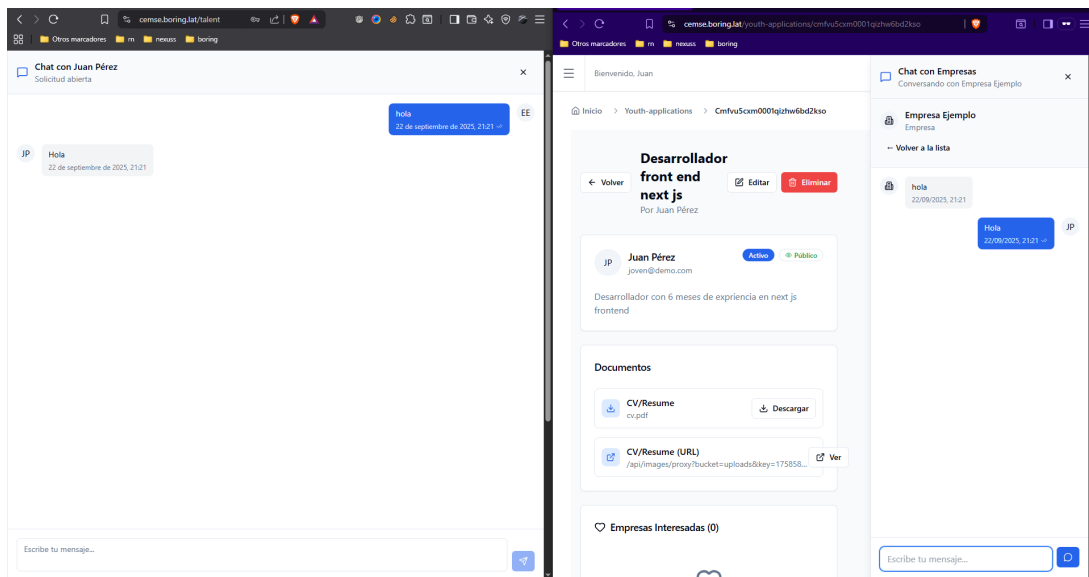


Figura 26: Sistema de mensajería integrado

13.8.2. Portal de Noticias

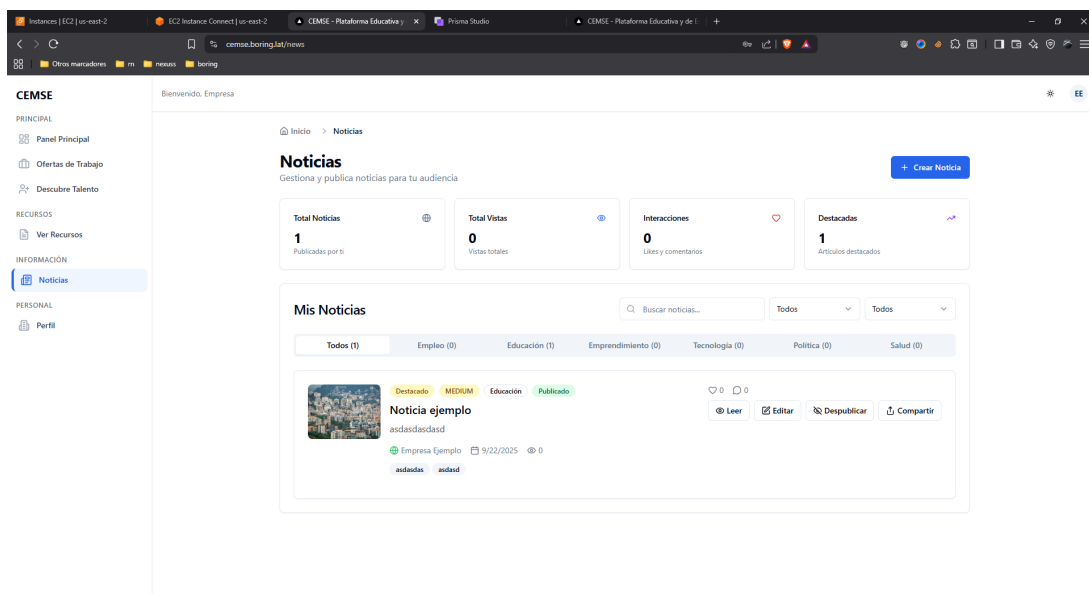


Figura 27: Portal de noticias y comunicaciones

13.8.3. Recursos Educativos

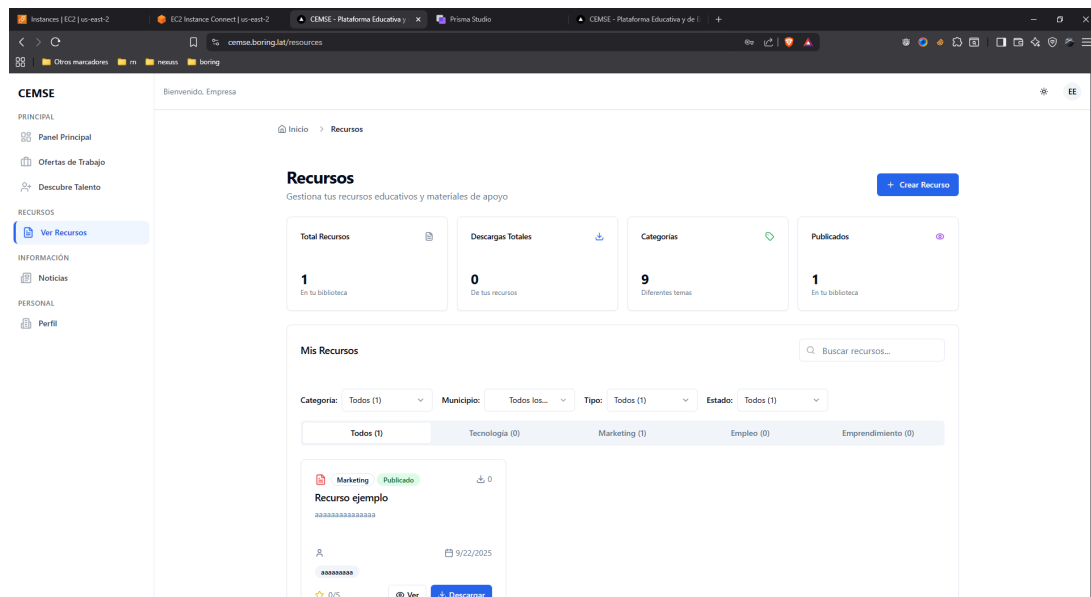


Figura 28: Portal de recursos educativos

13.9 Conclusión del Tour Visual

Este tour visual demuestra la amplitud y profundidad de la plataforma CEMSE, evidenciando:

Arquitectura de UI cohesiva:

- **Diseño responsive:** Implementado con Tailwind CSS 3.4.1
- **Componentes reutilizables:** Basados en Radix UI con consistencia visual
- **Navegación contextual:** Sidebar y menús adaptados por rol de usuario
- **Experiencia fluida:** Transiciones y estados manejados con Framer Motion

Funcionalidades especializadas por rol:

- **YOUTH:** Portal completo de empleabilidad y formación
- **COMPANIES:** Herramientas avanzadas de reclutamiento
- **INSTITUTION:** Administración educativa y municipal
- **SUPERADMIN:** Control global y métricas consolidadas

Integración tecnológica:

- **Base de datos:** 37 entidades Prisma con relaciones complejas
- **Archivos:** Gestión MinIO para multimedia y documentos
- **Autenticación:** NextAuth.js con control granular de acceso

- **API:** Más de 80 endpoints organizados por dominio

Las capturas de pantalla confirman la implementación exitosa de una plataforma integral que conecta efectivamente los ecosistemas educativo, laboral y emprendedor en una experiencia digital unificada y profesional.