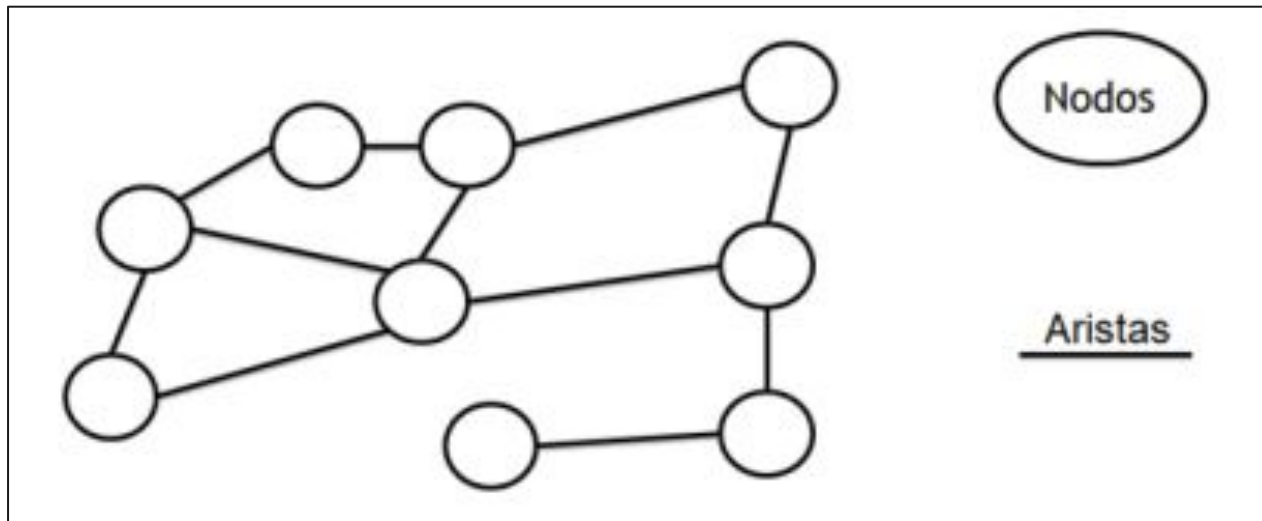




# Implementación del camino más corto en una red de servidores

Santiago García & Daniela León

# Definiciones





## Planteamiento del problema

En la empresa X hay un nuevo edificio con  $n$  servidores conectados por cables de red. Cada uno de los  $m$  cables conecta dos ordenadores y tiene una cierta latencia medida en segundos necesaria para enviar un archivo de 10mb a través de FTP.

Para los miembros de la junta directiva, un archivo de este peso debe poder pasarse en menos de 1 minuto.

# Diseño del grafo

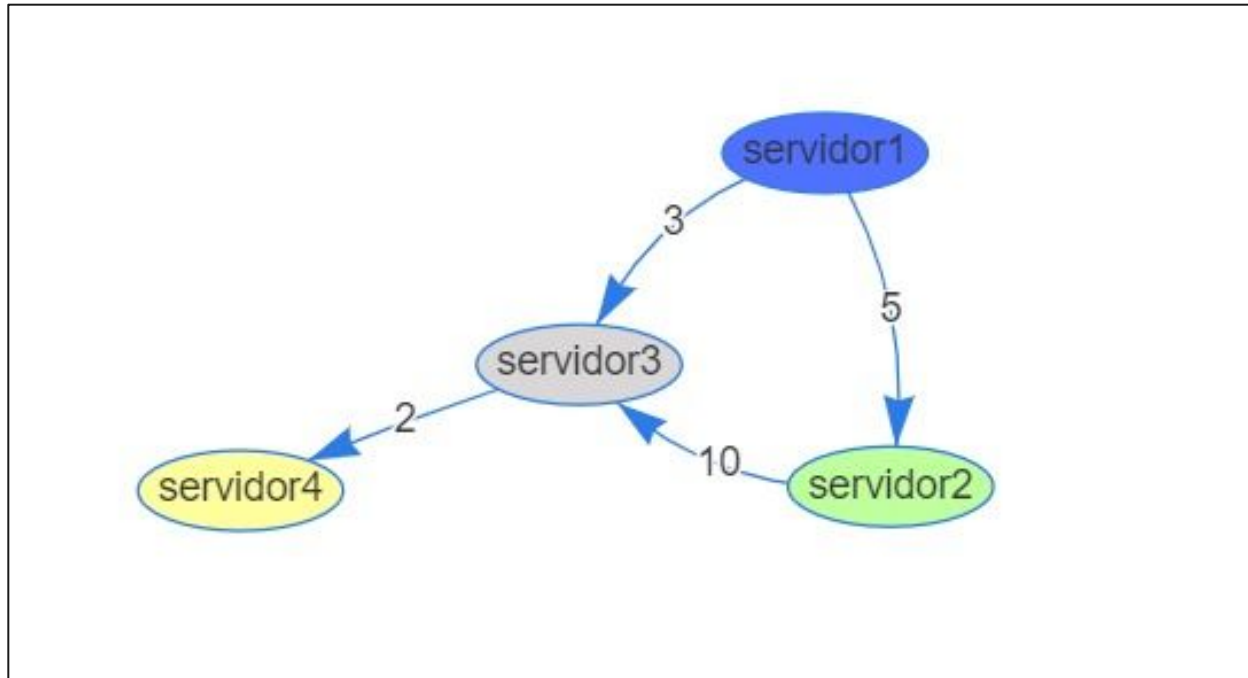
Para la implementación de los grafos utilizamos una aplicación desarrollada en PHP, esta nos permite agregar los nodos (servidores) y vértices (cables) con las etiquetas y peso deseados.

```
grafo.php
4 class Grafo
5 {
6
7     private $matrizA; // aristar [].[].[]
8     private $vectorV; // nodos A
9     private $dirigido; // -> vertice
10    private $cola = null; // cola de adyacentes
11
12    public function __construct($dir = true){
13
14        $this->matrizA = null; // matriz de aristas || Array de 2 posiciones
15        $this->vectorV = null; // vector de nodos || Array de 1 posición
16        $this->dirigido = $dir; // dirigido o no dirigido
17    }
18
19    /// vertices igual nodos
20    public function agregarVertice($v)
21    {
22        // si el vector de nodos no tiene el nodo asociado (ejemplo ["B"]=10)
23        // "si no existe"
24        if (!isset($this->vectorV[$v->getId()])) {
25            $this->matrizA[$v->getId()] = null;
26            $this->vectorV[$v->getId()] = $v; // se agrega el objeto al
27            vector de nodos
28        }
29    }
30 }
```

# Recorrido más corto

```
index.php x grafo.php x +
grafo.php
215 public function CaminoMasCorto($a,$b){
216     //Validamos si existe los vertice
217     if(isset($this->vectorV[$a]) AND isset($this->vectorV[$b])){
218         $S = array();
219         $Q = array();
220         foreach(array_keys($this->matrizA) as $val) $Q[$val] = 99999;
221         $Q[$a] = 0;
222         //inicio calculo
223         while(!empty($Q)){
224             $min = array_search(min($Q), $Q);
225             if($min == $b) break;
226             foreach($this->matrizA[$min] as $key=>$val) if(!empty($Q[$key]) && $Q[$min] + $val < $Q[$key]) {
227                 $Q[$key] = $Q[$min] + $val;
228                 $S[$key] = array($min, $Q[$key]);
229             }
230             unset($Q[$min]);
231         }//End While
232         $path = array();
233         $pos = $b;
234         while($pos != $a){
235             $path[] = $pos;
236             $pos = $S[$pos][0];
237         }
238         $path[] = $a;
239         $path = array_reverse($path);
240         return $path;
241     }else{
```

Para resolver el problema planteado, debemos utilizar el algoritmo del recorrido más corto,



Grafo planteado en la aplicación.