Plantilla de Especificación Operacional

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | Programa de prueba de grafos | **No. de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número de escenario** | **1** | **Objetivo del usuario** | Probar funcionalidad basica | |
| **Objetivo del escenario** | | Describir interacción sin errores de entrada de datos | | |
| **Fuente** | **Paso #** | **Acción** | | **Comentarios** |
| Programa | 1 | Pedir al usuario un valor de peso para el vértice 1 | |  |
| Usuario | 2 | Teclea un valor entero positivo | |  |
| Programa | 3 | Pedir al usuario un valor de peso para el vértice 2 | |  |
| Usuario | 4 | Teclea un valor entero positivo | |  |
| Programa | 5 | Pedir al usuario un valor de peso para el vértice 3 | |  |
| Usuario | 6 | Teclea un valor entero positivo | |  |
| Programa | 7 | Pedir al usuario un valor de peso para el vértice 4 | |  |
| Usuario | 8 | Teclea un valor entero positivo | |  |
| Programa | 9 | Inserta 3 Arcos (1,2), (1,4) y (2,3) | |  |
| Programa | 10 | Despliega la información del grafo | |  |
| Programa | 11 | Borra el arco (1,2) | |  |
| Programa | 12 | Despliega la información del grafo | |  |
| Programa | 13 | Borra el vértice 3 | |  |
| Programa | 14 | Borra el vértice 2 | |  |
| Programa | 15 | Despliega la información del grafo | |  |
| Programa | 16 | Borra el vértice 4 | |  |
| Programa | 17 | Borra el vértice 1 | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |

Plantilla de Especificación Funcional

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Estudiante** | |  | | | **Fecha** | 24/abril/2024 |
| **Programa** | | TDA NodoGrafo | | | **# de Pgma** |  |
| **Instructor** | |  | | | **Lenguaje** |  |
|  | | | | | | |
| **Nombre de la clase** | | | NodoGrafo | | | |
| **Clase(s) padre(s)** | | | No hay, pero si tiene a Grafo como clase amiga | | | |
|  | | |  | | | |
|  | | |  | | | |
|  | | |  | | | |
|  | | | | | | |
| **Atributos** | | | | | | |
|  | **Declaración** | | | **Descripción** | | |
|  | T informacion | | | Información del vértice (tipo T) | | |
|  | ListaEnlazada de enteros adyacencias | | | Lista de adyacencias del vértice (valores enteros) | | |
|  | entero estado | | | Estado del vértice, se utiliza para los recorridos, valor entero | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  | | | | | | |
| **Elementos** | | | | | | |
|  | **Declaración** | | | **Descripción** | | |
|  | NodoGrafo() | | | Constructor, crea un vértice con estado en 0 | | |
|  | NodoGrafo(T info) | | | Constructor, crea un vértice con info, y estado en 0 | | |
|  | Booleano insertaEnLista(entero valor) | | | Trata de insertar un nuevo valor al final de la lista de adyacencias, regresa verdadero si pudo hacerlo, falso si no pudo | | |
|  | Entero buscaEnLista(entero valor) | | | Regresa la posición de valor en la lista de adyacencias, si no lo encuentra regresa 0 | | |
|  | Sin tipo borraEnLista(entero posicion) | | | Borra el elemento que está en la posición de la lista de adyacencias. | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |

Plantilla de Especificación Funcional

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Estudiante** | |  | | | **Fecha** | 17/Octubre/2023 |
| **Programa** | | TDA Grafo | | | **# de Pgma** |  |
| **Instructor** | |  | | | **Lenguaje** |  |
|  | | | | | | |
| **Nombre de la clase** | | | Grafo | | | |
| **Clase(s) padre(s)** | | |  | | | |
|  | | |  | | | |
|  | | |  | | | |
|  | | |  | | | |
|  | | | | | | |
| **Atributos** | | | | | | |
|  | **Declaración** | | | **Descripción** | | |
|  | ListaEncadenada de NodoGrafo vertices | | | Lista de vértices | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  |  | | |  | | |
|  | | | | | | |
| **Elementos** | | | | | | |
|  | **Declaración** | | | **Descripción** | | |
|  | booleano insertaNodo(T info) | | | Crea un nuevo nodo al que le asigna info y lo añade a vertices | | |
|  | booleano insertaArco(entero origen, entero destino) | | | Crea un nuevo arco entre el nodo número origen y el nodo número destino. Lo inserta en la lista de adyacencias del nodo número origen | | |
|  | booleano borraNodo(T info) | | | Busca el nodo cuya información sea info y lo elimina de la lista de vértices | | |
|  | booleano borraArco(entero origen, entero posicion) | | | Borra el arco numero posición de la lista de adyacencias del vértice número origen | | |
|  | Sin tipo recorridoAnchura() | | | Realiza el recorrido en anchura del gráfo | | |
|  | Sin tipo recorridoProfundidad() | | | Realiza el recorrido en profundidad del gráfo | | |
|  |  | | |  | | |
|  |  | | |  | | |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | TDA NodoGrafo | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Especificación Funcional del TDA NodoGrafo |
| **De Diseño** | Constructor |
|  | NodoGrafo(T info) |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | T info |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Asignar info a informacion |
| Asignar cero a estado |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | TDA NodoGrafo | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Especificación Funcional del TDA NodoGrafo |
| **De Diseño** | Constructor |
|  | NodoGrafo() |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | No tiene |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Asignar cero a estado |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | TDA NodoGrafo | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Especificación Funcional del TDA NodoGrafo |
| **De Diseño** | booleano insertaEnLista |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | entero valor |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar bandera como booleano |
| Asignarle a bandera el resultado de llamar a la función insertarElementoFinal de adyacencias pasándole como parámetro valor |
| Regresar el valor de bandera |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | TDA NodoGrafo | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Especificación Funcional del TDA NodoGrafo |
| **De Diseño** | Entero buscaEnLista |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | entero valor |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Regresar el resultado de llamar a la función buscarPosicionElemento de la lista de adyacencias enviándole como parámetro valor |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | TDA NodoGrafo | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Especificación Funcional del TDA NodoGrafo |
| **De Diseño** | Sin tipo borraEnLista(entero posicion) |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | No tiene |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Llamar a la función borrarElemento de la lista de adyacencias mandándole posición como parámetro |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | TDA Grafo | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Especificación Funcional del TDA Grafo |
| **De Diseño** | booleano insertaNodo(T info) |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | T info |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar la variable bandera como booleano |
| Declarar la variable auxiliar como NodoGrafo e inicializarlo con info |
| Asignar a bandera el resultado de llamar a la función insertarElementoFinal de vertices pasándole como parámetro auxiliar |
| Regresar bandera |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | TDA Grafo | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Especificación Funcional del TDA Grafo |
| **De Diseño** | booleano insertaArco(entero origen, entero destino) |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | entero origen |
|  | entero destino |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar variable auxiliar como apuntador a NodoGrafo |
| Declarar variable bandera como booleana |
| Declarar variable tamanio como entera |
| Declarar variable ady como ListaEncadenada de enteros |
| Asignar a tamanio vertices.obtenerTamanio() |
| Si ((origen <= tamanio) Y (destino <= tamanio) Y (tamanio !=0)) |
| Asignar a auxiliar vertices.traerDatosPosicion(origen) |
| Asignar a ady auxiliar->adyacencias |
| Asignar a bandera ady.insertarElementoFinal(destino) |
| else |
| Asignar falso a bandera |
| Regresar bandera |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | TDA Grafo | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Especificación Funcional del TDA Grafo |
| **De Diseño** | booleano borraNodo(entero posicion) |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | entero posicion |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar variable bandera como booleana |
| Declarar variable tamanio como entera |
| Asignar a tamanio vertices.tamanio() |
| Si (posición <= tamanio) |
| Asignar verdadero a bandera |
| vertices.borrarElemento(posición) |
| Sino |
| Asignar falso a bandera |
| FinSi |
| Regresar bandera |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | TDA Grafo | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Especificación Funcional del TDA Grafo |
| **De Diseño** | booleano borraArco(entero origen, entero posicion) |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | entero origen |
|  | entero posicion |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar variable auxiliar como apuntador a Nodografo |
| Declarar variable bandera como booleana |
| Declarar variable tamanio como entera |
| Declarar variable ady como ListaEncadenada de enteros |
| Asignar a tamanio vértices.tamanio() |
| Si ((origen <= tamanio) Y (posicion <= tamanio) Y (tamanio != 0)) |
| Asignar a auxiliar vertices.traerDatosPosicion(origen) |
| Asignar a ady auxiliar->adyacencias |
| Asignar a bandera ady.borrarElemento(posicion) |
| else |
| Asignar falso a bandera |
| Regresar bandera |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | TDA Grafo | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Especificación Funcional del TDA Grafo |
| **De Diseño** | despliegaInformacion() |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | No tiene |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar variable auxiliar como apuntador a NodoGrafo |
| Declarar variable tamanio como entera |
| Declarar variable ady como ListaEncadenada de enteros |
| Declarar variable tamAdy como entera |
| Asignar a tamanio vertices.tamanio() |
| Ciclarse sobre i de 1 a tamanio |
| Asignar a auxiliar vértices.traerDatosPosicion(i) |
| Desplegar en pantalla en una línea “Informacion del vértice “,i |
| Desplegar en pantalla auxiliar->informacion |
| Asignar a ady auxiliar->adyacencias |
| Asignar a tamAdy ady.tamanio() |
| Ciclarse sobre j de 1 a tamAdy |
| Desplegar en pantalla en una línea  “Arco con vértice no”,\*(ady.traerDatosPosicion(j)) |
| FinCiclo j |
| FinCiclo i |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | TDA Grafo | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Especificación Funcional del TDA Grafo |
| **De Diseño** | recorridoAnchura() |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | No tiene |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar variable auxiliar como apuntador a NodoGrafo |
| Declarar variable tamanio como entera |
| Declarar variable ady como ListaEncadenada de enteros |
| Declarar variable tamAdy como entera |
| Declarar variable fila como ListaEncadenada de apuntadores a NodoGrafo |
| Declarar variable auxiliar2 como apuntador a NodoGrafo |
| Declarar variable índice como entera |
| Asignar a tamanio vertices.tamanio() |
| Ciclarse sobre i de 1 a tamanio |
| Asignar a auxiliar vértices.traerDatosPosicion(i) |
| Asignar 0 al estado de auxiliar |
| FinCiclo i |
| Ciclarse sobre i de 1 a tamanio |
| Asignar a auxiliar vértices.traerDatosPosicion(i) |
| Insertar auxiliar al final de fila |
| Mientras fila no este vacia |
| Asignar a auxiliar el inicio de la fila |
| Borrar el inicio de la fila |
| Asignar 2 al estado de auxiliar |
| Asignar a ady auxiliar->adyacencias |
| Asignar a tamAdy ady.tamanio() |
| Ciclarse sobre j de 1 a tamAdy |
| Asignar a índice el valor apuntado por ady.traerDatosPosicion(j) |
| Asignar a auxiliar2 vertices.traerDatosPosicion(índice) |
| Si estado de auxiliar2 es 0 |
| Asignar 1 a estado de auxiliar2 |
| Insertar auxiliar 2 al final de fila |
| FinCiclo j |
| FinMientras |
| FinCiclo i |

Plantilla de Especificación Lógica

|  |  |  |  |
| --- | --- | --- | --- |
| **Estudiante** |  | **Fecha** |  |
| **Programa** | TDA Grafo | **# de Pgma** |  |
| **Instructor** |  | **Lenguaje** |  |

|  |  |
| --- | --- |
| **Referencias** | Especificación Funcional del TDA Grafo |
| **De Diseño** | recorridoProfundidad() |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Parámetros** | No tiene |
|  |  |
|  |  |
|  |  |
|  |  |

|  |
| --- |
| Declarar variable auxiliar como apuntador a Vertice |
| Declarar variable tamanio como entera |
| Declarar variable ady como apuntador a ListaEncadenada de enteros |
| Declarar variable tamAdy como entera |
| Declarar variable pila como ListaEncadenada de apuntadores a Vertice |
| Declarar variable auxiliar2 como apuntador a Vertice |
| Declarar variable índice como entera |
| Asignar a tamanio vertices.tamanio() |
| Ciclarse sobre i de 1 a tamanio |
| Asignar a auxiliar vértices.traerDatosPosicion(i) |
| Asignar 0 al estado de auxiliar |
| FinCiclo i |
| Ciclarse sobre i de 1 a tamanio |
| Asignar a auxiliar vértices.traerDatosPosicion(i) |
| Si estado de auxiliar es 0 |
| Insertar auxiliar al inicio de pila |
| Mientras pila no este vacia |
| Asignar a auxiliar el inicio de la pila |
| Borrar el inicio de la pila |
| Si el estado de auxiliar es 0 |
| Asignar 1 al estado de auxiliar |
| Asignar a ady auxiliar->traeAdyacencias() |
| Asignar a tamAdy ady->tamanio() |
| Ciclarse sobre j de 1 a tamAdy |
| Asignar a índice el valor apuntado por ady->traerDatosPosicion(j) |
| Asignar a auxiliar2 vertices.traerDatosPosicion(índice) |
| Si estado de auxiliar2 es 0 |
| Insertar auxiliar2 al inicio de pila |
| FinSi |
| FinCiclo j |
| FinSi |
| FinMientras |
| FinSi |
| FinCiclo i |