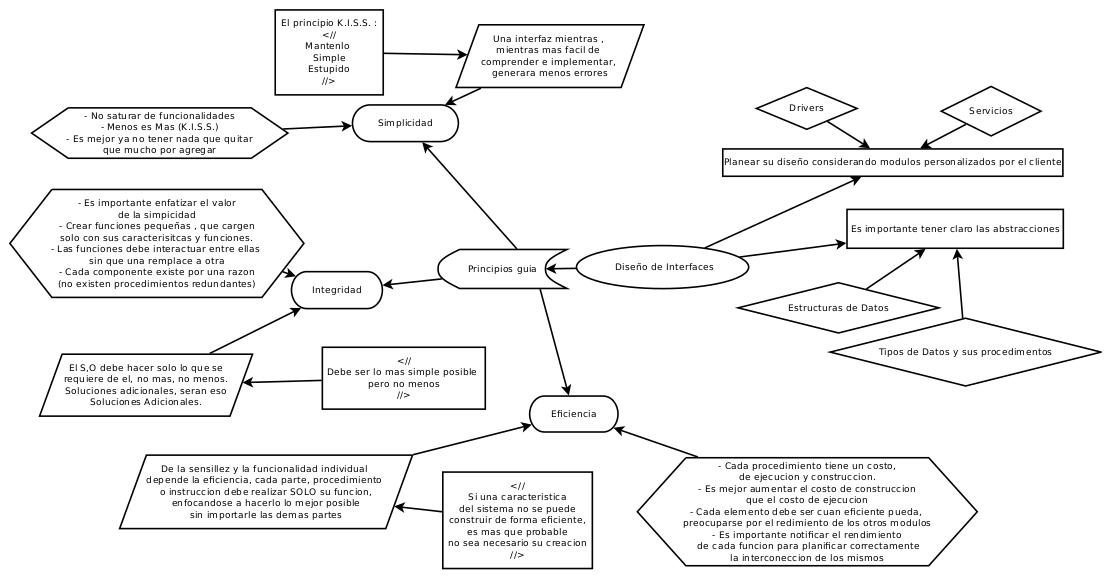
Integrantes :

* Medina Carbajal , Diego
* Medina Lopez, Jahir
* Marquina Gonzales, Alessandra
* Mendoza Campos , Renato

1. **Crear un mapa conceptual del tema.**

1. **Explicar cada principio.**

* **Simplicidad :**

Este principio busca garantizar que durante las etapas de desarrollo se planee cumplir con las funciones mínimas necesarias, de forma que se reduzca lo máximo posible un debug futuro.

***Ejm. Visual***

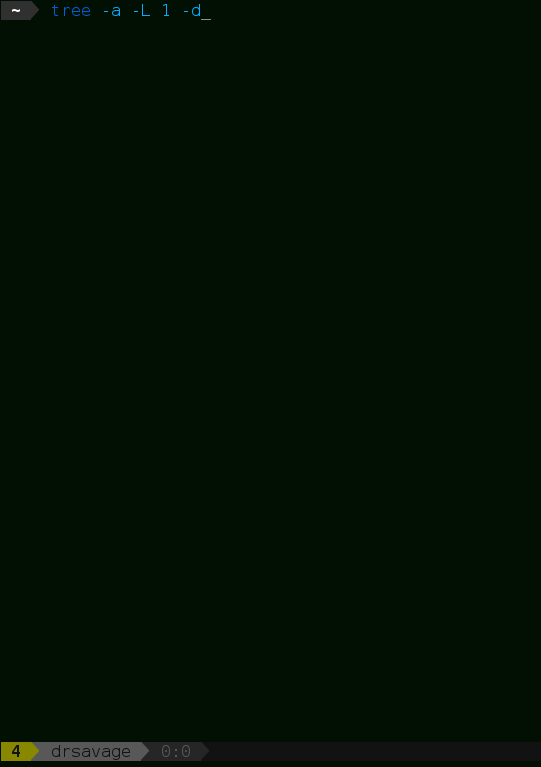
*En este ejemplo se ejecuta el comando* **tree** en linux, este comando genera una vista de árbol de todos el contenido de un directorio, con solo 3 parámetros :

-a : All , muestra todo el contenido

-L: Level, identificar hasta qué profundidad se debe generar el árbol

-d: Dir, especifica solo se debe mostrar los directorios.

Se consigue generar un listado de todos las carpetas que existen en el directorio actual , incluidas las carpetas ocultas.

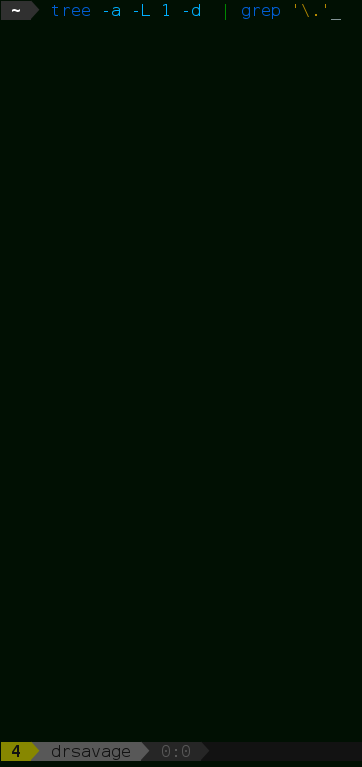
******

* **Integridad :**

Cada componente del sistema operativo debe cumplir su función, de forma que al interactuar con las diferentes partes o módulos del sistema , garantice un correcto funcionamiento.

***Ejm. Visual***

*Se conecta dos procesos :* **tree y grep** *, estos micro procedimientos permiten, el primero generar un árbol del contenido de una carpeta, mientras el segundo filtra la salida según un formato de verificación, este caso ,* **grep** *busca todas las carpetas que tienen un* **“.”** *al comienzo de su nombre (carpetas ocultas)*

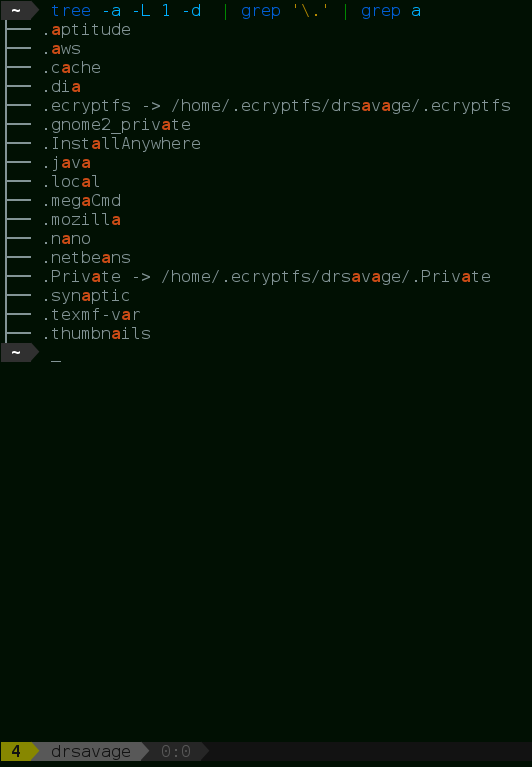
**

* **Eficiencia :**

Tal cual se intuye por simplicidad e integridad, la eficiencia debe garantizar que cada módulo , de forma independiente se ejecute lo más rápido posible sin importar la eficiencia de los demás.

***Ejm. Visual***

*En esta parte final se juntan 3 procedimientos : tree, grep y nuevamente grep, siendo este último el encargado del filtrado final:* **Buscar todas las carpetas dentro un directorio (solo el primer nivel) que están ocultas (comienzan con .) y contengan una letra “a” en su nombre***. Estos 3 módulos simples, que individualmente se ejecutan rápidamente, construyeron un procedimiento complejo en tiempo record. La construcción de la operación no tomó mucho tiempo y retorno el resultado deseado correctamente.*

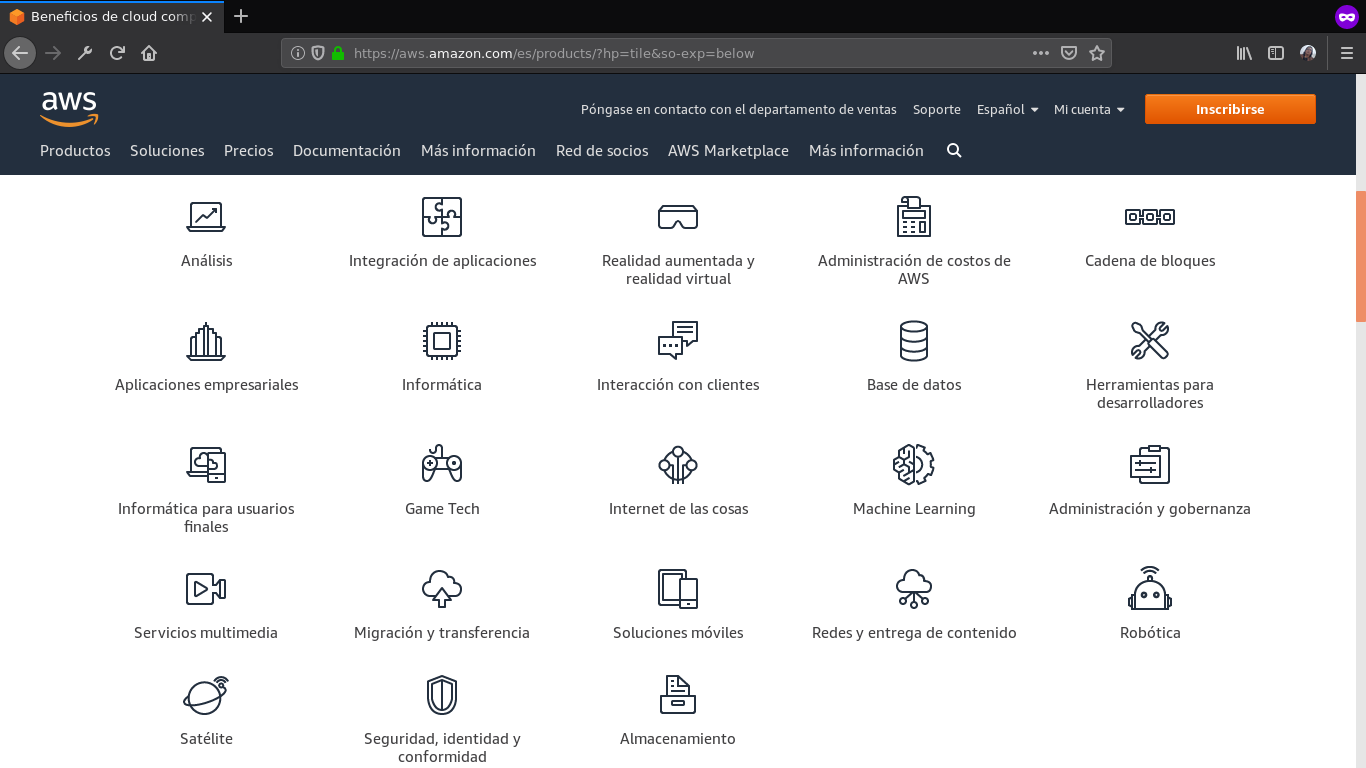
******

1. **Explicar 3 ejemplos de la aplicación o uso de estos principios.**

* **Ejemplo 1:**

La construcción los servicios AWS (amazon cloud) , cada “producto” , aunque reside sobre una misma estructura de servidores, está planificado de tal forma que puedan comunicarse entre sí como si de entidades independientes se trataran, esto implica, que de fallar uno, no se lleva por delante los demás servicios.

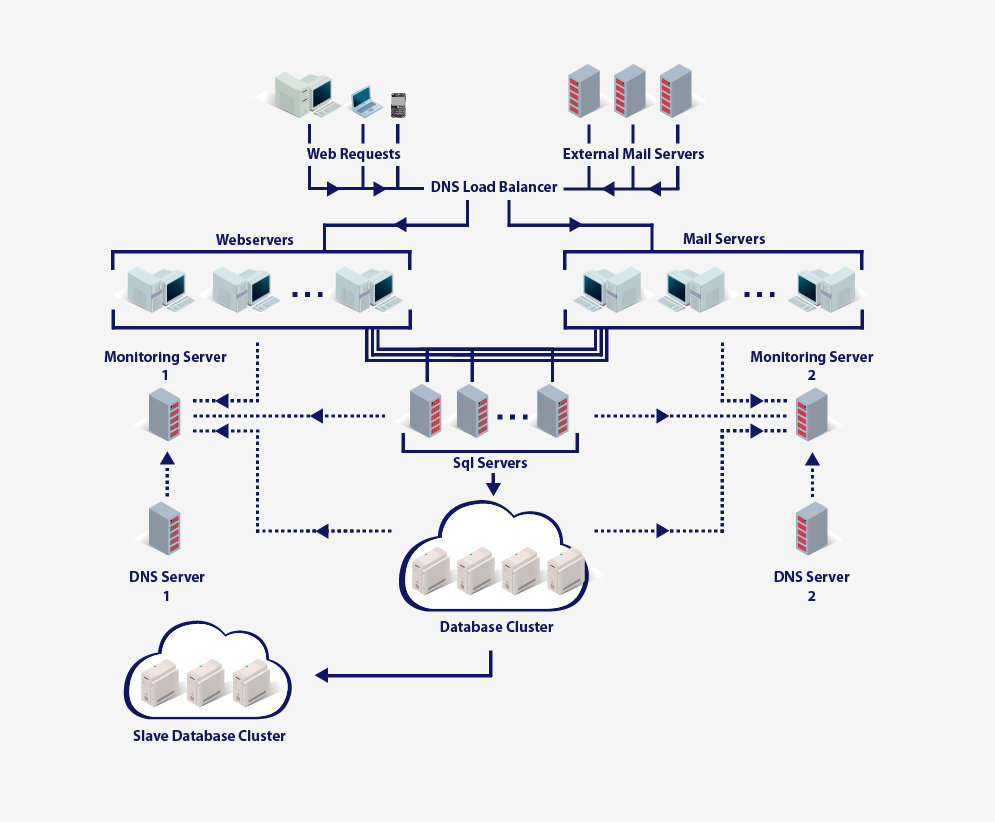
Cada servicio garantiza su propia eficiencia y calidad de servicio.



* **Ejemplo 2:**

La implementación de un servidor de almacenamiento de correo electrónico, si bien es cierto, esto reside en un S.O. y no es como tal uno, su construcción hace uso de los 3 principios, pues debe garantizar una alta confiabilidad.

Cada modulo (recepcion de email, edición, filtrado, etc) debe operar sin afectar al otro pero sí apoyando y constituyendo en conjunto un servicio confiable.



* **Ejemplo 3:**

El sistema operativo **Free DOS** , este sistema operativo es una variante de la arquitectura MS-DOS , el cual busca la ejecución de programas asociados a esta arquitectura pero sin ser privativo (como el caso de windows, creador de la arquitectura MS-DOS).

En la creación de este sistema operativo fue importante trabajar usando los 3 principios, pues no solo se debe garantizar su funcionamiento sino también la interoperabilidad con otro S.O. : Microsoft Windows, pues este S.O. se presenta como una alternativa libre a Windows.

