

---

### Pseudocódigo para el recorrido en profundidad de un grafo

---

Los tipos utilizados son los siguientes:

- Para el grafo:

```
grafo = tabla [1..n] de vértice;  {n = número de vértices del grafo}
vértice = registro
    vert: TipoVertice;
    listaAdy: ptNodo;      {puntero a la lista de vértices adyacentes}
fregistro
ptNodo = puntero a nodo;
nodo = registro
    vertAdy: TipoVertice;
    sig: ptNodo;
fregistro
```

- La estructura auxiliar *pila* puede ser del tipo *ptNodo*
- Mantendremos 2 tablas de tamaño *n*: una para los nodos visitados y otra para los nodos que van formando el recorrido:

```
visitados = tabla [1..n] de booleanos
recorrido = tabla [1..n] de tipoVertice
```

```

accion recorrido_profundidad (G:grafo; verticeOrigen:tipoVertice);
var
    i, ind: entero; P: Pila; elemento: tipoVertice; aux: ptNodo;
fvar

inicio
    para i=1 hasta n hacer
        visitados[i]:= falso;
    fpara

    creaVacia(P);
    apilar(P, verticeOrigen);
    ind:= indice(verticeOrigen);
    visitados[ind]:= verdad;
    i:=1;
    repetir
        elemento:= cima(P);
        desapilar(P);
        recorrido[i]:= elemento;
        i:= i+1;
        ind:= indice(elemento);
        aux:= G[ind].listaAdy;
        mientras aux <> nulo hacer
            si visitados [indice(aux^.vertAdy)]= falso entonces
                apilar(P, aux^.vertAdy);
                visitados[indice(aux^.vertAdy)]:= verdad;
            fsi
                aux:= aux^.sig;
        fmientras
    hasta esvacia(P);
faccion

```