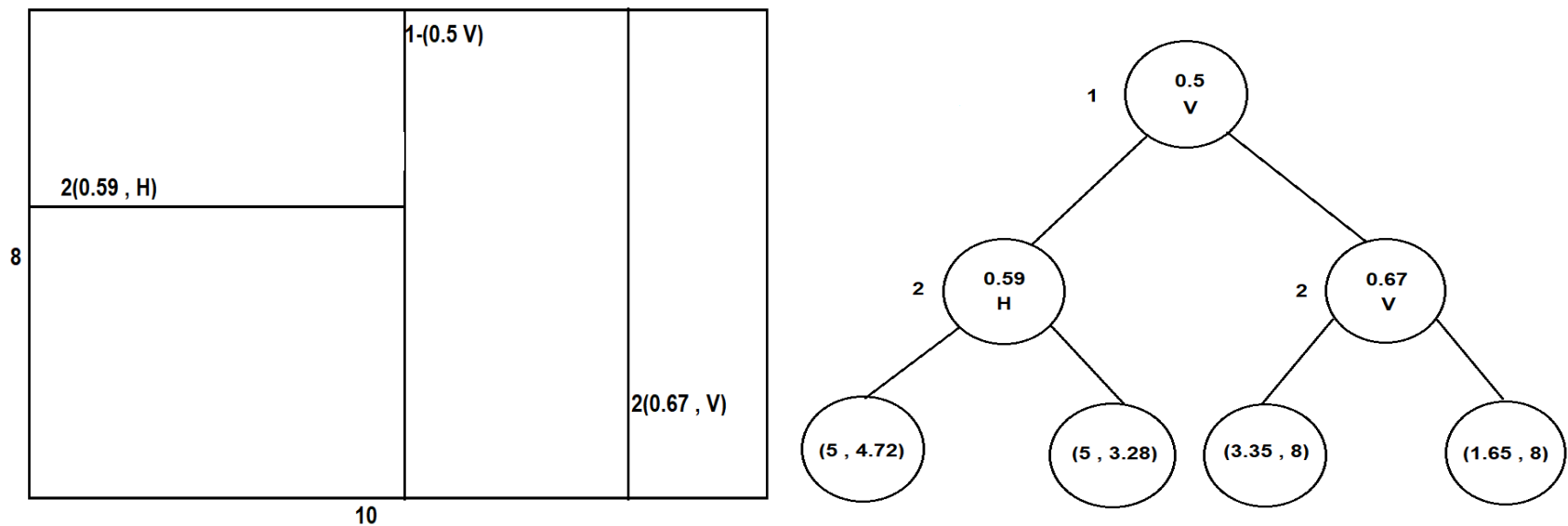


# Estructuras Árbol Guillotina y Árbol de Ordenamiento

En este documento tratare de explicar las estructuras y métodos usados para el desarrollo del Algoritmo Cúmulo de Partículas.

## Árbol Guillotina:

El árbol Guillotina es un árbol binario, este contara con nodos intermedio y hoja, este realiza cortes en el contenedor principal ( cortes que pueden ser horizontales o verticales ) y generar subcontenedores de menor tamaño cada nodo intermedio contiene un valor entre 0 y 1 ( un porcentaje de corte ) y también el tipo de corte tanto Horizontal como Vertical. Los nodos hoja, contienen la dimencion del nuevo contenedor que es resultado de realizar cortes y este se usara para ingresar las cajas a ordenar.



En las anteriores figuras podemos ver un contenedor de ( 10 , 8 ), al cual se le realizaron cortes comenzando por 1.(0.5 V) el cual genera un corte de 0.5 Vertical, este genera 2 subcontenedores de ancho 5 y altura 8, esto se puede ver en el árbol guillotina como el nodo de nivel 1. luego los siguientes cortes fueron sobre los 2 nuevos subcontenedores dejando 4 nuevos contenedores de tamaño (5,4.72) , (5,3.28) , (3.35,8) , (1.65,8) los cuales son los nodos hojas en el Árbol de corte guillotina, estos serán las que almacenen las cajas que queremos ordenar

El uso de esta estructura se debe a la facilidad de manipular los cortes que generamos sobre el contenedor principal, al cambiar valores de los nodos intermedios, tanto su porcentaje como su tipo de corte(V o H), obtendremos nuevos subcontenedores de diferentes dimensiones, entonces el problema se convierte en algo mas simple, encontrando los porcentajes y tipos de cortes que encajen mejor con los grupos de cajas a ordenar, esta estructura sera usada tanto en el Algoritmos Genéticos como en Cúmulo de Partículas.

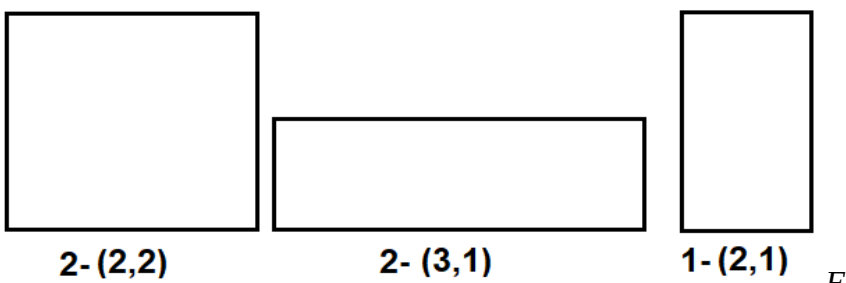
## Árbol de Ordenamiento Simple:

Esta estructura sera usada para representar los espacios disponibles dentro de un contenedor, ingresar ,si es posible, en esos espacios disponibles cajas a ordenar y ver que espacios se quedaran disponibles y ser espacios desperdiciado.

El algoritmo sigue unos pasos simples , sabiendo que las cajas a ordenar se pueden rotar 90 grados, entonces:

- 1: Ordenamos las cajas a ingresar de mayor a menor área
- 2: Dado un contenedor tratamos de ingresar la caja de mayor área en esta , si no ingresa en su estado normal, entonces rotamos 90 grados y verificamos
- 3: En el caso de que esta caja no entre probamos con la siguiente caja, hasta encontrar una que encaje , en el caso de que no encaje ninguna este contenedor queda como espacio disponible.
- 4: Suponiendo de que se ingreso una caja en el contenedor, este generara 2 subcontenedores nuevos uno de arriba y otro de alado , comenzamos con el subcontenedor de alado y luego de arriba, realizamos este proceso de nuevo desde el paso1.

Siguiendo estos pasos ordenamos las cajas dentro del contenedor mayor



igura 1: Cajas a Ingresar ordenadas de mayor a menor área

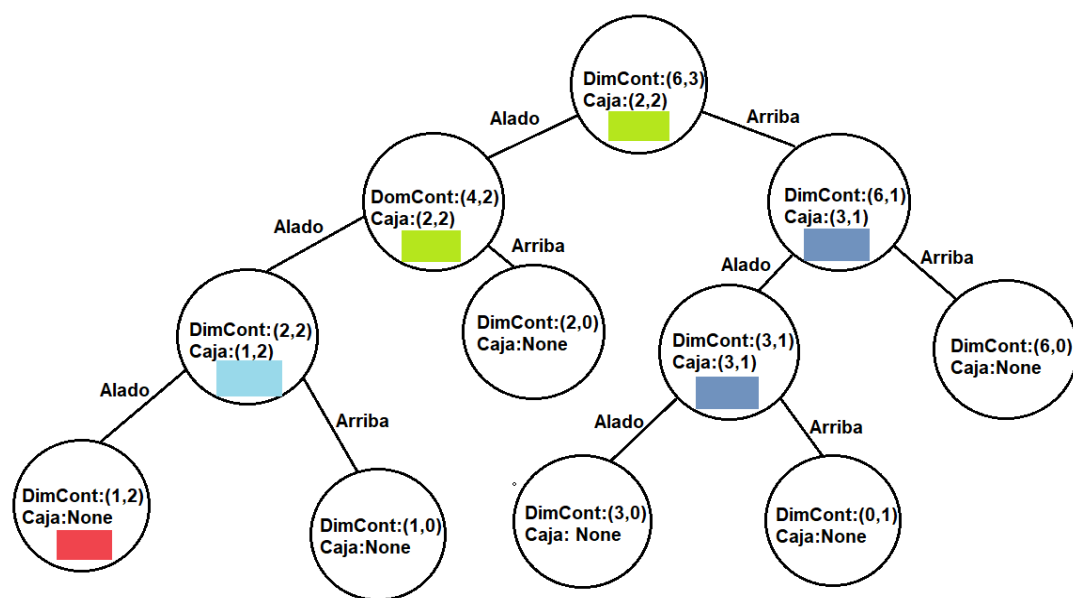


Figura 2: Estructura Árbol de Ordenamiento luego de ordenar cajas en Contenedor

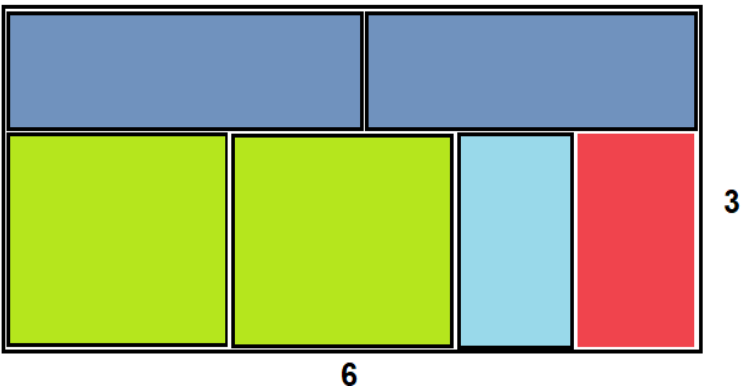


Figura 3: Contenedor con cajas ingresadas usando algoritmo Árbol Ordenamiento

En el árbol de Ordenamiento, Figura 2 , podemos ver como los nodos intermedios contiene un atributo “Caja” el cual contiene la caja, que ingresamos, y 2 nodos hijos , Arriba y Alado , los cuales representan los subcontenedores nuevos generados al ingresar una Caja en el contenedor. Los nodos hoja representan los espacios disponibles. Este árbol representa el Contenedor con las cajas ingresadas como en la Figura 3, y los espacios disponibles como contenedores de color Rojo