

Manual de Usuario

Proyecto:

Ciencias de la Computación II

Autores:

Tomás Alejandro Delgado Ortiz - 20221020045

Juan Pablo Mosquera Marín - 20221020026

Este documento contiene el instructivo tecnico detallado de los modulos del proyecto explicando su funcionalidad enfocado en proporcionar una guia para el usuario final y garantizar el correcto uso del aplicativo a traves de su intefaz.

Tabla de Contenidos

- Manual de Búsquedas Internas
 - Busqueda Binaria
 - Busqueda Lineal
 - Funciones Hash
 - Arbol Digital
 - Arbol de Huffman
 - Residuo Multiple
 - Residuo Simple
 - Controles y Campos
- Búsquedas Externas
 - Búsqueda Lineal Externa
 - Búsqueda Binaria Externa
 - Búsqueda Hash Externa
 - Estructuras Dinámicas
 - Índices Externos
- Grafos
 - Árboles de Expansión
 - Distancia entre Árboles de Expansión
 - Algoritmo de Floyd-Warshall
 - Matrices y Adyacencias
 - Matrices y Circuitos
 - Operaciones entre N Grafos
 - Operaciones en un Grafo
- Fin del Manual de Usuario

Manual de Usuario - Manual Búsquedas Internas

Esta sección del documento contiene los manuales de usuario internos para las vistas y módulos del aplicativo. Cada sección replica exactamente la estructura y profundidad del manual de [AlgoritmoFloyd.md](#)

y está adaptada a la funcionalidad de cada módulo. Las secciones aparecen secuencialmente y están pensadas para un usuario sin conocimientos previos de algoritmos.

Busqueda lineal

Descripción General

La vista **Busqueda Lineal** provee una interfaz para practicar la búsqueda lineal dentro de bloques (simulación de búsqueda externa por bloques) y operaciones relacionadas: insertar, buscar y eliminar. El objetivo pedagógico es mostrar cómo funciona la búsqueda secuencial dentro de bloques y el impacto de los corrimientos al insertar o eliminar.

Crear la Estructura

Paso 1: Configuración Inicial

1. **Capacidad total:** número máximo de elementos del archivo.
2. **Tamaño de bloque (B):** número de elementos por bloque (puede calcularse automáticamente como $L/\text{capacidad } J$).
3. **Tipo de clave:** **numérico** o **alfabético**.

-#### Paso 2: Crear la Estructura

- Presione "**Crear estructura**" para generar los bloques vacíos.
- Se renderizarán los bloques en la vista, cada uno con sus índices internos.

Alternativa: Importar Estructura

- Use "**Abrir desde archivo**" para cargar un JSON previamente exportado.
-

Gestionar Elementos

Agregar Elementos

1. **Campo de clave:** ingrese el valor a insertar.
2. **Insertar:** ubica el bloque objetivo (comparando con último elemento del bloque) y realiza corrimientos si es necesario.
 - Si el bloque está lleno, **insertarConCorrimiento** desplaza elementos a bloques siguientes.
 - Se validan duplicados y se muestra un mensaje claro si ya existe la clave.

Buscar Elementos

- La búsqueda determina el bloque candidato y realiza búsqueda lineal (recorrido secuencial) dentro del bloque hasta encontrar la clave o llegar a una celda vacía.
- Muestra el bloque y la posición dentro del bloque.

Eliminar Elementos

- Localiza la clave y aplica `eliminarConCorrimiento` para compactar el bloque hacia la izquierda.

Visualización

- Muestra bloques verticales con índices globales y locales.
- Resalta el bloque en búsqueda y los desplazamientos durante inserciones/eliminaciones.

Guardar y Cargar Estructuras

Guardar

- Botón "**Guardar**" descarga un JSON con `type: 'lineal-externa'`, `config` (capacidad, B) y `data` (bloques).

Abrir

- Botón "**Abrir**" permite seleccionar un JSON válido y restaurar la estructura.

Ejecutar Operaciones

Ejecutar la Búsqueda

1. Inserte o cargue la estructura.
2. Ingrese la clave a buscar y pulse "**Buscar**".
3. El sistema determinará bloque candidato y efectuará búsqueda lineal dentro de él.

Interpretar Resultados

1. Bloque candidato

- Se indica cuál bloque fue elegido por la heurística (comparación con último elemento del bloque).

2. Posición en bloque

- Si se encuentra, se muestra el índice global y local; si no, se indica la ausencia.

Resetear la Estructura

- Use "**Resetear**" para borrar la estructura y volver a la configuración inicial. Confirme la acción.

Resumen de Botones y Acciones

Botón/Acción	Ubicación	Función
Crear estructura	Configuración inicial	Genera bloques vacíos según parámetros

Botón/Acción	Ubicación	Función
Abrir desde archivo (initial)	Debajo de "Crear estructura"	Importa una estructura antes de crear una nueva
Guardar Estructura	Barra de operaciones	Exporta los bloques a JSON
Abrir Estructura (operaciones)	Barra de operaciones	Importa estructura reemplazando la actual
Insertar	Sección de inserción	Inserta en bloque con corrimiento si aplica
Buscar	Sección de búsqueda	Encuentra la clave por búsqueda lineal en bloque
Eliminar	Sección de borrado	Elimina y compacta el bloque
Resetear	Final de la página	Elimina la estructura actual

Busqueda Binaria

Descripción General

La vista **Binaria** proporciona una interfaz interactiva para comprender y practicar la búsqueda binaria en un arreglo ordenado. El usuario puede crear una estructura de datos ordenada (lista/array), configurar parámetros de prueba, insertar elementos en orden, y realizar operaciones de búsqueda y eliminación. La interfaz visualiza el arreglo dividido en índices y resalta pasos de la búsqueda binaria para que el usuario siga el algoritmo paso a paso.

Crear la Estructura

Paso 1: Configuración Inicial

Al ingresar a la vista se presenta un formulario con los campos principales:

- Capacidad (tamaño del arreglo):** Ingrese el número máximo de elementos que la estructura podrá contener (ej.: 20).
 - Recomendación: usar valores mayores a 10 para prácticas más ilustrativas.
- Tipo de datos:** Seleccione el tipo de claves a manejar (por ejemplo, **numérico** o **alfabético**).
- Orden de inserción:** Opción para definir si la inserción mantiene el arreglo **ascendente** o **descendente**.

Paso 2: Crear la Estructura

- Haga clic en el botón "**Crear estructura**".
- El sistema generará un arreglo vacío con la **capacidad** indicada y mostrará la vista del arreglo con índices (0..capacidad-1) listo para insertar elementos.

Alternativa: Importar una Estructura Existente

Si ya tiene un arreglo guardado previamente:

- Haga clic en "**Abrir desde archivo**".
 - Seleccione el archivo JSON con extensión **.json** que contiene la estructura serializada.
 - El arreglo se cargará con su configuración y elementos, y la vista se actualizará automáticamente.
-

Gestionar Elementos

Agregar Elementos

Una vez creada la estructura aparece la sección "**Insertar**" con los campos:

1. **Campo de valor**: Ingrese la clave a insertar (ej.: **42** o **A**).
 2. **Botón "Insertar"**: Inserta el valor manteniendo el orden. La inserción se realiza en lugar correcto desplazando elementos si es necesario.
- La vista valida duplicados y muestra un mensaje en caso de intento de insertar un elemento ya existente.
 - Si el arreglo está lleno el sistema alerta que no se puede insertar.

Buscar Elementos

- Campo "**BuscarBuscar**".
- La interfaz ejecuta la búsqueda binaria y resalta los índices **izquierda**, **derecha** y **medio** en cada paso, mostrando comparaciones y resultado final.
- Si el elemento existe se resalta su posición; si no, se muestra la posición donde debería insertarse.

Eliminar Elementos

- Campo "**ClaveEliminar**".
- El sistema localiza la posición (usando búsqueda binaria) y elimina el elemento desplazando los elementos restantes para mantener el orden.

Visualización del Arreglo

La visualización muestra:

- **Índices**: filas numeradas 0..n-1
- **Valores**: cada celda con su contenido o **null** si está vacía
- **Resaltados**: durante la búsqueda se indican **izquierda**, **derecha** y **medio** en cada iteración

Puede interactuar con la vista para seguir la animación paso a paso o ejecutar la búsqueda automáticamente.

Guardar y Cargar Estructuras

Guardar la Estructura

1. Haga clic en "**Guardar**".
2. El sistema generará un archivo JSON que contiene:

- Configuración (**capacidad, tipo, orden**)
 - Contenido del arreglo
 - Fecha y etiqueta en el nombre del archivo
3. El archivo se descargará automáticamente.

Abrir una Estructura Guardada

1. Haga clic en "**Abrir**" (operaciones)
 2. Seleccione el archivo JSON previamente guardado
 3. La estructura se cargará y la vista se actualizará
-

Ejecutar la Búsqueda Binaria

Ejecutar el Algoritmo

1. Inserte los datos o cargue una estructura.
2. Ingrese la clave a buscar y pulse "**Buscar**".
3. El sistema realizará la búsqueda binaria y mostrará cada paso de comparación.

Interpretar los Resultados

1. Paso a paso de la Búsqueda

- Se muestran las variables **izquierda, derecha y medio** en cada iteración.
- Se indica si **valor_en_medio < objetivo, > objetivo o == objetivo**.

2. Resultado final

- Si el elemento existe: se muestra la posición exacta.
 - Si no existe: se indica **No encontrado** y la posición sugerida para la inserción.
-

Resetear la Estructura

Para empezar desde cero:

1. Desplácese hasta el final de la vista
2. Haga clic en "**Resetear**" (botón rojo)
3. Confirme la acción
4. Volverá a la pantalla de configuración inicial

Nota: Esta acción elimina la estructura actual. Guarde antes si desea conservarla.

Resumen de Botones y Acciones

Botón/Acción	Ubicación	Función
--------------	-----------	---------

Botón/Acción	Ubicación	Función
Crear estructura	Configuración inicial	Genera el arreglo con los parámetros especificados
Abrir desde archivo (initial)	Debajo de "Crear estructura"	Importa una estructura antes de crear una nueva
Guardar Estructura	Barra de operaciones	Exporta el arreglo actual a JSON
Abrir Estructura (operaciones)	Barra de operaciones	Importa una estructura reemplazando la actual
Insertar	Sección de inserción	Inserta un elemento manteniendo el orden
Buscar	Sección de búsqueda	Ejecuta búsqueda binaria mostrando pasos
Eliminar	Sección de borrado	Elimina el elemento y compacta el arreglo
Resetear	Final de la página	Elimina la estructura actual y vuelve al inicio

Funciones Hash

Descripción General

La vista **FuncionesHash** implementa una interfaz para explorar hashing por módulo y técnicas de resolución de colisiones (área de colisiones o estructura secundaria). Es una herramienta didáctica para experimentar con parámetros como **capacidad**, **digitosClave** y la estrategia de colisiones, observando cómo se distribuyen las claves en bloques principales y áreas de colisión.

Crear la Estructura

Paso 1: Configuración Inicial

- Capacidad:** número total de posiciones que la estructura puede contener (mínimo recomendado 10).
- Dígitos por clave:** cantidad de dígitos/carácteres esperados en la clave (ej.: 1 para claves simples).
- Resolución de colisiones:** seleccione **estructura-secundaria** o **area-colisiones**.

Paso 2: Crear la Estructura

- Pulse "**Crear estructura**" para inicializar bloques principales y, según la estrategia, bloques secundarios o áreas de colisiones.

Alternativa: Importar

- Use "**Abrir desde archivo**" para restaurar desde un JSON exportado previamente.

Gestionar Elementos

Agregar Elementos

1. **Campo de clave:** ingrese la clave a insertar (entera positiva o formato definido por la vista).
 2. **Insertar:** se aplica `hashModulo(clave, numeroBloques)` para localizar el bloque principal.
 - Si el bloque principal tiene espacio, se coloca en la primera posición libre.
 - Si está lleno, se usa `areaColisiones` o se inserta en `estructuraSecundaria` según la estrategia.
- Mensajes claros informan sobre duplicados, inserciones exitosas o errores de capacidad.

Buscar Elementos

- Calcula el mismo hash y busca por orden: bloque principal → área de colisiones → bloque secundario (si aplica).
- Muestra la ruta de búsqueda y la posición si se encuentra.

Eliminar Elementos

- Localiza la clave y la elimina si existe; actualiza contadores de ocupación.

Visualización

- Muestra bloques principales y, cuando corresponde, áreas de colisión o bloques secundarios.
- Resalta el bloque objetivo y las posiciones exploradas durante búsquedas o corrimientos.

Guardar y Cargar Estructuras

Guardar

- "**Guardar**" descarga JSON con `type: 'hash-modulo-externo'`, config y estructuras internas.

Abrir

- "**Abrir**" valida el JSON y restaura la estructura si el `type` y `config` son compatibles.

Ejecutar Operaciones

Insertar/Buscar/Eliminar

- Uso típico: crear estructura → insertar varias claves → buscar para verificar posiciones → eliminar si se necesita.
- Interpretación: la distribución muestra cuántas claves van al bloque principal vs. área de colisiones, permitiendo evaluar la eficiencia del hash.

Resetear la Estructura

- Botón "**Resetear**" borra toda la estructura; confirme para proceder.

Resumen de Botones y Acciones

Botón/Acción	Ubicación	Función
Crear estructura	Configuración inicial	Inicializa bloques principales y secundarios/áreas según estrategia
Abrir desde archivo (inicial)	Debajo de "Crear estructura"	Importa una estructura antes de crear una nueva
Guardar Estructura	Barra de operaciones	Exporta la estructura a JSON
Abrir Estructura (operaciones)	Barra de operaciones	Importa estructura reemplazando la actual
Insertar	Sección de inserción	Inserta clave usando hash por módulo
Buscar	Sección de búsqueda	Localiza clave por recorrido de áreas
Eliminar	Sección de borrado	Elimina la clave si existe
Resetear	Final de la página	Elimina la estructura actual

Arbol Digital

Descripción General

La vista **Arbol Digital** implementa un árbol digital (trie/árbol binario por bits) para indexar claves basadas en su representación binaria. Permite insertar letras/clave, buscar, eliminar y visualizar el árbol en forma gráfica con **vis-network**. Es útil para comprender búsquedas por prefijos y navegación bit a bit.

Crear la Estructura

Paso 1: Configuración Inicial

- Profundidad / dígitos por clave:** define cuántos bits se usan para representar cada clave.
- Opciones de visualización:** activar/desactivar **Vista gráfica**.

Paso 2: Crear la Estructura

- Inserte una clave en el campo correspondiente y pulse "**Insertar**" para crear la raíz y comenzar a construir el árbol.

Alternativa: Importar

- "**Abrir**" permite cargar un JSON previamente exportado con la estructura del árbol y códigos almacenados.

Gestionar Elementos

Agregar Elementos

1. **Campo de clave:** ingrese una letra o clave compatible.
 2. **Insertar:** la vista convierte la letra a `binary code` (usando `letterToCode`), divide en grupos y recorre el árbol asignando la clave en la hoja correspondiente.
- El sistema informa `duplicado` o `insertado` con la ruta en índices de hijos.

Buscar Elementos

- Convierte la clave a código binario y recorre los grupos para localizar la hoja; si existe muestra la ruta y resalta el nodo.

Eliminar Elementos

- Localiza la hoja y borra `key` y `code` (eliminación lógica), actualizando la vista.

Visualización

- `vis-network` renderiza conectores y hojas; los conectores muestran `labelBits` y las hojas muestran la `key`.
 - La vista permite centrar y seleccionar nodos según la ruta calculada.
-

Guardar y Cargar Estructuras

Guardar

- "**Guardar**" descarga un JSON con `type: 'residuo-digital'`, config y el árbol serializado.

Abrir

- "**Abrir**" valida e importa el JSON restaurando la plantilla y claves.
-

Ejecutar Operaciones

Insertar/Buscar/Eliminar

- Pasos típicos: crear plantilla → insertar letras → buscar para comprobar ubicación → eliminar si es necesario.
 - Interpretación: la ruta y los `groups` binarios explican cómo se accede a cada hoja.
-

Resetear la Estructura

- "**Resetear**" borra la estructura actual y reinicia la configuración. Confirme antes de continuar.
-

Resumen de Botones y Acciones

Botón/Acción	Ubicación	Función
Crear estructura	Configuración inicial	Genera plantilla del árbol según parámetros
Abrir (inicial)	Debajo de "Crear estructura"	Importa árbol antes de crear uno nuevo
Guardar	Barra de operaciones	Exporta el árbol a JSON
Abrir (operaciones)	Barra de operaciones	Importa árbol reemplazando el actual
Insertar	Sección de inserción	Inserta clave recorriendo grupos binarios
Buscar	Sección de búsqueda	Encuentra la clave y resalta la ruta
Borrar	Sección de borrado	Elimina la clave de la hoja correspondiente
Reiniciar	Final de la página	Elimina la estructura actual

Arbol de Huffman

Descripción General

La vista **Arbol de Huffman** ofrece una interfaz para construir y experimentar con códigos de Huffman aplicados a cadenas de texto. Con esta vista el usuario aprende a calcular frecuencias, generar códigos prefijo, codificar y decodificar textos, y evaluar métricas de eficiencia. Incluye export/import del árbol y la representación serializable para ejercicios y pruebas.

Crear la Estructura

Paso 1: Configuración Inicial

1. **InputText:** área de texto donde el usuario pega la cadena a codificar.
2. **Opciones de normalización:** por ejemplo, eliminar saltos de línea o considerar mayúsculas/minúsculas.

Paso 2: Construir Huffman

- Haga clic en "**Construir**" para calcular frecuencias, formar el árbol y generar códigos.
- Se mostrará la tabla de frecuencias y el mapa **codes**.

Alternativa: Importar

- Use "**Abrir**" para cargar un JSON con **type: 'huffman'** y restaurar árbol, códigos y texto original.

Gestionar Elementos

Calcular Frecuencias

- Tras pegar **inputText**, pulse el botón para calcular frecuencias; la tabla mostrará símbolo y frecuencia.

Generar Códigos y Codificar

- **Generar códigos:** ejecuta la construcción de Huffman.
- **Codificar:** convierte `inputText` a `encodedBits` usando el mapa `codes`.
- **Decodificar:** con el árbol construido se puede recuperar el texto a partir de la secuencia binaria.

Visualización

- Tabla de frecuencias ordenada por frecuencia descendente.
 - Vista gráfica del árbol con `vis-network`, aristas etiquetadas `0/1`.
-

Guardar y Cargar

Guardar

- "**Guardar**" descarga un JSON con `type: 'huffman'`, `frequencies`, `tree` y `config`.

Abrir

- "**Abrir**" valida `type === 'huffman'` y restaura la vista con árbol, códigos y texto.
-

Ejecutar Operaciones

Codificar/Decodificar

- Paso a paso: pegar texto → construir → codificar → descargar o decodificar para verificar integridad.

Interpretar Resultados

1. Tabla de frecuencias

- Muestra cada símbolo y su frecuencia.

2. Longitud ponderada L

- $L = \sum(\text{frecuencia} * \text{longitud_codigo})$ y el total de bits codificados; se compara con el tamaño en bits original.
-

Reiniciar la Vista

- "**Reiniciar**" borra `inputText` y el `huffmanResult`. Confirme antes de proceder.
-

Resumen de Botones y Acciones

Botón/Acción	Ubicación	Función
Construir	Sección principal	Calcula frecuencias y construye árbol
Codificar	Controles	Convierte texto a bits según códigos

Botón/Acción	Ubicación	Función
Decodificar	Controles	Reconstruye texto desde bits usando el árbol
Guardar	Barra de operaciones	Exporta datos a JSON
Abrir	Barra de operaciones	Importa JSON y restaura estado
Reiniciar	Final de la página	Limpia la vista

Residuo Simple

Descripción General

ResiduoSimple muestra la versión básica de tabla por residuos: $h(k) = k \bmod m$. Sirve para entender particionamiento por módulo y comparar estrategias de resolución de colisiones como encadenamiento y sondeo.

Crear la Estructura

Paso 1: Configuración Inicial

1. **m (tamaño de tabla)**: número de casillas en la tabla.
2. **Estrategia de colisión**: seleccione encadenamiento o sondeo.

Paso 2: Crear la Tabla

- Pulse "Crear Tabla" para inicializar las casillas vacías.

Alternativa: Importar

- "Abrir" restaura una tabla desde JSON con type: 'residuo-simple'.

Gestionar Elementos

Insertar

1. **Campo clave**: ingrese valor numérico o letra (según la conversión definida).
2. **Insertar**: calcula $h(k)$ y aplica la estrategia seleccionada.
 - Encadenamiento: añade el elemento a la lista en $table[h]$ si no está duplicado.
 - Sondeo: busca la siguiente posición libre $i = (i+1)\%m$ hasta encontrar espacio o duplicado.

Buscar

- Calcula $h(k)$ y recorre la lista (encadenamiento) o la secuencia de sondeo para localizar la clave.

Eliminar

- Elimina la clave y actualiza métricas de carga.

Visualización

- Tabla con casillas mostrando listas o valores según la estrategia.
 - Resalta la casilla objetivo y pasos de sondeo si aplica.
-

Guardar y Cargar

Guardar

- "**Guardar**" exporta JSON con `type: 'residuo-simple'`, `config` y `data`.

Abrir

- "**Abrir**" valida y restaura la tabla.
-

Ejecutar Operaciones

- Crear tabla → insertar claves → buscar/ eliminar → observar métricas (load factor).
 - Interpretación: la distribución en la tabla y la longitud de las listas permiten evaluar la eficiencia.
-

Reiniciar la Tabla

- "**Reiniciar**" elimina la tabla; confirme antes de realizar la acción.
-

Resumen de Botones y Acciones

Botón/Acción	Ubicación	Función
Crear Tabla	Configuración inicial	Inicializa la tabla con <code>m</code> casillas
Abrir Tabla (inicial)	Debajo de "Crear Tabla"	Importa una tabla antes de crear una nueva
Guardar	Barra de operaciones	Exporta la tabla a JSON
Abrir (operaciones)	Barra de operaciones	Importa tabla reemplazando la actual
Insertar	Sección de inserción	Inserta clave aplicando <code>h(k)</code> y la estrategia
Buscar	Sección de búsqueda	Localiza la clave según la estrategia
Borrar	Sección de borrado	Elimina la clave y actualiza métricas
Reiniciar	Final de la página	Borra la tabla

Residuo Multiple

Descripción General

Residuo Multiple implementa una estructura de residuos múltiples (árbol multinivel) que agrupa claves según su codificación binaria dividida en grupos de m bits. Es una herramienta pedagógica para visualizar cómo los bits de una clave determinan la ruta dentro de un árbol y cómo se almacenan claves en hojas.

Crear la Estructura

Paso 1: Configuración Inicial

1. **m (bits por grupo)**: valor entre 1 y 3 que define el número de hijos por conector (2^m).
2. **Selector de vista**: **Vista gráfica** activable para mostrar el árbol con **vis-network**.

Paso 2: Crear la Plantilla

- Pulse "**Crear estructura**" para generar el **buildRMTemplate(m)** y habilitar operaciones.

Alternativa: Importar

- "**Abrir**" restaura un JSON exportado con **type: 'residuo-multiple'**.
-

Gestionar Elementos

Insertar

1. **Campo clave (A-Z)**: ingrese una letra y pulse "**Insertar**".
 2. La vista convierte la letra a código binario (**letterToCode**), lo divide en grupos (**groupBits**) y recorre el árbol asignando la clave en la hoja.
- Mensajes informan **duplicado** o **insertado** con la ruta de índices.

Buscar

- Similar al insert: se calcula **groups** y se recorre el árbol; si se encuentra muestra ✓ **Encontrada** con ruta.

Borrar

- Busca la hoja y limpia **key** y **code**, mostrando la ruta del nodo afectado.

Visualización

- **buildGraphData** genera nodos y aristas para **vis-network**.
 - Conectores muestran **labelBits**, hojas muestran la **key**.
 - Al buscar/insertar se centra la vista en el nodo objetivo.
-

Guardar y Cargar

Guardar

- "**Guardar**" descarga `residuo-multiple` JSON con `config.base` y `data` (raíz del árbol).

Abrir

- "**Abrir**" valida con `validateResidueMultipleImport` y restaura `m`, `root` y el estado.
-

Ejecutar Operaciones

- Pasos: crear plantilla → insertar varias claves → usar **Buscar** para comprobar ubicación → **Borrar** si se desea.
 - Interpretación de mensajes: se muestran grupos y ruta → realizada en el árbol.
-

Reiniciar la Estructura

- "**Reiniciar**" destruye la plantilla y limpia la vista.
-

Resumen de Botones y Acciones

Botón/Acción	Ubicación	Función
Crear estructura	Panel inicial	Genera <code>buildRMTemplate(m)</code>
Abrir (inicial)	Panel inicial	Importa estructura antes de crear una nueva
Guardar	Herramientas	Exporta árbol a JSON
Abrir (operaciones)	Herramientas	Importa árbol reemplazando el actual
Insertar	Controles	Inserta letra recorriendo grupos binarios
Buscar	Controles	Busca letra y resalta ruta
Borrar	Controles	Elimina la clave en la hoja correspondiente
Reiniciar	Final	Borra la plantilla

Controles y Campos (Sección adicional)

Esta sección agrega una descripción detallada y accesible de los controles e inputs comunes en todas las vistas documentadas. Se incluye para facilitar al usuario final sin conocimientos técnicos.

Campos comunes

- **Capacidad / m / tamaño:** número que define el tamaño de la estructura; siempre debe ser entero positivo. Si no está seguro, usar un valor moderado (ej. 20) para evitar vistas saturadas.
- **Clave / Valor / InputText:** campo donde se ingresa la clave a insertar/buscar; para letras use una sola letra (A-Z), para números use enteros positivos.
- **Tamaño de bloque (B):** en vistas externas, define cuántos elementos hay por bloque; se sugiere usar la raíz cuadrada de la capacidad como punto de partida.

- **Estrategia de colisión:** opciones típicas: **encadenamiento, sondeo, area-colisiones, estructura-secundaria**. La elección afecta visiblemente la distribución de datos.

Botones comunes

- **Crear / Construir / Crear Estructura:** inicializa la estructura según parámetros.
- **Insertar / Agregar:** agrega un elemento, aplicando la lógica interna (orden, hash o recorrido).
- **Buscar:** ejecuta la operación de búsqueda y muestra resultados paso a paso.
- **Borrar / Eliminar:** quita la clave si se encuentra.
- **Guardar / Exportar:** descarga la estructura actual en JSON para compartir o preservar el estado.
- **Abrir / Importar:** carga una estructura previamente exportada; el archivo debe cumplir el **type** y la forma esperada.
- **Resetear:** borra la estructura actual; confirmar antes de proceder.

Mensajes y validaciones

- El sistema valida entradas (ej.: capacidad ≥ 2 , claves no vacías) y muestra mensajes de ayuda cuando una acción no es posible (duplicado, estructura llena, JSON inválido).
- Lea siempre el mensaje mostrado en la interfaz tras cada operación: son el registro inmediato del éxito o fallo.

Fin de la sección de Búsquedas Internas

Manual de Usuario - Busquedas Externas

Esta sección del documento agrupa las vistas relacionadas con Busquedas externas del proyecto. Cada sección sigue la misma plantilla: descripción, creación, gestión de aristas/nodos, importación/guardado, ejecución de algoritmos (si aplica), interpretación de resultados, reinicio y resumen de botones con los textos exactos de la interfaz.

Búsqueda Lineal Externa

Introducción

La herramienta de **Búsqueda Lineal Externa** te permite trabajar con grandes conjuntos de datos que están organizados en bloques de memoria. Esta técnica es útil cuando manejas información que no cabe toda en la memoria principal y necesitas buscar elementos de forma ordenada y eficiente.

¿Qué es la Búsqueda Lineal Externa?

Imagina que tienes una biblioteca muy grande donde los libros están ordenados alfabéticamente en diferentes estantes (bloques). Para encontrar un libro específico, revisas cada estante en orden hasta encontrar el que buscas. La búsqueda lineal externa funciona de manera similar: divide los datos en bloques ordenados y los busca secuencialmente.

Esta técnica mantiene tus datos **ordenados** en cada bloque, lo que facilita su búsqueda y gestión.

Paso 1: Configurar la Capacidad

Configuración Inicial

1. **Ingresar la capacidad total:** Escribe cuántos elementos en total puede almacenar tu estructura

- Ejemplo: **100** significa que puedes guardar hasta 100 elementos

2. **Hacer clic en "Crear":** El sistema calculará automáticamente:

- **Capacidad del bloque:** Cuántos elementos caben en cada bloque
- **Número de bloques:** Cuántos bloques se necesitan para tu capacidad total

Información Calculada

Después de crear la estructura, verás:

- **Capacidad total:** El valor que ingresaste
- **Capacidad por bloque:** Cuántos elementos caben en cada bloque (calculado automáticamente)
- **Número de bloques:** Cuántos bloques se crearán para almacenar todos los datos

Ejemplo:

```
Si ingresas capacidad = 100
El sistema podría calcular:
- Capacidad por bloque = 10
- Número de bloques = 10
```

Paso 2: Insertar Elementos

Una vez configurada la estructura, puedes comenzar a agregar elementos.

Cómo Insertar Elementos

1. **Ingresar el elemento:** Escribe el número que deseas agregar

- Solo acepta números enteros positivos
- Ejemplo: **25, 150, 7**

2. **Hacer clic en "Insertar":** El elemento se agregará automáticamente en:

- Su posición correcta dentro del bloque correspondiente
- Manteniendo el orden de menor a mayor

Características de la Inserción

Ordenamiento automático:

- Los elementos se insertan en orden creciente
- Si insertas: 50, 20, 80, 30
- Se almacenarán como: 20, 30, 50, 80

Distribución por bloques:

- El sistema decide automáticamente en qué bloque guardar cada elemento
- Cada bloque mantiene su propio orden interno
- Cuando un bloque se llena, el siguiente elemento va al siguiente bloque

Visualización de Bloques

Verás una representación visual de tus bloques:

Cada bloque muestra:

- **Número del bloque:** Bloque 0, Bloque 1, etc.
- **Elementos almacenados:** Los números dentro del bloque en orden
- **Capacidad usada:** Cuántos elementos hay vs. cuántos caben

Ejemplo visual:

```
Bloque 0: [5, 12, 23, 45] (4/10)
Bloque 1: [67, 89, 91] (3/10)
Bloque 2: [102, 150] (2/10)
```

Paso 3: Buscar Elementos

Puedes buscar si un elemento específico existe en tu estructura.

Cómo Realizar una Búsqueda

1. **Ingresar el elemento a buscar:** Escribe el número que deseas encontrar
 - Ejemplo: 45
2. **Hacer clic en "Buscar":** El sistema realizará la búsqueda

Proceso de Búsqueda

El sistema seguirá estos pasos automáticamente:

1. **Revisa bloque por bloque:** Empieza desde el primer bloque
2. **Busca dentro de cada bloque:** Examina los elementos en orden
3. **Se detiene al encontrar:** Cuando encuentra el elemento o determina que no existe

Resultados de la Búsqueda

Si encuentra el elemento:

- Mensaje: "Elemento encontrado"
- Muestra: En qué bloque está
- Muestra: En qué posición dentro del bloque
- Ejemplo: "Encontrado en Bloque 1, posición 2"

Si no encuentra el elemento:

- Mensaje: "Elemento no encontrado"
- Indica que el elemento no existe en ningún bloque

Visualización del Proceso

La búsqueda te mostrará:

- **Bloques revisados:** Cuántos bloques examinó
 - **Accesos realizados:** Cuántas veces tuvo que leer datos
 - **Bloque final:** Dónde encontró el elemento o dónde terminó la búsqueda
-

Paso 4: Guardar y Abrir Estructuras

Guardar tu Trabajo

Cuando hayas creado tu estructura con sus elementos, puedes guardarla para usarla después.

Pasos para guardar:

1. **Hacer clic en "Guardar"**
2. **Elegir ubicación:** Selecciona dónde guardar el archivo
3. **Nombrar el archivo:** Dale un nombre descriptivo
 - Ejemplo: `busqueda_lineal_100elementos.json`
4. **Confirmar:** El archivo se guardará en formato JSON

Qué se guarda:

- Configuración completa (capacidad, bloques)
- Todos los elementos insertados
- Distribución en bloques

Abrir una Estructura Guardada

Si quieras continuar trabajando con una estructura anterior:

Opción 1: Abrir al inicio

- Haz clic en "**Abrir**" antes de crear una nueva estructura
- Selecciona el archivo JSON guardado previamente
- La estructura se cargará con todos sus elementos

Opción 2: Abrir después de crear

- También puedes abrir una estructura después de haber creado una nueva

- Los datos actuales serán reemplazados por los del archivo

Qué se restaura:

- Configuración original (capacidad, número de bloques)
 - Todos los elementos en sus posiciones
 - Visualización completa de bloques
-

Consejos y Mejores Prácticas

Planificación de Capacidad

Elige una capacidad adecuada:

- Considera cuántos elementos necesitarás almacenar
- Deja espacio extra para crecimiento futuro
- Ejemplo: Si necesitas 80 elementos, considera usar capacidad de 100

Inserción Eficiente

Orden de inserción:

- No importa en qué orden insertes los elementos
- El sistema siempre los mantendrá ordenados
- Insertar en cualquier orden: 90, 10, 50, 30
- Resultado: 10, 30, 50, 90

Evita duplicados:

- Verifica que un elemento no exista antes de insertarlo
- Usa la función de búsqueda para confirmarlo

Búsqueda Efectiva

Búsqueda lineal es mejor cuando:

- Tienes conjuntos pequeños o medianos de datos
- Los datos están naturalmente ordenados
- No necesitas búsquedas extremadamente rápidas

Limitaciones:

- La búsqueda puede ser lenta con muchos bloques
- Debe revisar cada bloque secuencialmente hasta encontrar el elemento

Gestión de Archivos

Nombres descriptivos:

- Usa nombres que indiquen el contenido
- Ejemplos:

- `lineal_ordenado_mayo2024.json`
- `elementos_100_ordenados.json`

Respaldos regulares:

- Guarda versiones de tu trabajo periódicamente
 - Mantén copias de seguridad importantes
-

Ejemplo Completo Paso a Paso

Caso Práctico: Sistema de Inventario

Objetivo: Crear una estructura para buscar códigos de productos ordenados.

Paso 1: Configurar

```
Capacidad total: 50
→ Resultado: 5 bloques de 10 elementos cada uno
```

Paso 2: Insertar productos

```
Insertamos códigos: 105, 203, 178, 145, 290
→ Sistema los ordena: 105, 145, 178, 203, 290
→ Distribución:
  Bloque 0: [105, 145, 178, 203, 290]
```

Paso 3: Buscar un producto

```
Buscar: 178
→ Proceso:
  1. Revisa Bloque 0
  2. Encuentra 178 en posición 3
→ Resultado: "Encontrado en Bloque 0, posición 3"
```

Paso 4: Guardar

```
Guardar como: inventario_productos_mayo.json
→ Archivo guardado con 5 elementos en 1 bloque
```

Solución de Problemas Comunes

No puedo insertar más elementos

Problema: El mensaje dice que la estructura está llena.

Solución:

- Has alcanzado la capacidad máxima
- Opciones:
 1. Crear una nueva estructura con mayor capacidad
 2. Eliminar elementos innecesarios (si la herramienta lo permite)
 3. Guardar la estructura actual y crear una nueva

La búsqueda es muy lenta

Problema: Tarda mucho en encontrar elementos.

Solución:

- La búsqueda lineal revisa bloque por bloque
- Es normal con muchos bloques
- Considera:
 1. Usar búsqueda binaria externa si necesitas mayor velocidad
 2. Reducir el número total de elementos
 3. Verificar que los datos estén correctamente ordenados

No se visualizan los bloques

Problema: Los bloques no aparecen después de crear la estructura.

Solución:

- Verifica que ingresaste una capacidad válida (número positivo)
- Asegúrate de hacer clic en "Crear"
- Actualiza la página si es necesario

Error al abrir archivo guardado

Problema: El archivo no se carga correctamente.

Solución:

- Verifica que el archivo sea un JSON válido
- Asegúrate de seleccionar un archivo guardado por esta herramienta
- El archivo no debe haber sido modificado manualmente

Resumen de Funciones

Función	Descripción	Cuándo usarla
Crear	Configura la capacidad y bloques	Al inicio, antes de insertar elementos
Insertar	Agrega elementos ordenadamente	Para poblar la estructura con datos

Función	Descripción	Cuándo usarla
Buscar	Encuentra elementos específicos	Para verificar si un elemento existe
Guardar	Exporta la estructura a archivo	Para preservar tu trabajo
Abrir	Importa estructura guardada	Para continuar trabajo previo

Conceptos Clave

Bloque:

- Unidad de almacenamiento que contiene varios elementos
- Tiene capacidad limitada
- Mantiene sus elementos ordenados

Ordenamiento:

- Los elementos se mantienen de menor a mayor
- Facilita la búsqueda y gestión
- Se realiza automáticamente al insertar

Búsqueda secuencial:

- Revisa elementos uno por uno
- Examina bloques en orden
- Se detiene al encontrar el elemento o llegar al final

Capacidad:

- Límite máximo de elementos que puede almacenar
- Se divide entre los bloques disponibles
- Define el tamaño de tu estructura

Esta herramienta de búsqueda lineal externa te proporciona una forma sencilla y ordenada de gestionar conjuntos de datos, manteniéndolos organizados en bloques para facilitar su búsqueda y manipulación. Recuerda que el orden automático y la división en bloques son las características clave que hacen eficiente esta técnica para trabajar con datos externos.

Búsqueda Binaria Externa

Introducción

La herramienta de **Búsqueda Binaria Externa** te permite trabajar con grandes conjuntos de datos organizados en bloques, utilizando una técnica de búsqueda más rápida que la búsqueda lineal. Esta técnica es especialmente útil cuando manejas muchos elementos y necesitas encontrarlos rápidamente.

Paso 1: Configurar la Capacidad

Configuración Inicial

1. **Ingresar la capacidad total:** Escribe cuántos elementos en total puede almacenar tu estructura

- Ejemplo: 200 significa que puedes guardar hasta 200 elementos

2. **Hacer clic en "Crear":** El sistema calculará automáticamente:

- **Capacidad del bloque:** Cuántos elementos caben en cada bloque
- **Número de bloques:** Cuántos bloques se necesitan

Información Calculada

Después de crear la estructura, verás:

- **Capacidad total:** El valor que ingresaste
- **Capacidad por bloque:** Elementos por bloque (calculado automáticamente)
- **Número de bloques:** Total de bloques creados

Ejemplo:

```
Si ingresas capacidad = 200  
El sistema podría calcular:  
- Capacidad por bloque = 10  
- Número de bloques = 20
```

Paso 2: Insertar Elementos

Una vez configurada la estructura, puedes comenzar a agregar elementos que se mantendrán ordenados automáticamente.

Cómo Insertar Elementos

1. **Ingresar el elemento:** Escribe el número que deseas agregar

- Solo acepta números enteros positivos
- Ejemplo: 45, 178, 23

2. **Hacer clic en "Insertar":** El elemento se agregará:

- En su posición correcta para mantener el orden
- Dentro del bloque correspondiente

Características de la Inserción

Ordenamiento global:

- Todos los elementos se mantienen ordenados en toda la estructura

- No solo dentro de cada bloque, sino entre bloques
- Si insertas: 100, 50, 150, 25
- Se almacenarán como: 25, 50, 100, 150

Distribución por bloques:

- Los elementos se distribuyen en bloques manteniendo el orden global
- Los bloques anteriores contienen elementos menores
- Los bloques posteriores contienen elementos mayores

Ejemplo de distribución ordenada:

```
Bloque 0: [5, 12, 23, 35, 41]      ← Elementos más pequeños  
Bloque 1: [56, 67, 78, 89, 95]    ← Elementos medianos  
Bloque 2: [102, 145, 167, 189]    ← Elementos más grandes
```

Visualización de Bloques

Verás una representación visual de tus bloques ordenados:

Cada bloque muestra:

- **Número del bloque:** Bloque 0, Bloque 1, etc.
- **Elementos en orden:** Los números dentro del bloque
- **Capacidad usada:** Cuántos elementos tiene vs. cuántos caben

Paso 3: Buscar Elementos

Cómo Realizar una Búsqueda

1. **Ingresar el elemento a buscar:** Escribe el número que deseas encontrar

- Ejemplo: 89

2. **Hacer clic en "Buscar":** El sistema realizará la búsqueda binaria

Proceso de Búsqueda Binaria

El sistema sigue este proceso optimizado:

1. **Determina el bloque correcto:**

- Examina los rangos de cada bloque
- Identifica en qué bloque debería estar el elemento

2. **Búsqueda binaria dentro del bloque:**

- Compara con el elemento del medio
- Descarta la mitad que no contiene el elemento
- Repite hasta encontrarlo o determinar que no existe

Resultados de la Búsqueda

Si encuentra el elemento:

- Mensaje: "Elemento encontrado"
- Indica: Bloque donde está
- Indica: Posición dentro del bloque
- Ejemplo: "Encontrado en Bloque 5, posición 3"

Si no encuentra el elemento:

- Mensaje: "Elemento no encontrado"
-

Paso 5: Guardar y Abrir Estructuras

Guardar tu Trabajo

Cuando hayas creado tu estructura con sus elementos ordenados, puedes guardarla.

Pasos para guardar:

1. Hacer clic en "Guardar"
 - Ejemplo: [busqueda_binaria_200elementos.json](#)

Qué se guarda:

- Configuración completa (capacidad, bloques)
- Todos los elementos en orden
- Distribución en bloques

Abrir una Estructura Guardada

Para continuar con un trabajo anterior:

Opción 1: Abrir al inicio

- Haz clic en "**Abrir**" antes de crear una nueva estructura
- Selecciona el archivo JSON guardado
- La estructura se cargará completamente

Opción 2: Abrir después de crear

- Puedes abrir después de crear una estructura
- Los datos actuales serán reemplazados

Qué se restaura:

- Configuración original
- Todos los elementos en sus posiciones
- Orden global mantenido
- Visualización completa

Búsqueda Hash Externa

Introducción

La herramienta de **Búsqueda Hash Externa** te permite trabajar con grandes conjuntos de datos utilizando una función hash para organizar y encontrar elementos muy rápidamente.

Paso 1: Configurar la Estructura Hash

Configuración Inicial

Antes de usar la estructura, debes configurar varios parámetros importantes:

1. **Capacidad total:** Cuántos elementos puedes almacenar en total
 - Ejemplo: **100**
2. **Dígitos de la clave:** Cuántos dígitos tiene cada número que insertarás
 - Ejemplo: **3** (para números de 3 dígitos como 123, 456, 789)
 - Todos los elementos deben tener esta cantidad de dígitos
3. **Método de resolución de colisiones:** Cómo manejar cuando dos elementos quieren la misma posición
 - **Estructura secundaria:** Cada posición puede tener múltiples elementos en una lista
 - **Área de colisiones:** Los elementos que colisionan van a un área especial
4. **Hacer clic en "Crear":** La estructura se inicializará

Información Calculada

Después de crear, verás:

- **Número de cubetas:** Cuántas posiciones principales hay
- **Capacidad por cubeta:** Cuántos elementos caben en cada posición
- **Capacidad área de colisiones** (si aplica): Espacio para elementos que colisionan

Ejemplo:

```
Capacidad: 100
Dígitos de clave: 3
Resolución: Estructura secundaria
→ Resultado:
  - 10 cubetas
  - 10 elementos por cubeta
```

Paso 2: Entender los Métodos de Resolución de Colisiones

Opción 1: Estructura Secundaria

Cómo funciona:

- Cada posición (cubeta) puede contener múltiples elementos
- Los elementos que colisionan se almacenan juntos en una lista
- Cada cubeta tiene su propia capacidad

Opción 2: Área de Colisiones

Cómo funciona:

- Cada cubeta principal tiene espacio para un solo elemento
- Si hay colisión, el elemento extra va a un área especial de colisiones
- El área de colisiones es un espacio adicional separado

Paso 3: Insertar Elementos

Cómo Insertar Elementos

1. Ingresar el elemento:

- Escribe el número que deseas agregar
- **Importante:** Debe tener exactamente el número de dígitos configurado
 - Si configuraste 3 dígitos, solo acepta números de 3 dígitos (100-999)
 - Ejemplo válido: **234, 567, 890**
 - Ejemplo inválido: **23** (solo 2 dígitos), **1234** (4 dígitos)

2. Hacer clic en "Insertar":

- El sistema:
- Calcula la posición usando función módulo
 - Verifica si hay espacio disponible
 - Coloca el elemento en la posición correcta o en área de colisiones

Proceso de Inserción

Paso a paso:

1. Cálculo de hash:

Sistema aplica el hash seleccionado al número

Ejemplo: $234 \% 10 = 4$
→ El elemento debería ir a la cubeta 4

2. Verificación de espacio:

- Si la cubeta tiene espacio → Inserta ahí
- Si la cubeta está llena → Maneja la colisión según configuración

3. Manejo de colisiones:

- **Estructura secundaria:** Agrega a la lista de la cubeta (si hay espacio)
 - **Área de colisiones:** Mueve a área especial de colisiones
-

Paso 4: Buscar Elementos

La búsqueda hash es extremadamente rápida porque calcula directamente dónde debe estar el elemento.

Cómo Realizar una Búsqueda

1. **Ingresar el elemento a buscar:** Escribe el número

- Debe tener el número correcto de dígitos
- Ejemplo: 234

2. **Hacer clic en "Buscar":** El sistema realiza la búsqueda

Proceso de Búsqueda

Búsqueda directa:

1. **Calcular hash:** Sistema aplica función hash seleccionada previamente

Ejemplo: $234 \% 10 = 4$
→ Busca directamente en cubeta 4

2. **Revisar cubeta:**

- **Estructura secundaria:** Revisa todos los elementos en la lista de esa cubeta
- **Área de colisiones:** Primero revisa cubeta, luego área de colisiones si es necesario

3. **Resultado inmediato:**

- Encuentra el elemento o determina que no existe
- Muy pocas comparaciones necesarias

Resultados de la Búsqueda

Si encuentra el elemento:

- Mensaje: "Elemento encontrado"
- Ubicación: Qué cubeta lo contiene
- Si está en área de colisiones, lo indica

Si no encuentra el elemento:

- Mensaje: "Elemento no encontrado"
- Sistema verificó solo la cubeta correspondiente
- Muy rápido incluso cuando no existe

Paso 5: Guardar y Abrir Estructuras

Guardar tu Trabajo

Cuando hayas creado tu estructura hash con sus elementos, puedes guardarla.

Pasos para guardar:

1. Hacer clic en "Guardar"

Qué se guarda:

- Configuración completa (capacidad, dígitos, método de colisiones)
- Todas las cubetas con sus elementos
- Área de colisiones (si aplica)
- Función hash utilizada

Abrir una Estructura Guardada

Para continuar con un trabajo anterior:

Opción 1: Abrir al inicio

- Haz clic en "Abrir" antes de crear
- Selecciona el archivo JSON
- La estructura se cargará completamente

Opción 2: Abrir después de crear

- Puedes abrir después de crear una estructura
- Los datos actuales serán reemplazados

Qué se restaura:

- Configuración original
- Todos los elementos en sus posiciones
- Método de resolución de colisiones
- Estado completo de cubetas y colisiones

Estructuras Dinámicas

Introducción

La herramienta de **Estructuras Dinámicas** te permite trabajar con tablas hash que pueden crecer y reducirse automáticamente según la cantidad de datos.

Paso 1: Entender la Estructura Dinámica

Componentes Principales

Cubetas:

- Contenedores numerados que almacenan los elementos
- Se organizan usando función hash módulo
- Ejemplo: Con 8 cubetas, están numeradas de 0 a 7

Registros por Cubeta:

- Cuántos elementos caben en cada cubeta
- Define la capacidad base antes de usar desbordamiento
- Ejemplo: 2 registros por cubeta

Desbordamientos:

- Área adicional cuando una cubeta se llena
- Los elementos extra se almacenan temporalmente aquí
- Se resuelven con la próxima expansión

Conceptos

Expansión:

- Cuando la ocupación supera 75%, la estructura crece
- Agrega más cubetas para dar más espacio
- Redistribuye elementos según nueva función hash

Reducción:

- Cuando la densidad es baja (≤ 0.8 registros/cubeta), la estructura se reduce
- Elimina cubetas innecesarias
- Redistribuye elementos en menos cubetas

Paso 2: Configurar la Estructura Dinámica

Configuración Inicial

Para crear una estructura dinámica, configura:

1. **Cantidad de cubetas:** Cuántas cubetas iniciales tendrá

- Ejemplo: 8 cubetas
- La estructura puede crecer o reducirse después

2. **Registros por cubeta:** Capacidad de cada cubeta

- Ejemplo: 2 registros por cubeta
- Determina cuándo usar desbordamiento

3. **Tamaño de la clave:** Cuántos dígitos tiene cada elemento

- Ejemplo: 2 (para números de 2 dígitos: 10-99)
- Todos los elementos deben tener esta cantidad de dígitos

4. **Tipo de expansión/reducción:** Cómo crece o se reduce

- **Total:** Duplica o divide a la mitad de una vez ($8 \rightarrow 16$ o $16 \rightarrow 8$)
- **Parcial:** Agrega o quita cubetas gradualmente (2 parciales = 1 total)

5. **Hacer clic en "Crear Estructura":** La estructura se inicializa

Información Calculada

Después de crear, verás:

- **Capacidad total:** Cubetas × Registros por cubeta
 - **Umbral de expansión:** 75% de ocupación
 - **Umbral de reducción:** Densidad ≤ 0.8 registros/cubeta
 - **Función hash:** Módulo sobre el número actual de cubetas
-

Paso 3: Insertar Elementos

Cómo Insertar Elementos

1. **Ingresar el elemento:** Escribe el número

- Debe tener el número correcto de dígitos
- Ejemplo: Si configuraste 2 dígitos, usa 10-99

2. **Hacer clic en "Insertar"** (o presiona Enter): El sistema:

- Calcula la cubeta con hash módulo
- Inserta en la cubeta correspondiente
- Usa desbordamiento si la cubeta está llena
- Expande si se alcanza el umbral

Proceso de Inserción Normal

Cuando hay espacio en la cubeta: Insertar en cubeta: Coloca el elemento en el primer espacio libre

Inserción con Desbordamiento

Cuando la cubeta está llena:

1. **Intentar insertar:** Sistema detecta cubeta llena
2. **Usar desbordamiento:** Elemento va a área de desbordamiento
3. **Marcar para expansión:** Si se alcanza 75% de ocupación

Expansión Automática

Expansión Total:

- Duplica el número de cubetas ($8 \rightarrow 16$)
- Redistribuye todos los elementos

Expansión Parcial:

- Agrega cubetas gradualmente
- 2 expansiones parciales = 1 total
- Menos cambios drásticos
- Mejor para inserciones distribuidas

Paso 4: Buscar Elementos

Cómo Realizar una Búsqueda

1. **Ingresar el elemento a buscar:** Escribe el número
2. **Hacer clic en "Buscar":** El sistema realiza la búsqueda

Proceso de Búsqueda

Búsqueda directa:

1. **Calcular hash:** `elemento % cubetas`
2. **Ir a la cubeta:** Directamente a la cubeta calculada
3. **Buscar en cubeta:** Revisar registros
4. **Buscar en desbordamiento:** Si no está en cubeta principal

Resultados de la Búsqueda

Si encuentra el elemento:

- Mensaje: "Elemento encontrado en Cubeta X"

Si no encuentra el elemento:

- Mensaje: "Elemento no encontrado"
- Sistema verificó cubeta y desbordamiento

Paso 5: Eliminar Elementos

Cómo Eliminar Elementos

1. **Ingresar el elemento a eliminar:** Escribe el número
2. **Hacer clic en "Eliminar":** El sistema:
 - Busca el elemento
 - Lo elimina de cubeta o desbordamiento
 - Verifica si debe reducir la estructura

Reducción Automática

Reducción Total:

- Divide a la mitad el número de cubetas ($16 \rightarrow 8$)
- Cuando densidad ≤ 0.8 registros/cubeta
- Redistribuye todos los elementos

Reducción Parcial:

- Elimina cubetas gradualmente
- 2 reducciones parciales = 1 total
- Menos redistribución

Paso 6: Visualizar la Estructura

Componentes de la Visualización

Información de estado:

- Cubetas actuales
- Registros por cubeta
- Capacidad total
- Registros ocupados
- Porcentaje de ocupación
- Densidad

Cubetas:

- Numeradas desde 0
- Muestra registros dentro de capacidad
- Registros vacíos se muestran como "-"
- Elementos insertados recientemente resaltados

Desbordamientos:

- Se muestran debajo de cada cubeta
- Solo si la cubeta tiene desbordamientos
- Numerados como D0, D1, D2...

Paso 7: Guardar y Abrir Estructuras

Guardar tu Trabajo

Pasos para guardar:

1. Hacer clic en "Guardar Estructura"
2. Elegir ubicación
3. Nombrar archivo:
 - Ejemplo: [dinamica_8cubetas.json](#)
4. Confirmar

Qué se guarda:

- Configuración completa
- Todas las cubetas con sus elementos
- Desbordamientos
- Contadores de expansiones/reducciones
- Estado actual completo

Abrir una Estructura Guardada

Opción 1: Abrir al inicio

- Haz clic en "**Abrir Estructura**" antes de crear
- Selecciona archivo JSON
- La estructura se carga completamente

Opción 2: Abrir después de crear

- Usa el botón "**Abrir Estructura**" en operaciones
- Reemplaza estructura actual

Qué se restaura:

- Configuración original
 - Todos los elementos en sus posiciones
 - Desbordamientos
 - Contadores
 - Visualización completa
-

Índices Externos

Introducción

La herramienta de **Índices Externos** te permite crear estructuras de índice para acceder eficientemente a archivos de datos grandes. Los índices actúan como una guía que te dice dónde encontrar información sin tener que revisar todo el archivo.

Paso 1: Tipos de Índices

Índice Primario vs. Secundario

Índice Primario:

- El archivo de datos está ordenado por la clave principal
- Necesita una entrada por cada bloque

Índice Secundario:

- El archivo de datos no está ordenado por el campo indexado
- Necesita una entrada por cada registro

Estructura Simple vs. Multinivel

Simple:

- Un solo nivel de índice que apunta directamente al archivo de datos
- Menos accesos pero requiere más espacio

Multinivel:

- Varios niveles de índices (índice del índice)
 - Usa menos espacio en memoria
 - Útil cuando el índice simple es muy grande
-

Paso 2: Configurar Índices

Parámetros de Configuración

1. **Tipo de índice:** Primario o Secundario
2. **Estructura:** Simple o Multinivel
3. **Cantidad de registros (r):** Total de registros en el archivo
4. **Longitud registro dato (bytes):** Tamaño de cada registro en el archivo principal
5. **Longitud registro índice (bytes):** Tamaño de cada entrada en el índice
6. **Capacidad del bloque (B - bytes):** Tamaño de cada bloque
7. **Hacer clic en "Crear Estructura"**

Cálculos Automáticos

El sistema calculará:

Para el archivo de datos:

- **bfr:** Registros por bloque = $\lceil \frac{LB}{\text{Longitud registro dato}} \rceil$
- **b:** Bloques necesarios = $\lceil \frac{r}{bfr} \rceil$

Para el índice:

- **bfri:** Entradas de índice por bloque = $\lceil \frac{LB}{\text{Longitud registro índice}} \rceil$
- **bi:** Bloques de índice
 - Primario: $\lceil \frac{b}{bfri} \rceil$
 - Secundario: $\lceil \frac{r}{bfri} \rceil$

Para multinivel (si aplica):

- Número de niveles necesarios
 - Bloques por cada nivel
-

Paso 3: Entender los Índices

Índice Primario

- El archivo está ordenado por la clave
- El índice guarda solo el primer registro de cada bloque

- Para buscar: encuentra el bloque correcto en el índice, luego busca dentro del bloque
- Más eficiente en espacio porque no necesita una entrada por cada registro

Índice Secundario

- El archivo no está ordenado por el campo del índice
- El índice necesita una entrada por cada registro
- Para buscar: encuentra todos los bloques que contienen el valor buscado
- Permite buscar por campos que no son la clave principal

Índice Multinivel

- Cuando el índice simple es muy grande, se crea un índice del índice
- Reduce el espacio en memoria principal
- Requiere más accesos pero cada acceso es a estructuras más pequeñas
- El nivel superior se mantiene en memoria para mayor velocidad

Paso 4: Visualizar la Estructura

Vista Simple

Muestra dos columnas lado a lado:

- **Izquierda:** Estructura de Índices (bloques de índice con sus entradas)
- **Derecha:** Estructura Principal (bloques del archivo de datos)

Cada entrada del índice apunta a un bloque o registro en el archivo principal.

Vista Multinivel

Muestra múltiples columnas:

- **Niveles superiores** a la izquierda (nivel más alto primero)
- **Nivel 1** apunta a la estructura principal
- **Estructura principal** a la derecha

Solo se muestran el primer, medio y último bloque de cada estructura para claridad.

Paso 5: Guardar y Abrir

Guardar

1. Hacer clic en "Guardar"
2. Elegir ubicación y nombre
3. Se guarda la configuración completa y estructura

Abrir

Opción 1: Antes de crear

- Usar botón "Abrir Estructura" en la pantalla inicial
- Seleccionar archivo JSON

Opción 2: Despues de crear

- Usar botón "Abrir Estructura" en las operaciones
- Reemplaza la estructura actual

Paso 6: Resetear Estructura

Si necesitas empezar de nuevo:

1. Hacer clic en "Resetear"
2. Confirmar en el modal
3. Se borra toda la configuración y datos

Fin de la sección de Búsquedas Externas

Manual de Usuario - Grafos

Esta sección del documento agrupa las vistas relacionadas con grafos del proyecto. Cada sección sigue la misma plantilla: descripción, creación, gestión de aristas/nodos, importación/guardado, ejecución de algoritmos (si aplica), interpretación de resultados, reinicio y resumen de botones con los textos exactos de la interfaz.

Árboles de Expansión

Introducción

La herramienta de **Árboles de Expansión** te permite crear un grafo ponderado y calcular su árbol de expansión mínimo o máximo. Un árbol de expansión es un subconjunto de las aristas del grafo que conecta todos los nodos sin formar ciclos, y el árbol de expansión mínimo es aquel cuya suma de pesos de las aristas es la menor posible.

¿Qué es un Árbol de Expansión?

Imagina que tienes varias ciudades y quieres conectarlas todas con carreteras, pero cada carretera tiene un costo diferente. Un árbol de expansión mínimo te ayuda a encontrar la manera más económica de conectar todas las ciudades, asegurándote de que todas estén conectadas pero sin gastar de más en rutas redundantes.

Paso 1: Crear tu Grafo

Configuración Inicial

- 1. Ingresar cantidad de nodos:** Escribe cuántos nodos (puntos) deseas en tu grafo. Por ejemplo, si escribes "6", tendrás 6 nodos numerados del 1 al 6.
- 2. Hacer clic en "Crear Grafo":** Una vez que hayas ingresado la cantidad de nodos, haz clic en el botón para crear tu grafo.

Importar un Grafo Existente

Si ya tienes un grafo guardado anteriormente, puedes importarlo:

- Haz clic en "**Abrir Grafo**"
- Selecciona el archivo JSON que guardaste previamente
- El grafo se cargará automáticamente con todas sus aristas

Paso 2: Agregar Aristas (Conexiones)

Una vez creado el grafo, puedes conectar los nodos mediante aristas con pesos.

Cómo Agregar una Arista

- 1. Seleccionar nodos a conectar:** En el campo de texto, escribe los dos nodos que quieras conectar.
 - Puedes escribir: **12** (sin espacio) o **1 2** (con espacio)
 - Ambos formatos crean una conexión entre el nodo 1 y el nodo 2
- 2. Ingresar el peso:** En el campo "Peso", escribe un número entero positivo (por ejemplo: 5, 10, 15).
 - **Importante:** El peso debe ser un número entero mayor o igual a 1
- 3. Agregar la arista:** Haz clic en el botón "**Agregar Arista**" o presiona Enter

Visualización del Grafo

- Verás tu grafo representado visualmente con:
 - **Círculos naranjas:** Representan los nodos
 - **Líneas grises:** Representan las aristas
 - **Números en las líneas:** Muestran los pesos de cada arista

Información del Grafo

Debajo de la visualización verás:

- **V** = Conjunto de vértices (nodos)
- **A** = Conjunto de aristas con sus conexiones
- **Cantidad de aristas:** Número total de conexiones

Paso 3: Calcular Árbol de Expansión

Árbol de Expansión Mínimo

1. Haz clic en el botón "**Árbol de Expansión Mínimo**"
2. El sistema calculará automáticamente el árbol con el menor peso total posible
3. Verás:
 - Un mensaje indicando el peso total del árbol
 - La visualización se actualizará mostrando solo las aristas seleccionadas

Árbol de Expansión Máximo

1. Haz clic en el botón "**Árbol de Expansión Máximo**"
2. El sistema calculará el árbol con el mayor peso total posible
3. Similar al mínimo, verás el resultado con el peso total

Información mostrada:

- **T** = Conjunto de aristas del árbol de expansión
 - **Peso Total**: Suma de todos los pesos de las aristas seleccionadas
-

Paso 4: Generar Árbol Complemento

El árbol complemento (T') contiene todas las aristas del grafo original que NO están en el árbol de expansión.

Cómo Generar el Complemento

1. Primero debes calcular un árbol de expansión (mínimo o máximo)
2. Haz clic en "**Generar Árbol Complemento (T')**"
3. Verás una nueva visualización con:
 - Las aristas que NO fueron seleccionadas en el árbol de expansión
 - T' = Conjunto de aristas complementarias

¿Para qué sirve?

El árbol complemento te muestra las conexiones "alternativas" o "redundantes" que no se necesitan para conectar todos los nodos. Es útil para:

- Identificar rutas de respaldo
 - Analizar la estructura del grafo original
 - Estudiar las diferencias entre el árbol mínimo y el grafo completo
-

Paso 5: Guardar y Abrir Grafos

Guardar tu Trabajo

1. Haz clic en "**Guardar Grafo**"
2. Se descargará un archivo JSON con:
 - La configuración del grafo
 - Todos los nodos y aristas
 - El árbol de expansión calculado (si existe)
3. El archivo se guardará con un nombre automático que incluye la fecha

Abrir un Grafo Guardado

1. Haz clic en "**Abrir Grafo**"
 2. Selecciona el archivo JSON que guardaste
 3. Todo el grafo se restaurará:
 - Nodos y aristas originales
 - Árbol de expansión (si estaba calculado)
 - Árbol complemento (si estaba generado)
-

Paso 6: Reiniciar

Si deseas empezar de nuevo:

1. Haz clic en "**Resetear Grafo**" (botón rojo)
 2. Se borrarán todos los datos:
 - Nodos y aristas
 - Árboles calculados
 - Visualizaciones
 3. Podrás crear un nuevo grafo desde cero
-

Restricciones

- Los pesos de las aristas deben ser **números enteros positivos** (1, 2, 3, etc.)
 - No puedes crear aristas de un nodo consigo mismo (bucles)
 - No puedes crear aristas duplicadas entre los mismos nodos
 - Debes tener al menos 2 nodos para crear aristas
-

Distancia entre Árboles de Expansión

Introducción

La herramienta de **Distancia entre Árboles de Expansión** te permite comparar dos grafos diferentes calculando sus árboles de expansión mínimos y determinando qué tan diferentes son entre sí. Esta métrica de distancia es útil para analizar similitudes y diferencias entre estructuras de red.

¿Qué es la Distancia entre Árboles?

Imagina que tienes dos diseños diferentes de red de transporte para conectar las mismas ciudades. La distancia entre árboles te dice qué tan diferentes son estos dos diseños, considerando tanto las rutas que comparten como las que son únicas de cada uno.

Un valor de distancia **bajo** significa que los árboles son muy similares, mientras que un valor **alto** indica que son muy diferentes.

Paso 1: Crear los Dos Grafos

Configuración Inicial

1. **Ingresar cantidad de nodos para Grafo 1:** Escribe cuántos nodos deseas en el primer grafo (ejemplo: 5 nodos)
2. **Ingresar cantidad de nodos para Grafo 2:** Escribe cuántos nodos deseas en el segundo grafo (ejemplo: 6 nodos)
 - Los grafos pueden tener diferente cantidad de nodos
3. **Hacer clic en "Crear Grafos":** Ambos grafos se crearán simultáneamente

Importar Grafos Existentes

Si ya tienes grafos guardados:

Antes de crear:

- Haz clic en "**Abrir Grafos**"
- Selecciona el archivo JSON guardado previamente
- Ambos grafos se cargarán automáticamente

Después de crear:

- También puedes importar grafos después de haberlos creado
- Usa el botón "**Abrir Grafos**" en la sección de operaciones

Paso 2: Seleccionar el Grafo a Modificar

Antes de agregar aristas, debes elegir en cuál de los dos grafos deseas trabajar.

Botones de Selección

Verás dos botones claramente marcados:

- **Grafo 1:** Selecciona el primer grafo para modificarlo
- **Grafo 2:** Selecciona el segundo grafo para modificarlo

El botón activo se mostrará resaltado para indicar cuál grafo estás editando actualmente.

Paso 3: Agregar Aristas a los Grafos

Cómo Agregar Aristas

1. **Asegúrate de tener el grafo correcto seleccionado** (verás "Crear Aristas - Grafo 1" o "Crear Aristas - Grafo 2")
2. **Ingresar los nodos a conectar:**
 - Escribe dos números para conectar esos nodos
 - Formatos aceptados: **12** o **1 2**

- Ejemplo: **34** conecta el nodo 3 con el nodo 4

3. Ingresar el peso:

- Escribe un número entero positivo (1, 2, 3, etc.)
- **Importante:** Solo se aceptan números enteros mayores o igual a 1

4. Agregar la arista:

- Haz clic en "**Agregar Arista**" o presiona Enter

5. Cambiar de grafo si es necesario:

- Haz clic en el botón del otro grafo
- Repite el proceso para agregar aristas al segundo grafo

Visualización en Tiempo Real

Verás dos paneles de visualización lado a lado:

Panel Izquierdo - Grafo 1:

- Círculos naranjas representan los nodos
- Líneas grises representan las aristas
- Números en las líneas muestran los pesos

Panel Derecho - Grafo 2:

- Mismo formato de visualización
- Independiente del Grafo 1

Debajo de cada visualización verás:

- **V₁ / V₂** = Conjunto de vértices (nodos)
- **A₁ / A₂** = Conjunto de aristas con sus conexiones

Paso 4: Generar Árboles de Expansión

Una vez que ambos grafos tengan suficientes aristas, puedes calcular sus árboles de expansión mínimos.

Cómo Generar los Árboles

1. Haz clic en "**Generar Árbol de Expansión**"

- Este botón genera ambos árboles simultáneamente
- Cada grafo se transformará mostrando solo las aristas de su árbol de expansión mínimo

2. Visualización Actualizada:

- Los paneles ahora muestran los árboles de expansión en lugar de los grafos completos
- El título cambia a "Árboles de Expansión Mínimos"
- Los subtítulos indican "Árbol de Expansión - Grafo 1" y "Árbol de Expansión - Grafo 2"

3. Información Mostrada:

- T_1 = Conjunto de aristas del árbol 1
- T_2 = Conjunto de aristas del árbol 2
- Los árboles permanecen en el mismo lugar donde estaban los grafos originales

Requisitos para Generar Árboles

Para que el árbol de expansión se pueda calcular correctamente:

- El grafo debe tener al menos $n-1$ aristas (donde n es el número de nodos)
 - Todos los nodos deben estar conectados entre sí (grafo conexo)
-

Paso 5: Calcular la Distancia entre Árboles

Una vez que ambos árboles de expansión han sido generados, puedes calcular qué tan diferentes son.

Cómo Calcular la Distancia

1. Haz clic en "Calcular Distancia de Árboles"

2. Validación Automática:

- El sistema verifica que ambos grafos sean árboles válidos
- Un árbol válido tiene exactamente $n-1$ aristas y no tiene ciclos
- Si alguno no es un árbol válido, recibirás un mensaje de error

3. Resultado:

- Si ambos son árboles válidos, verás un cuadro destacado con:
 - **"Distancia entre Árboles"**
 - El valor numérico de la distancia

¿Cómo se Calcula la Distancia?

La distancia se calcula usando la siguiente fórmula:

$$\text{Distancia} = (\text{Suma de pesos de aristas únicas}) / 2$$

Donde las "aristas únicas" son aquellas que están en uno de los árboles pero no en el otro, considerando tanto las aristas diferentes como las diferencias en sus pesos.

Interpretación del Resultado

- **Distancia = 0:** Los árboles son idénticos (mismas aristas con mismos pesos)
 - **Distancia baja (1-5):** Los árboles son muy similares
 - **Distancia media (6-15):** Los árboles tienen algunas diferencias significativas
 - **Distancia alta (>15):** Los árboles son muy diferentes
-

Paso 6: Guardar y Abrir Grafos

Guardar tu Trabajo

1. **Haz clic en "Guardar Grafos"**
2. Se descargará un archivo JSON que contiene:
 - Configuración de ambos grafos
 - Todos los nodos y aristas de ambos grafos
 - Fecha y hora del guardado
3. El archivo tiene un nombre automático como **distancia-arboles-1733123456789.json**

Abrir Grafos Guardados

1. **Haz clic en "Abrir Grafos"**
 2. Selecciona el archivo JSON
 3. Ambos grafos se restaurarán completamente con:
 - Todos los nodos y aristas
 - Las mismas configuraciones originales
 - Visualizaciones listas para trabajar
-

Paso 7: Reiniciar

Si deseas empezar de nuevo con grafos diferentes:

1. **Haz clic en "Resetear Grafos"** (botón rojo al final)
 2. Se borrarán:
 - Ambos grafos completos
 - Árboles de expansión calculados
 - Distancia calculada
 - Todas las visualizaciones
 3. Volverás al inicio para crear nuevos grafos
-

Flujo de Trabajo Recomendado

1. **Planificación** → Define cuántos nodos necesita cada grafo
 2. **Creación** → Crea ambos grafos con "Crear Grafos"
 3. **Construcción Grafo 1** → Selecciona Grafo 1 y agrega todas sus aristas
 4. **Construcción Grafo 2** → Selecciona Grafo 2 y agrega todas sus aristas
 5. **Guardar** → Guarda los grafos
 6. **Generar Árboles** → Calcula ambos árboles de expansión
 7. **Calcular Distancia** → Obtén la métrica de diferencia
 8. **Analizar** → Interpreta el resultado según tus necesidades
-

Algoritmo de Floyd-Warshall

Descripción General

El Algoritmo de Floyd-Warshall es una herramienta interactiva para calcular las distancias mínimas entre todos los pares de vértices en un grafo ponderado o no ponderado. Esta vista permite crear grafos personalizados, agregar aristas con o sin pesos, y calcular métricas importantes como excentricidades, distancias de vértices, mediana y diámetro del grafo.

Crear un Grafo

Paso 1: Configuración Inicial

Al ingresar a la vista, se presenta un formulario con tres campos de configuración:

1. **Cantidad de Nodos:** Ingrese el número de nodos que tendrá su grafo (ej: 5)

- Los nodos se numerarán automáticamente del 1 al número especificado

2. **Tipo de Grafo:** Seleccione entre:

- **No Dirigido:** Las conexiones entre nodos son bidireccionales ($A \leftrightarrow B$)
- **Dirigido:** Las conexiones tienen una dirección específica ($A \rightarrow B$)

3. **Ponderado:** Seleccione si desea asignar pesos a las aristas:

- **Sí:** Las aristas tendrán valores numéricos (distancias, costos, etc.)
- **No:** Todas las aristas tendrán peso 1

Paso 2: Crear el Grafo

- Haga clic en el botón "**Crear Grafo**"
- El sistema generará los nodos especificados y mostrará la visualización inicial del grafo

Alternativa: Importar un Grafo Existente

Si ya tiene un grafo guardado previamente:

- Haga clic en el botón "**Abrir Grafo**" ubicado debajo del botón "Crear Grafo"
 - Seleccione el archivo JSON con la extensión **.json**
 - El grafo se cargará automáticamente con su configuración y aristas
-

Gestionar Aristas

Agregar Aristas

Una vez creado el grafo, aparecerá la sección "**Crear Aristas**" con los siguientes campos:

1. **Campo de Conexión:** Ingrese los dos nodos que desea conectar

- Formato aceptado: **12** o **1 2** (conecta el nodo 1 con el nodo 2)
- Ejemplo: Para conectar nodos 3 y 5, escriba **35** o **3 5**

2. **Campo de Peso** (solo si el grafo es ponderado):

- Ingrese el valor numérico del peso de la arista
- Ejemplo: [4.5](#) o [10](#)
- Este campo es obligatorio para grafos ponderados

3. Botón "Agregar Arista": Haga clic para confirmar la conexión

- La arista aparecerá inmediatamente en la visualización
- Un mensaje confirmará la operación exitosa

Visualización del Grafo

La visualización interactiva muestra:

- **Nodos:** Círculos naranjas numerados con su identificador
- **Aristas:** Líneas grises que conectan los nodos
- **Pesos:** Números sobre las aristas (si el grafo es ponderado)
- **Direcciones:** Flechas en las aristas (si el grafo es dirigido)

Puede:

- Hacer clic y arrastrar los nodos para reorganizar la visualización
- Acercar o alejar usando la rueda del mouse
- Pasar el cursor sobre elementos para resaltarlos

Guardar y Cargar Grafos

Guardar un Grafo

En la sección de botones de operaciones:

1. Haga clic en "**Guardar Grafo**"
2. El sistema generará un archivo JSON con:
 - Configuración del grafo (tipo, nodos, ponderación)
 - Todas las aristas y sus propiedades
 - Fecha de exportación en el nombre del archivo
3. El archivo se descargará automáticamente con el formato:
 - [floyd_dirigido_ponderado_2025-12-02.json](#)
 - [floyd_no-dirigido_no-ponderado_2025-12-02.json](#)

Abrir un Grafo Guardado

Después de crear un grafo:

1. Haga clic en "**Abrir Grafo**" (al lado de "Guardar Grafo")
2. Seleccione el archivo JSON previamente guardado
3. El grafo se cargará con todas sus configuraciones y aristas
4. La visualización se actualizará automáticamente

Calcular Distancias con el Algoritmo de Floyd-Warshall

Ejecutar el Algoritmo

Una vez que su grafo tenga al menos una arista:

1. Localice el botón "**Calcular Distancias**" debajo de la visualización
2. Haga clic en el botón
3. El sistema ejecutará el algoritmo y mostrará los resultados

Interpretar los Resultados

Los resultados se dividen en varias secciones:

1. Matriz Final del Algoritmo

Una tabla que muestra las distancias mínimas entre todos los pares de vértices:

Matriz	1	2	3	4
1	0	2	5	7
2	2	0	3	5
3	5	3	0	2
4	7	5	2	0

- **Diagonal (0):** Distancia de un vértice a sí mismo
- **Números:** Distancia mínima entre dos vértices
- **∞ (Infinito):** No existe camino entre los vértices

2. Excentricidades por Vértice

La excentricidad de un vértice es la máxima distancia desde ese vértice a cualquier otro vértice del grafo.

Ejemplo:

- Vértice 1: 7
- Vértice 2: 5
- Vértice 3: 5
- Vértice 4: 7

Interpretación: El vértice 2 tiene excentricidad 5, lo que significa que el vértice más lejano está a distancia 5.

3. Distancia de Vértice (Suma de Caminos Mínimos)

Para cada vértice, se muestra la suma de todas las distancias mínimas hacia los demás vértices.

Ejemplo:

- Vértice 1: 14 (suma: $0+2+5+7$)
- Vértice 2: 10 (suma: $2+0+3+5$)
- Vértice 3: 10 (suma: $5+3+0+2$)

- Vértice 4: 14 (suma: 7+5+2+0)

Interpretación:

- Vértices con menor distancia total están más "centrales" en el grafo
- Útil para identificar nodos estratégicos en redes de transporte o comunicación

4. Mediana (Excentricidad Mínima)

La mediana es el valor mínimo entre todas las excentricidades.

Visualización:

- Se muestra un pequeño grafo con el(los) vértice(s) centro
- **Centro:** Un solo vértice con excentricidad mínima (un nodo naranja)
- **Bicentro:** Dos vértices con excentricidad mínima (dos nodos naranjas conectados)

Ejemplo:

- Mediana: 5
- Centro: Vértices 2, 3 (Bicentro)

Interpretación: Los vértices centro son los más centrales del grafo, minimizando la distancia máxima a cualquier otro vértice.

5. Diámetro (Excentricidad Máxima)

El diámetro es el valor máximo entre todas las excentricidades, representando la mayor distancia mínima entre dos vértices cualesquiera del grafo.

Ejemplo:

- Diámetro: 7

Interpretación: La mayor distancia mínima en el grafo es 7, entre los vértices 1 y 4 (o 4 y 1).

Resetear el Grafo

Para empezar desde cero:

1. Desplácese hasta el final de la página
2. Haga clic en el botón "**Resetear Grafo**" (botón rojo)
3. Confirme la acción
4. Volverá a la pantalla de configuración inicial

Nota: Esta acción elimina permanentemente el grafo actual. Asegúrese de guardarla antes si desea conservarla.

Resumen de Botones y Acciones

Botón/Acción	Ubicación	Función
Crear Grafo	Configuración inicial	Genera el grafo con los parámetros especificados
Abrir Grafo (inicial)	Debajo de "Crear Grafo"	Importa un grafo antes de crear uno nuevo
Guardar Grafo	Barra de operaciones	Exporta el grafo actual a JSON
Abrir Grafo (operaciones)	Barra de operaciones	Importa un grafo reemplazando el actual
Agregar Arista	Sección de aristas	Crea una conexión entre dos nodos
Calcular Distancias	Debajo de visualización	Ejecuta el algoritmo de Floyd-Warshall
Resetear Grafo	Final de la página	Elimina el grafo actual y vuelve al inicio

Matrices y Adyacencias

Descripción General:

Vista centrada en mostrar matrices dinámicas relacionadas con un grafo: matriz de adyacencia (nodos), adyacencia de aristas y matriz de incidencia. Permite crear grafos, agregar/borrar aristas y nodos, ver notación en teoría de conjuntos y exportar/importar grafos.

Crear un Grafo

- **Botón:** Crear Grafo
- Campos: Cantidad de Nodos, Tipo de Grafo (No Dirigido / Dirigido), ¿Es Ponderado? (Sí / No)
- Importación inicial: botón Importar Grafo (siempre visible)

Gestionar Aristas y Nodos

- Campo conexión: placeholder Ej: 12 para conectar nodo 1 y 2
- Campo peso: placeholder Peso (si ¿Es Ponderado? está activo)
- Botones: Agregar Arista, Agregar Nodo (etiquetado en UI como Agregar Nodo), Eliminar (para eliminar nodo), Borrar (para arista)

Paneles de Matrices

- Matriz de Adyacencia de Nodos
- Adyacencia de Aristas
- Matriz de Incidencia
- Control dinámico de tamaño: slider con etiqueta Tamaño matrices y valor en px

Operaciones / Guardado

- Botones: Resetear Grafo, Guardar Grafo

Resumen de botones

- Importar Grafo, Crear Grafo, Agregar Arista, Agregar Nodo, Eliminar, Borrar, Resetear Grafo, Guardar Grafo

Detalles técnicos

- Matrices disponibles:
 - **Matriz de adyacencia (nodos)**: matriz A de tamaño $n \times n$ ($n = |V|$). Para grafos no ponderados $A[i][j] \in \{0,1\}$; para grafos ponderados $A[i][j]$ contiene el peso de la arista o $0/\infty$ cuando no existe conexión. En grafos dirigidos la matriz no es necesariamente simétrica.
 - **Adyacencia de aristas**: matriz $E \times E$ donde cada fila/columna representa una arista; la entrada indica si dos aristas comparten un extremo. Útil para construir el grafo línea.
 - **Matriz de incidencia**: matriz B de tamaño $n \times m$ ($m = |E|$). Convención usada en la UI: en grafos no dirigidos $B[i][j] \in \{0,1\}$; en dirigidos se usa $\{1,-1,0\}$ según la orientación (1 si el nodo es origen, -1 si es destino, 0 si no participa).
- Usos prácticos en la vista:
 - **Comprobación de conectividad y grados**: grados de nodos se calculan sumando filas/columnas de la matriz de adyacencia o sumando columnas en la matriz de incidencia.
 - **Caminos de longitud k**: la k-ésima potencia de la matriz de adyacencia (A^k) da el número de caminos de longitud k entre pares (operación teórica, costosa para n grandes).
 - **Visualización**: las matrices se muestran con scroll y control de tamaño para mantener legibilidad en grafos grandes.

Recomendaciones de uso

- Para grafos grandes preferir la representación por listas de adyacencia (no visualizada) y regenerar matrices solo bajo demanda (export/import o al abrir la pestaña de matrices).
- Los archivos JSON exportados contienen nodos y aristas explícitos: conservarlos para importación sin pérdida de información.

Matrices y Circuitos

Descripción General:

Vista especializada en análisis de circuitos y cortes de un grafo ponderado. Calcula todos los circuitos, circuitos fundamentales, matrices de circuitos, conjuntos de corte y sus matrices, además del árbol de expansión máximo (ramas/cuerdas).

Crear un Grafo

- **Botón: Crear Grafo**
- Importación siempre disponible: botón **Abrir Grafo**
- Campos: **Cantidad de Nodos**, **Tipo de Grafo**, **¿Es Ponderado?** (bloqueado a **Sí** en la UI)

Gestionar Aristas y Nodos

- Campo conexión: placeholder **Ej: 12 para conectar nodo 1 y 2**
- Campo peso: placeholder **Peso**
- Botones: **Agregar Arista**, **Insertar Nodo**, **Eliminar**, **Borrar**

Paneles y Salidas

- Ramas (T) – Árbol de expansión máximo
- Cuerdas (C)
- Todos los Circuitos y Matriz de Circuitos
- Circuitos Fundamentales y Matriz de Circuitos Fundamentales
- Conjuntos de Corte y Matriz de Conjuntos de Corte
- Cortes Fundamentales y su matriz

Operaciones / Guardado

- Botones: Resetear Grafo, Guardar Grafo

Resumen de botones (texto literal en UI)

- Abrir Grafo, Crear Grafo, Agregar Arista, Insertar Nodo, Eliminar, Borrar, Resetear Grafo, Guardar Grafo
-

Detalles técnicos y definiciones

- Definiciones clave:
 - **Círculo (ciclo simple)**: recorrido cerrado sin repetir aristas ni vértices (excepto vértice inicial=final). La vista lista "Todos los Circuitos" detectados por algoritmos basados en búsqueda DFS y eliminación de duplicados por rotación y reflejo.
 - **Círculo fundamental**: dado un árbol generador (spanning tree) T, cada arista fuera de T define exactamente un circuito fundamental (la arista más el camino en T entre sus extremos). El conjunto de circuitos fundamentales forma una base del espacio de ciclos.
 - **Conjunto de corte (cut-set)**: conjunto de aristas que separa el grafo en dos componentes; los cortes fundamentales se definen respecto a un árbol generador.
- Matrices y convenciones usadas en la vista:
 - **Matriz de circuitos**: filas = circuitos, columnas = aristas; entradas en {1,-1,0}. La convención de signo indica orientación relativa con la arista de referencia (1 si la arista aparece en el circuito con la misma orientación, -1 si con orientación inversa, 0 si no participa).
 - **Matriz de cortes**: similar a la matriz de circuitos pero con 1/0 según inclusión de arista en el corte; en cortes dirigidos puede usarse el signo para distinguir origen/destino.
- Algoritmos y complejidad:
 - Encontrar "todos los circuitos" es una tarea combinatoria (puede crecer exponencialmente con m). La vista aplica heurísticas y límites prácticos para evitar explosión computacional en grafos grandes.
 - Construcción de circuitos fundamentales y matrices de cortes/fundamentales se hace en O(n + m) tras obtener un árbol generador (DFS/BFS + procesamiento de aristas restantes).
 - El número de circuitos independientes (dimensión del espacio de ciclos) es el número ciclomático: $\mu = m - n + c$ (c = componentes conectadas).

Recomendaciones de uso

- Evitar solicitar "Todos los Circuitos" en grafos con más de ~20 aristas — la operación puede tomar mucho tiempo. Para análisis exhaustivos, prefiera primero generar un árbol (árbol de expansión) y trabajar con los circuitos fundamentales.

Operaciones entre N Grafos

Descripción General:

Vista para crear N grafos (configurable), gestionar cada grafo por separado y ejecutar operaciones binarias entre grafos: unión, intersección, suma, suma anillo, productos y composición.

Crear Grafos

- **Botón:** [Crear Grafos](#)
- Importación: [Abrir Grafos](#) (siempre visible)
- Configuración: [Cantidad de Grafos](#), y la cantidad de nodos para cada grafo

Gestionar Aristas y Nodos

- Selector dinámico de grafo: botones con etiquetas G_1, G_2, \dots (en UI aparecen como [Grafo \$G\\$\{i\}\$](#) y botones seleccionables $G\$\{i\}$)
- Campo conexión: placeholder [Ej: 12 para conectar nodo 1 y 2](#)
- Botones: [Aregar Arista](#), [Insertar Nodo](#), [Eliminar](#), [Borrar](#)

Operaciones entre Grafos

- [Unión \(\$G_1 \cup G_2\$ \)](#)
- [Intersección \(\$G_1 \cap G_2\$ \)](#)
- [Suma \(\$G_1 + G_2\$ \)](#)
- [Suma Anillo \(\$G_1 \oplus G_2\$ \)](#)
- [Producto Cartesiano \(\$G_1 \times G_2\$ \)](#)
- [Producto Tensorial \(\$G_1 \otimes G_2\$ \)](#)
- [Composición \(\$G_1 \circ G_2\$ \)](#)

Resultados

- Panel de resultado con visualización del grafo resultado y notación (V, A)
- Cerrar panel: [Cerrar Resultado](#)

Reset / Guardado

- Botones: [Resetear Grafos](#), [Guardar Grafos](#)

Definiciones y detalles técnicos

- Operaciones elementales (definiciones usadas como referencia):
 - **Unión ($G_1 \cup G_2$):** $V = V_1 \cup V_2, A = A_1 \cup A_2$. Si las etiquetas de vértice coinciden se mantienen; en implementaciones donde se crean grafos nuevos puede realizarse reenumeración para evitar colisiones.

- **Intersección ($G_1 \cap G_2$):** $V = V_1 \cap V_2$, $A = A_1 \cap A_2$ (solo se mantienen los vértices y aristas presentes en ambos grafos).
 - **Suma ($G_1 + G_2$):** en este contexto se interpreta como **unión disjunta** (disjoint union) — los vértices se renumeran/etiquetan para mantener grafos independientes y luego se combinan; resultado: $|V| = |V_1| + |V_2|$, $|A| = |A_1| + |A_2|$.
 - **Suma Anillo ($G_1 \oplus G_2$):** diferencia simétrica de aristas sobre $V = V_1 \cup V_2$, $A = (A_1 \cup A_2) \setminus (A_1 \cap A_2)$.
 - **Producto Cartesiano ($G_1 \times G_2$):** $V = V_1 \times V_2$; hay arista entre (u_1, v_1) y (u_2, v_2) si $(u_1 = u_2 \text{ and } v_1, v_2) \in A_2$ o $(v_1 = v_2 \text{ and } (u_1, u_2) \in A_1)$. Tamaño crece multiplicativamente: $|V| = |V_1| \cdot |V_2|$.
 - **Producto Tensorial ($G_1 \otimes G_2$):** $V = V_1 \times V_2$; hay arista entre (u_1, v_1) y (u_2, v_2) si $(u_1, u_2) \in A_1 \text{ and } (v_1, v_2) \in A_2$.
 - **Composición ($G_1 \circ G_2$):** operación que combina relaciones de adyacencia por composición de relaciones; implementación y semántica pueden variar, la UI ofrece una versión comúnmente usada en teoría de grafos (consulte la referencia en la sección de ayuda interna si necesita la fórmula exacta usada en esta vista).
- Complejidad y consideraciones:
 - Muchas operaciones tienen coste $O(|V_1| \cdot |V_2| + |A_1| \cdot |A_2|)$ en worst-case (especialmente productos), por lo que la vista limita o advierte cuando los tamaños combinados son grandes.
 - Al usar operaciones que amplían el conjunto vértice, tenga en cuenta memoria/tiempo y prefiera trabajar con grafos pequeños o con ejemplos reducidos.

Ejemplos rápidos

- Unión: G_1 con $V_1 = \{1, 2\}$, $A_1 = \{(1, 2)\}$ y G_2 con $V_2 = \{2, 3\}$, $A_2 = \{(2, 3)\} \rightarrow G_1 \cup G_2$ tiene $V = \{1, 2, 3\}$ y $A = \{(1, 2), (2, 3)\}$.
- Suma Anillo: con los mismos $G_1, G_2 \rightarrow A$ resultado es $\{(1, 2), (2, 3)\} \setminus \{(2, 3)\}$ si estuviera en ambos} = diferencia simétrica.

Operaciones en un Grafo

Descripción General:

Vista de operaciones avanzadas sobre un grafo único: fusionar vértices, contraer aristas, generar grafo línea y generar complemento del grafo. Incluye import/export y control de transformaciones reversibles.

Crear un Grafo

- **Botón:** [Crear Grafo](#)
- Importación siempre disponible: [Abrir Grafo](#)
- Campos: [Cantidad de Nodos](#), [Tipo de Grafo](#), [¿Es Ponderado?](#)

Gestionar Aristas y Nodos

- Campo conexión: placeholder [Ej: 12 para conectar nodo 1 y 2](#)
- Campo peso: placeholder [Peso](#) (si aplica)
- Botones: [Agregar Arista](#), [Insertar Nodo](#), [Eliminar](#), [Borrar](#)

Operaciones Avanzadas (textos literales)

- Fusionar vértices: sección **Fusionar nodos** → botón **Fusionar** (ingresar pares en placeholder **Ej: 12**)
- Contraer arista: sección **Contraer Arista** → botón **Contraer** (placeholder **Ej: 12**)
- Generar grafo línea: **Generar Línea** (botón dentro de **Generar Grafo Línea**)
- Generar complemento: **Generar Complemento** (botón dentro de **Generar Complemento del Grafo**)
- Reversión: **Revertir Grafo Línea**, **Revertir Complemento** (botones de reversión)

Reset / Guardado

- Botones: **Resetear Grafo**, **Guardar Grafo**
-

Detalles técnicos y definiciones

- Transformaciones soportadas (definiciones y efectos):
 - **Fusionar vértices (identificación)**: identificar dos vértices u,v en uno solo ($u \approx v$). Todas las aristas incidentes a u o v pasan a incidir en el vértice resultante; pueden generarse bucles (autoincidencias) y aristas paralelas, que la vista puede mantener o normalizar según opción.
 - **Contraer arista**: operación que elimina una arista e identifica sus extremos (equivalente a fusionar los dos vértices que dicha arista une). Es básica en algoritmos de conectividad y en reducción de grafos para obtener menores topológicos.
 - **Grafo Línea ($L(G)$)**: vértices de $L(G)$ = aristas de G ; hay arista entre dos vértices de $L(G)$ si las aristas correspondientes en G comparten un extremo. En la UI la generación de grafo línea convierte la lista de aristas en nodos y reconstruye las conexiones mediante comprobación de incidencia.
 - **Complemento de G (\bar{G})**: mismo conjunto de vértices; aristas = pares no conectados en G . Efectos: $|E(\bar{G})| = n(n-1)/2 - |E(G)|$ (para grafos simples no dirigidos).
- Propiedades y reversibilidad:
 - **Contracción vs reversión**: contraer una arista es generalmente irreversible si no se guarda el histórico (porque se pierde la distinción entre las aristas paralelas o el orden original); la vista ofrece botones **Revertir Grafo Línea** y **Revertir Complemento** para deshacer transformaciones específicas cuando se dispone del grafo original en memoria.
 - **Cambio en métricas**: la contracción reduce n y típicamente reduce la distancia entre nodos; el grafo línea aumenta el número de nodos ($|V(L(G))| = |E(G)|$) y cambia la estructura utilizada para algoritmos basados en aristas.

Recomendaciones de uso

- Antes de aplicar transformaciones destructivas (fusionar/contraer), guarde el estado original con **Guardar Grafo** si piensa revertir.
- Al generar el grafo línea o el complemento, revisar visualmente la correspondencia entre nodos/aristas para validar que la operación aplicada corresponde a la transformación matemática esperada.

Notas finales y recomendaciones

- Antes de ejecutar operaciones destructivas (p. ej. **Resetear Grafo**) guarde el grafo con **Guardar Grafo** o **Guardar Grafos** según corresponda.

- Para importar, use archivos `.json` generados por las propias vistas (las validaciones internas advierten si el formato no coincide).
-

Fin de la sección de Grafos

Fin del Manual de Usuario