|  |  |
| --- | --- |
| **TAD GRAFO** | |
| **Invariantes:**  El grafo G consiste en un conjunto de vértices (V) y un conjunto de aristas (E). Cada arista conecta dos vértices y ambos conjuntos son no vacíos. | |
| Operaciones Primitivas:  **1.AgregarVértice(): -> Vértice**   * Descripción: Agrega un nuevo vértice al grafo. * Entrada: Ninguna. * Salida: Un nuevo vértice agregado al grafo.   **2.AgregarArista(v1, v2): -> Arista**   * Descripción: Crea una arista que conecta dos vértices existentes en el grafo. * Entrada: Dos vértices v1 y v2. * Salida: Una nueva arista que conecta los vértices v1 y v2.   **3.EliminarVértice(v):**   * Descripción: Elimina un vértice específico del grafo. * Entrada: El vértice v a ser eliminado. * Salida: Ninguna.   **4.EliminarArista(a):**   * Descripción: Elimina una arista específica del grafo. * Entrada: La arista a a ser eliminada. * Salida: Ninguna.   **5.ObtenerVértices(): ->Vértices**   * Descripción: Devuelve el conjunto de vértices presentes en el grafo. * Entrada: Ninguna. * Salida: Conjunto de vértices del grafo.   **6.ObtenerAristas(): ->Arista**   * Descripción: Devuelve el conjunto de aristas presentes en el grafo. * Entrada: Ninguna. * Salida: Conjunto de aristas del grafo.   **7.VerificarConexión(v1, v2): -> Booleano**   * Descripción: Verifica si hay una arista que conecta dos vértices. * Entrada: Dos vértices v1 y v2. * Salida: Valor booleano indicando si existe una conexión entre v1 y v2. | |
| AgregarVértice():   * Precondición: Ninguna. * Postcondición: Se agrega un vértice válido al grafo | AgregarArista(v1, v2):   * Precondición: v1 y v2 son vértices válidos en el grafo. * Postcondición: Se crea una arista que conecta los vértices v1 y v2, generando una conexión válida. |
| EliminarVértice(v):   * Precondición: v es un vértice existente en el grafo. * Postcondición: El vértice v es eliminado del grafo junto con las aristas asociadas a él. | EliminarArista(a):   * Precondición: a es una arista existente en el grafo. * Postcondición: La arista a es eliminada del grafo. |
| ObtenerVértices()   * Precondición: No hay ninguna precondición. * Postcondición: Se devuelve un conjunto válido de vértices presentes en el grafo. | ObtenerAristas()   * Precondición: No hay ninguna precondición. * Postcondición: Se devuelve un conjunto válido de aristas presentes en el grafo. |
| VerificarConexión(v1, v2)   * Precondición: v1 y v2 son vértices válidos en el grafo. * Postcondición: Se devuelve 'Verdadero' si existe una arista que conecta v1 y v2; de lo contrario, se devuelve 'Falso'. | |