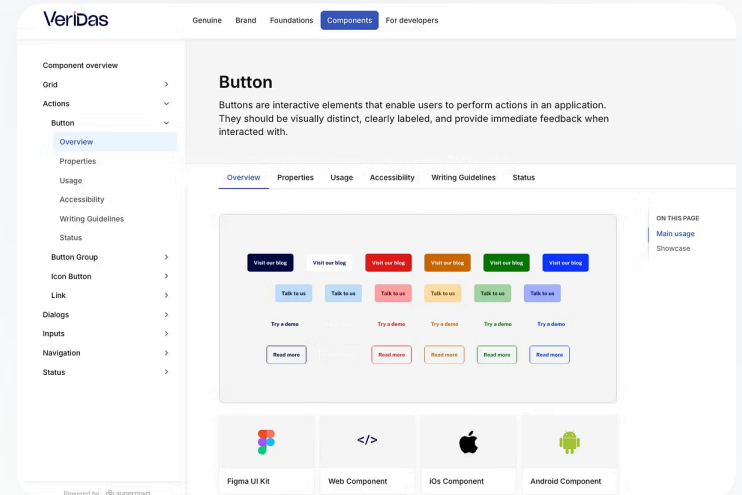


Cómo montar un sistema de diseño desde cero

De la necesidad a la estructura: así nació Genuine DS, el sistema de diseño de Veridas.

Un recorrido realista por decisiones, retos y aprendizajes clave.



¿Qué hacemos?



Identidad Digital

Veridas ayuda a empresas y administraciones a verificar la identidad de las personas mediante biometría e inteligencia artificial. Su tecnología previene el fraude de identidad y permite acceder a servicios digitales de forma segura, rápida y sencilla.



Talento local e innovación

Tecnología 100% propia made in Pamplona, con más de 45 patentes internacionales.



Cambio de paradigma



Antes: lanzar rápido para llegar al mercado

- Enfoque en lanzar producto rápido.
- Soluciones ad hoc, foco en lo funcional.
- Diseño reactivo a necesidades inmediatas.

Ahora: habilitar negocio de forma escalable y sostenible

- Enfoque en procesos, escalabilidad y reutilización.
- Diseño sistémico, foco en consistencia.
- Diseño proactivo desde la investigación con clientes y usuarios.

¿Por qué decidimos crear un DS?

7+

Productos

Con interfaces gráficas únicas.

40+

Variaciones

De un mismo componente visual

1

Rebranding

Punto de inflexión para nuestra coherencia visual

La fragmentación visual entre nuestros productos reveló la necesidad urgente de unificar criterios. El rebranding evidenció que actualizar interfaces inconsistentes era extremadamente complejo.

Primeros pasos

1

Auditoría de Interfaces

Queríamos responder a 4 preguntas que nos permitieran trazar una estrategia:

1. ¿Qué tenemos?
2. ¿Qué se repite?
3. ¿Qué es inconsistente?
4. ¿Qué merece la pena sistematizar?

2

Decisión de Tecnologías

¿En qué tecnologías deberían estar nuestros componentes?

3

Creación de Foundations

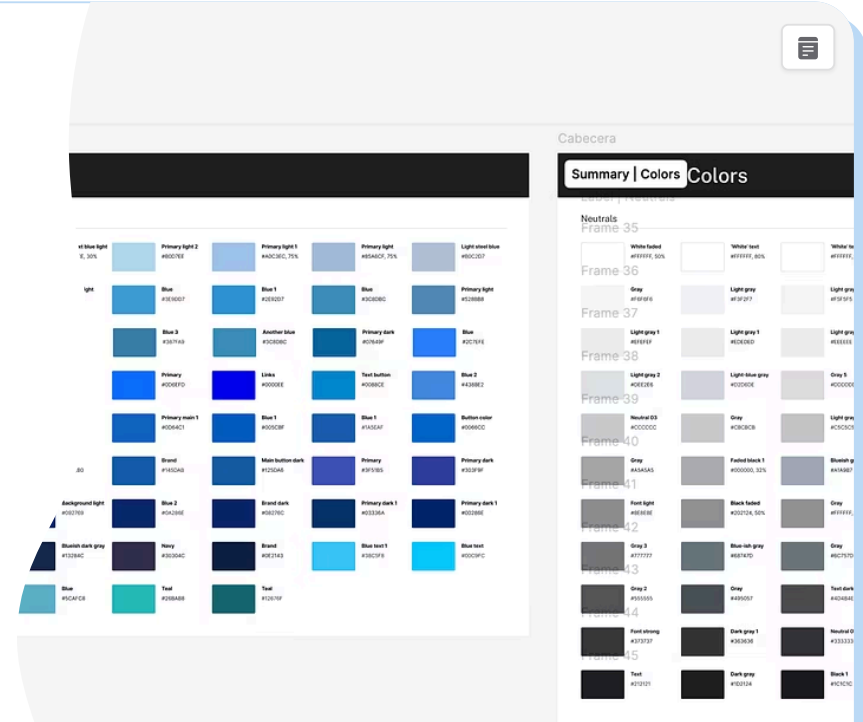
Creación de las bases que conforman el Design System. Aprovechamos el rebranding e hicimos 2×1.

1. Resultados de la auditoría

Nuestra auditoría reveló:

- Inconsistencias significativas en componentes.
- Fragmentación de color.
- Caos tipográfico.
- Experiencias de usuario muy diferentes entre nuestros productos.

Resultados de la auditoría



2. Decisión de tecnologías



Web Components

Creamos componentes base con wrappers específicos para React, asegurando máxima compatibilidad.



React Native

Implementación específica para nuestro producto wallet (NEXUS), manteniendo coherencia visual.



iOS y Android

Desarrollo nativo para componentes de verificación de identidad (IDV).

Nuestra estrategia tecnológica permite mantener una experiencia consistente independientemente de la plataforma.



3. Definición de Foundations

Las Foundations son los elementos fundamentales que constituyen la base de un sistema de diseño. Representan las decisiones de diseño más básicas y se utilizan como bloques de construcción para crear componentes más complejos y experiencias coherentes.



Colores

Paletas primarias, secundarias y neutras de la marca.



Tipografía

Fuentes, tamaños, pesos y estilos que garantizan legibilidad y jerarquía visual.



Espaciado

Sistema de espaciado consistente para márgenes, padding y distribución de elementos.



Iconografía

Biblioteca de iconos.

[Ver Foundations](#)

Construcción del Sistema

1

Creación de tokens

Definición de los tokens de diseño.

2

Diseño de componentes

Seguimos los principios de Atomic Design para crear elementos desde átomos hasta organismos complejos.

3

Grid Dinámico

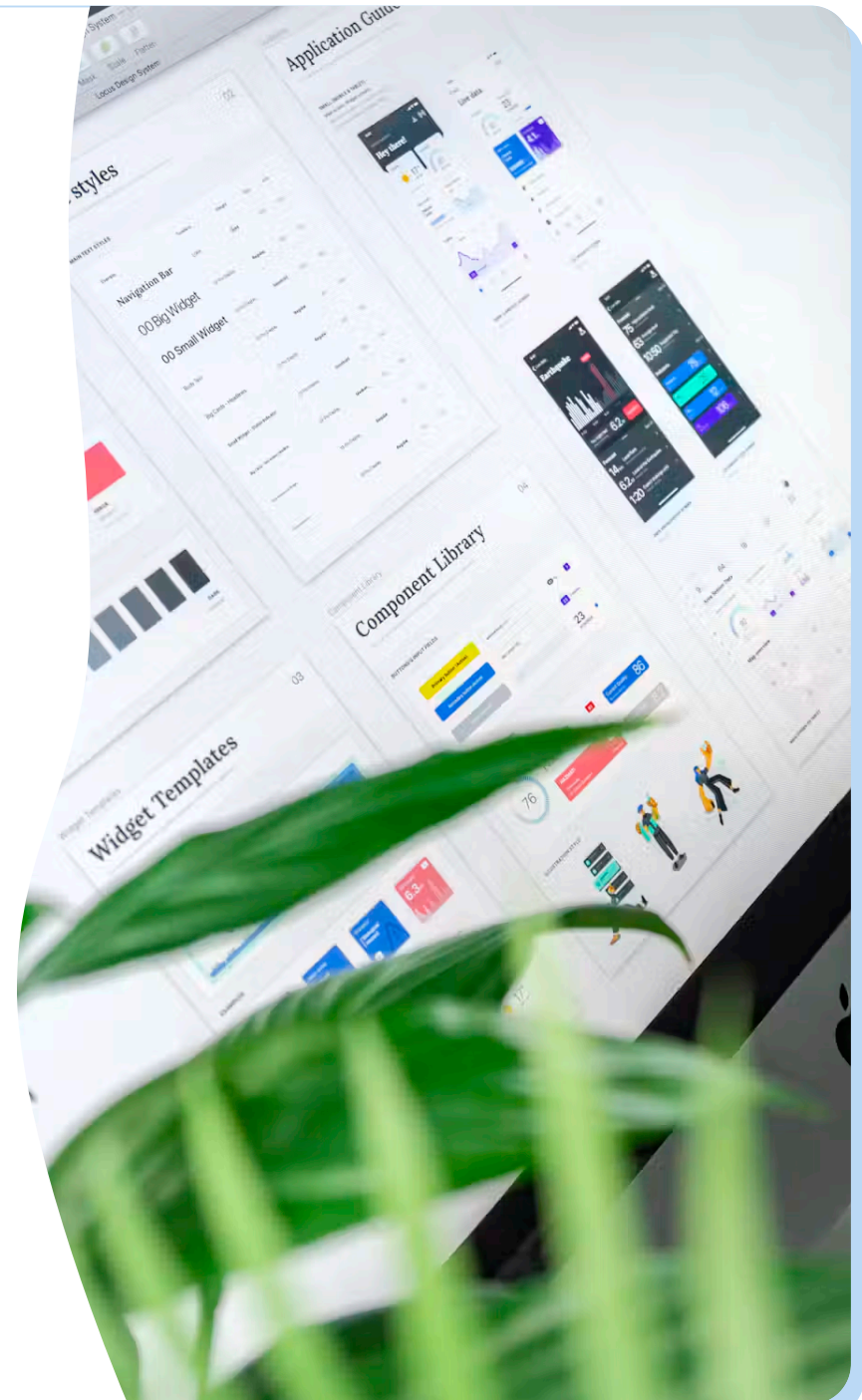
Determina comportamientos responsive y espaciados consistentes entre dispositivos.

4

Documentación

Creamos guías completas en Supernova con indicaciones de uso y accesibilidad.

Priorizamos la construcción de componentes relevantes para nuestras necesidades, no simplemente replicando lo que tienen otros sistemas.





Definición de Tokens

Los tokens de diseño son la capa de abstracción que traduce nuestras foundations en valores utilizables por los diseñadores y los desarrolladores.



Semántica

Nombres significativos que reflejan su propósito.



Consistencia

Valores unificados en todas las plataformas.



Implementación

Variables para CSS, JavaScript, iOS y Android.



Lenguaje común

Crea un lenguaje común entre diseñadores y desarrolladores.

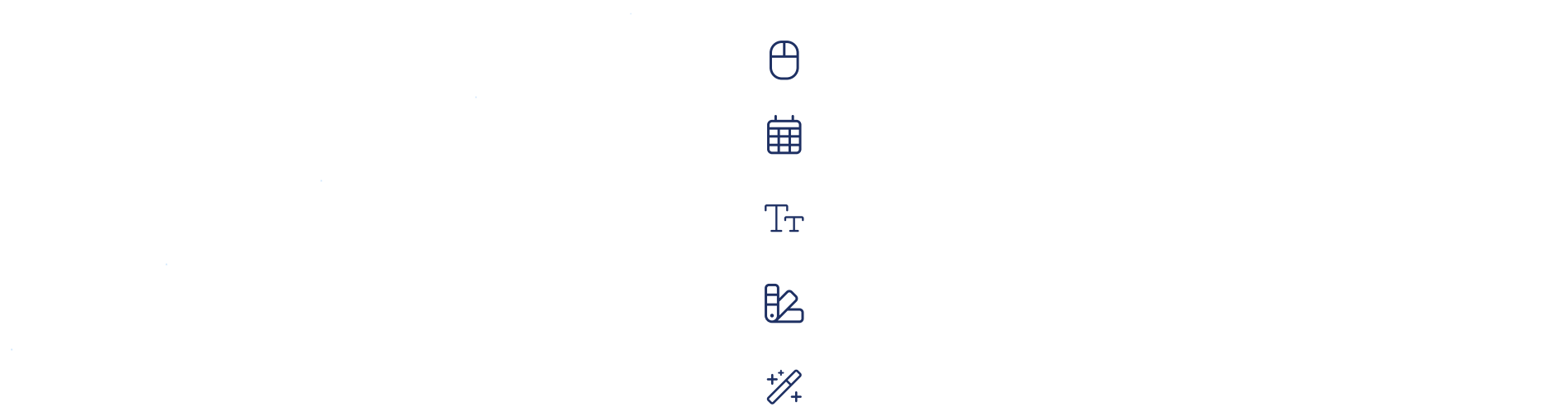
5

Mantenimiento simplificado

Permite realizar cambios globales modificando solo los tokens

1. Definición de Tokens

Las foundations se implementan como tokens de diseño que permiten su aplicación coherente y mantenible a través de todos los productos.



 **States**
Estados específicos de interacción

 **Componente**
Específicos para cada elemento UI

 **Semánticos**
Referentes al contexto de uso

 **Custom**
Personalizaciones para clientes

 **Core**
Valores clave fundamentales.

Implementamos cinco niveles de tokens en lugar de los tres tradicionales. Esto respondió perfectamente a nuestras necesidades de personalización para clientes.

Jerarquía de Tokens

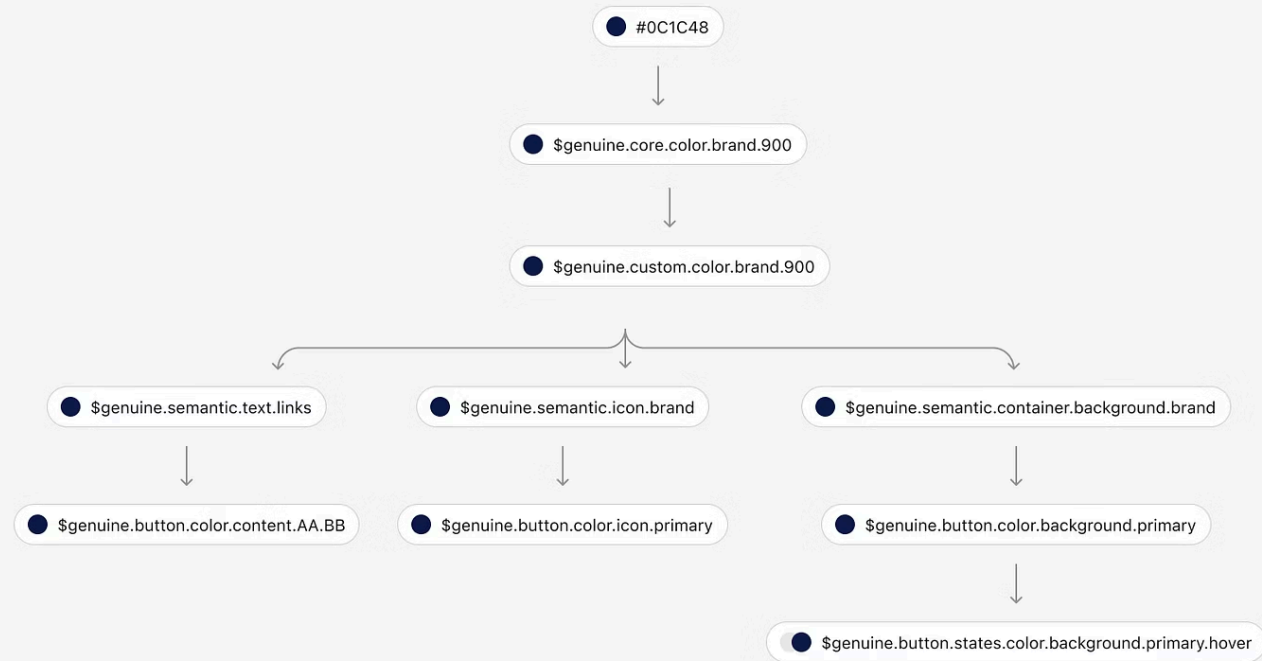
1 Core Tokens

2 Custom Tokens

3 Semantic Tokens

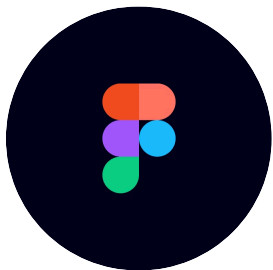
4 Component Tokens

5 States Tokens





Proceso de Creación de Tokens



1. Figma: Definición de estilos y tokens

Establecemos decisiones visuales en Figma usando nomenclatura semántica.

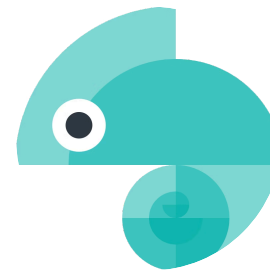
Organizamos colores, tipografías y estilos por función, no por apariencia.



2. Tokens Studio: estructura de variables

Transformamos decisiones visuales en tokens estructurados mediante este plugin.

Organizamos por categorías, gestionamos versiones y configuramos modos dark/light.



3. Style Dictionary: implementación.

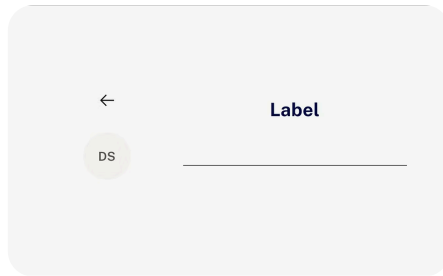
Convertimos tokens JSON en código para múltiples plataformas.

Generamos variables SCSS, CSS, Swift y XML con nomenclatura consistente.

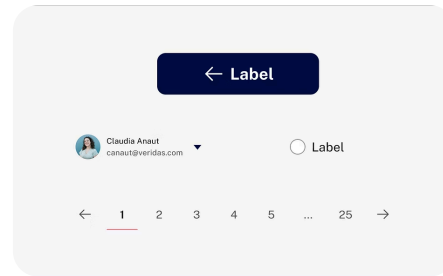


2. Diseño de componentes

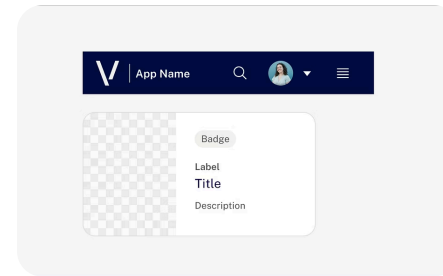
Utilizamos la metodología Atomic Design (Brad Frost) para estructurar nuestro sistema de componentes de manera escalable.



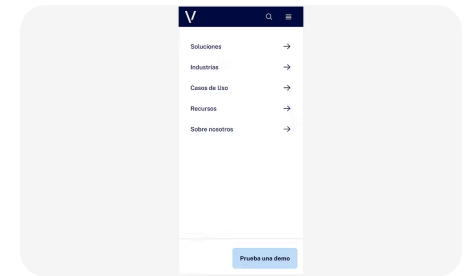
Átomos



Moléculas



Organismos

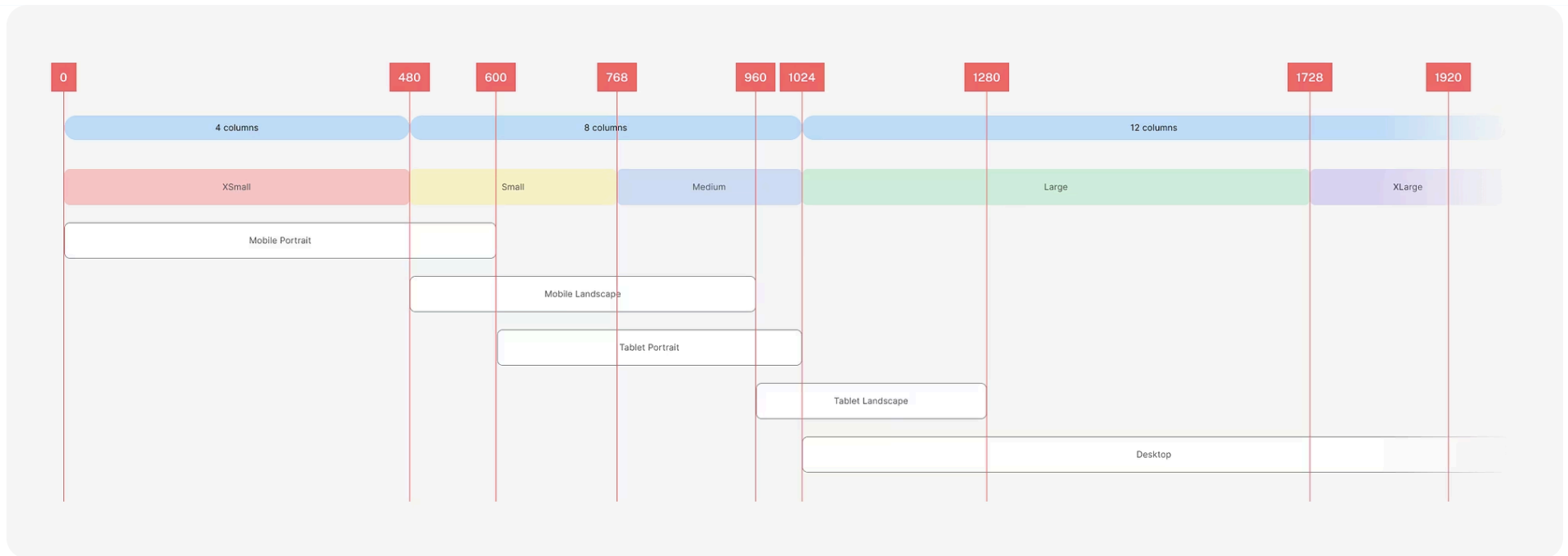


Plantillas

[Ver componentes](#)

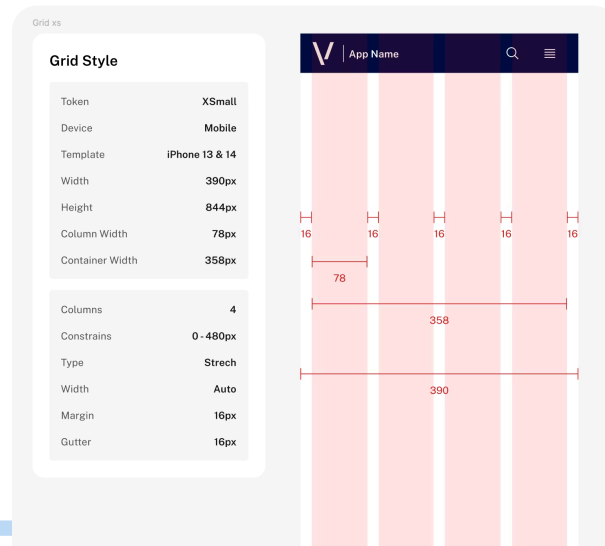
3. Creación de grid dinámico

1. Definimos los breakpoints



Creación de grid dinámico

2. Definimos margins, gutter, column widths dentro del breakpoint.



4. Documentación del sistema

Todo lo referente al sistema de diseño, se va documentando en una web llamada Supernova, que se integra con:

- Figma
- Tokens Studio

A nivel de implementación, usamos [Storybook](#).



veridas-genuine.supernova-docs.io



Overview | Made with Supernova

Foundations are the building blocks we have created for our design system. They include layout, color, typography...

Desafíos y soluciones

Accesibilidad desde el inicio

- Revisión de requisitos WCAG.
- Implementación de contraste adecuado.
- Etiquetado correcto con ARIA.
- Documentación específica.

Personalización para clientes

- Sistema de tokens específico.
- Componentes configurables.
- Solución adaptada a nuestro modelo.

Transición y mantenimiento

- Modelo de contribución estructurado.
- Gestión como producto, no como proyecto.
- Proceso continuo de mejora.

Accesibilidad: Diseño para Todos



Revisión de Estándares

Incorporamos requisitos WCAG desde las primeras fases del diseño. Las guidelines nos marcan el camino para crear interfaces inclusivas.



Diseño Universal

Aplicamos estrictos criterios de contraste, dimensiones adecuadas de texto y áreas de interacción amplias en cada componente.



Implementación Técnica

Cada componente incluye etiquetado semántico correcto, atributos ARIA y compatibilidad con lectores de pantalla y tecnologías de asistencia.



Testing

Integramos tests de accesibilidad automáticos, además de los que ya incorpora Storybook.

Así trabajamos la accesibilidad

Keyboard Accessibility

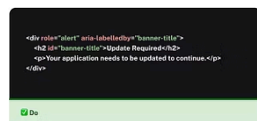
- Guideline: Banners must be navigable and dismissible using keyboard-only input. Interactive elements (e.g., links, buttons, close icons) must follow the standard keyboard focus order.
- Rationale: Keyboard navigation is essential for users with mobility impairments or who rely on assistive devices like switch controls.
- WCAG Standard 2.1.1-Keyboard (Level A)

Identify Regions for Assistive Technologies

- Guideline: Use ARIA roles (role="alert", role="status") to identify the Banner and communicate its purpose to assistive technologies.
- Rationale: Properly identifying the Banner's role helps screen readers announce its presence and prioritize its content.
- WCAG Standard 4.1.2-Name, Role, Value (Level A)

Identify Regions for Assistive Technologies

- Guideline: Use ARIA roles (role="alert", role="status") to identify the Banner and communicate its purpose to assistive technologies.
- Rationale: Properly identifying the Banner's role helps screen readers announce its presence and prioritize its content.
- WCAG Standard 4.1.2-Name, Role, Value (Level A)



Provide Clear Dismissal Controls

Revisión de estándares

```
test("should have a visible focus indicator", async () => {
  const el = await fixture(html`<button>Update Required</button>`);
  const buttonElement = el.shadowRoot?.querySelector("button");

  if (buttonElement instanceof HTMLElement) {
    buttonElement.focus();
    const focusStyle = getComputedStyle(buttonElement);
    expect(focusStyle.outlineWidth).not.toBe("0px");
  } else {
    throw new Error("Button element not found or not focusable");
  }
});
```

Implementación en código

Controls	10	Actions	Interactions	Design	Accessibility
0 Violations	15 Passes	0 Incomplete			
> Elements must only use supported ARIA attributes > ARIA role should be appropriate for the element > ARIA attributes must be used as specified for the element's role > Deprecated ARIA roles must not be used > ARIA hidden element must not be focusable or contain focusable elements					
VisualizaciónReproducciónVentanaAyuda					

Testing en Storybook



Design System: puerta de entrada a la accesibilidad

Un Design System no resuelve la accesibilidad por sí solo, pero la habilita y visibiliza, forzándonos a trabajarla sistemáticamente.



Punto de partida, no destino final

- **Diseñar componentes accesibles** es solo el comienzo.
- Un botón accesible no garantiza una **pantalla accesible**.
- La accesibilidad real depende también del **ensamblado correcto**: jerarquías, roles ARIA, foco, contraste, orden lógico...

“A design system isn’t a project. It’s a product serving products” – Nathan Curtis.

El verdadero desafío: mantener vivo el sistema

1 Construir es solo el inicio

La creación inicial representa apenas el 20% del esfuerzo total.

"Un sistema de diseño es como un jardín, no como un edificio. No se termina, se cultiva constantemente."

2 Crecimiento constante

Un sistema abandonado se vuelve irrelevante en menos de un año.

3 Alimentación continua

Requiere recursos dedicados y un compromiso organizacional permanente.



Modelo de Contribución

Se necesita un modelo de gobernanza y guía colaborativo, que sirva para decidir cuándo y cómo se realizan incorporaciones en el sistema de diseño.

Niveles de contribución

Ver más ▾



Pasos para contribuir

Ver más ▾



Niveles de contribución



Contribución Pequeña

- Cambio de diseño en un componente existente.
- Cambio de token.
- Mejoras de accesibilidad
- Solucionar problemas de diseño, tokens o accesibilidad.



Contribución Media

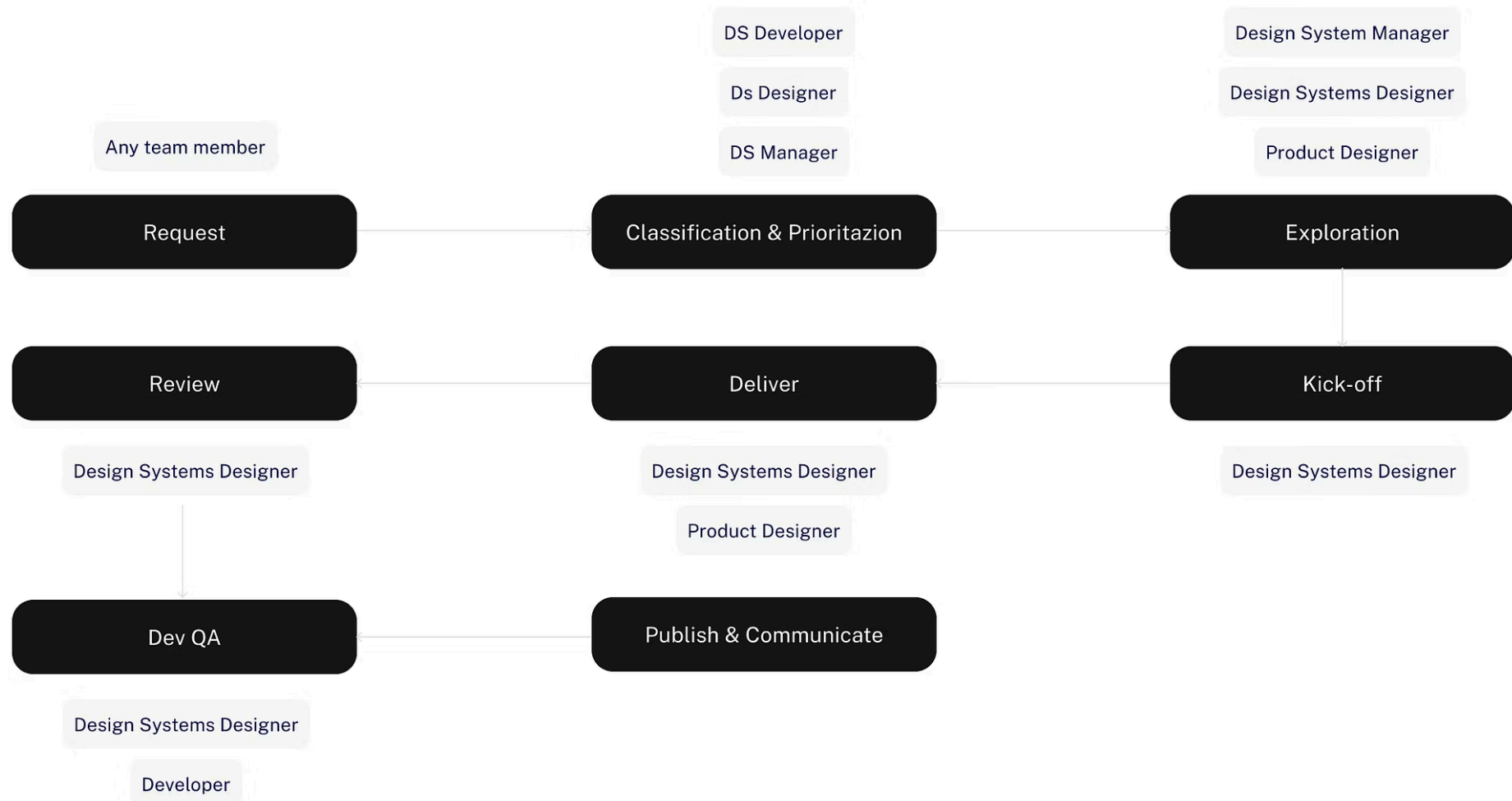
- Actualización de un UX Pattern.
- Nuevas variantes o mejoras estructurales.



Contribución Grande

- Creación de un nuevo componente o patrón

Pasos para contribuir





Minimizar el tiempo de reintegración = clave



Flujos optimizados

Los cambios pequeños necesitan procesos autónomos y rápidos.

Cada día de retraso aumenta el riesgo de "snowflakes" (soluciones paralelas).



Motivación del equipo

Los colaboradores se sienten valorados cuando ven sus aportes integrados rápidamente.

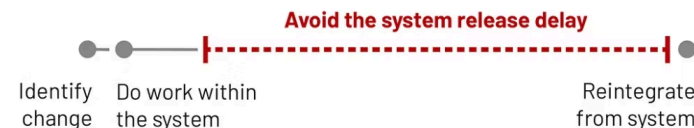
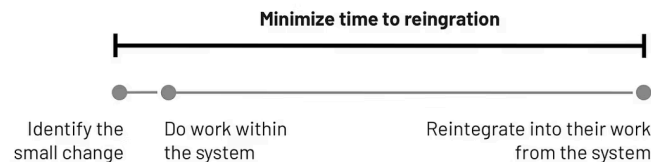
La agilidad fomenta una cultura de mejora continua.



Consistencia garantizada

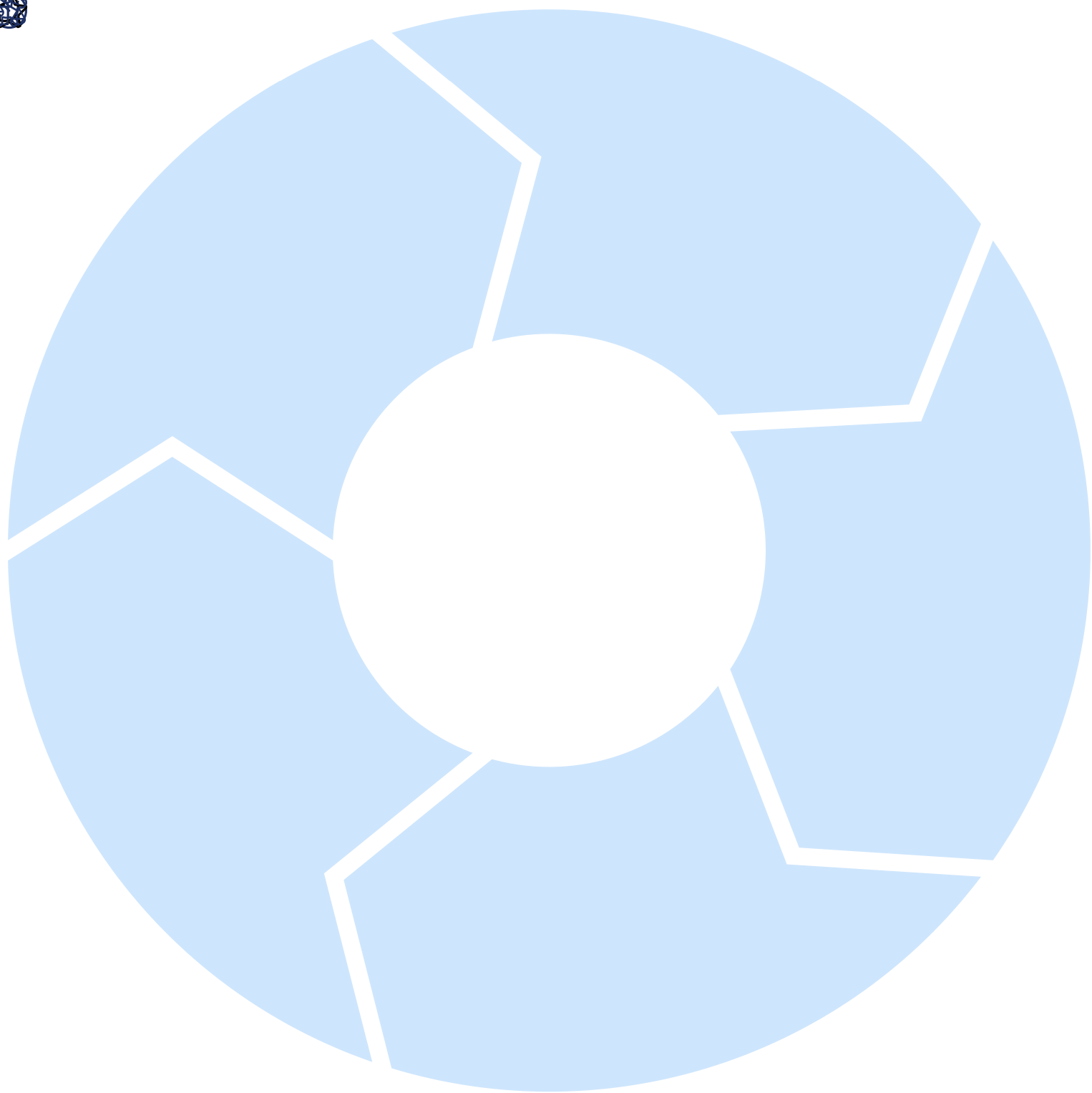
La integración rápida previene inconsistencias visuales y técnicas.

El sistema permanece como fuente única de verdad.



Images' source: Brad Frost article "[Defining Design System contributions](#)".

Beneficios Obtenidos



Consistencia visual

Experiencia unificada en todos los productos



Ahorro de tiempo

Desarrollo acelerado de nuevas interfaces



Mejor colaboración

Lenguaje común entre diseño y desarrollo



Accesibilidad mejorada

Prácticas incorporadas en todos los componentes



Escalabilidad

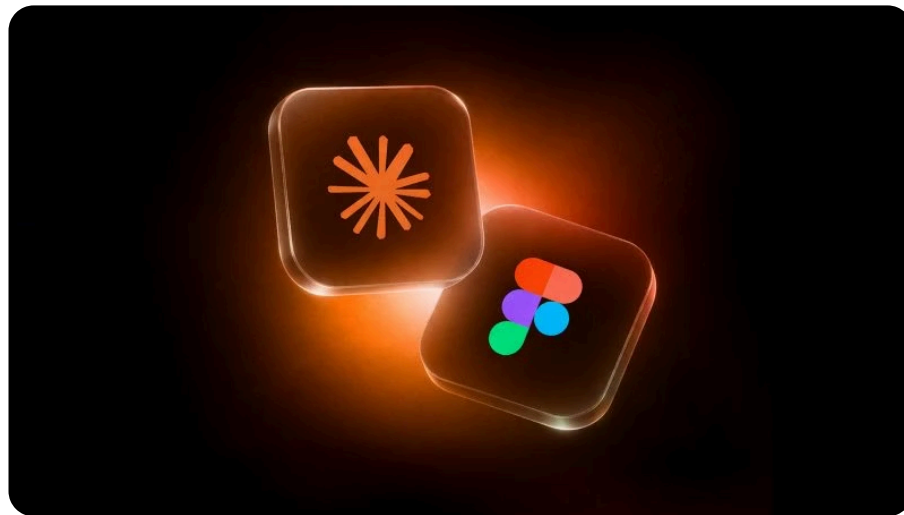
Coherencia mantenida al crecer los equipos

La implementación ha reducido significativamente nuestra deuda técnica. Los cambios ahora son más rápidos y seguros.

¿Y si tuviésemos que crear Genuine DS hoy?

Qué haríamos diferente

- Auditoría → Claude + GitLab + Figma.
- Foundations → validación.
- Tokens → revisión y gobernanza.
- Componentes → documentación y testing.
- Contribution Model → IA como primer reviewer.
- Claude + Genuine → prototipado basado en el sistema.



01 Productos auditados

La auditoría cubre superficies muy distintas, cada una con su propia sección de Color, Tipografía, Botones e Inputs. El problema de fondo: **cada producto trae su propio look heredado de una plantilla diferente.**

Native Apps (iOS / Android)

App nativa de captura biométrica (incl. vdVoice)

Tipografía: **Poppins + Arial**

Base visual: **Estilo propio mobile; mezcla de fuentes y radios en botones**



dasGate — Visitas / Onboarding Express / Stadiums / Gambling

Plataforma web con 4 verticales

Tipografía: **Montserrat + Open Sans**

Base visual: **Bootstrap / Vuxxy admin**



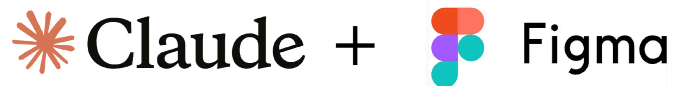
Si realizáramos esta auditoría hoy: IA al rescate

Antes

- Inventario manual de interfaces
- Clasificación manual de patrones
- Identificación manual de inconsistencias

Hoy (con IA)

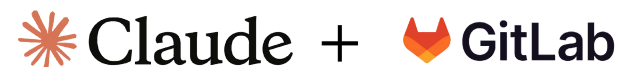
- Análisis de repositorios automatizado (Claude + GitLab y Claude + MCP Figma)
- Agrupación automática de patrones visuales
- Detección de inconsistencias entre plataformas
- Priorización de componentes basada en impacto



Diseño: Figma MCP + Claude

- Agrupar automáticamente cientos de pantallas por patrones de interfaz.
- Detectar componentes candidatos aunque todavía no existan como componentes.

- Identificar variantes redundantes y oportunidades de consolidación.



Desarrollo: Claude Code + GitLab

- Inventariar componentes reales utilizados en producción.
- Detectar duplicidades, forks y deuda técnica.
- Identificar estilos hardcodeados fuera del sistema.



Cómo utilizaría la IA podría para la gestión de tokens



Detectar redundancias



Revisar nomenclatura



Encontrar tokens huérfanos



Validar accesibilidad



Detectar hardcodes en los productos



Generar documentación automáticamente



Claude conoce Genuine



Le damos el contexto

Para que Claude pueda diseñar utilizando Genuine, primero tiene que aprender Genuine.



Nos aporta ideas de diseño

Genera propuestas utilizando únicamente componentes y patrones del sistema.



Actúa como reviewer

Detecta inconsistencias, problemas de accesibilidad, duplicados o desviaciones respecto al sistema.



Nos ayuda a documentar

Genera especificaciones, casos de uso, criterios de accesibilidad y escenarios de prueba.



El Sistema de Diseño es un motor de transformación organizacional



Mentalidad

Evolución del pensamiento: de soluciones aisladas a sistemas completos.

Pasamos de diseñar pantallas a definir patrones reutilizables y escalables.



Procesos

Formalización del ciclo diseño-desarrollo-documentación.

Decisiones basadas en principios establecidos, no en opiniones personales.



Cultura

Colaboración genuina entre equipos multidisciplinares.

Democratización que permite contribuciones de calidad desde cualquier rol.



Negocio

Aceleración del time-to-market manteniendo estándares de calidad.

Reducción significativa de la deuda técnica a largo plazo.

Cada sistema de diseño debe adaptarse a las necesidades específicas de la empresa. No existe una fórmula universal.