

Module Digrafo (Digrafo(G), vértices, arcos, nvertices, narcos, sucesores, antecesores, gradosal, gradoent, depthFirstSearch)

Where

Data Digrafo v = G[v] (v->[v])

Vértices::Digrafo v -> [v]

Vertices (G vs _) = vs

Arcos::Digrafo v -> [(v,v)]

Arcos (G vs suc) = [(x,y) | x <- vs, y <- suc x]

Nvertices::Digrafo v -> Int

Nvertices = length.vertices

Narcos::Digrafo v -> int

Narcos = length.arcos

Sucesores::Digrafo v -> v -> [v]

Sucesores ::Digrafo(G vs suc) x

| elem x vs = suc x

| otherwise = error "vértice no encontrado"

Antecesores::Digrafo v -> v -> [v]

Antecesores (G us suc) y

| elem y vs = [x | x <- vs, elem y (suc x)]

| otherwise = error "vértice no encontrado"

Gradosal::Digrafo v -> v -> Int

Gradosal (G vs suc) x

| elem x vs = length (suc x)

| otherwise = error "vértice no encontrado"

Gradoent::Digrafo v -> v -> Int

Gradoent g@(G vs suc) x

| elem x vs = length(antecesoires g x)

| otherwise = error "vértice no encontrado"

depthFirstSearch::Digrafo v -> v -> [v]

depthFirstSearch (G vs suc) orig = reverse(dfs[orig][[]])

where

dfs[] vis = vis

dfs (c:cs) vis

| elem c vis = dfs cs vis

| otherwise = dfs ((suc c)++cs) (c:vis)