

Informe Tecnico y Visual de Avance Gasex - Dia de Hoy

Resumen detallado de los avances tecnicos, visuales y estructurales realizados hoy en el proyecto Gasex. Incluye diagramas y esquemas de la estructura del theme, jerarquia de categorias, flujo y relacion de bloques.

1. Bloques Gutenberg Personalizados

Tecnico:

- Conversion de secciones HTML/CSS a bloques Gutenberg editables (con atributos dinamicos y estructuramodular).
- Compilacion y build del bloque con npm y @wordpress/scripts para uso profesional y portable.
- Registro automatico del bloque via functions.php.

Visual:

- Bloque visible en Gutenberg, editable visualmente y con vista previa animada.

2. Iconografia y Experiencia de Usuario

Tecnico:

- Vectorizacion exacta del isotipo Gasex como SVG embebido, uso correcto en el registro del bloque.
- Aclaracion sobre soporte de SVG/Dashicons como icono de bloque.

Visual:

- Icono propio en selector y barra del editor.
- Entrega de iconos alternativos en PNG y variantes visuales.

3. Mejoras Visuales y UX en Gutenberg

Tecnico:

- Uso del atributo 'example' en block.json para vista previa realista y animada.

Visual:

- Vista previa profesional y representativa al buscar/insertar el bloque.

4. Escalabilidad y Estructura Avanzada (CPT, Taxonomias, Plantillas)

Tecnico:

- Registro de Custom Post Type jerarquico para categorias/subcategorias.
- Plantillas de bloques asociadas al CPT para carga automatica de bloques en cada nuevo registro.
- Soporte total para subniveles y edicion visual flexible.

Visual:

- Edicion visual y pre-carga de bloques base en cada categoria/subcategoria, anidacion intuitiva en el admin.

5.Codigo, Portabilidad y Buenas Practicas

Tecnico:

- Tema independiente, sin plugins ni dependencias externas para usuarios finales.
- Estructura modular y soporte para edicion futura.

- Ejemplo de estructura de carpetas profesional.

Visual:

- Experiencia visual intuitiva y portable, igual o superior a cualquier constructor visual premium.

Resumen visual del flujo

1. Admin crea una nueva categoria/subcategoria.
2. Plantilla pre-carga los bloques basicos (ej: Tenemos Todo).
3. El usuario personaliza contenido y subniveles desde el editor visual Gutenberg.
4. El icono propio Gasex y la preview refuerzan la identidad y facilidad de uso.
5. El codigo queda limpio, escalable y apto para entregar a cualquier cliente o equipo.

Diagrama: Estructura de Carpetas del Theme Gasex

```
/wp-content/themes/gasex-theme/
|-- blocks/
|   |-- tenemos-todo/
|       |-- block.json
|       |-- style.css
|       |-- build/
|           |-- index.js
|           |-- src/
|               |-- index.js
|               |-- edit.js
|               |-- save.js
|-- inc/
|   |-- blocks.php
|-- functions.php
|-- style.css
```

1. Estructura de Carpetas del Theme Gasex

```
lua
/wp-content/themes/gasex-theme/
|-- blocks/
|   |-- tenemos-todo/
|       |-- block.json
|       |-- style.css
|       |-- build/
|           |-- index.js
|           |-- src/
|               |-- index.js
|               |-- edit.js
|               |-- save.js
|-- inc/
|   |-- blocks.php
|-- functions.php
|-- style.css
```

Diagrama: Jerarquia de Categorías y Subcategorías (CPT)

Categoria Gasex (CPT)

```
-- Subcategoria 1
|   |-- Sub-Subcategoria 1.1 |
|-- Sub-Sub-Subcategoria 1.1.1
|-- Subcategoria 2
```

2. Jerarquía de Categorías y Subcategorías (CPT)

```
lua
Categoria Gasex (CPT)
|-- Subcategoria 1
|   |-- Sub-Subcategoria 1.1
|       |-- Sub-Sub-Subcategoria 1.1.1
|-- Subcategoria 2
```

Diagrama: Flujo de Edición Visual en Gutenberg

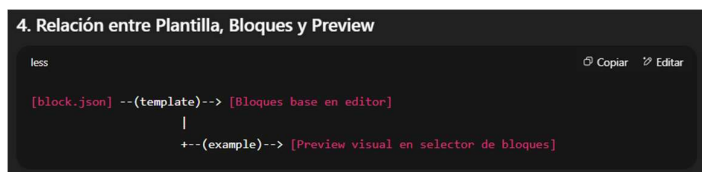
```
[Admin] --> [Crear nueva Categoria Gasex]
|           v [Editor Gutenberg]
|           v [Plantilla pre-cargada con
bloques]      |           v [Usuario
edita bloques y guarda]
```

3. Flujo de Edición Visual en Gutenberg

```
less
[Admin] --> [Crear nueva Categoria Gasex]
|
| v
[Editor Gutenberg]
|
| v
[Plantilla pre-cargada con bloques]
|
| v
[Usuario edita bloques y guarda]
```

Diagrama: Relación entre Plantilla, Bloques y Preview

```
[block.json] --(template)--> [Bloques base en editor]
|
+--(example)--> [Preview visual en selector de bloques]
```



Informe de Avance Diario – Desarrollo Gasex “Elementos”

Fecha: 2025-07-30

Responsable: Ariel Antonio Jara Fuentes

1. Revisión y Optimización CPT "Elementos"

- Se revisó y ajustó el archivo principal `cpt-elementos.php` encargado de registrar el Custom Post Type “Elementos”.
- Se verificaron y optimizaron las funciones para agregar **metaboxes** personalizados, permitiendo la edición de todos los datos necesarios para los elementos químicos del sitio.
- Se estandarizó el uso de funciones de seguridad (`esc_attr`, `esc_textarea`, `sanitize_text_field`, etc.) tanto en el guardado como en el despliegue de datos.

2. Panel de Edición WordPress – Mejoras

- Estructuración clara de campos para ingresar toda la información relevante de cada elemento:
 - Datos principales (símbolo, número atómico, descripción, núcleo, órbitas, electrones, imágenes, PDF).
 - Características perceptibles.
 - Configuración electrónica.
 - Detalles extendidos: tablas para fases sólida y líquida, fase gaseosa y aplicaciones.
 - Carga de imágenes y PDFs mediante Media Library, todo con uploaders modernos e intuitivos.

3. Sección Nueva: “Ejemplos Prácticos Resumen”

- **Nueva funcionalidad solicitada:** Permitir el ingreso de hasta 3 “ejemplos prácticos resumen” (textos cortos) en el editor del CPT.
- Se definieron los **campos personalizados** `_ejemplo_resumen_1`, `_ejemplo_resumen_2` y `_ejemplo_resumen_3` para estos textos.
- Se generó el **llamado PHP dinámico** para mostrar estos ejemplos como una lista `<ul><li>...</li></ul>` en la sección “Ejemplos Prácticos” de la plantilla, solo si existen datos.

4. Integración Front-End

- Se adaptó el código HTML de la sección “Ejemplos Prácticos” para que consuma dinámicamente los 3 ítems del panel, optimizando la semántica y el estilo (`<ul class="texto-ejemplos">...</ul>`).
- El sistema ya permite que el usuario final vea solo los ejemplos que fueron completados, sin mostrar campos vacíos.

5. Soporte y Próximos Pasos

- Queda habilitado el flujo desde la carga en el backend hasta el renderizado en el frontend de todos los campos principales, incluyendo la nueva sección de ejemplos prácticos resumen.
- Siguiendo etapa: continuar con otras secciones o iteraciones según lo que vayas priorizando en la ficha del elemento y su visualización.

Informe de Avance — Integración de Secciones CPT Aplicaciones (WordPress)

Fecha: 31 de Julio 2025

1. Limpieza y Orden del Código CPT

- Se depuró y ordenó el archivo `cpt-aplicaciones.php` para mayor claridad y mantenibilidad.
 - Se definieron y organizaron los *metaboxes* para cada sección que el usuario debe editar al crear/editar una Aplicación.
 - Se centralizó la función de guardado de todos los campos personalizados, minimizando duplicación de código.
-

2. Secciones Integradas y Sus Campos Personalizados

a. Subtítulo de la Cabecera

- **Campo:** Texto para subtítulo (single input).

b. Sección: Contenido 1

- **Campos:**
 - Título
 - Imagen 1
 - Texto 1
 - Subtítulo
 - Texto 2
 - Imagen 2

c. Sección: GASEX-UTILIZADOS

- **Campos:**
 - Título general
 - Para cada columna (2):
 - Imagen de fondo
 - Título interior
 - Subtítulo 1 + Texto 1
 - Subtítulo 2 + Texto 2
 - Texto final inferior

d. Sección: Equipos y Tamaños

- **Campos:**
 - Título
 - Imagen de fondo (background)
 - 7 imágenes en posiciones variables (overlay)
 - Texto general + 4 pares Subtítulo/Texto + Texto final

e. Sección: SECCION-IMAGENES-APLICACIONES

- **Campos:**

- Título
- 4 Imágenes y sus textos correspondientes

f. Sección: EMPLEO-OXICORTE

- **Campos:**
 - Título 1
 - Texto 1, Texto 2
 - 3 imágenes y texto para cada una
 - Título 2 + imagen + texto
 - 4 pares Subtítulo/Texto
 - Título 3 + texto final

g. Sección: APLICACIONES

- **Campos:**
 - Título
 - 4 tarjetas: cada una con Imagen, Subtítulo, Texto

h. Sección: SERVICIOS-GASEX

- **Campos:**
 - Título
 - 4 servicios: imagen, título, texto cada uno
 - Texto inferior (diferencial)

i. Sección: IMAGEN-APLICACIONES

- **Campos:**
 - Imagen de fondo (background)
 - Texto largo

3. Ejemplo de Llamados para Mostrar los Campos en el Frontend

Se entregaron **snippets PHP** para reemplazar los contenidos estáticos en cada sección del template (`single-aplicacion.php`), incluyendo:

- Obtención de imágenes vía `wp_get_attachment_image_url($id, 'full')`
 - Impresión de textos con `esc_html()`, `nl2br()`
 - Inserción dinámica de textos, títulos, imágenes, y textos de etiquetas, según corresponda.
-

4. Mejoras Visuales y CSS

- Se sugirió CSS para overlays, transparencias, y ajuste de imágenes de fondo usando `background-image`, para mayor control visual y responsividad.
 - Se recomendó CSS para evitar deformaciones de imagen (`object-fit: cover;` para `<img>`, `background-size: cover;` para `background`).
-

5. Sugerencias/Notas Finales

- **Pruebas:** Se recomendó probar la inserción de contenido desde el admin para verificar que todos los campos aparecen y se guardan correctamente.
- **Extensión:** Si alguna sección requiere más campos (por ejemplo, subtítulos adicionales, campos WYSIWYG, etc.), el sistema está preparado para agregar fácilmente.
- **Soporte Multimedia:** El código sugerido permite vista previa de imágenes al seleccionar desde la Mediateca de WordPress, con cambio inmediato en el admin.
- **Centralización:** Todo el JS para subir imágenes se centralizó en un solo lugar para evitar conflictos y recargas innecesarias.

Conclusión

Hoy se avanzó significativamente en la estructuración profesional del Custom Post Type Aplicaciones, integrando todos los bloques/secciones requeridos por tu landing. Quedaron listos:

- Los campos para el backend,
- Los llamados para el frontend,
- Las recomendaciones CSS para el resultado visual,
- Y la arquitectura de guardado centralizado para todos los metaboxes.

DETALLE DE IMPLEMENTACIÓN CPT “APLICACIONES”

1. Registro del Custom Post Type

- El archivo `cpt-aplicaciones.php` registra el post type `aplicacion`.
 - Soporta: **Título, editor, imagen destacada, extracto**.
 - Se habilita el editor de bloques (Gutenberg) y se bloquean los bloques para mantener controlado el contenido.
 - Se personaliza el menú de WordPress y el slug (`aplicacion`).
-

2. Estructura de Campos Personalizados y Secciones

Para cada sección del diseño de página de aplicación, se integraron **metaboxes** en el admin. Cada uno agrupa los campos (textos, imágenes, subtítulos, etc) relevantes para su bloque correspondiente en el frontend.

Metaboxes destacados:

- **Subtítulo Cabecera:** Input sencillo para el subtítulo bajo el título principal.
 - **Contenido 1:** Título, 2 imágenes, 2 textos, subtítulo.
 - **Gasex Utilizados:** Título general, 2 columnas (cada una con fondo, título interior, 2 subtítulos, 2 textos), texto final.
 - **Equipos y Tamaños:** Título, imagen fondo (background), 7 imágenes sobrepuestas, 4 subtítulos/textos, texto general y final.
 - **Sección Imágenes Aplicaciones:** Título y 4 imágenes + textos.
 - **Empleo Oxicorte:** Títulos, textos, 4 imágenes con textos, varios subtítulos y textos para recomendaciones y precauciones.
 - **Aplicaciones:** Título y 4 tarjetas (imagen, subtítulo, texto).
 - **Servicios Gasex:** Título, 4 servicios (imagen, título, texto), texto diferencial inferior.
 - **Imagen Aplicaciones:** Imagen de fondo y texto largo.
-

3. Implementación en el Admin (Metaboxes y Media Uploader)

- Todos los campos se agregan mediante metaboxes personalizados.
 - Para imágenes, se usa el **WordPress Media Uploader**, con preview en vivo y guardado del ID del adjunto.
 - Todos los campos se guardan en la función `save_post_aplicacion`.
 - El JavaScript para seleccionar imágenes se centralizó para no duplicar scripts ni recargar la página.
-

4. Guardado Centralizado

- Todos los campos se procesan y guardan en un único `add_action('save_post_aplicacion', ...)`.
 - Se utiliza `sanitize_text_field` para la mayoría de los campos y `sanitize_textarea_field` para los textos largos.
 - Esto permite que la adición de nuevos campos sea fácil y segura.
-

5. Llamados Dinámicos en el Frontend

- En el archivo `single-aplicacion.php`, los campos se imprimen dinámicamente:
 - **Texto:** `get_post_meta(get_the_ID(), 'campo', true)`
 - **Imágenes:** `wp_get_attachment_image_url($id, 'full')`
 - **Ejemplo:**

```
php
CopiarEditar
$img_id = get_post_meta(get_the_ID(), 'equipos_tamano_img1_id', true);
$img_url = $img_id ? wp_get_attachment_image_url($img_id, 'full') : '';
```

- Se usan clases y estructuras HTML listas para CSS personalizado (overlay, grid, etc).
 - Los textos largos usan `esc_html` y `nl2br` para saltos de línea.
-

6. Estilos CSS

- Se sugiere CSS con clases como `.gases-utilizados`, `.equipos-tamano`, `.imagenes-cabecera-aplicaciones`, `.aplicaciones`, etc., para permitir overlays, fondos transparentes, y responsividad.
 - Se evita la deformación de imágenes usando `background-size: cover` y/o `object-fit: cover`.
-

7. Usabilidad en el Admin

- Los campos más importantes aparecen arriba en el admin para facilitar la carga de datos al usuario.
 - Etiquetas y previews ayudan a visualizar cómo quedará el contenido antes de publicarlo.
 - Se ocultó el panel de bloques izquierdo del editor para que el usuario solo rellene los campos definidos.
-

8. Escalabilidad

- Puedes agregar nuevas secciones o campos solo replicando el patrón de metabox + guardado + llamado PHP.
 - El sistema está listo para ampliar funcionalidades, como relaciones con productos, sliders de imágenes, campos condicionales, etc.
-

Resumen Visual del Flujo

1. **El usuario crea/edita una Aplicación** desde el admin y rellena los campos personalizados en los metaboxes.
2. **Las imágenes** se cargan desde la galería con vista previa en vivo.
3. **El contenido se imprime en el frontend** con los llamados PHP, listo para el diseño visual.

¿Qué es el sistema CPT Aplicaciones y cómo funciona?

El sistema CPT Aplicaciones permite que puedas crear y editar páginas especiales dentro del sitio web, pensadas para mostrar diferentes “Aplicaciones” de tus productos o servicios de forma profesional, ordenada y muy visual.

¿Qué ventajas ofrece este sistema?

- **Fácil de usar:**
No necesitas saber de código ni diseño. Solo debes llenar los campos que aparecen al crear una nueva “Aplicación”, como si estuvieras llenando un formulario.
 - **Todo el contenido personalizado:**
Para cada Aplicación puedes ingresar títulos, textos, imágenes y más, en las secciones que verás en la página final.
Así, cada página de Aplicación es única y se adapta a la información que tú quieras mostrar.
 - **Previsualización de imágenes:**
Puedes subir imágenes desde la galería de WordPress y ver cómo quedarán directamente antes de publicar.
 - **Secciones claras y organizadas:**
El contenido está dividido en bloques como:
 - Cabecera (título y subtítulo)
 - Galerías de imágenes
 - Listados de equipos y servicios
 - Textos destacados y recomendaciones
 - Aplicaciones específicas
 - Y mucho más...
 - **Ahorra tiempo y errores:**
No puedes romper el diseño ni agregar elementos donde no corresponde. El sistema solo te deja completar los campos necesarios y mantiene la coherencia visual en todo el sitio.
-

¿Cómo lo uso paso a paso?

1. **Ingresar al panel de administración de WordPress.**
 2. **Ir al menú “Aplicaciones” y hacer clic en “Añadir nueva”.**
 3. **Completar los campos que aparecen:**
 - Título, subtítulos, imágenes, textos, etc. (verás una sección para cada bloque del diseño).
 4. **Subir imágenes con un solo clic, seleccionando desde tu computador o la biblioteca de medios.**
 5. **Guardar y/o publicar la Aplicación.**
 6. **¡Listo!** La nueva página de Aplicación aparecerá automáticamente en el sitio, con el diseño predefinido y toda la información que ingresaste.
-

¿Qué logras con esto?

- Mantienes una imagen profesional y homogénea en todas tus páginas.
- Puedes actualizar o crear nuevas Aplicaciones fácilmente, sin depender de un diseñador o programador.
- Todo el contenido es 100% autogestionable.

1. Usar un CPT jerárquico para categorías

- El CPT puede llamarse, por ejemplo, `categoria_custom` o similar.
- Al definirlo con `'hierarchical' => true`, el usuario podrá crear páginas tipo “categorías” con estructura padre-hijo, simulando categorías, subcategorías y sub-subcategorías.
- Esto facilita la gestión escalable y permite usar plantillas y bloques específicos para cada nivel.

2. Elegir categoría padre al crear la página

- En el editor clásico o bloque, el usuario seleccionará la categoría padre (o ninguna para categoría raíz).
- Esto creará automáticamente la estructura jerárquica.
- Se puede agregar un campo personalizado (meta field) para especificar el tipo o nivel de categoría si se requiere más control.

3. Plantillas automáticas basadas en el tipo/nivel

- Se pueden usar **plantillas PHP tradicionales** (`single-categoria_custom.php`) que detecten el nivel jerárquico con funciones como `get_post_ancestors()`.
- En Full Site Editing (FSE), se pueden crear plantillas y plantillas parciales (template parts) que se asignen automáticamente por la plantilla base o mediante condicionales en bloques dinámicos.
- Ejemplo:
 - Categoría raíz → plantilla “category-root”
 - Subcategoría → plantilla “category-child”
 - Sub-subcategoría → plantilla “category-grandchild”
- Esto otorga escalabilidad visual y funcional.

4. Bloques editables y contenido flexible

- Cada página de categoría puede editarse con el editor de bloques, permitiendo un diseño 100% personalizado y dinámico.
- Se pueden crear bloques reutilizables para mostrar productos, posts, sliders, banners, etc.
- Con ACF o metacampos nativos, se puede extender con datos extra que ayuden a mostrar contenido específico.

5. Integración con productos y posts

- Se pueden relacionar productos y posts con las categorías personalizadas mediante taxonomías o campos personalizados.
- Esto permitirá mostrar listados dinámicos filtrados dentro de la plantilla de la categoría.

6. Ventajas

- Máxima flexibilidad para administrar contenido jerárquico.
- Control de diseño muy granular.
- Escalable y adaptable a futuros requerimientos.
- Facilita experiencia visual coherente para usuario final.

7. Consideraciones

- Incrementa complejidad de desarrollo y mantenimiento.
- Requiere buena planificación para evitar estructuras huérfanas o rotas.
- El rendimiento puede ser afectado si hay muchas consultas anidadas (optimización necesaria).

Resumen rápido para implementar:

Paso	Descripción
1	Registrar CPT jerárquico para categorías personalizadas.
2	Permitir seleccionar categoría padre (jerarquía).
3	Crear plantillas específicas para cada nivel jerárquico.
4	Usar bloques dinámicos y/o ACF para contenido personalizado.
5	Relacionar productos y posts con categorías.
6	Integrar lógica para mostrar contenido relacionado dinámicamente.

dividir tu `assets.php` en módulos pequeños y ordenados

/wp-content/themes/tu-tema

functions.php ← carga los módulos

/inc

enqueue.php ← fuentes + CSS + JS (incluye newsletter.js y CSS de unsubscribe)

admin-ui.php ← scripts admin y CSS del editor

newsletter-ajax.php ← opción + handler AJAX + mail

unsubscribe-route.php ← helper + rewrite + template

/css/_unsubscribe.css ← estilos de la card de baja (ya lo tienes)

/js/newsletter.js ← frontend AJAX (ya lo tienes)