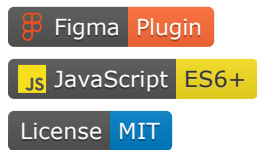


Simple Smells Detector - Plugin de Figma



Un plugin extensible para Figma que detecta automáticamente **usability smells** (indicios de problemas de usabilidad) en prototipos de interfaces de usuario, proporcionando retroalimentación accionable para mejorar la experiencia de usuario en etapas tempranas del diseño.

Tabla de Contenidos

- [Características](#)
- [Instalación](#)
- [Detectores Implementados](#)
- [Uso](#)
- [Arquitectura](#)
- [Configuración](#)
- [Contribución](#)
- [Licencia](#)
- [Citas y Referencias](#)

Características

Detección Automática

- **7 detectores de usability smells** implementados con viabilidad alta
- **Análisis en tiempo real** de prototipos estáticos de Figma
- **Retroalimentación inmediata** durante el proceso de diseño

Detectores Implementados

Detector	Descripción	Severidad
S01	Campos con ancho inadecuado para su tipo de contenido	Media-Alta
S02	Inconsistencias dimensionales en formularios	Media
S03	Campos de fecha/teléfono sin formato estructural	Alta
S04	Enlaces y vínculos con texto ambiguo	Media
S05	Valores limitados usando campos de texto libre	Media
S06	Formularios excesivamente complejos	Alta
S07	Flujos de prototipo lineales demasiado extensos	Media

Funcionalidades Avanzadas

- **Sistema de ignorados persistente:** Marca elementos para excluir de futuras detecciones
- **Presets por industria:** Configuraciones optimizadas para e-commerce, banca, etc.
- **Exportación de resultados:** CSV y Markdown para documentación y seguimiento
- **Arquitectura modular:** Fácil extensión con nuevos detectores
- **Configuración personalizable:** Umbrales y parámetros ajustables por proyecto

Instalación

Desde Figma Community

1. Abre Figma y navega a **Plugins > Browse all plugins**
2. Busca "Simple Smells Detector"
3. Haz clic en **Install**

Instalación Manual para Desarrollo

1. Clona este repositorio:

```
git clone https://github.com/[usuario]/simple-smells-detector.git
cd simple-smells-detector
```

2. En Figma Desktop:

- Ve a **Plugins > Development > Import plugin from manifest...**
- Selecciona el archivo `manifest.json` del proyecto

3. El plugin estará disponible en **Plugins > Development > Simple Smells Detector**

Detectores Implementados

S01: Análisis de Tamaño de Campos

Detecta campos de entrada con anchos inadecuados según el tipo de contenido esperado.

Tipos Soportados:

- Fecha (100-150px)
- Teléfono (120-200px)
- Código Postal (70-120px)
- Email (200-350px)
- Nombres (150-250px)

Ejemplo de Detección:

- ✗ Campo de Email con 80px de ancho
- ✓ Campo de Email con 250px de ancho

S02: Consistencia Dimensional

Identifica inconsistencias en anchos de campos dentro del mismo formulario.

Heurística: Detecta desviaciones significativas en grupos de campos relacionados por proximidad.

S03: Campos Sin Formato Estructural

Encuentra campos que requieren componentes especializados (como selectores de fecha) pero están implementados como texto libre.

Detecta:

- Campos de fecha sin componente de calendario
- Campos de teléfono sin máscara de formato
- Campos que requieren validación específica

S04: Enlaces Confusos

Identifica vínculos con texto genérico o ambiguo.

Frases Detectadas:

- "Click aquí", "Ver más", "Leer más"
- "Descargar", "Ingresar aquí"
- "Click here", "Learn more", "Download"

S05: Valores Limitados

Detecta campos de texto libre que deberían usar selectores para conjuntos de valores limitados.

Campos Detectados:

- País, Ciudad, Provincia
- Género, Categoría
- Tipo de documento

S06: Complejidad de Formularios

Analiza formularios con excesiva cantidad de campos en una sola pantalla.

Umbral por Defecto: 8 campos (configurable por industria)

S07: Flujos Lineales Extensos

Detecta secuencias de prototipo con demasiados pasos lineales.

Umbral por Defecto: 30 pasos (configurable)



Análisis Básico

1. Abre tu prototipo en Figma
2. Ejecuta el plugin: **Plugins > Simple Smells Detector**
3. Haz clic en "**Analizar Diseño Completo**"
4. Revisa los hallazgos en la interfaz del plugin

Análisis Específico

- **Por Detector:** Ejecuta detectores individuales para análisis focalizados
- **Por Scope:** Analiza solo la página actual o todo el archivo
- **Filtros:** Filtra resultados por severidad o tipo de detector

Sistema de Ignorados

1. Selecciona un elemento con un smell detectado
2. Haz clic en "**Ignorar este elemento**"
3. El elemento se excluirá de futuras detecciones

Exportación de Resultados

- **CSV:** Para análisis cuantitativo y tracking
- **Markdown:** Para documentación y reportes



Arquitectura

Estructura del Proyecto

```
├─ manifest.json      # Configuración del plugin
├─ code.js            # Lógica principal y coordinación
├─ ui.html            # Interfaz de usuario
├─ analysis-engine/   # Motor de análisis modular
│   ├── core/         # Componentes centrales
│   │   ├── runner.js  # Coordinador de ejecución
│   │   ├── registry.js # Registro de detectores
│   │   └── normalizer.js # Normalización de datos
│   ├── detectors/     # Implementación de detectores
│   │   ├── sizeDetector.js
│   │   ├── consistencyDetector.js
│   │   ├── formatDetector.js
│   │   ├── linkDetector.js
│   │   ├── valuesDetector.js
│   │   ├── complexityDetector.js
│   │   └── flowDetector.js
│   └── utilities/    # Utilidades compartidas
│       ├── geometry.js # Cálculos geométricos
│       ├── semantics.js # Análisis semántico
│       ├── grouping.js  # Agrupación de elementos
│       └── flows.js     # Análisis de flujos
└─ docs/              # Documentación técnica
```

Arquitectura de Capas

1. **Capa de Presentación:** Interfaz de usuario (ui.html)
2. **Capa de Lógica:** Motor de análisis modular
3. **Capa de Datos:** API de Figma y persistencia local

Patrones Implementados

- **Strategy Pattern:** Detectores intercambiables
- **Chain of Responsibility:** Procesamiento secuencial
- **Observer Pattern:** Comunicación asíncrona UI-Engine

Configuración

Presets por Industria

E-commerce

```
{
  UMBRAL_CAMPOS_FORMULARIO: 6,
  MAX_DISTANCIA_VERTICAL: 50,
  MAX_DESVIACION_HORIZONTAL: 15,
  MAX_PASOS_FLOW: 25
}
```

Banca

```
{
  UMBRAL_CAMPOS_FORMULARIO: 5,
  MAX_DISTANCIA_VERTICAL: 30,
  MAX_DESVIACION_HORIZONTAL: 8,
  MAX_PASOS_FLOW: 20
}
```

Configuración Personalizada

Accede a **Settings** en el plugin para:

- Ajustar umbrales de detección
- Crear tipos de datos personalizados
- Configurar frases prohibidas específicas
- Exportar/importar configuraciones

Contribución

¡Las contribuciones son bienvenidas! Por favor lee [CONTRIBUTING.md](#) para conocer:

- Estándares de código
- Proceso de pull requests
- Guías para agregar nuevos detectores
- Estructura de testing

Agregando Nuevos Detectores

1. Crea un archivo en `analysis-engine/detectors/`
2. Implementa la interfaz estándar del detector
3. Registra el detector en `registry.js`
4. Añade tests y documentación

Licencia

Este proyecto está licenciado bajo la Licencia MIT - ver el archivo [LICENSE](#) para detalles.

Citas y Referencias

Este plugin está basado en la investigación académica sobre **usability smells** y detección temprana de problemas de usabilidad:

Referencias Principales

- Grigera, J., et al. (2017). "Automatic Detection of Usability Smells in Web Applications"
- Garrido, A., Rossi, G., & Distante, D. (2010). "Refactoring for Usability in Web Applications"
- Nielsen, J., & Molich, R. (1990). "Heuristic evaluation of user interfaces"

Estándares Aplicados

- ISO 9241-11:2018 - Ergonomics of human-system interaction
- ISO/IEC 25010 - Systems and software Quality Requirements and Evaluation (SQuaRE)

Contexto Académico

Este plugin forma parte de una tesina de grado que aborda la **detección temprana y automatizada de usability smells en prototipos de interfaces digitales**. La investigación se enfoca en:

- Taxonomía de usability smells automatizables
- Arquitectura modular para detectores extensibles
- Integración en flujos de trabajo de diseño ágil
- Reducción de deuda de UX mediante detección temprana

Soporte

- **Issues:** [GitHub Issues](#)
- **Discusiones:** [GitHub Discussions](#)
- **Email:** [tu-email@ejemplo.com]

☆ Si este plugin te resulta útil, considera darle una estrella en GitHub para apoyar el proyecto.