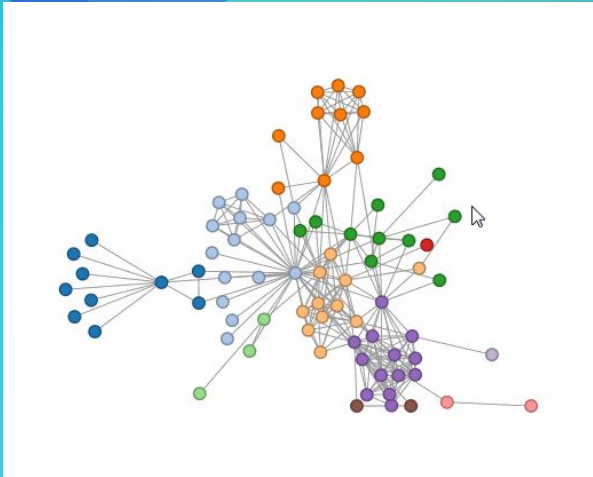


Ordenamiento Topológico-Puntos de articulación

Curso: Estructura de datos

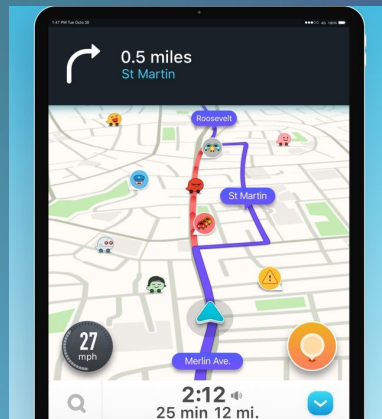
- Antúñez Palomino ,Kori Xiomara(3)
- Gómez Caverro, Mishell(3)
- Santiago Arapa, Naysha Solange (3)
- Pérez Fonseca, Juan Diego (3)



Descripción del aplicativo

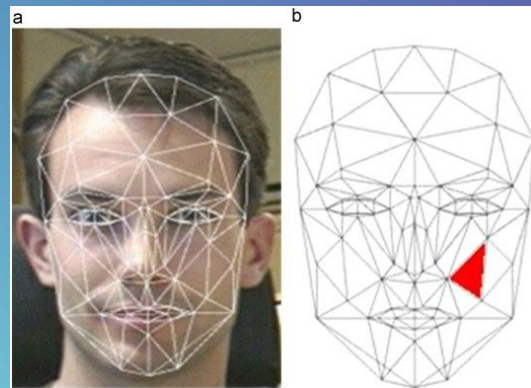
Motivación

Las aplicaciones de los grafos actualmente son muy requeridos en diferentes ámbitos. Desde conseguir la ruta mínima para llegar de una dirección a otra hasta organizar la malla curricular de una carrera profesional



Problema

Actualmente los avances tecnológicos nos demandan a desarrollar tanto en la genética, en aspectos de seguridad, una lógica de optimización, es por ello importante contar con un modelo abstracto, grafos, que nos permita realizar tal lógica



FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA - UNMSM

MAPA CURRICULAR DEL PLAN DE ESTUDIOS 2018 INGENIERIA DE SOFTWARE

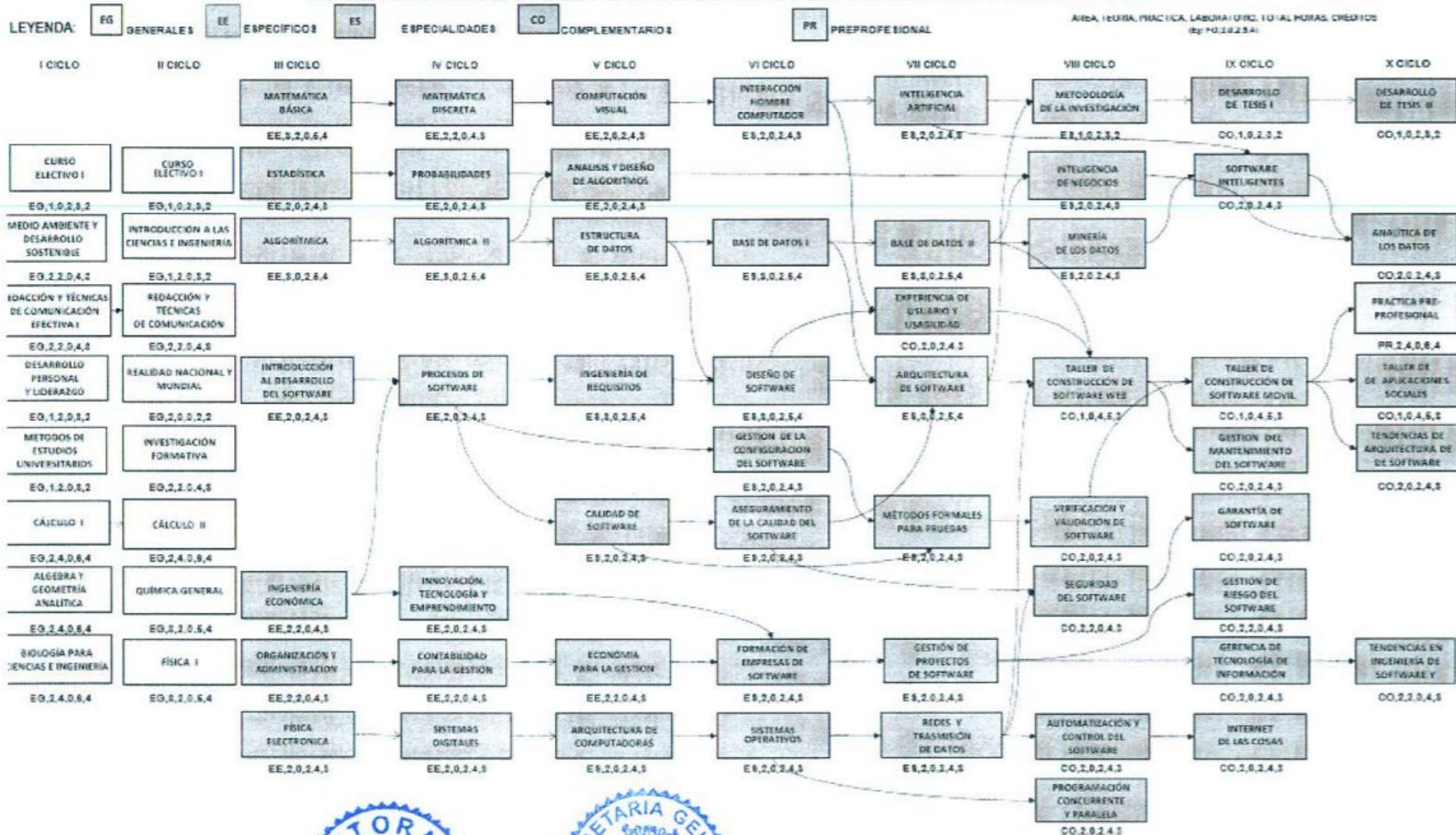
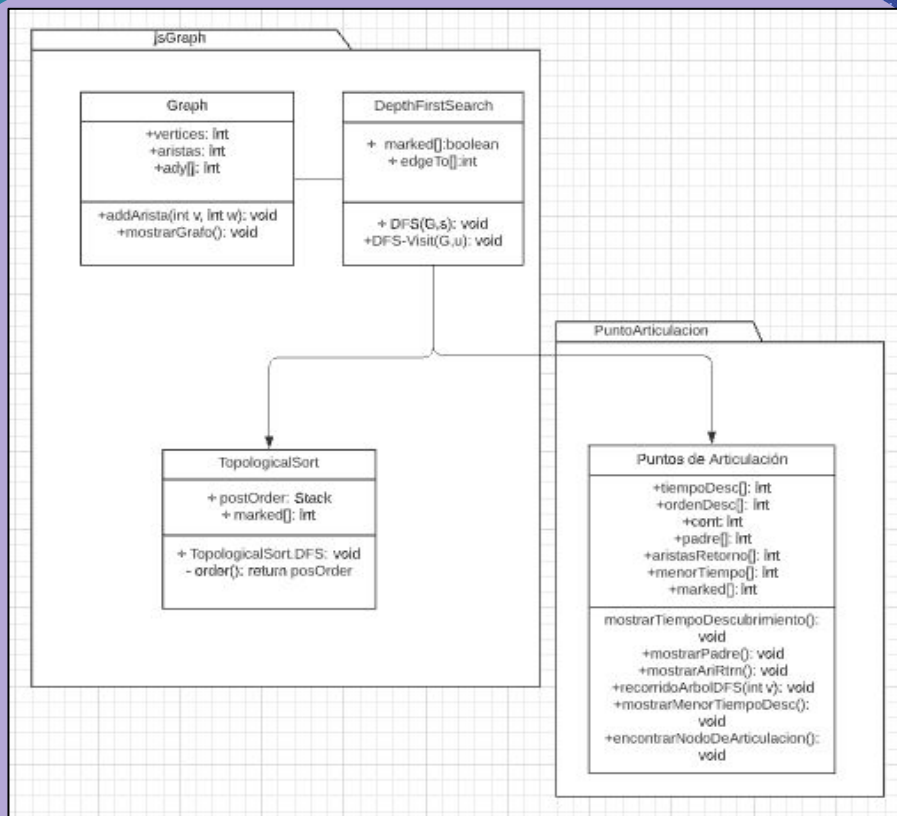


Diagrama de clases



Herramientas y técnicas usadas



Herramientas y técnicas usadas

Algoritmo Ordenamiento topológico DFS

```
DFS(G)
  PARA CADA vértice  $v \in V[G]$  HACER
    estado[v]  $\leftarrow$  NO_VISITADO
  FIN PARA
  PARA CADA vértice  $v \in V[G]$  HACER
    SI estado[v] = NO_VISITADO ENTONCES
      DFS_Visitar(G,v)
  FIN PARA
FIN

DFS_Visitar (G, u)
  estado[u]  $\leftarrow$  VISITADO
  PARA CADA  $v \in \text{adj}_v[u]$  HACER
    SI estado[v] = NO_VISITADO ENTONCES
      DFS_Visitar(G,v)

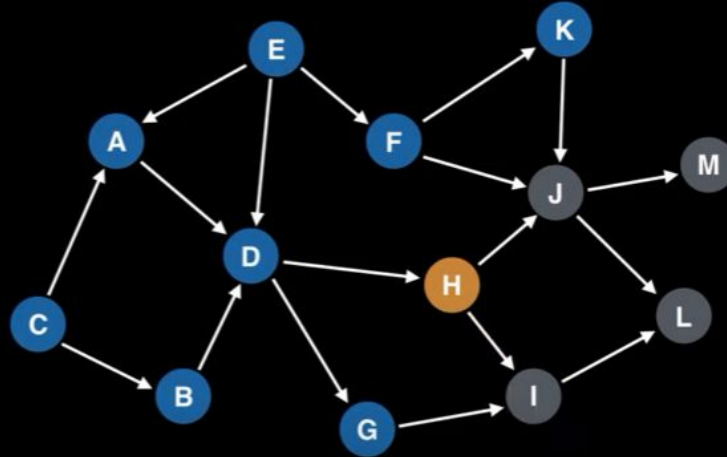
  FIN SI
  FIN PARA
  postOrder(v)
FIN
```

Herramientas y técnicas usadas

Algoritmo Ordenamiento topológico DFS

Topological Sort Algorithm

DFS recursion
call stack:
Node H



Topological ordering:

----- I J L M

Algoritmo Puntos de Articulación

Método DFS (nodo)

nodoVisitado[nodo] = true;

Si (nodo tiene adyacentes) entonces

 tiempoDescubrimiento[nodo] = cont

 ordenDescubrimiento[cont] = nodo

 cont++

Fin Si

Para w desde 1 hasta cantidadAdyacentes hacer

 Si(adyacentes[w] no fue visitado) entonces

 ady = adyacentes[w]

 padre[ady] = nodo

 DFS(ady)

 Si no

 Si(ady != padre[nodo]) entonces

 adyacente es arista de retorno de nodo

 Fin Si

Fin Si

Fin Para

Fin DFS

Algoritmo Puntos de Articulación

Método recorridoDelArbloDFS 0

Para i desde 0 hasta numeroVertices hacer

 entero nActual = ordenDescubri[i]

 menorTiempoDescubri[nActual] = i

 Si (nActual tiene aristas de retorno) entonces

 entero j = 1

 Mientras (aristasRetorno[nActual][j] != vacío) hacer

 entero nRtrn = aristasRetorno[nActual][j]

 Si (menorTiempoDescubri[nActual] > tiempoDescubri[nRtrn])

 menorTiempoDescubri[nActual] = tiempoDescubri[nRtrn]

 entero x = nActual

 Mientras (padre[x] != nRtrn)

 entero pa = padre[x]

 menorTiempoDescubri[pa] = menorTiempoDescubri[x]

 x = pa

 Fin Mientras

 Fin Si

 j++

 Fin Mientras

Fin Para

Fin Metodo

Algoritmo Puntos de Articulación

Método encontrarNodo0

Para i desde 2 hasta v-1 hacer

 hijo <- ordenDescubri[i]

 padre <- padre[hijo]

 Si menorTiempoDescubri[hijo] >= tiempoDescubri[padre]

 entonces

 nodosDeArticulacion[padre] <- verdadero

 Fin Si

Fin Para

Fin Metodo

Presentación del aplicativo

Inicio

Ordenamiento Topológico

Puntos de articulación

Universidad Nacional Mayor de
San Marcos

Proyecto Estructura de datos

Ordenamiento Topológico y Puntos de Articulación



Presentación del aplicativo

Inicio

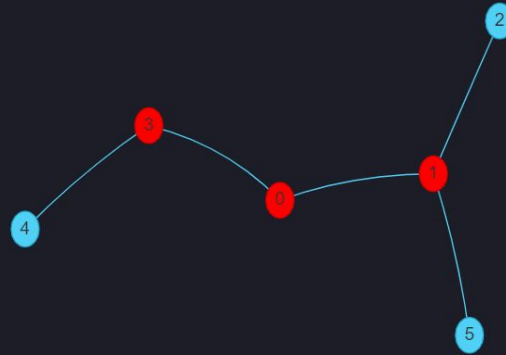
Puntos de articulación

Ordenamiento Topológico

Puntos de articulación

Todos los nodos ingresados deben estar conectados

Nodos Rojos: Puntos de articulación

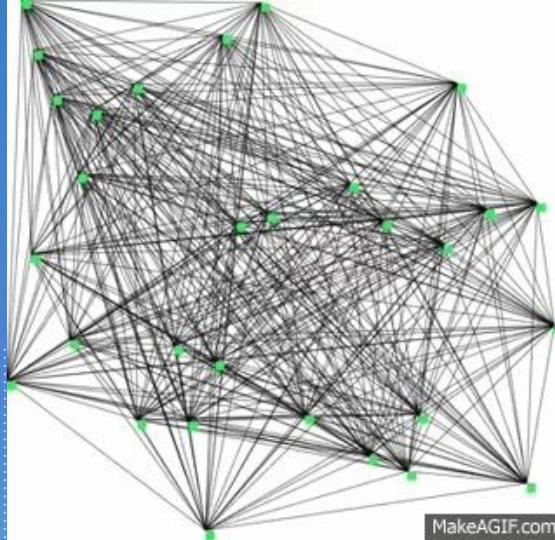


Tiempo de duracion: 12996 milisegundos

Pseudocódigo

Conclusiones

En conclusión se logró implementar el desarrollo de los grafos y el coloreado tanto de los vértices como las aristas , gracias al uso de la librería Vis.js que nos proporciona JavaScript.



Gracias

