

Taller 1: Recursión

Caso Base y Caso Recursivo

Caso base: El caso base de la recursión ocurre cuando el nodo actual no tiene hijos válidos para continuar el camino (ya sea porque no tiene hijos o todos los hijos están prohibidos). En este caso, se imprime el camino recorrido hasta el momento.

```
if (!tiene_hijo_valido) {  
    for (int i = 0; i <= nivel; i++) {  
        printf("%s ", camino[i]);  
    }  
    printf("\n");  
}
```

Caso recursivo: En cada llamada, se agrega el nombre del nodo actual al camino y se recorre recursivamente cada hijo que no esté prohibido.

```
int tiene_hijo_valido = 0;  
for (int i = 0; i < nodo->num_hijos; i++) {  
    if (!es_prohibido(nodo, nodo->hijos[i]->nombre)) {  
        tiene_hijo_valido = 1;  
        caminos_validos(nodo->hijos[i], camino, nivel + 1);  
    }  
}
```

es_prohibido: Es la función que se utiliza en el caso recursivo para evaluar si el nombre del hijo actual está en la lista de prohibidos del nodo padre. Si no lo está, se permite continuar el recorrido hacia ese hijo de esta forma, el algoritmo construye e imprime todos los caminos válidos desde el nodo raíz hasta las hojas, evitando seguir caminos con restricciones.

Reflexión personal

Durante la creación del código, una de las dificultades fue comprender cómo manejar la relación entre los nodos padre e hijo, y cómo incluir las restricciones de los prohibidos entre ciertos nodos.

Otro reto fue estructurar correctamente la recursividad. Al principio, fue confuso ya que no manejo muy bien C, pero con la ayuda de YouTube, herramientas de aprendizaje y el conocimiento previo gracias a otros lenguajes, aprendí como hacer bien un código o al menos que funcionara.

En cuanto a mi aprendizaje, reforcé mi entendimiento sobre cómo funciona el recorrido en estructuras de árbol y cómo gestionar arreglos de cadenas en C.

Como podría mejorar, sería seguir aprendiendo C y también revisar si el código no tiene complejidades muy grandes utilizando Big O.