

# Composite

# Patrón Composite

---

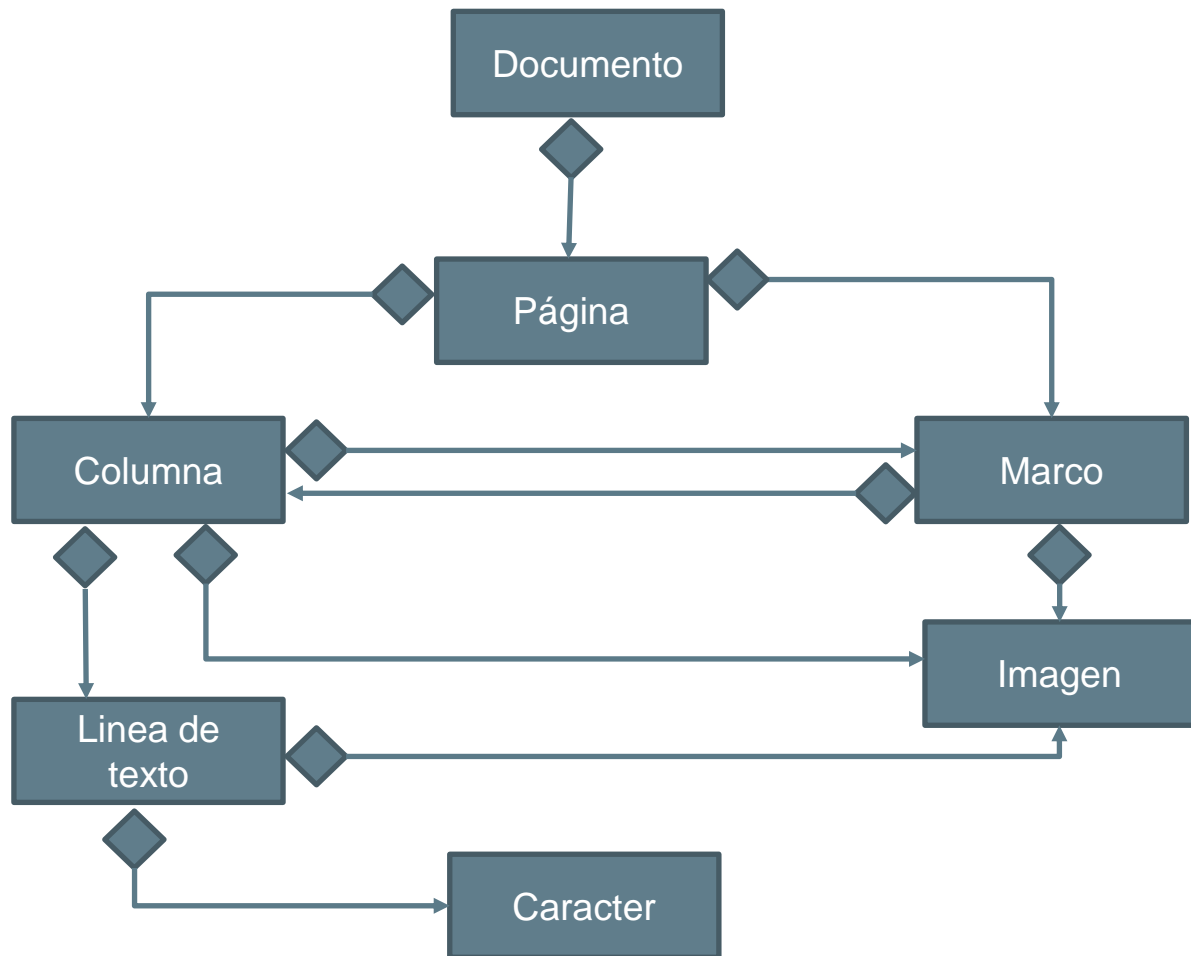
- **Intención:**
  - Componer objetos en jerarquías *todo-parte* y permitir a los clientes tratar objetos simples y compuestos de manera uniforme
- **Ventajas:**
  - Permite tratamiento uniforme de objetos simples y complejos así como composiciones recursivas
  - Simplifica el código de los clientes, que sólo usan una interfaz
  - Facilita añadir nuevos componentes sin afectar a los clientes
- **Inconvenientes:**
  - Es difícil restringir los tipos de los hijos
  - Las operaciones de gestión de hijos en los objetos simples pueden presentar problemas: seguridad frente a flexibilidad

# Patrón Composite

Un **Documento** está formado por varias **páginas**, las cuales están formadas por **columnas** que contienen **líneas de texto**, formadas por **caracteres**.

Las **columnas** y las **páginas** pueden contener **marcos**. Los **marcos** pueden contener **columnas**.

Las **columnas**, **marcos** y **líneas de texto** pueden contener **imágenes**.

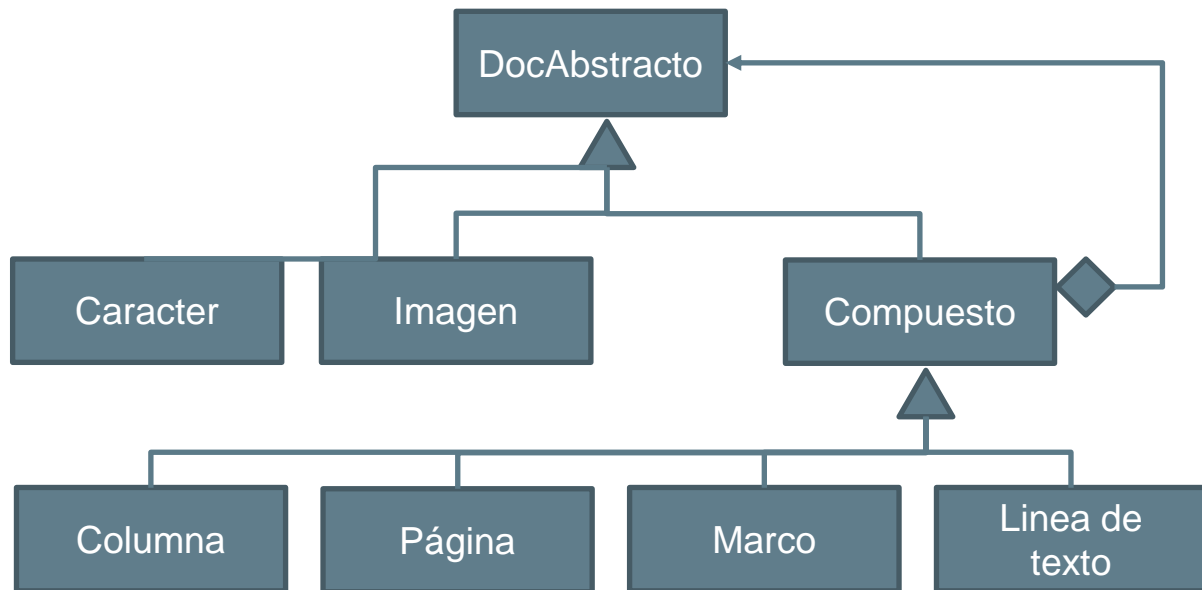


# Patrón Composite

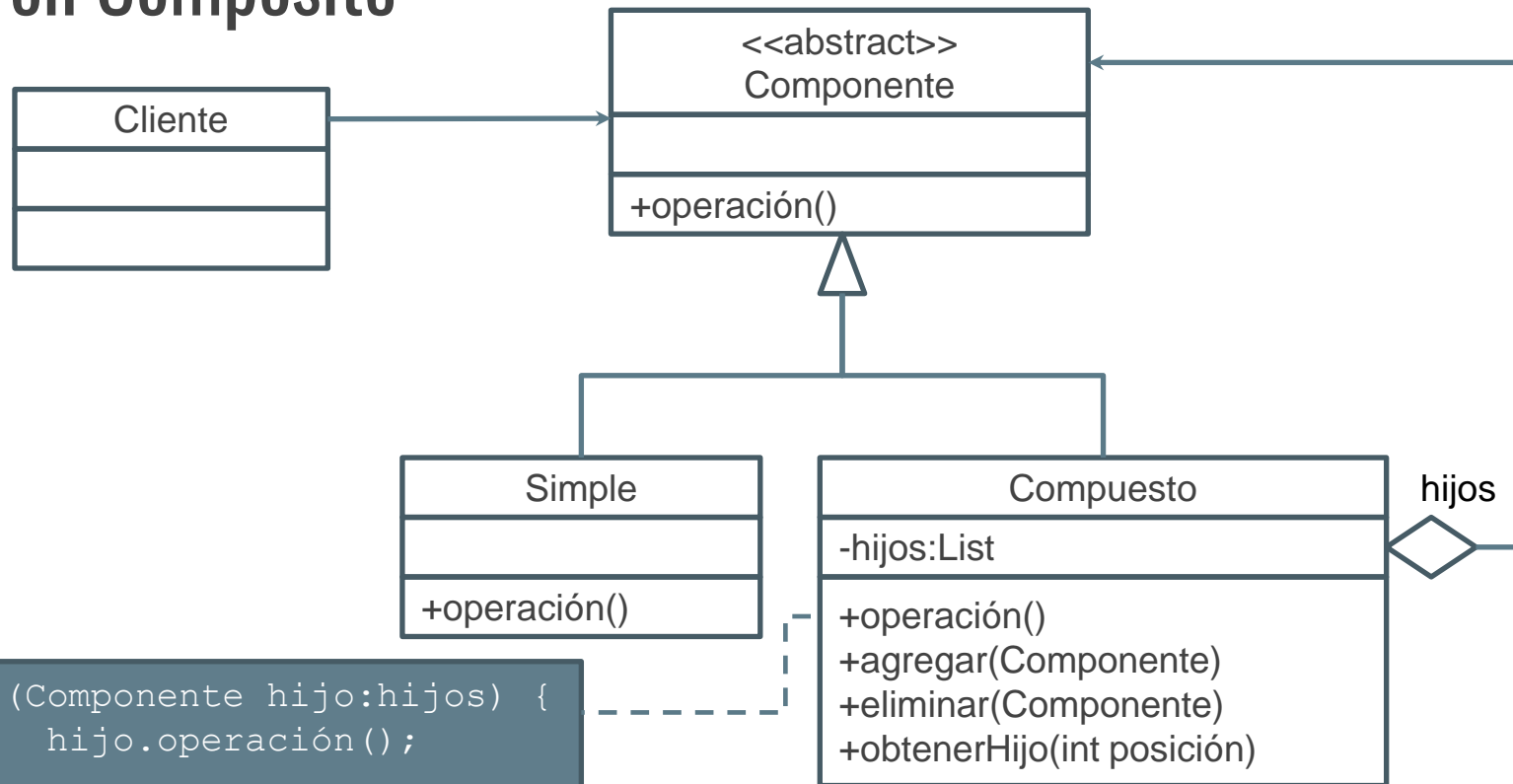
Un **Documento** está formado por varias **páginas**, las cuales están formadas por **columnas** que contienen **líneas de texto**, formadas por **caracteres**.

Las **columnas** y las **páginas** pueden contener **marcos**. Los **marcos** pueden contener **columnas**.

Las **columnas**, **marcos** y **líneas de texto** pueden contener **imágenes**

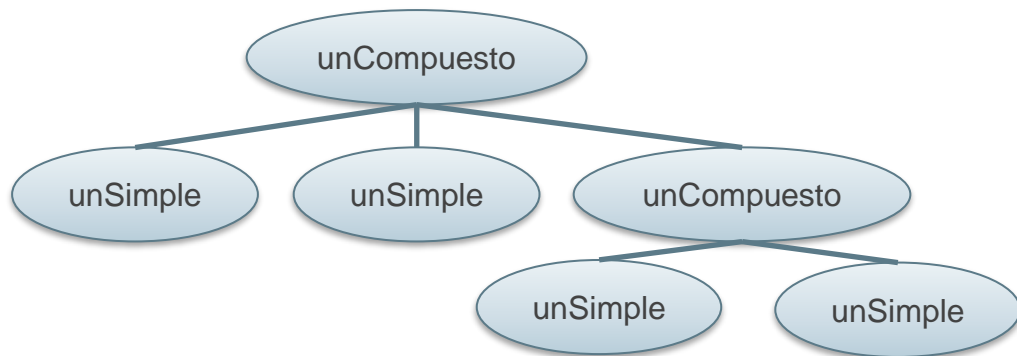


# Patrón Composite



```
for (Componente hijo:hijos) {  
    hijo.operación();  
}
```

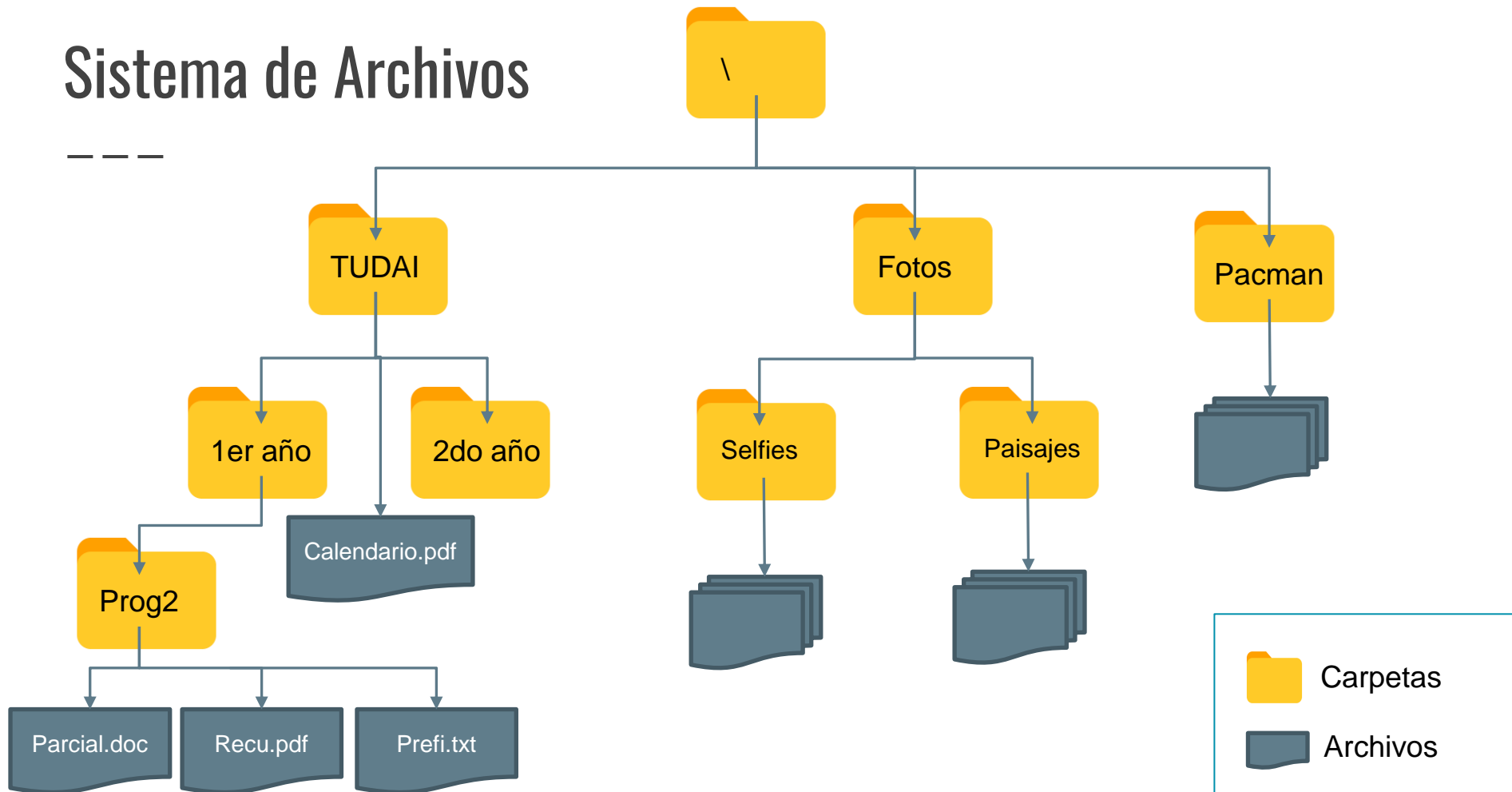
# Patrón Composite



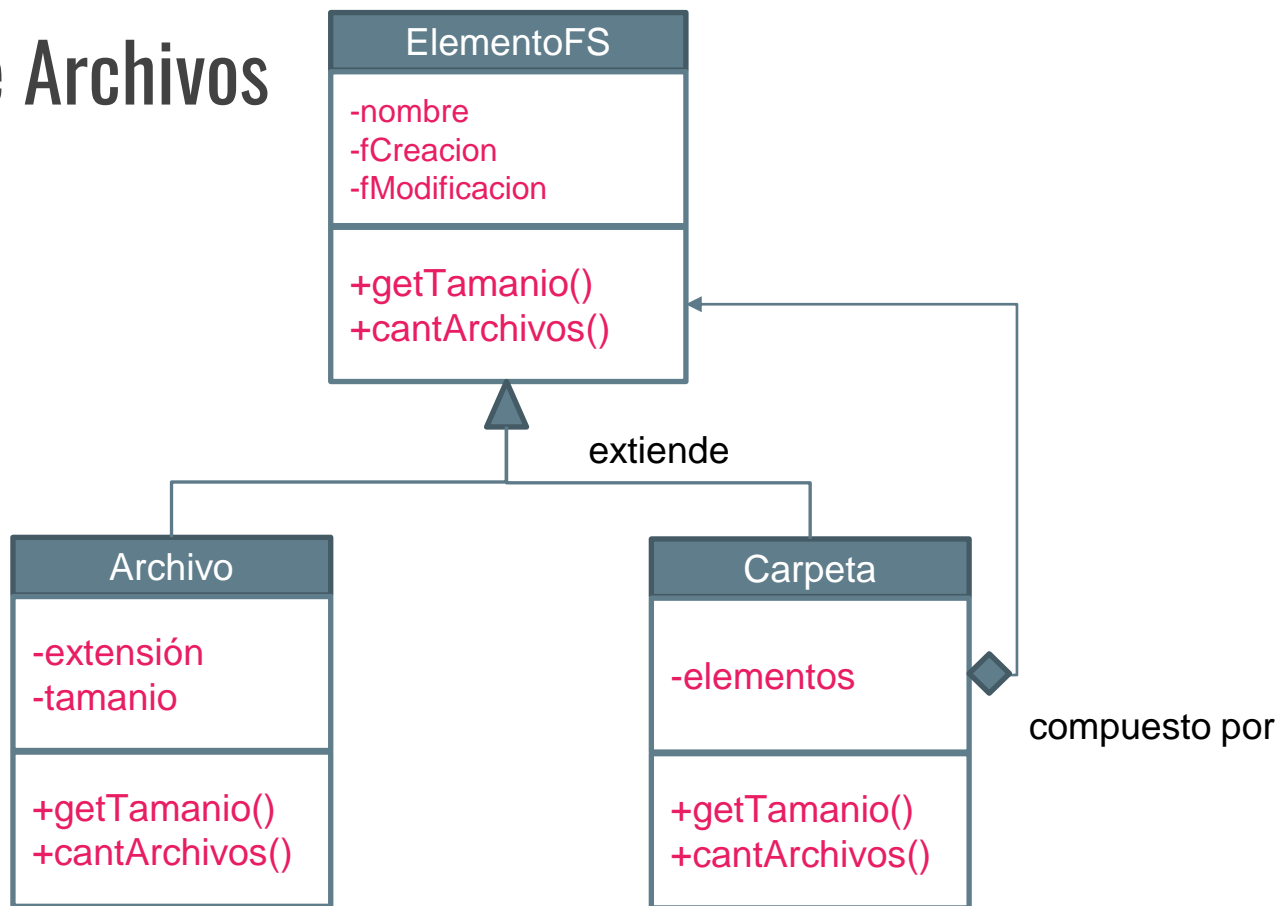
- Participantes:

- **Componente:** Declara una clase abstracta para la composición de objetos
- **Simple:** Representa los objetos de la composición que no tienen hijos e implementa sus operaciones.
- **Compuesto:** implementa las operaciones para los componentes con hijos y almacena a los hijos
- **Cliente:** utiliza objetos de la composición mediante la interface de Componente

# Sistema de Archivos



# Sistema de Archivos





# Patrón Composite

— — —

- Ejemplos:
  - File System, formado por carpetas y archivos
  - Documento, formado por páginas que contienen párrafos, imágenes, etc.

# Patrón Composite - Ejercicio

---

Una empresa multinacional se organiza en sucursales regionales (ej suc. regional Argentina, España, etc.) y sus sucursales regionales se componen de grupos de trabajo ubicados en distintas ciudades dentro de la sucursal regional, y cada uno de ellos se compone a su vez por cierto número de empleados. Cada uno de los empleados posee una especialidad y un sueldo.

# Patrón Composite - Ejercicio

---

Se debe responder a los siguientes servicios:

- Determinar la cantidad de empleados de la empresa, sucursal, o grupo que tienen una determinada especialidad.
- Obtener la especialidad de un empleado, grupo, sucursal y empresa. La especialidad de los grupos, sucursales y empresas se determinan como la especialidad que es mayoría entre sus miembros
- Obtener todos los empleados de una empresa, sucursal o grupo que tienen una determinada especialidad.