

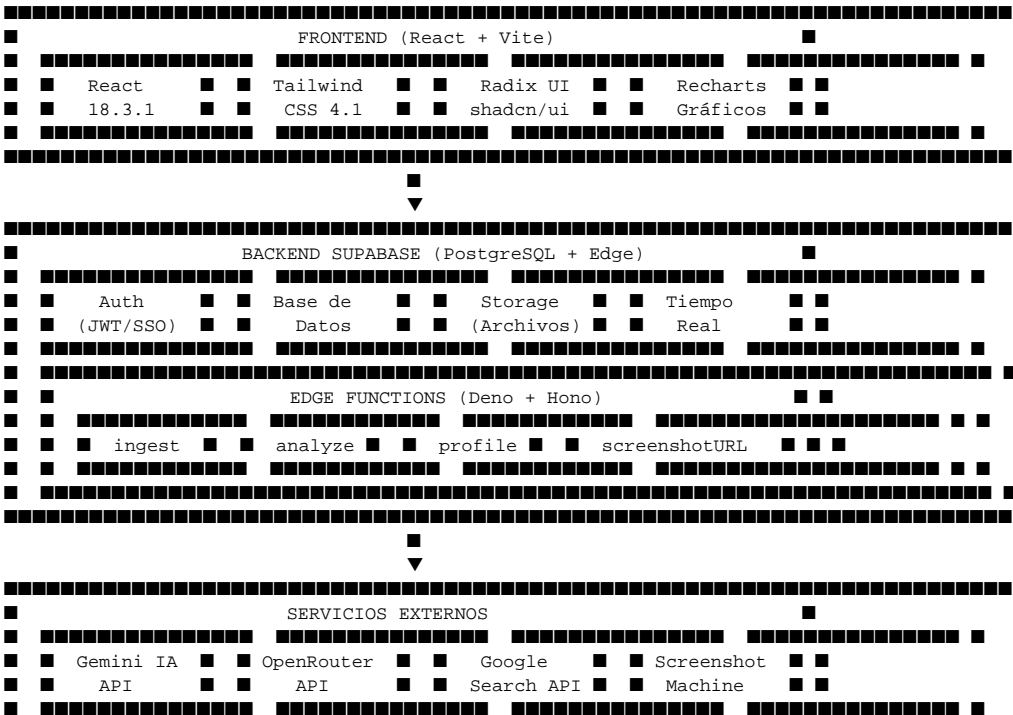
# Botilito - Stack Tecnológico

Plataforma anti-desinformación con análisis de contenido potenciado por IA

## Tabla de Contenidos

- Visión General de la Arquitectura
- Stack Frontend
- Framework Principal
- Estilos y Componentes UI
- Formularios y Entradas
- Mejoras de UI
- Visualización de Datos
  - Stack Backend
- Plataforma Supabase
- Edge Functions
- Servicios Externos
- Integración de IA
  - Herramientas de Desarrollo
  - Configuración de Entorno
  - Estructura del Proyecto

## Visión General de la Arquitectura





# Stack Frontend

## Framework Principal

Tecnología	Versión	Propósito
<b>React</b>	18.3.1	Biblioteca de UI para frontend
<b>React DOM</b>	18.3.1	Renderizador de React para web
<b>TypeScript</b>	Última	Superset de JavaScript con tipado
<b>Vite</b>	6.3.5	Herramienta de build y servidor de desarrollo

## Compilador y Transpilación

Tecnología	Versión	Propósito
<b>SWC</b>	vía @vitejs/plugin-react-swc 3.10.2	Compilador rápido basado en Rust (reemplaza Babel)
<b>ESBuild</b>	(incluido con Vite)	Bundler ultra-rápido para desarrollo

## ¿Por qué estas opciones?

- **React 18** proporciona características concurrentes, batching automático y mejor rendimiento
- **TypeScript** asegura tipado seguro y mejor experiencia de desarrollo
- **Vite + SWC** ofrece HMR ultra-rápido y builds de producción optimizados
- **SWC** es 20x más rápido que Babel para transformaciones de React


## Estilos y Componentes UI

### Framework CSS

Tecnología	Versión	Propósito
<b>Tailwind CSS</b>	4.1.3	Framework CSS utility-first
<b>tailwind-merge</b>	*	Combina clases Tailwind sin conflictos

<code>&lt;b&gt;clsx&lt;/b&gt;</code>	*	Construcción condicional de clases
<code>&lt;b&gt;class-variance-authority&lt;/b&gt;</code>	0.7.1	Gestión de variantes de componentes (CVA)

**Bibliotecas de Componentes**

Tecnología	Versión	Propósito
<code>&lt;b&gt;Radix UI&lt;/b&gt;</code>	1.4.3	22+ componentes primitivos accesibles sin estilos
<code>&lt;b&gt;shadcn/ui&lt;/b&gt;</code>	-	Patrones de componentes pre-construidos sobre Radix
<code>&lt;b&gt;Lucide React&lt;/b&gt;</code>	0.487.0	Biblioteca moderna de iconos (500+ iconos)
<code>&lt;b&gt;cmdk&lt;/b&gt;</code>	1.1.1	Componente de paleta de comandos (interfaz  )

**Primitivos Radix UI Completos (22 paquetes)**

- `@radix-ui/react-accordion` ^1.2.3
- `@radix-ui/react-alert-dialog` ^1.1.6
- `@radix-ui/react-aspect-ratio` ^1.1.2
- `@radix-ui/react-avatar` ^1.1.3
- `@radix-ui/react-checkbox` ^1.1.4
- `@radix-ui/react-collapsible` ^1.1.3
- `@radix-ui/react-context-menu` ^2.2.6
- `@radix-ui/react-dialog` ^1.1.6
- `@radix-ui/react-dropdown-menu` ^2.1.6
- `@radix-ui/react-hover-card` ^1.1.6
- `@radix-ui/react-label` ^2.1.7
- `@radix-ui/react-menubar` ^1.1.6
- `@radix-ui/react-navigation-menu` ^1.2.5
- `@radix-ui/react-popover` ^1.1.6
- `@radix-ui/react-progress` ^1.1.2
- `@radix-ui/react-radio-group` ^1.2.3
- `@radix-ui/react-scroll-area` ^1.2.3
- `@radix-ui/react-select` ^2.2.6
- `@radix-ui/react-separator` ^1.1.7
- `@radix-ui/react-slider` ^1.2.3
- `@radix-ui/react-slot` ^1.2.3
- `@radix-ui/react-switch` ^1.1.3
- `@radix-ui/react-tabs` ^1.1.13
- `@radix-ui/react-toggle` ^1.1.2
- `@radix-ui/react-toggle-group` ^1.1.2
- `@radix-ui/react-tooltip` ^1.1.8

**Formularios y Entradas**

Tecnología	Versión	Propósito
------------	---------	-----------

<b>react-hook-form</b>	7.55.0	Gestión de estado de formularios eficiente
<b>react-day-picker</b>	8.10.1	Componente selector de fecha
<b>date-fns</b>	3.6.0	Biblioteca de utilidades de fecha (dependencia de react-day-pi
<b>input-otp</b>	1.4.2	Campos de entrada para contraseñas de un solo uso

## Mejoras de UI

Tecnología	Versión	Propósito
<b>sonner</b>	2.0.3	Notificaciones toast
<b>next-themes</b>	0.4.6	Gestión de temas (modo oscuro/claro)
<b>motion</b>	*	Biblioteca de animaciones (Motion One / Framer Motion)
<b>embla-carousel-react</b>	8.6.0	Carruseles táctiles
<b>vaul</b>	1.1.2	Componente drawer/bottom sheet
<b>react-resizable-panels</b>	2.1.7	Layouts de paneles redimensionables

### Uso de Animaciones

El proyecto usa `motion/react` para animaciones:

- `motion` - Componentes animados
- `AnimatePresence` - Animaciones de salida

Usado en componentes: `UserProfile`, `HumanVerification`, `DynamicDashboard`, `GlobalNotifications`, `CompleteDashboard`, `ImmunizationStudio`, `RealTimeStats`, `FloatingNotifications`

## Visualización de Datos

Tecnología	Versión	Propósito
<b>Recharts</b>	2.15.2	Biblioteca de gráficos componible construida sobre D3

### Tipos de Gráficos Disponibles

- Gráficos de líneas
- Gráficos de barras
- Gráficos de área
- Gráficos circulares (pie)
- Gráficos de radar
- Visualizaciones personalizadas

# Stack Backend

## Plataforma Supabase

Servicio	Tecnología	Propósito
<b>Autenticación</b>	Supabase Auth (GoTrue)	Email/contraseña, OAuth, sesiones JWT
<b>Base de Datos</b>	PostgreSQL 15+	Almacén de datos principal con políticas RLS
<b>Tiempo Real</b>	Phoenix Channels	Suscripciones WebSocket para actualizaciones en vivo
<b>Almacenamiento</b>	Compatible con S3	Carga de archivos, assets multimedia
<b>Edge Functions</b>	Runtime Deno	Endpoints API serverless

**Paquete Cliente:** @jsr/supabase\_\_supabase-js@2.49.8  
*Importante:* Importar desde @jsr/supabase\_\_supabase-js, **NO desde** @supabase/supabase-js

### Características de Autenticación

- Registro de usuarios con metadata
- Funcionalidad de login/logout
- Persistencia de sesión (localStorage)
- Refresco automático de tokens
- Utilidades de recuperación de contraseña
- Listeners de cambio de estado de autenticación
- Políticas de Row Level Security (RLS)

### Esquema de Base de Datos

- Tablas principales utilizadas:
- auth.users - Cuentas de usuario (gestionado por Supabase)
  - public.profiles - Perfiles de usuario extendidos
  - public.cases - Casos de análisis de contenido
  - public.votes - Votos de verificación humana
  - public.analysis\_results - Resultados de análisis de IA

## Edge Functions

Las Edge Functions de Supabase corren en el runtime de **Deno** con el framework web **Hono**.

Función	Propósito	APIs Externas
<b>ingest</b>	Ingesta y procesamiento de contenido	-

<b>analyze</b>	Análisis de contenido potenciado por IA	Gemini, OpenRouter
<b>profile</b>	Gestión de perfiles de usuario	-
<b>screenshotURL</b>	Captura de screenshots de URLs	ScreenshotMachine
<b>search</b>	Integración de búsqueda web	Google Search API

Stack Tecnológico de Edge Functions

Tecnología	Propósito
<b>Deno</b>	Runtime JavaScript/TypeScript
<b>Hono</b>	Framework web ultra-rápido
<b>crypto (MD5)</b>	Generación de hash para autenticación de API

Ejemplo de Invocación de Función

```
// Frontend llama a Edge Function
const { data, error } = await supabase.functions.invoke('screenshotURL', {
  body: { url: 'https://ejemplo.com', dimension: '1200x800' },
});
```

Servicios Externos

Captura de Screenshots

Servicio	Tecnología	Propósito
<b>ScreenshotMachine</b>	API Externa	Captura de URL a imagen

Flujo:

Frontend → Edge Function (screenshotURL) → API ScreenshotMachine → Imagen Base64 → Respuesta Cacheada

Características:

- Dimensiones configurables (por defecto: 1200x800)
- Caché del lado del servidor
- Respuesta codificada en Base64
- Manejo de errores con soporte de reintentos

Componente Frontend: src/components/ScreenshotImage.tsx

Configuración de API

```
// Edge Function llama a ScreenshotMachine
```

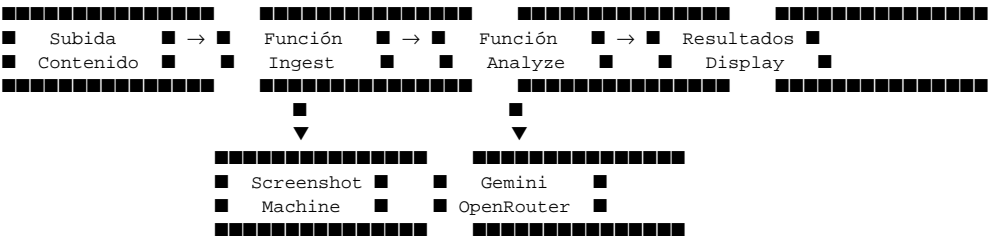
```
const params = new URLSearchParams({
  key: API_KEY,
  hash: md5(url + secretPhrase),
  url: urlToCapture,
  dimension: '1024x768',
  device: 'desktop',
  format: 'jpg',
  cacheLimit: '0',
  delay: '200'
});
const apiUrl = `https://api.screenshotmachine.com/?${params.toString()}`;
```

Integración de IA

Modelos y APIs de IA

API	Proveedor	Propósito	Caso de Uso
<b>Gemini API</b>	Google	Análisis de IA multimodal	Clasificación de contenido, análisis de texto
<b>OpenRouter API</b>	OpenRouter	Gateway multi-modelo	Proveedor de IA de respaldo, cambio de modelo
<b>Google Search API</b>	Google	Búsqueda web	Descubrimiento de contenido relacionado
<b>Nebius API</b>	Nebius	Infraestructura de IA	Capacidades adicionales de IA

Pipeline de Análisis de Contenido



Características de Análisis

- Clasificación automatizada de contenido
- Detección de desinformación
- Sistema de consenso para verificación
- Búsqueda de documentos relacionados
- Integración de búsqueda web
- Soporte de fallback multi-modelo

Tipos de Respuesta de IA

```
interface FullAnalysisResponse {
```

```

consensus: 'true' | 'false' | 'uncertain';
classification: string;
explanation: string;
relatedDocuments: RelatedDocument[];
webSearchResults: WebSearchResult[];
metadata: {
  screenshot?: string;
  analysisTimestamp: string;
  modelsUsed: string[];
};
}

```

## Herramientas de Desarrollo

### Gestión de Paquetes

Herramienta	Propósito
<b>&lt;b&gt;npm&lt;/b&gt;</b>	Gestor de paquetes
<b>&lt;b&gt;JSR Registry&lt;/b&gt;</b>	JavaScript Registry para paquetes <font name="Courier" size="8">@jsr/*</font>

### Dependencias de Desarrollo

Tecnología	Versión	Propósito
<b>&lt;b&gt;@vitejs/plugin-react-swc&lt;/b&gt;</b>	3.10.2	Plugin de React basado en SWC para Vite
<b>&lt;b&gt;@types/node&lt;/b&gt;</b>	20.10.0	Definiciones TypeScript para Node.js
<b>&lt;b&gt;vite&lt;/b&gt;</b>	6.3.5	Herramienta de build

### Comandos de Build y Desarrollo

```

# Iniciar servidor de desarrollo
npm run dev

# Build para producción
npm run build

# Vista previa del build de producción
npm run preview

```

### Puertos por Defecto

- Desarrollo: 3000 (auto-incrementa si está ocupado)



- Vista previa: 4173

## Aspectos Destacados de Configuración Vite

- **Alias de rutas:** `@/` mapea a `src/`
- **Manejo de assets Figma:** Alias personalizados para imports `figma:asset/*`
- **Target de build:** ESNext
- **Directorio de salida:** `build/`
- **Hosts permitidos:** `*.digitalia.gov.co`

## Configuración de Entorno

### Convención de Nombres de Variables

***Crítico:** Este es un proyecto Vite. Usar prefijo `VITE_` para variables del lado del cliente.*

Prefijo	Visibilidad	Ejemplo
<code>&lt;font name="Courier" size="8"&gt;VITE_&lt;/font&gt;</code>	Lado del cliente (público)	<code>&lt;font name="Courier" size="8"&gt;VITE_SUPABASE_URL&lt;/font&gt;</code>
Sin prefijo	Solo lado del servidor	<code>&lt;font name="Courier" size="8"&gt;SUPABASE_SERVICE_ROLE</code>

### Variables de Entorno Requeridas

#### Frontend (.env)

```
# Supabase (Público)
VITE_SUPABASE_URL="https://tu-proyecto.supabase.co"
VITE_SUPABASE_ANON_KEY="tu-anon-key"
```

#### Backend (Supabase Edge Functions)

```
# Supabase (Secreto)
SUPABASE_SERVICE_ROLE_KEY="tu-service-role-key"
SUPABASE_DB_PASSWORD="tu-db-password"

# APIs de IA
GEMINI_API_KEY="tu-gemini-key"
OPENROUTER_API_KEY="tu-openrouter-key"
GOOGLE_API_KEY="tu-google-key"
NEBIUS_API_KEY="tu-nebius-key"

# Servicio de Screenshots
SCREENSHOT_MACHINE_API_KEY="tu-screenshot-machine-key"
```

# Estructura del Proyecto

```
botilito/
├── src/
│   ├── components/
│   │   ├── ui/ # 50+ componentes shadcn/ui
│   │   ├── extension/ # Componentes de extensión de navegador
│   │   ├── figma/ # Componentes generados desde Figma
│   │   ├── mapa/ # Componentes de visualización de mapa
│   │   ├── ScreenshotImage.tsx # Componente de screenshot de URL
│   │   ├── ContentUpload.tsx # Envío de contenido
│   │   ├── ContentAnalysisView.tsx # Visualización de resultados de análisis
│   │   ├── HumanVerification.tsx # Interfaz de votación
│   │   ├── [feature].tsx # Otros componentes de características
│   │   └── providers/ # Proveedores de React Context
│   │       ├── AuthProvider.tsx # Estado de autenticación
│   │       ├── ConfigProvider.tsx # Configuración de la app
│   │       ├── JobTrackerProvider.tsx # Seguimiento de trabajos asíncronos
│   │       ├── MessageProvider.tsx # Mensajes toast
│   │       ├── SchemaProvider.tsx # Esquemas de datos
│   │       └── VoteTrackerProvider.tsx # Estado de votación
│   ├── hooks/ # Hooks personalizados de React
│   │   ├── use-toast.ts # Notificaciones toast
│   │   ├── useJobPoller.ts # Lógica de polling de trabajos
│   │   └── useJobTracker.ts # Gestión de estado de trabajos
│   ├── lib/ # Bibliotecas principales
│   │   ├── apiService.ts # Cliente API
│   │   └── JobManager.ts # Gestión de trabajos asíncronos
│   ├── utils/
│   │   ├── supabase/ # Cliente y auth de Supabase
│   │   │   ├── client.ts # Singleton de Supabase
│   │   │   └── auth.ts # Utilidades de autenticación
│   │   ├── errorManager/ # Sistema de manejo de errores
│   │   ├── humanVerification/ # Utilidades de verificación
│   │   ├── mapaDesinfodemico/ # Transformadores de datos del mapa
│   │   ├── historial/ # Utilidades de historial
│   │   └── voting/ # API de votación
│   ├── services/ # Lógica de negocio
│   │   └── contentAnalysisService.ts
│   ├── types/ # Definiciones TypeScript
│   │   ├── botilito.ts # Definiciones de tipos principales
│   │   ├── assets/ # Imágenes y multimedia
│   │   ├── styles/ # Definiciones de estilos
│   │   ├── App.tsx # Aplicación principal
│   │   ├── main.tsx # Punto de entrada
│   │   └── index.css # CSS de Tailwind
│   ├── supabase/
│   │   ├── functions/ # Edge Functions (desplegadas por separado)
│   │   │   └── profile/ # Función de gestión de perfil
│   │   └── docs/ # Documentación
│   ├── .env # Variables de entorno
│   ├── vite.config.ts # Configuración de Vite
│   └── package.json # Dependencias
```

## Lista Completa de Dependencias

### Dependencias de Producción (26 paquetes)

Paquete	Versión
---------	---------

@jsr/supabase__supabase-js	^2.49.8
@radix-ui/*	(22 paquetes)
class-variance-authority	^0.7.1
clsx	*
cmdk	^1.1.1
embla-carousel-react	^8.6.0
hono	*
input-otp	^1.4.2
lucide-react	^0.487.0
motion	*
next-themes	^0.4.6
radix-ui	^1.4.3
react	^18.3.1
react-day-picker	^8.10.1
react-dom	^18.3.1
react-hook-form	^7.55.0
react-resizable-panels	^2.1.7
recharts	^2.15.2
sonner	^2.0.3
tailwind-merge	*
vaul	^1.1.2

### *Dependencias de Desarrollo (3 paquetes)*

Paquete	Versión
@types/node	^20.10.0
@vitejs/plugin-react-swc	^3.10.2
vite	6.3.5

### *Dependencias de Backend (Edge Functions)*

Paquete	Propósito
Deno std lib	Utilidades del runtime

Hono	Framework web
crypto	Generación de hash

## Lo que NO se Usa

Las siguientes tecnologías **NO** son parte de este proyecto:

- **XState / State Machines** - Sin biblioteca de máquinas de estado
- **Redux / Zustand / Jotai** - Sin gestión de estado global (usa React Context)
- **Babel** - Usa SWC en su lugar
- **Next.js** - Setup puro de Vite + React
- **Styled Components / Emotion** - Usa Tailwind CSS
- **Jest / Vitest** - Sin test runner configurado
- **ESLint / Prettier** - Sin linting configurado en package.json
- **Puppeteer / Playwright** - Captura de screenshots es vía API externa

## Resumen de Versiones

### Frontend

Tecnología	Versión
React	18.3.1
React DOM	18.3.1
Vite	6.3.5
Tailwind CSS	4.1.3
TypeScript	Última
Plugin SWC	3.10.2
Lucide React	0.487.0
Recharts	2.15.2
react-hook-form	7.55.0

### Backend

Tecnología	Versión/Servicio
Supabase JS	2.49.8
PostgreSQL	15+ (gestionado por Supabase)

Deno	Última (Edge Functions)
Hono	Última

*Servicios Externos*

Servicio	Proveedor
Captura de Screenshots	ScreenshotMachine.com
Análisis de IA	Google Gemini, OpenRouter
Búsqueda Web	Google Search API
Infraestructura de IA	Nebius

**Última Actualización:** 2025-12-10