



**Aplicación del Atomic Design para mejorar la eficiencia de proyectos Frontend: un enfoque metodológico para optimizar el desarrollo de interfaces de usuario.**

Juan Manuel Grajales Urquijo

Ingeniero de Sistemas

Asesor

Jeysson Pérez Gómez, Especialista en Gerencia Integral

Universidad de Antioquia

Facultad de Ingeniería

Pregrado

Medellín

2023

<b>Cita</b>	(Grajales Urquijo, 2023)
<b>Referencia</b>	Grajales Urquijo J. M. (2023). <i>Aplicación del Atomic Design para mejorar la eficiencia de proyectos Frontend: un enfoque metodológico para optimizar el desarrollo de interfaces de usuario</i> , Ingeniería de Sistemas, Universidad de Antioquia, Medellín, 2023.
<b>Estilo APA 7 (2020)</b>	



Centro de Documentación Ingeniería (CENDOI)

**Repositorio Institucional:** <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - [www.udea.edu.co](http://www.udea.edu.co)

**Rector:** Jhon Jairo Arboleda Céspedes

**Decano de la facultad de ingeniería:** Julio César Saldarriaga Molina

**Jefe departamento:** Diego José Luis Botia

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicado a mis familiares y seres queridos que incondicionalmente me apoyaron en mis decisiones dentro de mi formación profesional.

Agradecimientos a  
Jeisson Pérez Gómez  
Daniel Saldarriaga  
PrevalentWare

## Tabla de contenido

Resumen .....	8
Abstract .....	9
Introducción .....	10
1 Objetivos .....	13
1.1 Objetivo general	13
1.2 Objetivos específicos	13
2 Marco teórico .....	14
3 Metodología .....	19
3.1 Análisis teórico y revisión de literatura	19
3.2 Evaluación de la eficiencia y calidad del Atomic Design	19
4. Resultados .....	21
4.1 Proceso Investigativo	21
4.2 Características Clave del Atomic Design	21
4.2.1 Reutilización de Componentes	21
4.2.2 Consistencia Visual	21
4.2.3 Escalabilidad	22
4.3.4 Facilita la Colaboración	22
4.3.5 Descomposición Jerárquica	22
4.3.6 Reutilización de componentes	22
4.3.7 Consistencia Visual	23
4.3.8 Facilita la Mantenibilidad	23
4.3.9 Documentación Clara	23
4.3.10 Flexibilidad en la Implementación	23
4.3.11 Enfoque Iterativo	23

4.4 Estructura de la Biblioteca de Componentes	24
4.4.1 Jerarquía de Componentes	24
4.4.2 Categorización Lógica	24
4.4.3 Nomenclatura Consistente	24
4.4.4 Vista Previa Visual	24
4.5 Casos de éxito	25
4.5.1 IBM	25
4.5.2 Salesforce	26
4.5.3 Shopify	26
4.5.4 Buffer	27
4.5.5 GitHub	27
4.6 Proyecto piloto	28
4.6.1 Átomos	30
4.6.2 Moléculas	30
4.6.3 Organismos	31
4.6.4 Pagina	34
4.7 Desarrollo	35
4.8 Evaluación de la eficiencia	37
4.8.1 Duplicación del código	37
4.8.2 Acoplamiento y cohesión	38
4.8.3 Matriz de dependencias	40
5 Análisis.....	41
6 Conclusiones .....	43
Referencias.....	44

## Tabla de figuras

<b>Figura 1</b> Ejemplo de ecuación química, mostrando a los átomos hidrógeno y al oxígeno combinándose en una molécula de agua. ....	15
<b>Figura 2</b> Atomic Design aplicado a la aplicación nativa de Instagram.....	16
<b>Figura 3</b> Diseño planteado para el proyecto piloto .....	29
<b>Figura 4</b> División de algunos átomos obtenidos del diseño planteado para el proyecto piloto ....	30
<b>Figura 5</b> División de algunas moléculas obtenidos del diseño planteado para el proyecto piloto .....	31
<b>Figura 6</b> División de algunos organismos obtenidos del diseño planteado para el proyecto piloto .....	32
<b>Figura 7</b> División de la página obtenida del diseño planteado para el proyecto piloto .....	34
<b>Figura 8</b> Arquitectura interna del proyecto .....	35
<b>Figura 9</b> Gráfica de SonarQube que resalta la cantidad total de líneas de código comparado con la cantidad de líneas de código duplicadas.....	38
<b>Figura 10</b> Gráfica de barras que muestra la comparación entre la cantidad promedio de dependencias de cada grupo de componentes, donde el eje X son los grupos de componentes y el eje Y son la cantidad promedio de dependencias.....	39
<b>Figura 11</b> Matriz de dependencias, tanto el eje Y como el eje X, alberga el nombre de los componentes del proyecto, mientras que las celdas coloreadas indican las dependencias entre ellos. ....	40

## **Siglas, acrónimos y abreviaturas**

<b>CRM</b>	Customer Relationship Management (Gestión de la Relación con el Cliente)
<b>HTML</b>	HyperText Markup Language
<b>UI</b>	User Interface (Interfaz de usuario)

## Resumen

En el contexto actual del desarrollo web, la optimización de los procesos de diseño y desarrollo de interfaces de usuario es esencial para mejorar la experiencia del usuario y la productividad de los desarrolladores. La calidad de la interfaz de usuario impacta directamente en la satisfacción del usuario, la fidelidad y las conversiones.

La metodología Atomic Design, propuesta por Brad Frost, se presenta como una solución eficiente. Al utilizar componentes modulares y reutilizables, se ahorra tiempo y esfuerzo, permitiendo a los desarrolladores centrarse en tareas más complejas. PrevalentWare busca adoptar este enfoque para optimizar sus proyectos Frontend, reducir la duplicación de esfuerzos y facilitar la incorporación de nuevos desarrolladores.

El proyecto se estructura en cuatro fases: investigación, análisis de experiencias exitosas, desarrollo del proyecto piloto y análisis cuantitativo. Estas fases buscan comprender, fortalecer y validar la implementación de Atomic Design. La estrategia integral no solo busca mejorar los tiempos de desarrollo, sino también potenciar la documentación y facilitar la rápida incorporación de nuevos desarrolladores, contribuyendo así a la optimización de proyectos Frontend en PrevalentWare.

*Palabras clave:* Atomic Design, desarrollo Frontend, componentes, eficiencia, interfaces de usuario, biblioteca de componentes, metodología, optimización, productividad.

### **Abstract**

In today's web development context, optimizing user interface design and development processes is essential to improve user experience and developer productivity. The quality of the user interface directly impacts user satisfaction, loyalty, and conversions.

The Atomic Design methodology, proposed by Brad Frost, is presented as an efficient solution. By using modular and reusable components, it saves time and effort, allowing developers to focus on more complex tasks. PrevalentWare seeks to adopt this approach to optimize its Frontend projects, reduce duplication of effort, and facilitate the onboarding of new developers.

The project is structured in four phases: research, analysis of successful experiences, pilot project development and quantitative analysis. These phases seek to understand, strengthen, and validate the implementation of Atomic Design. The integral strategy not only seeks to improve development times, but also to enhance documentation and facilitate the rapid incorporation of new developers, thus contributing to the optimization of Frontend projects in PrevalentWare.

*Keywords:* Atomic Design, Frontend development, components, efficiency, user interfaces, component library, methodology, optimization, productivity.

## Introducción

En el actual panorama del desarrollo web, la optimización de los procesos de diseño y desarrollo de interfaces de usuario se ha convertido en un aspecto crítico para las empresas porque permite mejorar la experiencia de usuario y la productividad de los desarrolladores.

La experiencia de usuario es uno de los factores más importantes para el éxito de un sitio web o aplicación. Una interfaz bien diseñada y optimizada es fácil de usar, intuitiva y eficiente, lo que facilita a los usuarios alcanzar sus objetivos. Esto se traduce en una mayor satisfacción del usuario, que puede generar un aumento de la fidelidad, el tráfico y las conversiones.

La productividad de los desarrolladores también se ve beneficiada por la optimización de los procesos de diseño y desarrollo de interfaces de usuario. Al utilizar componentes modulares y reutilizables, los desarrolladores pueden ahorrar tiempo y esfuerzo en la creación de nuevas interfaces. Esto les permite centrarse en tareas más complejas y creativas, lo que mejora la calidad del producto final.

En este contexto, el Atomic Design, una metodología propuesta por Brad Frost se presenta como una solución que permite construir interfaces de manera eficiente a través de la creación de componentes modulares y reutilizables. Al utilizar bloques de construcción predefinidos en lugar de diseñar y desarrollar cada elemento de forma individual, se logra ahorrar tiempo y esfuerzo considerablemente.

La empresa PrevalentWare se encuentra en un momento de búsqueda de estándares para la realización de sus proyectos, y el desarrollo Frontend ocupa un lugar central en este proceso. En su objetivo de alcanzar la excelencia en el código y mejorar la eficiencia de sus desarrolladores, la empresa ha identificado la oportunidad de explorar y aplicar la metodología del Atomic Design. Al adoptar este enfoque, se busca reducir la duplicación de esfuerzos y permitir que sus desarrolladores trabajen con componentes reutilizables, lo que a su vez fomenta la separación de responsabilidades y la adopción de buenas prácticas. Esta iniciativa busca no

solo mejorar los tiempos de desarrollo, sino también potenciar la documentación y facilitar la incorporación de nuevos desarrolladores en los proyectos de manera ágil.

En este proyecto, nos enfocamos en abordar el problema de falta de eficiencia en los proyectos Frontend, donde la creación de interfaces complejas desde cero suele ser una práctica recurrente. Este enfoque tradicional implica duplicar esfuerzos y limita la productividad del equipo. Nuestro objetivo es aplicar el Atomic Design como una solución para superar estos desafíos y alcanzar una mayor eficiencia en el proceso de desarrollo de interfaces, permitiendo a PrevalentWare optimizar sus proyectos Frontend y mejorar sustancialmente su rendimiento y calidad.

La solución propuesta es la implementación de Atomic Design, dividida en cuatro fases: investigación, análisis de experiencias exitosas, desarrollo del proyecto piloto y análisis cuantitativo.

La investigación inicial brinda una comprensión del marco de trabajo necesario para implementar el Atomic Design en un proyecto piloto. Este proyecto piloto sirve como prueba de eficiencia y adaptabilidad, permitiendo comprender sus limitaciones y beneficios.

El análisis de experiencias exitosas proporciona bases teóricas y ejemplos concretos, obteniendo experiencias positivas de otras empresas que han adoptado este enfoque. Esto fortalece el proyecto piloto, aumenta la confianza en la metodología y ofrece perspectivas valiosas a través de ejemplos reales.

El desarrollo del proyecto piloto capacita al equipo en la lógica de desglose de componentes, brindando experiencia práctica en la separación de responsabilidades. Esto facilita la construcción de interfaces sólidas, escalables y mantenibles, proporcionando conocimientos prácticos tangibles.

El análisis cuantitativo respalda la viabilidad de la implementación, con cifras que demuestran eficiencia, baja duplicación de código y estructura de bajo acoplamiento. Esto refuerza la justificación para la adopción de Atomic Design en futuros proyectos.

Estas fases, en conjunto, no solo buscan mejorar los tiempos de desarrollo y la eficiencia en la creación de interfaces, sino que también pretenden potenciar la documentación y facilitar la rápida incorporación de nuevos desarrolladores. Esta estrategia integral tiene como objetivo contribuir de manera significativa a la optimización de proyectos Frontend en PrevalentWare, mejorando la calidad del producto final y la experiencia tanto para los desarrolladores como para los usuarios finales.

## **1 Objetivos**

### **1.1 Objetivo general**

Investigar, analizar, implementar y evaluar los principios y fundamentos del Atomic Design en proyectos Frontend para el equipo de desarrollo web de la empresa PrevalentWare, con el fin de comprender su eficiencia y beneficios, mediante el estudio de casos de éxito de su implementación y la creación de una biblioteca de componentes para un proyecto piloto.

### **1.2 Objetivos específicos**

- Investigar y comprender los principios y fundamentos del Atomic Design enfocado en la organización de componentes.
- Analizar casos de estudio y experiencias exitosas de implementación del Atomic Design en proyectos Frontend.
- Implementar la biblioteca de componentes en un proyecto piloto de Frontend.
- Evaluar la eficiencia y los beneficios obtenidos mediante la aplicación del Atomic Design.

## 2 Marco teórico

El desarrollo web es un proceso complejo que requiere una variedad de habilidades y conocimientos como el diseño, la programación, la implementación y el mantenimiento. Una de las partes más importantes del desarrollo web es el diseño de la interfaz de usuario (UI).

La interfaz de usuario es la parte de un sitio web o aplicación que los usuarios ven e interactúan. Incluye todos los elementos visuales, como texto, imágenes, botones, menús y barras de desplazamiento. La interfaz de usuario es lo que permite a los usuarios navegar por un sitio web o aplicación y realizar tareas. “La Interfaz de Usuario es la parte del software que las personas pueden ver, oír, tocar, hablar; es decir, donde se pueden entender” (Albornoz, 2014) La meta es lograr sitios funcionales, accesibles y visualmente atractivos para los usuarios.

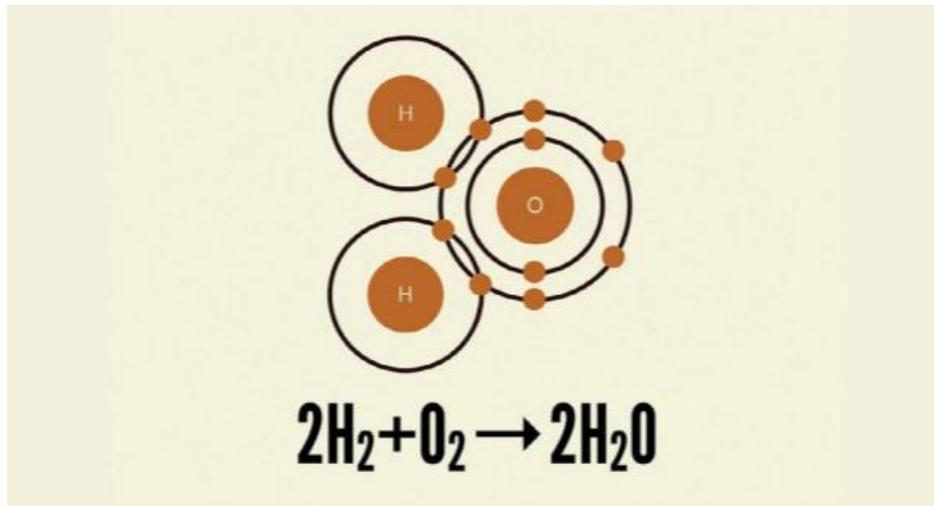
Dentro del desarrollo web, es común encontrar elementos visuales que se repiten en diferentes partes del sitio. El desafío es evitar duplicar esfuerzos y rehacer elementos existentes, por lo que es crucial identificar y separar estos elementos para su posterior reutilización. Aquí es donde entran en juego los componentes web, que no son más que un conjunto de tecnologías que permiten crear nuevos elementos en HTML, encapsulados y reutilizables para cualquier página o aplicación web (Web Components, 2020). Estos componentes encapsulan funcionalidades específicas, incluyendo código y elementos visuales, con estilos, interacciones y comportamientos predefinidos.

Al crear una biblioteca de componentes bien estructurada, se obtiene una colección de recursos listos para ser utilizados en distintas partes de una aplicación, lo que optimiza el proceso de desarrollo y mejora la consistencia de las interfaces. “El diseño atómico facilita una forma escalable y portátil de desarrollar aplicaciones de sistemas. Posteriormente, los componentes serán más fáciles de desarrollar o ampliar si ya cuentan con una base común sólida.” (Nghì, 2017) Para lograr esta organización y reutilización eficiente de componentes, se emplea el Diseño Atómico que "es una metodología compuesta por cinco etapas distintas que trabajan juntas para crear sistemas de diseño de interfaces de forma más deliberada y jerárquica." (Frost, 2013).

El creador de este marco de trabajo, nos indica que la inspiración del Atomic Design comienza como en la química, donde hay elementos que interactúan entre ellos y obtienen reacciones, se presenta a continuación cómo funcionan estas reacciones:

### Figura 1

*Ejemplo de ecuación química, mostrando a los átomos hidrógeno y al oxígeno combinándose en una molécula de agua.*

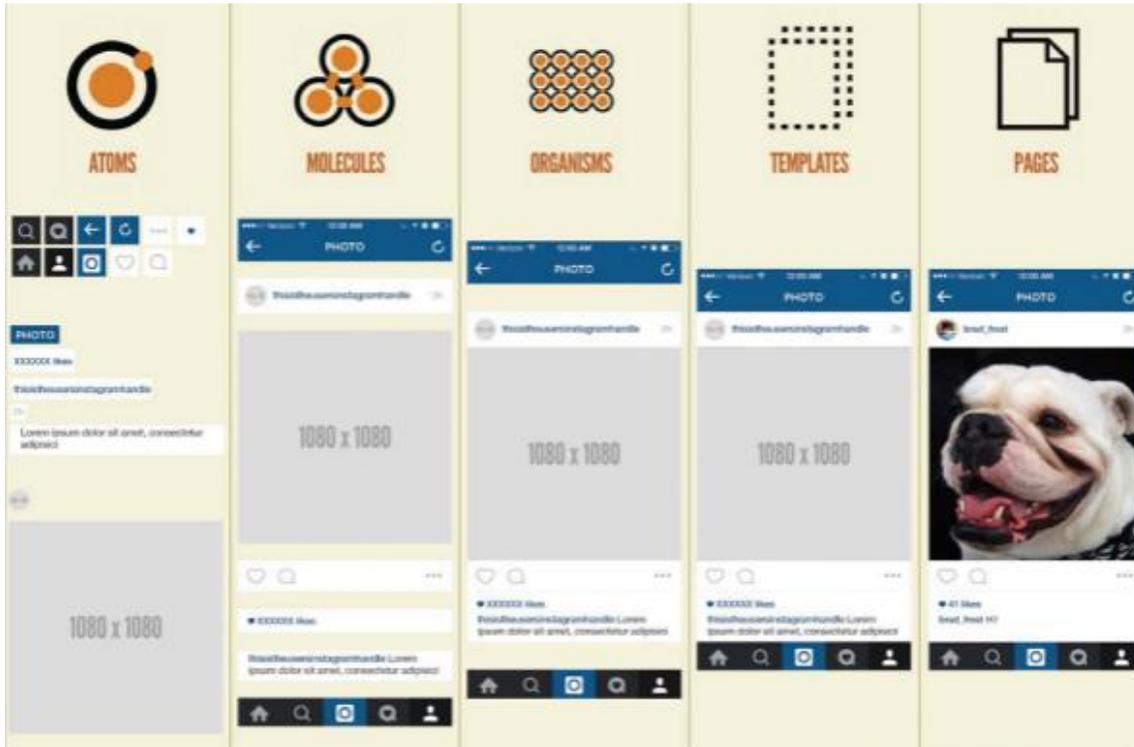


*Nota.* Tomado de Atomic Design (p. 38), por B. Frost, 2013

Podemos ver entonces cómo las moléculas interactúan entre sí y se suman para crear elementos más complejos, una suma de átomos nos permite obtener estructuras más complejas y la suma iterativa de estos elementos complejos nos permite obtener todos los componentes existentes en el universo, estas sustancias están representadas por ecuaciones que nos indican que la combinación de dos elementos, crean uno nuevo más complejo. Con base en esta lógica, podemos obtener lo siguiente:

**Figura 2**

*Atomic Design aplicado a la aplicación nativa de Instagram.*



*Nota.* Tomado de Atomic Design (p. 60), por B. Frost, 2013

Los átomos están formados por otros componentes más pequeños como protones, electrones y neutrones, pero entenderemos al átomo como la unidad funcional más pequeña de nuestro sistema (Frost, 2013) en el contexto del diseño, se consideran la unidad más básica y fundamental, estos pueden incluir elementos HTML como botones, campos de entrada, etiquetas y otros elementos básicos de la interfaz. Los átomos son fundamentales ya que constituyen la base sobre la cual se construyen componentes más complejos.

Las moléculas son un grupo de dos o más átomos unidos por enlaces. Estas unidades adquieren propiedades que le dan integridad y las vuelve tangibles y operativas (Frost, 2013). en el diseño, representan elementos que adquieren significado y funcionalidad autónoma. Por ejemplo, un botón junto con una etiqueta de texto puede formar una molécula que representa un campo de entrada con etiqueta. Estas moléculas son más complejas que los átomos y se utilizan para crear componentes más grandes.

“Los organismos son conjuntos de moléculas que funcionan como una unidad. Estas estructuras relativamente complejas van desde organismos unicelulares hasta organismos increíblemente sofisticados como el ser humano.” (Frost, 2013). Los organismos colaboran como una unidad integral entre átomos, moléculas u otros organismos. Su complejidad depende de la interacción entre los componentes más pequeños. En el diseño, esto se puede comparar con la combinación de elementos diversos para formar un conjunto coherente y funcional.

Por ejemplo, un organismo podría ser un formulario de contacto. Este formulario estaría compuesto por una serie de moléculas, como campos de texto, botones y cuadros de selección. Los átomos que componen estas moléculas podrían ser elementos de texto, iconos y colores.

Los organismos juegan un papel importante en el Atomic Design. Proporcionan una forma de crear interfaces de usuario complejas que son fáciles de entender y usar. Al agrupar los componentes en organismos, los diseñadores pueden reutilizarlos en diferentes contextos. Esto ahorra tiempo y esfuerzo, y ayuda a garantizar la coherencia de la interfaz de usuario.

Los templates son diseños de páginas que proporcionan una estructura básica para la disposición de los organismos. Las plantillas son objetos a nivel de página que colocan componentes en un diseño y articulan la estructura de contenido subyacente del diseño. Definen cómo se organizan los organismos en una página y pueden variar según las necesidades de diseño.

Los templates son útiles para mantener la coherencia visual en una aplicación o sitio web. Al utilizar un template, los diseñadores pueden asegurarse de que todas las páginas de la aplicación o sitio web tengan un aspecto similar. “Otra característica importante de los templates es que se centran en la estructura subyacente del contenido de la página y no en su contenido final. Los sistemas de diseño deben tener en cuenta la naturaleza dinámica del contenido, por lo que es muy útil articular propiedades importantes de componentes, como el tamaño de las imágenes y la longitud de los caracteres de los y pasajes de texto.” (Frost, 2013)

Las páginas son instancias específicas de plantillas que muestran el aspecto de una interfaz de usuario con contenido representativo real. (Frost, 2013) Las páginas son útiles para crear contenido personalizado para una aplicación o sitio web. Es en la fase de página donde podemos comprobar cómo se mantienen la coherencia entre todos los elementos dentro de la página. Justo en este punto podemos evaluar la interfaz y actualizar y retocar ponentes anteriores, permitiendo así iterar entre las distintas soluciones hasta alcanzar el diseño deseado.

### 3 Metodología

Para el desarrollo de este proyecto, se usó la metodología en cascada, separada en tres puntos:

#### 3.1 Análisis teórico y revisión de literatura

En esta etapa, se llevó a cabo un análisis exhaustivo de la teoría y literatura relacionada con el diseño atómico (Atomic Design) y su aplicación en proyectos Frontend. Se examinaron diversas fuentes para obtener un conocimiento sólido sobre los principios y mejores prácticas de Atomic Design.

La implementación del proyecto piloto se llevó a cabo de manera jerárquica, comenzando con la creación de los átomos y, una vez completados, procediendo con el desarrollo de las moléculas y así sucesivamente con el resto de los elementos hasta obtener la interfaz de usuario especificada en el diseño, para esta implementación, se estuvo usando tecnologías como TypeScript y React para la creación de las interfaces de usuario.

#### 3.2 Evaluación de la eficiencia y calidad del Atomic Design

Para evaluar la eficiencia y calidad del diseño atómico en el proyecto piloto, se utilizó diversas métricas:

- Duplicación de código: mide cuántos bloques de código se repiten dentro del código, para esto existen herramientas como el Linter para detectar estas repeticiones, con esta herramienta compararé el proyecto y qué tanto código se repite dentro de él.
- Acoplamiento y cohesión: Estas métricas evalúan qué tan fuerte está unido un componente a otros y cuánta funcionalidad está contenida en ellos. Se busca un acoplamiento bajo y una alta cohesión para favorecer la reutilización y el mantenimiento.

- Matriz de dependencias: Se realizó una matriz donde las filas y columnas representan diferentes componentes, y las celdas contienen información sobre si un componente depende de otro. Esto ayudó a visualizar el acoplamiento entre los distintos módulos.

## **4. Resultados**

### **4.1 Proceso Investigativo**

En el transcurso de la investigación acerca de este marco de trabajo, nos sumergimos en la comprensión de su forma de trabajar, contribuyendo así a la construcción de un marco teórico sólido. Además de esclarecer los principios fundamentales, este proceso investigativo no solo llenó el marco teórico, sino que también proporcionó una base robusta para la toma de decisiones estratégicas en relación con la implementación del Atomic Design. Las características identificadas nos brindan una visión clara y detallada, esencial para abordar con éxito la aplicación de este marco de trabajo en futuros proyectos web.

### **4.2 Características Clave del Atomic Design**

#### ***4.2.1 Reutilización de Componentes***

Una de las características más importantes del Atomic Design es la capacidad de reutilizar componentes en toda la aplicación. Los átomos, moléculas y organismos se crean una vez y pueden implementarse en múltiples partes de la interfaz de usuario. Esto reduce la duplicación de código y facilita la mantenibilidad del proyecto. “Escribir código de forma modularizada es muy importante tanto para el desde el punto de vista técnico y comercial, ya que reducirá los costos en lo que respecta al mantenimiento y el desarrollo posterior” (Nghì, 2017)

#### ***4.2.2 Consistencia Visual***

“La coherencia dentro de una aplicación debería facilitar la percepción humana y los procesos cognitivos como el escaneo visual, el aprendizaje y el recuerdo” (R. Mahajan y B. Shneiderman, 1997). El Atomic Design promueve la consistencia visual al definir patrones de diseño claros para cada nivel (átomos, moléculas, organismos, etc.). Esto garantiza que los componentes se vean y se comporten de la misma manera en toda la aplicación, lo que mejora la experiencia del usuario.

### ***4.2.3 Escalabilidad***

La estructura jerárquica del Atomic Design facilita la escalabilidad de proyectos. “...cada uno de los componentes permite la escalabilidad en el largo plazo debido a que divide la funcionalidad del aplicativo dándole a cada archivo una única responsabilidad” (M. Vega y K. Liliana, 2021). Los componentes pueden crecer en complejidad a medida que la aplicación se expande, y seguirán siendo manejables gracias a la organización en niveles claros.

### ***4.3.4 Facilita la Colaboración***

El Atomic Design promueve la colaboración entre diseñadores y desarrolladores, ya que ambos pueden trabajar en componentes individuales de manera independiente. Esto acelera el proceso de desarrollo y garantiza una mayor coherencia en la interfaz de usuario.

### ***4.3.5 Descomposición Jerárquica***

El Atomic Design comienza con la descomposición de la interfaz de usuario en átomos, que son los componentes más simples, y luego se agrupan en unidades más grandes como moléculas y organismos. Este enfoque jerárquico permite una organización estructurada y escalable de los componentes.

### ***4.3.6 Reutilización de componentes***

Uno de los principios clave es la reutilización de componentes. “. Los componentes son más versátiles, más modulares y auto contenidos, por tanto, mantenibles, adaptables y reutilizables en contextos muy diferentes” (E. Karlson, 1995). Los átomos y moléculas se crean una vez y se utilizan en múltiples partes de la interfaz de usuario. Esto reduce la duplicación de código y facilita la consistencia visual en toda la aplicación.

#### ***4.3.7 Consistencia Visual***

Atomic Design establece patrones de diseño claros para cada nivel de componente, lo que promueve una consistencia visual en toda la aplicación. Los componentes se diseñan y se comportan de la misma manera siempre que se utilicen, lo que mejora la experiencia del usuario y facilita el mantenimiento.

#### ***4.3.8 Facilita la Mantenibilidad***

La organización jerárquica y la reutilización de componentes hacen que los proyectos sean más fáciles de mantener. Cuando se necesita realizar cambios, es más sencillo actualizar un componente en lugar de buscar y modificar múltiples instancias dispersas por la aplicación.

#### ***4.3.9 Documentación Clara***

Cada nivel de componente se documenta de manera clara y se organiza en una biblioteca de diseño o sistema de diseño. Esto proporciona a los equipos una referencia centralizada para los componentes, sus variaciones y su uso.

#### ***4.3.10 Flexibilidad en la Implementación***

Atomic Design no dicta un lenguaje de programación o una tecnología específica. Puede aplicarse en diversas tecnologías web y frameworks, lo que brinda flexibilidad a los equipos de desarrollo.

#### ***4.3.11 Enfoque Iterativo***

El proceso de Atomic Design es iterativo. A medida que se crean y prueban componentes en diferentes contextos, se pueden realizar ajustes y mejoras continuas, lo que permite una evolución constante de la interfaz de usuario.

## **4.4 Estructura de la Biblioteca de Componentes**

### ***4.4.1 Jerarquía de Componentes***

Organiza la biblioteca en una jerarquía clara que refleje los niveles de Atomic Design, como átomos, moléculas, organismos, templates y páginas. Esto facilita la navegación y la búsqueda de componentes.

### ***4.4.2 Categorización Lógica***

Agrupar los componentes de manera lógica en categorías basadas en su función o uso. Por ejemplo, puedes tener secciones para botones, formularios, encabezados, etc.

### ***4.4.3 Nomenclatura Consistente***

Utiliza una nomenclatura coherente y descriptiva para los componentes. Esto ayuda a los diseñadores y desarrolladores a identificar rápidamente la utilidad y función de cada componente.

### ***4.4.4 Vista Previa Visual***

Ofrece vistas previas visuales de cada componente en la biblioteca. Las imágenes o miniaturas ayudan a los diseñadores a identificar rápidamente cómo se verá el componente en la interfaz de usuario.

Una vez comprendidos los principios y fundamentos del Atomic Design, centrados en la organización de componentes, es evidente que este enfoque ofrece una metodología sólida y altamente eficiente para el diseño y desarrollo web. La descomposición jerárquica, la reutilización de componentes, la consistencia visual y la facilitación de la colaboración entre equipos son solo algunas de las características clave que destacan en este marco de trabajo.

La estructura de la biblioteca de componentes, con su jerarquía clara, categorización lógica, nomenclatura consistente, metadatos detallados y vistas previas visuales, se revela como una herramienta esencial para la gestión y optimización de proyectos web.

## **4.5 Casos de éxito**

El Atomic Design ha demostrado ser una metodología revolucionaria en el diseño y desarrollo web, y varios casos de éxito respaldan su eficacia. A continuación, se muestran ejemplos destacados de empresas líderes que han implementado con éxito el Atomic Design, destacando los beneficios clave que han experimentado en términos de eficiencia, coherencia y escalabilidad.

### **4.5.1 IBM**

Una de las empresas líderes en tecnología a nivel global, ha realizado una transición exitosa hacia la implementación del diseño atómico (Atomic Design) en su sistema de diseño. Este enfoque se basa en la creación y utilización de patrones y componentes reutilizables en sus proyectos Frontend. La adopción de Atomic Design ha tenido un impacto significativo en la eficiencia de sus desarrollos. La reutilización de componentes se ha convertido en un pilar fundamental para mejorar la coherencia en sus aplicaciones y reducir de manera sustancial el tiempo de desarrollo.

En particular, IBM ha experimentado una mayor coherencia visual y funcional en sus aplicaciones gracias a la implementación de Atomic Design. La uniformidad en la apariencia y el comportamiento de los elementos de la interfaz de usuario ha mejorado la experiencia del usuario final. Además, el tiempo de desarrollo se ha visto significativamente reducido, lo que se traduce en una mayor agilidad y eficiencia en la entrega de proyectos. Esto ha permitido a IBM mantenerse a la vanguardia de la innovación tecnológica y ofrecer soluciones de alta calidad a sus clientes.

### ***4.5.2 Salesforce***

Empresa líder en soluciones de gestión de relaciones con los clientes (CRM), es otro ejemplo sobresaliente de la implementación exitosa de Atomic Design. La adopción de componentes atómicos en la creación de interfaces de usuario ha llevado a una mayor consistencia visual y funcional en todas sus aplicaciones. Esta consistencia no solo mejora la experiencia del usuario, sino que también facilita la escalabilidad de su sistema de diseño.

La implementación de Atomic Design en Salesforce ha permitido una coherencia visual que refuerza la identidad de la marca y aumenta la eficiencia en el desarrollo. La utilización de componentes reutilizables garantiza que los equipos de desarrollo mantengan una sincronización constante y que los cambios realizados en un componente se reflejen de manera coherente en todas las aplicaciones. Esto ha demostrado ser esencial para Salesforce, ya que la escalabilidad de su sistema de diseño es crucial para satisfacer las necesidades de una amplia gama de clientes y aplicaciones.

### ***4.5.3 Shopify***

Plataforma de comercio electrónico líder en el mercado, ha adoptado Polaris, un sistema de diseño basado en Atomic Design. Polaris proporciona directrices y componentes reutilizables que han mejorado significativamente la eficiencia en el desarrollo de productos. Además, ha contribuido a lograr una interfaz de usuario más coherente en todas las aplicaciones de Shopify.

La implementación de Atomic Design en Shopify ha permitido a los equipos de desarrollo aprovechar al máximo los componentes reutilizables de Polaris. Esto ha agilizado la creación de nuevas funcionalidades y la adaptación de la interfaz de usuario a las necesidades cambiantes del mercado. La coherencia en la interfaz de usuario garantiza una experiencia de usuario uniforme en todas las etapas de compra, lo que ha sido fundamental para el éxito continuo de Shopify como plataforma de comercio electrónico de renombre.

#### **4.5.4 Buffer**

Una empresa que ofrece servicios de gestión de redes sociales ha destacado por su exitosa implementación del Atomic Design en su proceso de desarrollo. La adopción de patrones de diseño reutilizables ha tenido un impacto significativo en su eficiencia y en la coherencia de sus aplicaciones. La implementación de Atomic Design ha permitido a Buffer desglosar su interfaz de usuario en componentes atómicos que pueden ser fácilmente reutilizados en diversos contextos.

Esta estrategia ha llevado a una notable aceleración en el desarrollo de nuevas características y mejoras. Los equipos de desarrollo de Buffer pueden aprovechar componentes previamente diseñados y probados, lo que reduce el tiempo requerido para implementar nuevas funcionalidades. Además, esta coherencia en la interfaz de usuario se traduce en una experiencia de usuario más uniforme y atractiva en todas las aplicaciones de Buffer.

La implementación exitosa de Atomic Design en Buffer ilustra cómo esta metodología puede ser aplicada con éxito en una variedad de contextos empresariales, no solo en el diseño web, sino también en el desarrollo de aplicaciones y servicios en línea.

#### **4.5.5 GitHub**

Una plataforma de desarrollo colaborativo de software no solo aplica el Atomic Design en su interfaz de usuario, sino que también fomenta activamente su adopción a través de numerosos proyectos alojados en la plataforma. Esto ha contribuido significativamente a fomentar la colaboración y la comunidad en torno al Atomic Design.

La adopción de Atomic Design en GitHub ha permitido a los desarrolladores que utilizan la plataforma aplicar los principios de Atomic Design en sus propios proyectos. La comunidad de código abierto que se ha formado alrededor de Atomic Design en GitHub ha compartido y colaborado en la creación de componentes atómicos y patrones de diseño reutilizables, lo que ha enriquecido la metodología y ha ampliado su adopción en la comunidad de desarrollo.

Esta iniciativa no solo ha fortalecido la comunidad de desarrollo en GitHub, sino que también ha demostrado cómo la colaboración y la promoción activa de las mejores prácticas pueden tener un impacto significativo en la adopción de nuevas metodologías en la industria del desarrollo de software.

En línea con estos éxitos, el próximo paso es aplicar las lecciones aprendidas y los beneficios observados en el propio entorno empresarial, en un proyecto piloto dentro de PrevalentWare, centrado en el desarrollo Frontend.

#### **4.6 Proyecto piloto**

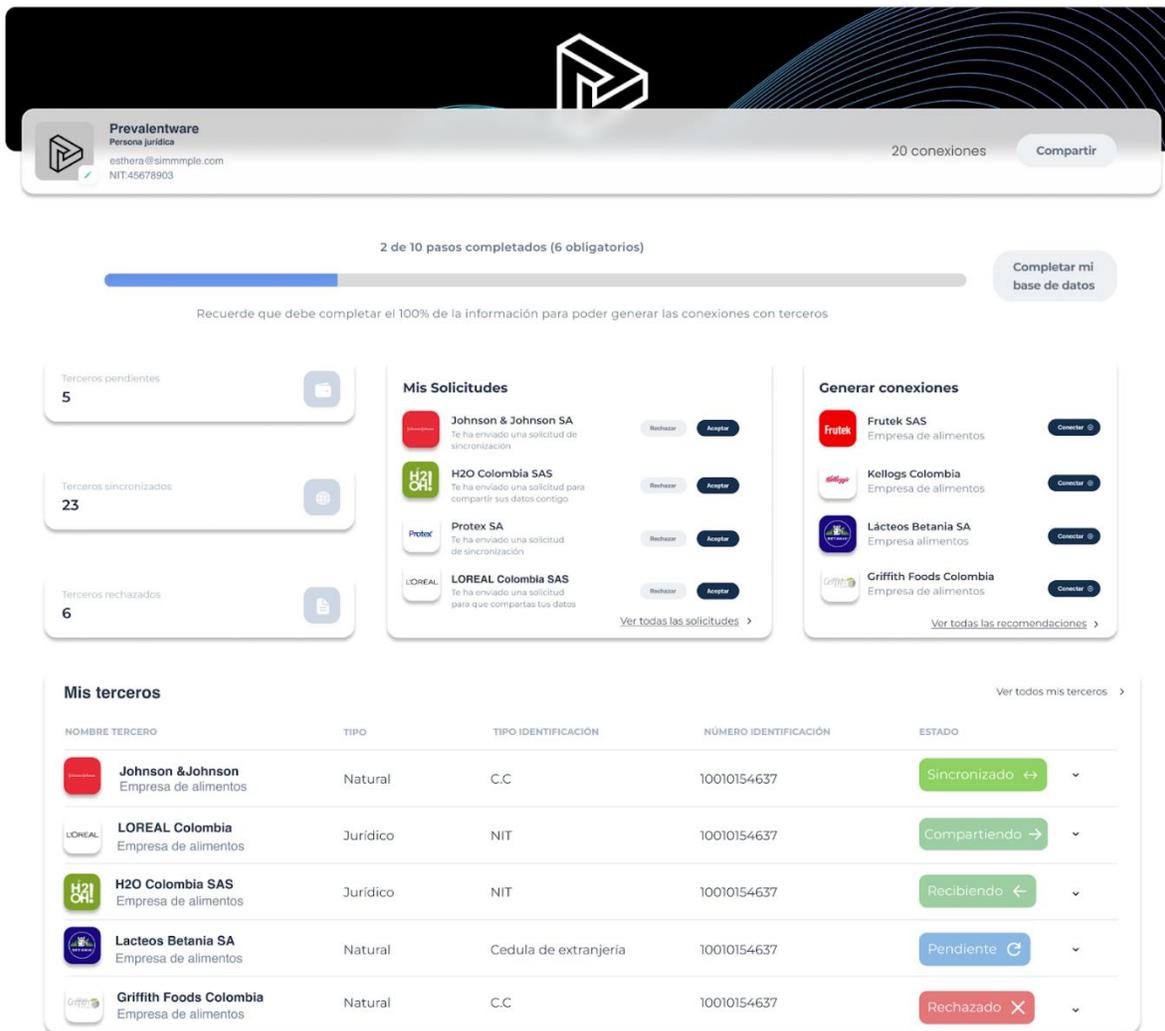
Dentro de los cimientos fundamentales de PrevalentWare se encuentra el desarrollo Frontend, una esfera que alberga una variedad de proyectos. Estos proyectos, aunque cumplen con sus funciones de manera efectiva, parten de una base no estructurada. A pesar de que esta falta de estructura no impide el desarrollo ni la ejecución de tareas, sí obstaculiza la creación de un orden definido, generando un impacto negativo a largo plazo en el mantenimiento de los proyectos.

Ante esta problemática, surgió la iniciativa de "reorganizar la casa", es decir, de establecer procesos y marcos de trabajo que dotaran a la estructura de una solidez necesaria. Al identificar la oportunidad de mejora dentro de los procesos de la empresa, se propuso un proyecto piloto: la "Gestión de terceros". Este proyecto interno ya contaba con un sistema de diseño preexistente; sin embargo, carecía de una estructura definida bajo el marco de trabajo del Atomic Design.

Se implementó un enfoque de ingeniería inversa para descomponer y reconstruir los componentes del sistema de diseño, asegurando que cada uno se alineara adecuadamente con los principios del Atomic Design. La reorganización se llevó a cabo de manera cuidadosa y metódica para no afectar la funcionalidad existente del proyecto.

**Figura 3**

*Diseño planteado para el proyecto piloto*



*Nota.* Imagen brindada por el equipo de diseño de PrevalentWare.

La reorganización del proyecto incluyó una revisión exhaustiva de la estructura de la página. Para garantizar una alineación jerárquica efectiva, se dividió el diseño en los elementos más básicos.

### 4.6.1 Átomos

Los átomos representan los elementos más básicos e indivisibles de la interfaz. En esta sección, se detallan y analizan los átomos utilizados en el proyecto "Gestión de terceros". Estos elementos fundamentales sirven como bloques de construcción esenciales para el diseño general de la interfaz. En este espacio definimos botones, inputs, avatares, imágenes, textos, iconos; elementos que individualmente no tienen sentido pero que son los cimientos para la construcción del proyecto piloto.

#### Figura 4

*División de algunos átomos obtenidos del diseño planteado para el proyecto piloto.*



*Nota.* Imagen obtenida por la separación de trozos del diseño.

### 4.6.2 Moléculas

Las moléculas son combinaciones de átomos que trabajan juntos para cumplir una función específica. Esta sección explora las moléculas integradas en el diseño de la interfaz. Cada molécula se construye mediante la composición de átomos para lograr una mayor complejidad y funcionalidad. Si unimos los elementos de navegación, con el icono de notificación, podemos obtener una barra de navegación, o si unimos un icono de lupa con un input, tendremos nuestro

buscador, son elementos que son más fuertes y tienen más sentido, pero aún carecen de una estructura.

### Figura 5

*División de algunas moléculas obtenidos del diseño planteado para el proyecto piloto.*



*Nota.* Imagen obtenida por la separación de trozos del diseño.

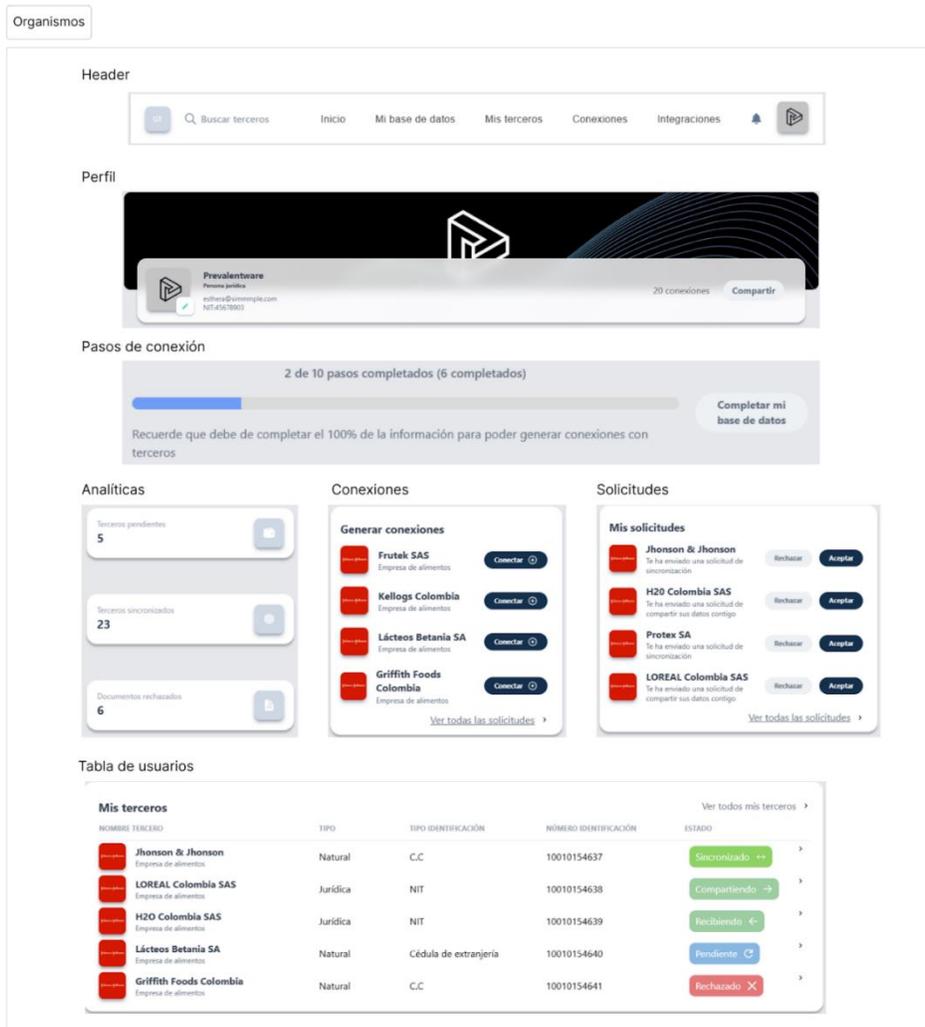
#### 4.6.3 Organismos

Los organismos representan componentes aún más complejos, formados por la combinación de moléculas y átomos. Estos elementos de nivel superior encapsulan funciones más avanzadas y completas dentro de la interfaz, son bloques más coherentes y con más sentido, la unión de estos nos permite subdividir la página en secciones que tienen coherencia por sí solas,

como lo son el Header, la información del perfil, el paso a paso de las conexiones o la tabla de los distintos usuarios con los que el usuario ha conectado.

**Figura 6**

*División de algunos organismos obtenidos del diseño planteado para el proyecto piloto.*



*Nota.* Imagen obtenida por la separación de trozos del diseño.

En el contexto de este proyecto, se optó por eludir la fase del template y pasar directamente al diseño de la página. Atomic Design, como marco de trabajo, demuestra su flexibilidad al permitir ajustes según las necesidades específicas de cada proyecto. En este caso, la implementación se beneficia de la eficiencia que implica dar este paso directo, agilizando así el desarrollo.

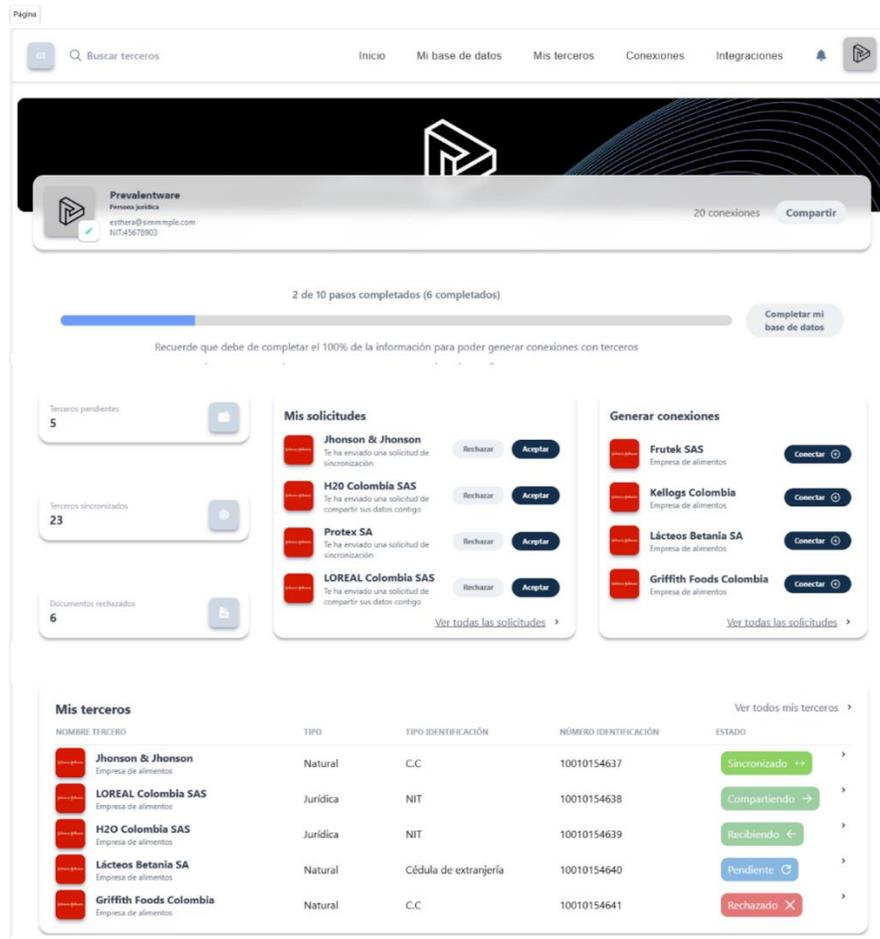
La filosofía del Atomic Design no impone rigidez, lo cual permite ajustes según las particularidades de cada proyecto. En esta ocasión, se evaluó que prescindir del paso del template se alineaba de manera más efectiva con las metas y requerimientos del proyecto "Gestión de terceros".

La decisión de omitir el template responde a la necesidad de acelerar el proceso de desarrollo. Este enfoque más directo en el diseño de la página permite una respuesta ágil a los cambios y facilita una iteración más rápida durante el desarrollo. Se privilegió la simplicidad y la eficiencia en la implementación. La omisión del paso del template simplifica el proceso, haciendo que el equipo pueda concentrarse directamente en la creación de la interfaz final, manteniendo la coherencia visual en todo momento.

#### 4.6.4 Pagina

**Figura 7**

*División de la página obtenida del diseño planteado para el proyecto piloto.*



*Nota.* Imagen obtenida por la separación de trozos del diseño.

Una vez se integran todos los organismos, el resultado es el producto final. En este estado, logramos la coherencia visual deseada, reflejando fielmente el diseño base. Este enfoque no solo respalda la eficiencia del desarrollo, sino que también asegura que el producto final se alinee estrechamente con las expectativas visuales establecidas desde el principio.

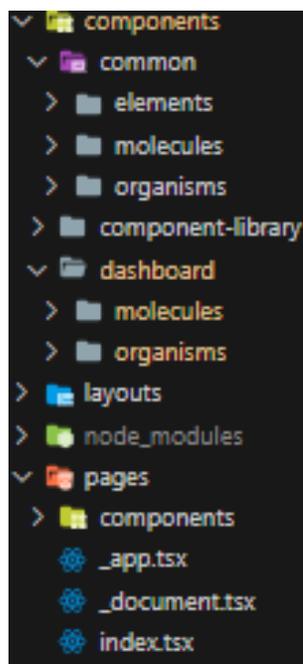
## 4.7 Desarrollo

El siguiente paso implica el desarrollo del proyecto. Se ha optado por seguir utilizando el conjunto de tecnologías convencionales para proyectos Frontend de la empresa, que incluye la librería React con el framework Next.js, junto con TypeScript para gestionar el tipado y proporcionar mayor seguridad al proceso de desarrollo.

En cuanto a la arquitectura interna del proyecto, se ha decidido seguir la convención de Next.js en su versión 13, la cual establece una estructura específica para desplegar páginas. En este enfoque, las páginas se almacenan en una carpeta designada, mientras que los elementos del Atomic Design se almacenan en la carpeta “*Components*”. Dentro de esta carpeta, se encuentra una subcarpeta llamada “*commons*” para los componentes compartidos en toda la aplicación, donde se colocan los átomos que pueden ser reutilizados en toda la aplicación.

### Figura 8

*Arquitectura interna del proyecto*



*Nota.* Imagen tomada del editor de código.

La carpeta "*pages*" contiene las distintas páginas del proyecto, siendo la página principal identificada en el archivo "*index.tsx*". En cuanto a los componentes, se ha establecido una clara separación entre los elementos comunes, los elementos de dashboard y los elementos relacionados con la biblioteca de componentes.

Después de definir la estructura de carpetas, se adapta un enfoque jerárquico en el desarrollo. Comenzando con los átomos, que son elementos individuales de la página. En nuestro contexto, llamamos "*elementos*" a los átomos para evitar confusiones con otras tecnologías que también utilizan el término "*átomo*". Los átomos representan elementos visuales que, por sí solos, no tienen un significado completo. Utilizando TypeScript, se garantiza que la funcionalidad de cada componente se mantenga sin pérdida.

Avanzamos hacia las moléculas, construyéndolas mediante la reutilización de elementos que no se modifican en el HTML, así como de átomos previamente creados. La separación de responsabilidades en el proceso de diseño es crucial, proporcionando un enfoque paso a paso para construir cada molécula y evitar confusiones en las decisiones de diseño.

Luego, se abordan los organismos, los cuales son más flexibles y permiten la inclusión de átomos, moléculas u otros organismos. Durante el desarrollo, observamos la oportunidad de optimizar el código al identificar repeticiones en componentes de tarjetas, como las de Análisis, Solicitudes o Conexiones. Se opta por separar el código duplicado en otro componente para evitar posibles fallos en la interfaz de usuario y reducir los tiempos de desarrollo. Esto es crucial puesto que el proceso de desarrollo del Atomic Design es iterativo, la primera versión del código no representa una camisa de fuerza para realizar modificaciones y separar y crear nuevos elementos, se debe primar la separación de responsabilidades y la optimización, antes que la deuda técnica.

Finalmente, se unen los organismos complejos para construir la página principal. La correcta división de cada componente facilitó la colocación de los elementos en el orden de la

página, logrando el resultado esperado. Posteriormente, se realizan ajustes finos, alineando cada elemento según el diseño y perfeccionando los detalles.

Después de finalizar la creación de la página, se procede a la fase de documentación. Se generan páginas adicionales que presentan un menú navegacional, permitiendo explorar diferentes secciones como átomos, moléculas, organismos y páginas. Cada una de estas secciones detalla los elementos que la componen. Este enfoque facilita la inmersión del equipo de desarrollo en el proyecto, posibilitándoles comprender de manera rápida los elementos existentes y agilizando el proceso de codificación.

Aunque mantener la documentación actualizada implica un esfuerzo adicional a corto plazo, a largo plazo se traduce en un ahorro significativo de tiempo al prevenir la duplicación de código entre los desarrolladores.

## **4.8 Evaluación de la eficiencia**

En la búsqueda constante de mejorar la eficiencia en el desarrollo de interfaces de usuario, se emplearon diversas herramientas y métricas para evaluar aspectos clave como la duplicación de código, el acoplamiento y la cohesión. A continuación, se detallan los resultados obtenidos y cómo la aplicación del Atomic Design contribuyó significativamente a optimizar estos aspectos.

### ***4.8.1 Duplicación del código***

Para abordar la duplicación de código, se utilizaron dos herramientas fundamentales: ESLint y SonarQube. ESLint desempeñó un papel crucial en el desarrollo individual de componentes al identificar errores, malas prácticas y violaciones de convenciones de codificación en tiempo real. Al corregir estos problemas de manera inmediata, se previnieron malas prácticas a largo plazo, mejorando la calidad del código. Por otro lado, SonarQube, configurado

específicamente para evaluar la duplicidad del código en todo el proyecto, proporcionó una visión más amplia del panorama, permitiendo abordar patrones repetitivos y optimizar la estructura global del código.

**Figura 9**

*Gráfica de SonarQube que resalta la cantidad total de líneas de código comparado con la cantidad de líneas de código duplicadas*



*Nota.* Tomado de la aplicación SonarQube.

La utilización de SonarQube para analizar el código proporcionó información valiosa sobre la duplicación de código en el proyecto. De las 2908 líneas de código evaluadas, se identificó que tan solo 45 líneas presentaban duplicación. Este resultado indica un nivel muy bajo de redundancia en el código, lo cual es una señal positiva de calidad y mantenibilidad. La baja incidencia de duplicación no solo mejora la legibilidad y comprensión del código, sino que también reduce el riesgo de errores y facilita las futuras actualizaciones y expansiones del proyecto.

**4.8.2 Acoplamiento y cohesión**

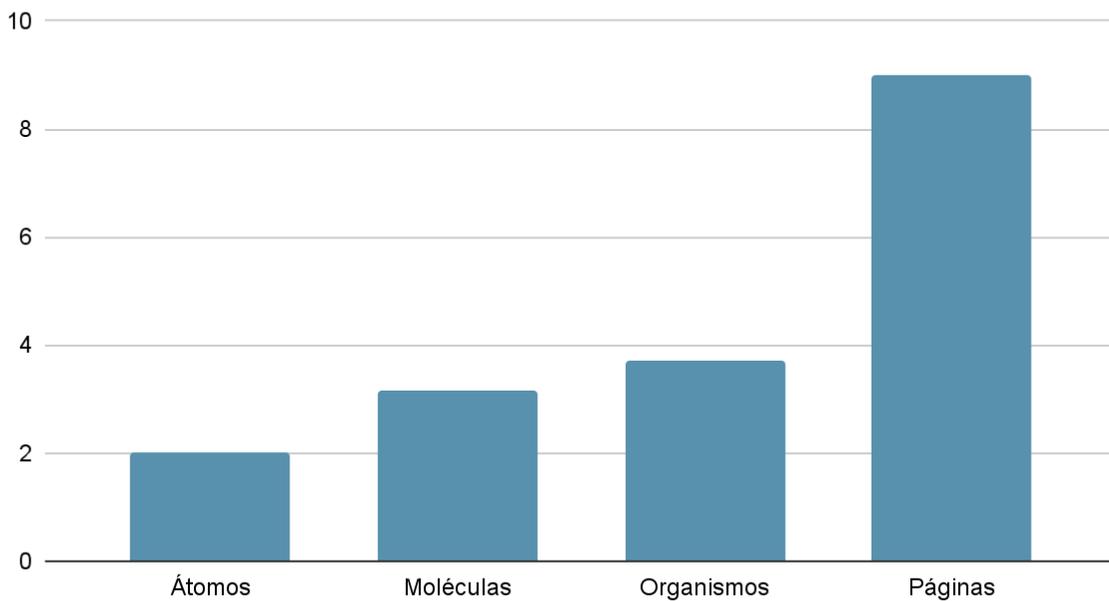
La adopción del Atomic Design facilitó la creación de una estructura que fomenta una alta cohesión y bajo acoplamiento. Al aplicar el principio de responsabilidad única, se logró una

distribución eficiente de tareas entre los diferentes elementos del diseño, optimizando así la cohesión de los componentes. En términos de acoplamiento, se evaluaron las dependencias entre componentes para garantizar un bajo nivel de interconexión. El gráfico de barras muestra claramente que, en promedio, cada grupo de componentes (átomos, moléculas, organismos y páginas) no depende de más de 5 componentes, indicando un acoplamiento reducido y una separación efectiva de responsabilidades.

**Figura 10**

*Gráfica de barras que muestra la comparación entre la cantidad promedio de dependencias de cada grupo de componentes, donde el eje X son los grupos de componentes y el eje Y son la cantidad promedio de dependencias.*

Dependencia de los componentes



*Nota.* Tomado de hoja de Excel.

El siguiente gráfico ilustra los promedios de dependencias para cada nivel de la jerarquía de Atomic Design. Destaca la baja dependencia, especialmente en átomos, moléculas y organismos, donde cada grupo de componentes muestra una conexión limitada con otros elementos. Este enfoque contribuye a la modularidad y facilita la mantenibilidad del código. En

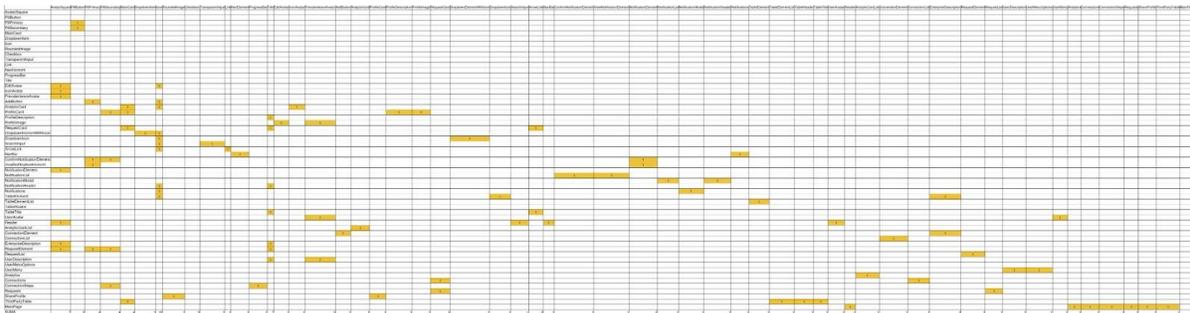
el caso de las páginas, una mayor cantidad de dependencias es esperada, ya que refleja la complejidad inherente a la presentación de información en la interfaz de usuario.

### 4.8.3 Matriz de dependencias

La matriz de dependencias proporciona una representación visual esclarecedora de las relaciones directas entre los componentes utilizados en el proyecto piloto.

**Figura 11**

*Matriz de dependencias, tanto el eje Y como el eje X, alberga el nombre de los componentes del proyecto, mientras que las celdas coloreadas indican las dependencias entre ellos.*



*Nota.* Tomada de hoja de Excel.

La observación detallada de la matriz revela que los casos de dependencia son limitados. Esta estructura refleja la implementación exitosa del principio de responsabilidad única y la baja interconexión entre los componentes. La ausencia de componentes con múltiples responsabilidades sugiere una distribución eficiente de tareas, cumpliendo así con los objetivos planteados para optimizar la cohesión del sistema.

La ventaja de esta representación gráfica radica en su capacidad para identificar claramente los responsables directos de cada dependencia. Esto facilita el análisis de la arquitectura del proyecto, permitiendo a los desarrolladores comprender rápidamente cómo se relacionan entre sí los diferentes elementos. Además, la identificación de dependencias limitadas respalda la idea de un acoplamiento bajo, ya que cada componente parece depender de un número reducido de elementos, lo que contribuye a la modularidad y mantenibilidad del código.

## 5 Análisis

La aplicación del Atomic Design en el proyecto piloto "Gestión de terceros" ha demostrado ser un enfoque efectivo para mejorar la eficiencia en el desarrollo de interfaces de usuario. A través de la descomposición jerárquica de componentes, la reutilización eficiente, y la promoción de la consistencia visual, se lograron resultados tangibles en términos de calidad del código, mantenibilidad y coherencia en la interfaz de usuario.

La combinación de ESLint y SonarQube ha sido clave para abordar la duplicación de código. La detección temprana de errores por parte de ESLint durante el desarrollo individual ha contribuido a la prevención de malas prácticas a largo plazo. El análisis exhaustivo realizado por SonarQube reveló que solo el 1.55% de las líneas de código evaluadas presentaban duplicación. Este resultado indica un alto nivel de calidad y mantenibilidad, lo que es esencial para garantizar la robustez a largo plazo del proyecto.

La adopción del principio de responsabilidad única en el Atomic Design ha facilitado una distribución eficiente de tareas, resultando en una alta cohesión entre los componentes. El bajo acoplamiento, evidenciado por la limitada cantidad de dependencias entre componentes, respalda la modularidad y mantenibilidad del código. La implementación exitosa de Atomic Design ha logrado un equilibrio entre la separación de responsabilidades y la conexión efectiva entre los diferentes elementos de la interfaz.

Si bien el Atomic Design ha demostrado ser una metodología eficaz en una variedad de proyectos, es importante reconocer que no es necesariamente la mejor opción en todas las circunstancias. Hay situaciones específicas en las cuales la aplicación del Atomic Design puede no ser la opción más adecuada. Por ejemplo, en proyectos pequeños con requisitos simples y una cantidad limitada de componentes, la implementación completa del Atomic Design podría resultar excesiva. La estructura jerárquica y modularidad que el Atomic Design ofrece puede ser más beneficiosa en proyectos más grandes y complejos. Además, para proyectos que tienen como objetivo principal la creación rápida de prototipos o demostraciones conceptuales, resulta innecesaria la estructura del proyecto enfocada al Atomic Design; Hay que tener en cuenta que

este marco de trabajo va de la mano de un equipo de diseño y realizarlo en proyectos rápidos podría resultar en sobrecostos innecesarios y mayores tiempos de desarrollo.

## 6 Conclusiones

El proceso investigativo ha proporcionado una valiosa comprensión de las limitaciones y beneficios del marco de trabajo Atomic Design. La adaptación exitosa de este marco a las herramientas de trabajo de la empresa ha permitido maximizar sus ventajas, especialmente al alinearse con la filosofía de componentización presente en las herramientas utilizadas. Este enfoque ha sentado las bases sólidas para los proyectos de la empresa, brindando la posibilidad de construir sobre una base consistente y compartida.

El análisis de los casos de estudio ha revelado la flexibilidad inherente del Atomic Design, demostrando que no es un marco rígido, sino adaptable a las necesidades específicas de la empresa. La clave reside en establecer un proceso definido que garantice resultados estandarizados y eficientes. La confianza generada al observar casos de éxito confirma que Atomic Design no es un experimento, sino una metodología respaldada por resultados tangibles, fortaleciendo así la decisión de implementarlo en PrevalentWare.

El desarrollo del proyecto piloto ha desempeñado un papel crucial al capacitar a todo el equipo de desarrollo web en la comprensión del desglose de componentes según la lógica de Atomic Design. Esta experiencia ha permitido reparar responsabilidades dentro de la interfaz, establecer jerarquías escalables y, en última instancia, construir una interfaz sólida, fácilmente escalable y mantenible. El aprendizaje obtenido se convierte en conocimiento práctico, allanando el camino para futuros proyectos basados en Atomic Design.

Además, el análisis del proyecto piloto proporciona las bases cuantitativas necesarias para respaldar la viabilidad de la implementación de Atomic Design en futuros proyectos. Las cifras obtenidas durante la evaluación de eficiencia, como la baja duplicación de código y la estructura de bajo acoplamiento, respaldan la eficacia del marco. Este análisis cuantitativo refuerza la justificación para la adopción de Atomic Design y allana el camino para futuros proyectos piloto, donde la integración con el equipo de diseño puede mejorar aún más la colaboración y la coherencia entre las disciplinas dentro de la empresa.

## Referencias

- Descripción general del componente de lienzo (contiene vídeo) - Power Apps. (s/f). Microsoft.com. Recuperado el 28 de julio de 2023, de <https://learn.microsoft.com/es-es/power-apps/maker/canvas-apps/create-component>
- Frost, B. (2016). Atomic Design. New York, NY: Brad Frost Web.
- Karlsson, E. A. (ed). (1995) Software Reuse: A Holistic Approach, John Wiley & Sons, Chichester, UK.
- M. Claudia Albornoz (2014). Diseño de Interfaz Gráfica de Usuario. Red de Universidades con Carreras en Informática
- M. Vega, K. Liliana (2021). Desarrollo del frontend para el aplicativo web Voice de la empresa TMSOFT. UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA
- Le, Nghi (2017). Creating software component using atomic design and test-driven development. Laurea University of Applied Sciences
- Palomeque-Zambrano, E. E., & Campoverde-Molina, M. (2023). Propuesta de una arquitectura de software para el desarrollo de sitios web accesibles. MQRInvestigar, 7(3), 1458-1474.
- R. Mahajan and B. Shneiderman, "Visual and textual consistency checking tools for graphical user interfaces," in IEEE Transactions on Software Engineering, vol. 23, no. 11, pp. 722-735, Nov. 1997, doi: 10.1109/32.637386.
- Roedavan, R., Pratondo, A., Pudjoatmodjo, B., & Siradj, Y. (2021). Adaptation Atomic Design Method for Rapid Game Development Model. development, 13, 14.
- Web Components. (2021, enero 30). MDN Web Docs. Recuperado el 6 de agosto de 2023, de [https://developer.mozilla.org/es/docs/Web/Web\\_Components](https://developer.mozilla.org/es/docs/Web/Web_Components)