

ESTRATEGIAS DE PROGRAMACIÓN Y ESTRUCTURAS DE DATOS

Estructuras de Datos Básicas (II)

Secuencias: Listas, Pilas y Colas

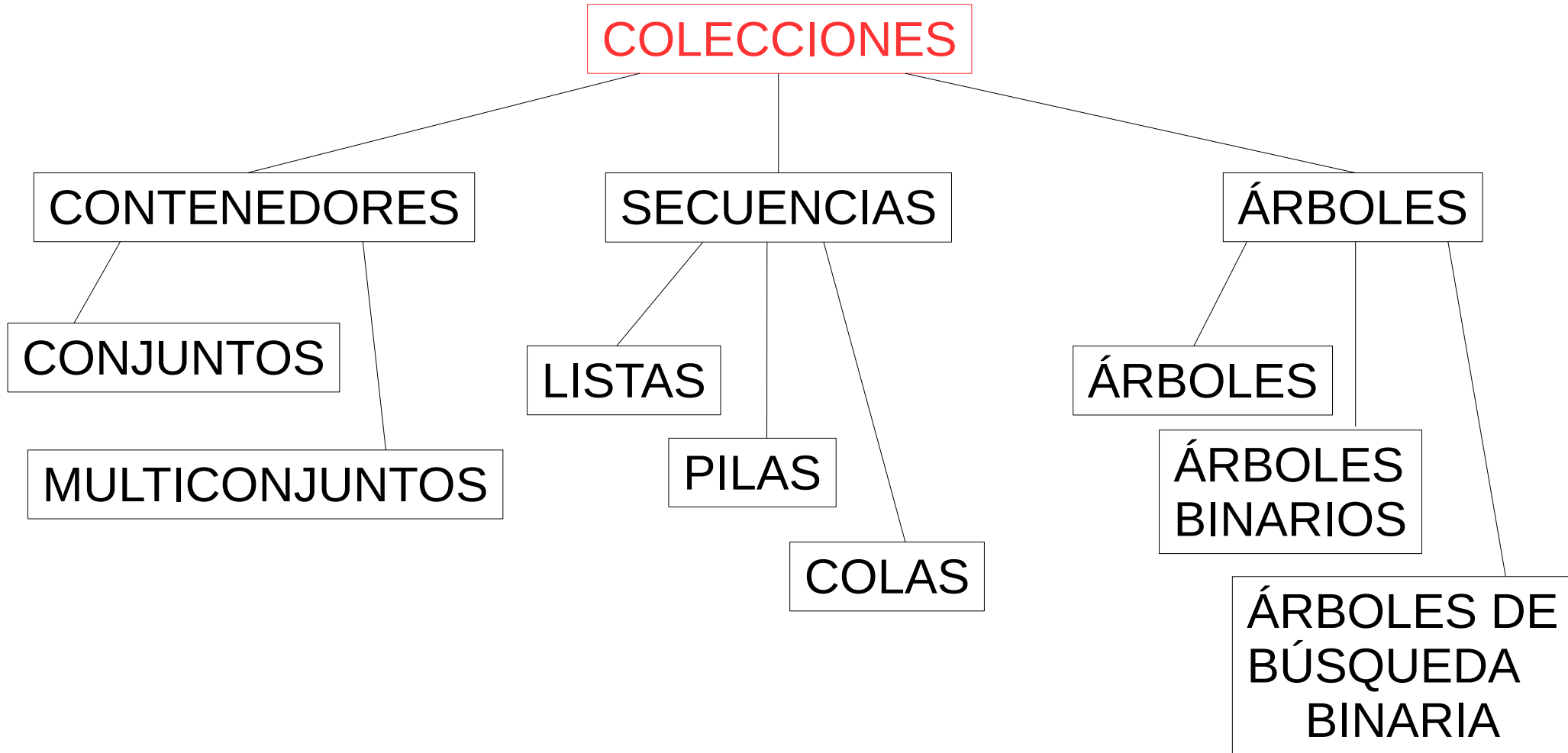
SECUENCIAS

Equipo Docente de la Asignatura. Departamento LSI.

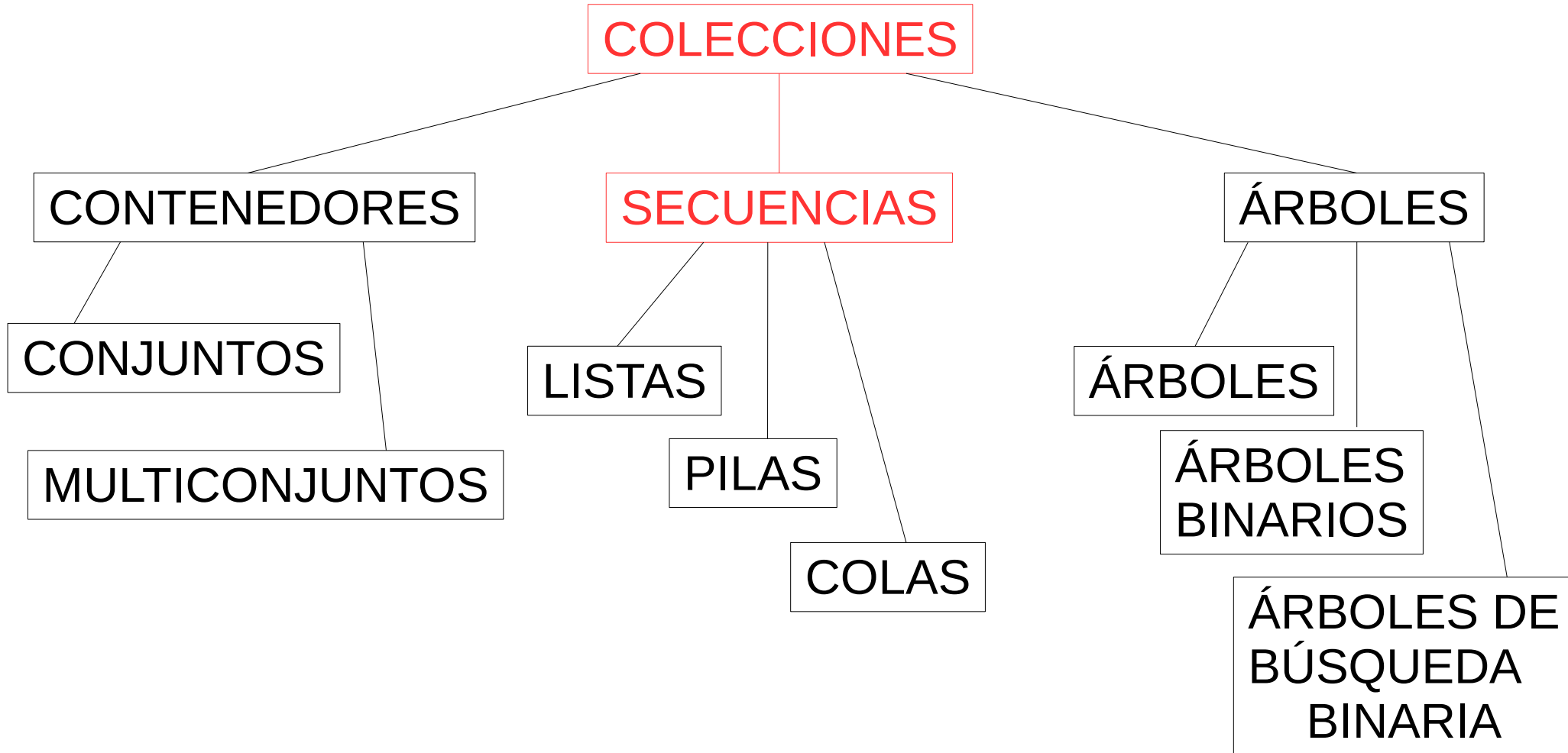
UNED

ETS de
Ingeniería
Informática

Tipos Abstractos de Datos estudiados en esta asignatura



Tipos Abstractos de Datos estudiados en esta asignatura



ESTRUCTURAS DE DATOS

SECUENCIAS

- Los datos forman una secuencia: elementos con orden lineal explícito
 - Relaciones de adyacencia: antecesión y sucesión
 - Elemento antecesor y elemento sucesor (si hay)
 - Organización no condiciona representación
 - No hay orden implícito (aunque las listas podrían)
- ¿Qué operaciones hacen falta?

ESTRUCTURAS DE DATOS

SECUENCIAS

- Los datos forman una secuencia: elementos con orden lineal explícito
 - Relaciones de adyacencia: antecesor y sucesor
 - Elemento antecesor y elemento sucesor (si hay)
 - Organización no condiciona representación
 - No hay orden implícito (aunque las listas podrían)
- ¿Qué operaciones hacen falta?
 - Recorrer los elementos

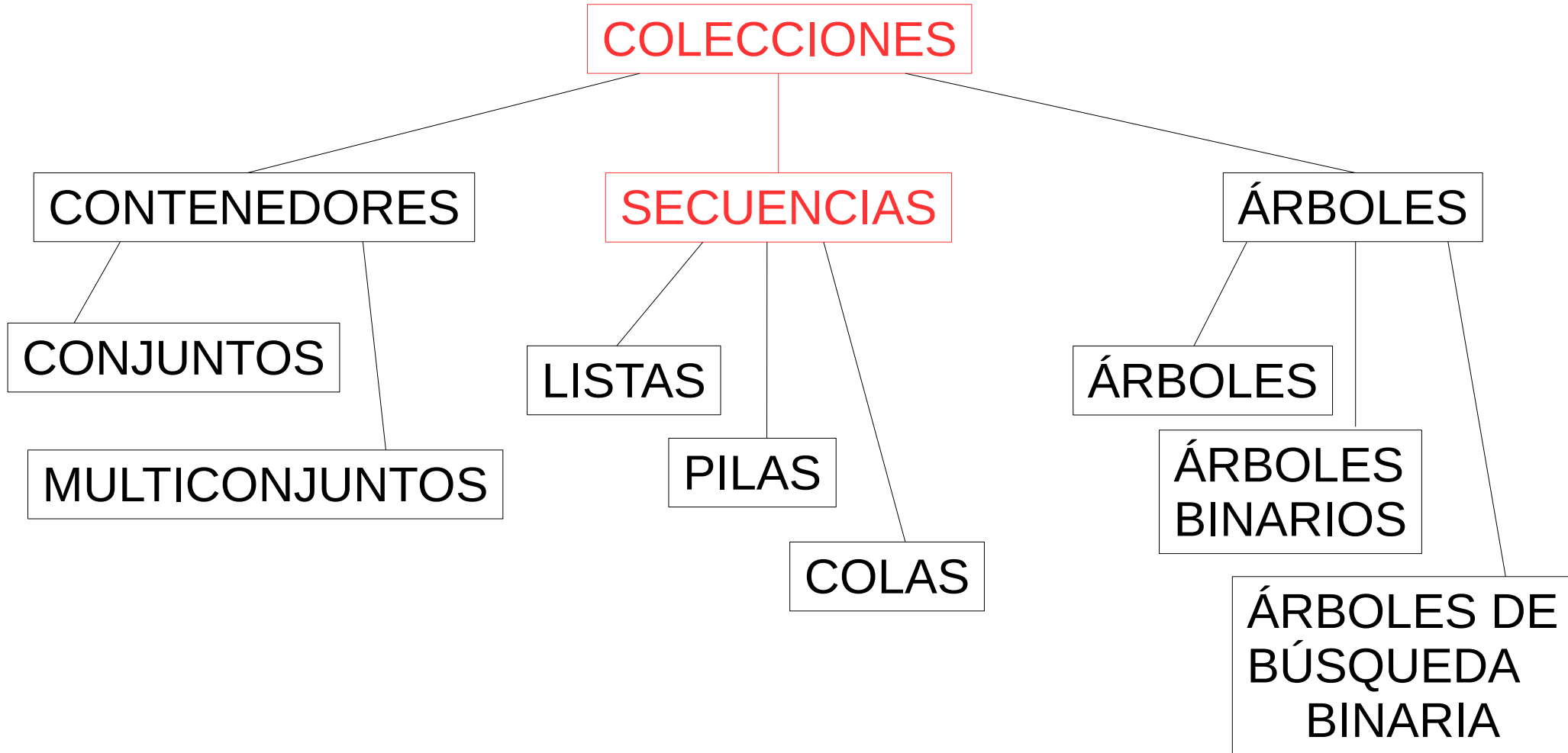
ESTRUCTURAS DE DATOS

SECUENCIAS

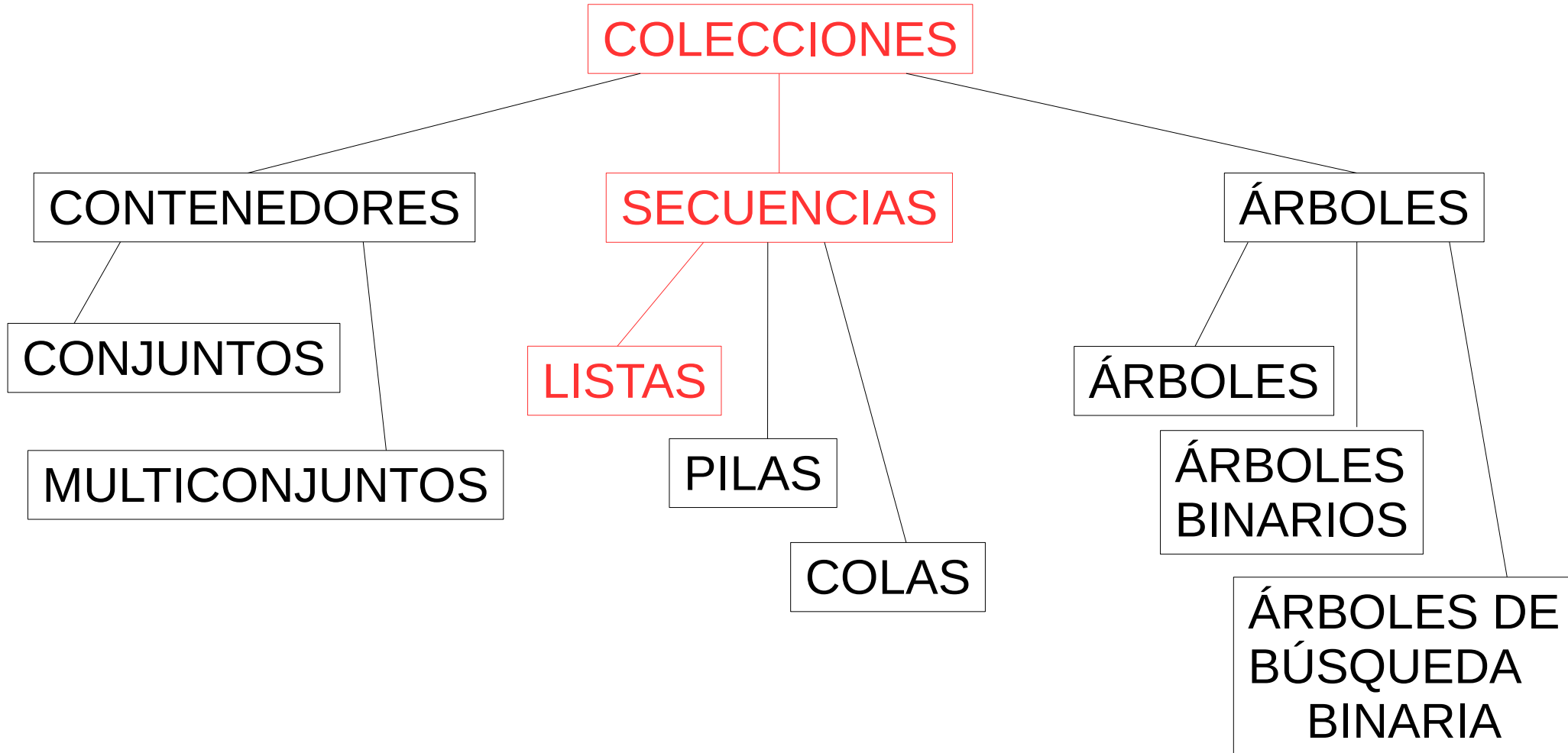
```
/* Representa una secuencia, que es una colección de          *  
 * elementos que se organizan linealmente.                    */  
public interface SequenceIF<E> extends CollectionIF<E> {  
  
    /* Devuelve el iterador sobre la secuencia. No necesita    *  
    * parámetros puesto que el recorrido es lineal y único.   */  
    public IteratorIF<E> iterator ();  
}
```

SECUENCIAS: LISTAS

Tipos Abstractos de Datos estudiados en esta asignatura



Tipos Abstractos de Datos estudiados en esta asignatura



ESTRUCTURAS DE DATOS LISTAS

- Acceso arbitrario a los elementos por un punto
- Recorrido NO destructivo
- Varios modelos de implementación
 - Primero y resto.
 - Punto de interés.
 - **Acceso por posición.**

ESTRUCTURAS DE DATOS

LISTAS

- Modelo: primero y resto
 - Lista = Lista vacía
 - Lista = Cabeza + Resto (lista)

ESTRUCTURAS DE DATOS

LISTAS

- Modelo: punto de interés
 - Puntero a una posición en la lista
 - Inserción, acceso y modificación a través de ese puntero

ESTRUCTURAS DE DATOS

LISTAS

- Modelo: acceso por posición
 - Cada elemento tiene una posición
 - Inserción y acceso a través de esa posición
- Operaciones:

ESTRUCTURAS DE DATOS

LISTAS

- Modelo: acceso por posición
 - Cada elemento tiene una posición
 - Inserción y acceso a través de esa posición
- Operaciones:
 - Insertar y eliminar elementos de la lista en una posición
 - Consultar y modificar elementos existentes en la lista

ESTRUCTURAS DE DATOS

LISTAS

```
/* Representa una lista de elementos. Una lista es una      *
 * secuencia que permite modificar o acceder a cualquiera de *
 * sus elementos de forma no destructiva.                  */
public interface ListIF<E> extends SequenceIF<E> {

    /* Devuelve el elemento de la lista que ocupa la posición *
     * indicada por el parámetro.                             *
     * @param pos la posición comenzando en 1.                 *
     * @Pre: 1 <= pos <= size().                               *
     * @return el elemento en la posición pos.                 */
    public E get (int pos);
```

...

ESTRUCTURAS DE DATOS

LISTAS

```
/* Modifica la posición dada por el parámetro pos para que*  
 * contenga el valor dado por el parámetro e.           *  
 * @param pos la posición cuyo valor se debe modificar,  *  
 * comenzando en 1.                                     *  
 * @param e el valor que debe adoptar la posición pos.  *  
 * @Pre: 1 <= pos <= size().                             */
```

```
public void set (int pos, E e);
```

```
/* Inserta un elemento en la Lista.                      *  
 * @param elem El elemento que hay que añadir.          *  
 * @param pos La posición en la que se debe añadir elem,*  
 * comenzando en 1.                                     *  
 * @Pre: 1 <= pos <= size()+1                           */
```

```
public void insert (E elem, int pos);
```

...

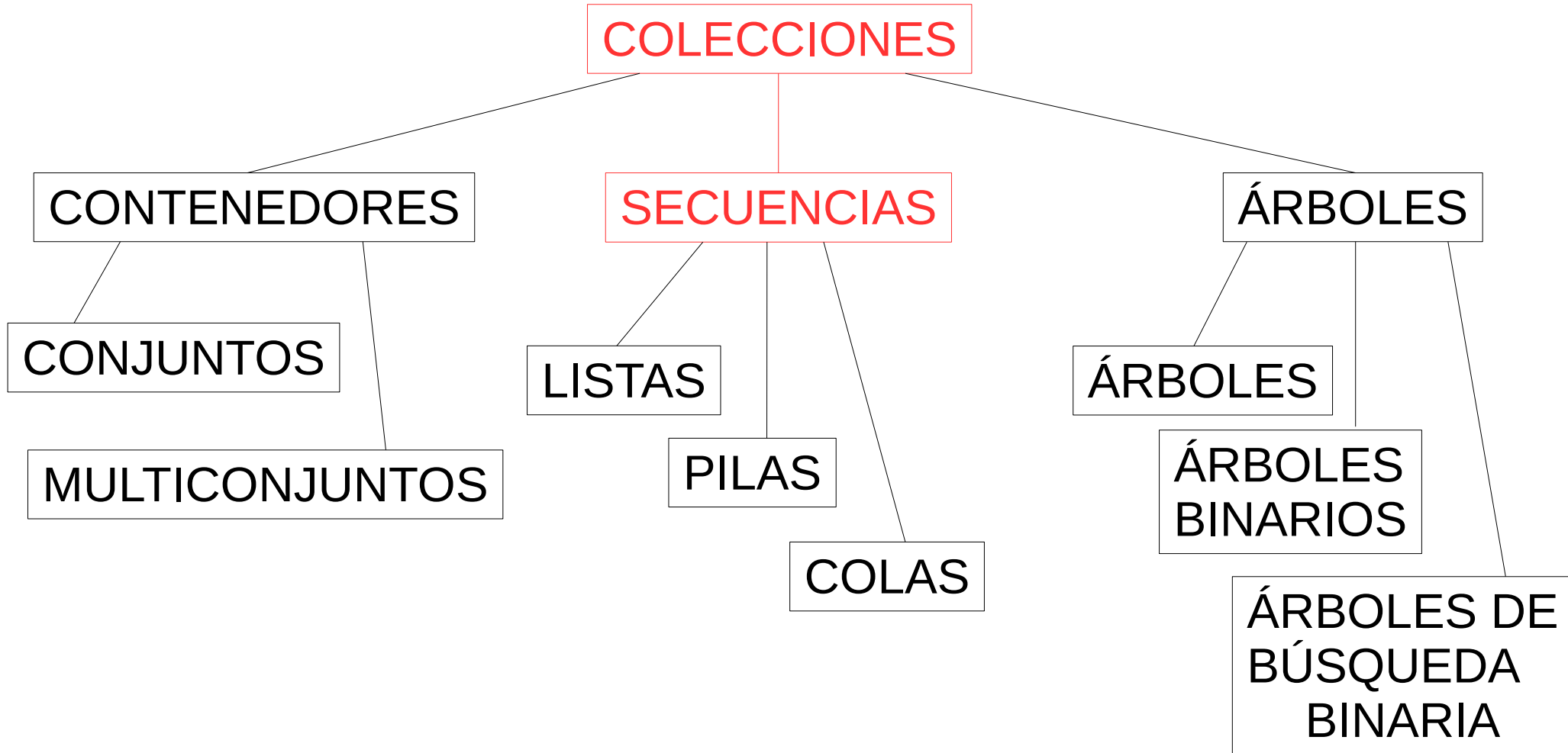
ESTRUCTURAS DE DATOS

LISTAS

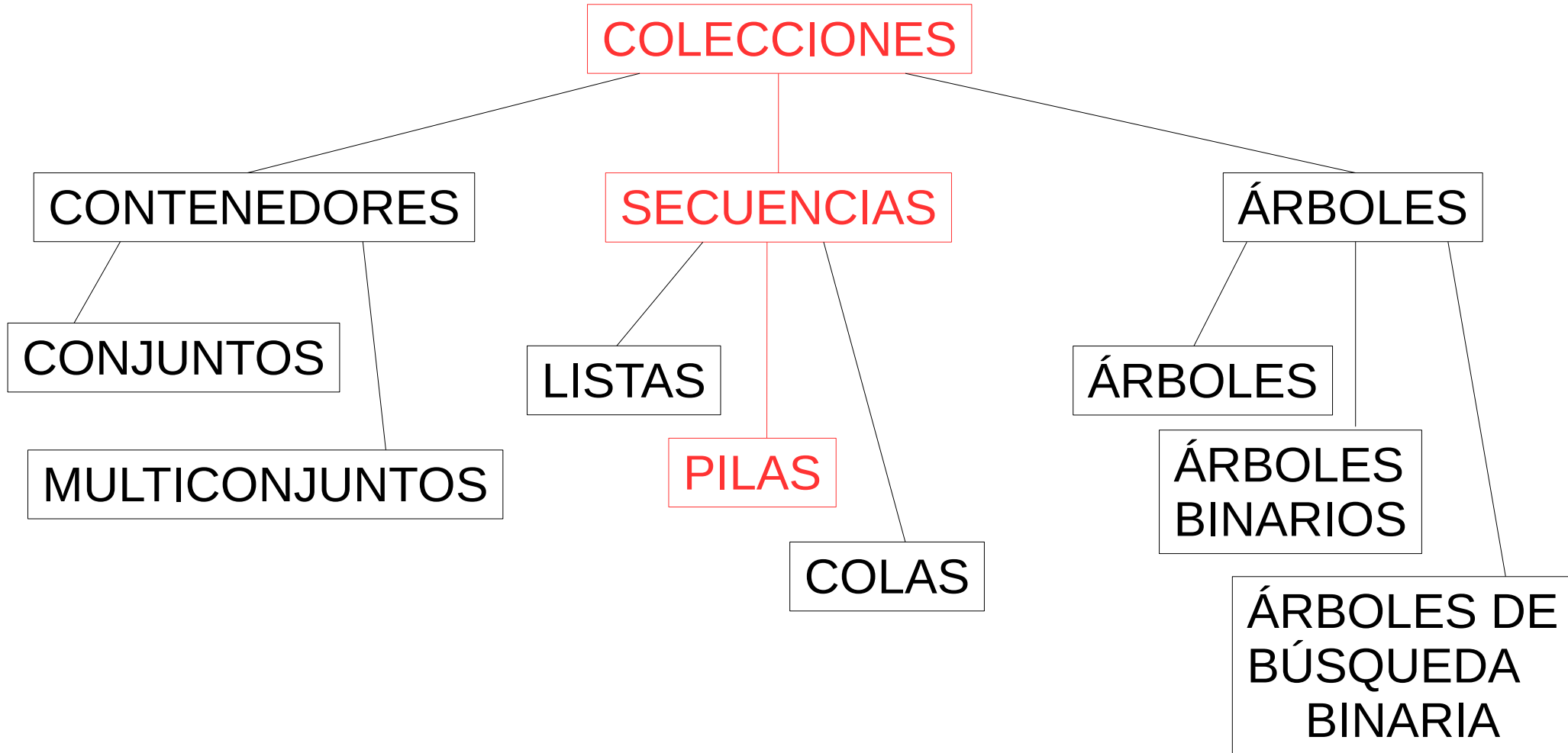
```
/* Elimina el elemento que ocupa la posición del parámetro*  
 * @param pos la posición que ocupa el elemento a eliminar*  
 * comenzando en 1 *  
 * @Pre: 1 <= pos <= size() */  
public void remove (int pos);  
}
```

SECUENCIAS: PILAS

Tipos Abstractos de Datos estudiados en esta asignatura



Tipos Abstractos de Datos estudiados en esta asignatura



ESTRUCTURAS DE DATOS

PILAS

- Acceso por un único punto → cima
- Se organizan inversamente a su inserción
 - Política LIFO (Last In First Out)
- No se puede modificar la cima → borrar + insertar
- Recorrido destructivo
- Operaciones:

ESTRUCTURAS DE DATOS

PILAS

- Acceso por un único punto → cima
- Se organizan inversamente a su inserción
 - Política LIFO (Last In First Out)
- No se puede modificar la cima → borrar + insertar
- Recorrido destructivo
- Operaciones:
 - Apilar, Desapilar y consultar cima

ESTRUCTURAS DE DATOS

PILAS

```
/* Representa una pila de elementos. Una pila es una          *  
 * especialización de una secuencia, que mantiene el orden de *  
 * almacenamiento de sus elementos y una política de acceso   *  
 * Last In First Out (LIFO).                                   *  
*/  
public interface StackIF <E> extends SequenceIF<E>{  
  
    /* Devuelve el elemento situado en la cima de la pila      *  
    * @Pre !isEmpty ();                                         *  
    * @return la cima de la pila                               */  
    public E getTop ();
```

...

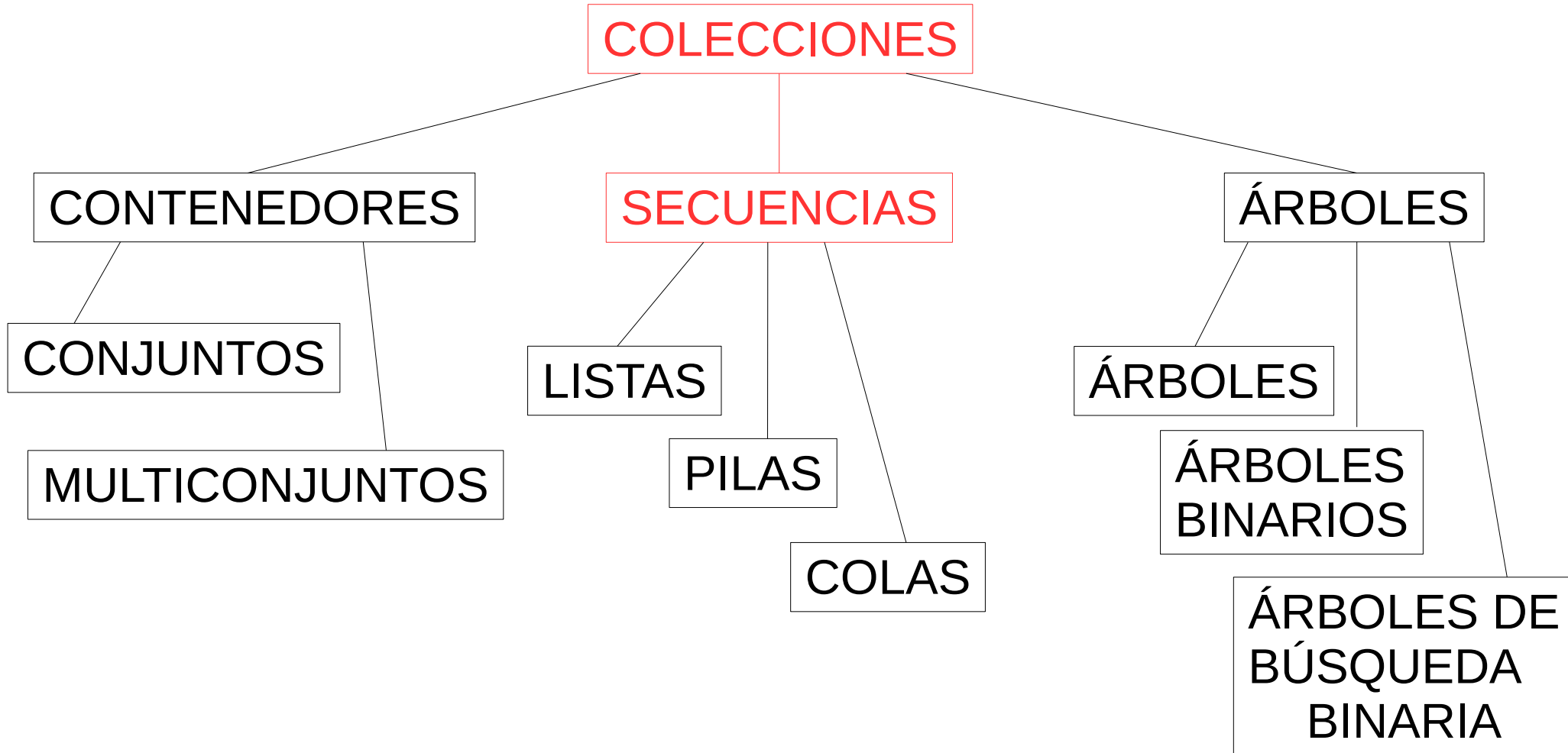
ESTRUCTURAS DE DATOS

PILAS

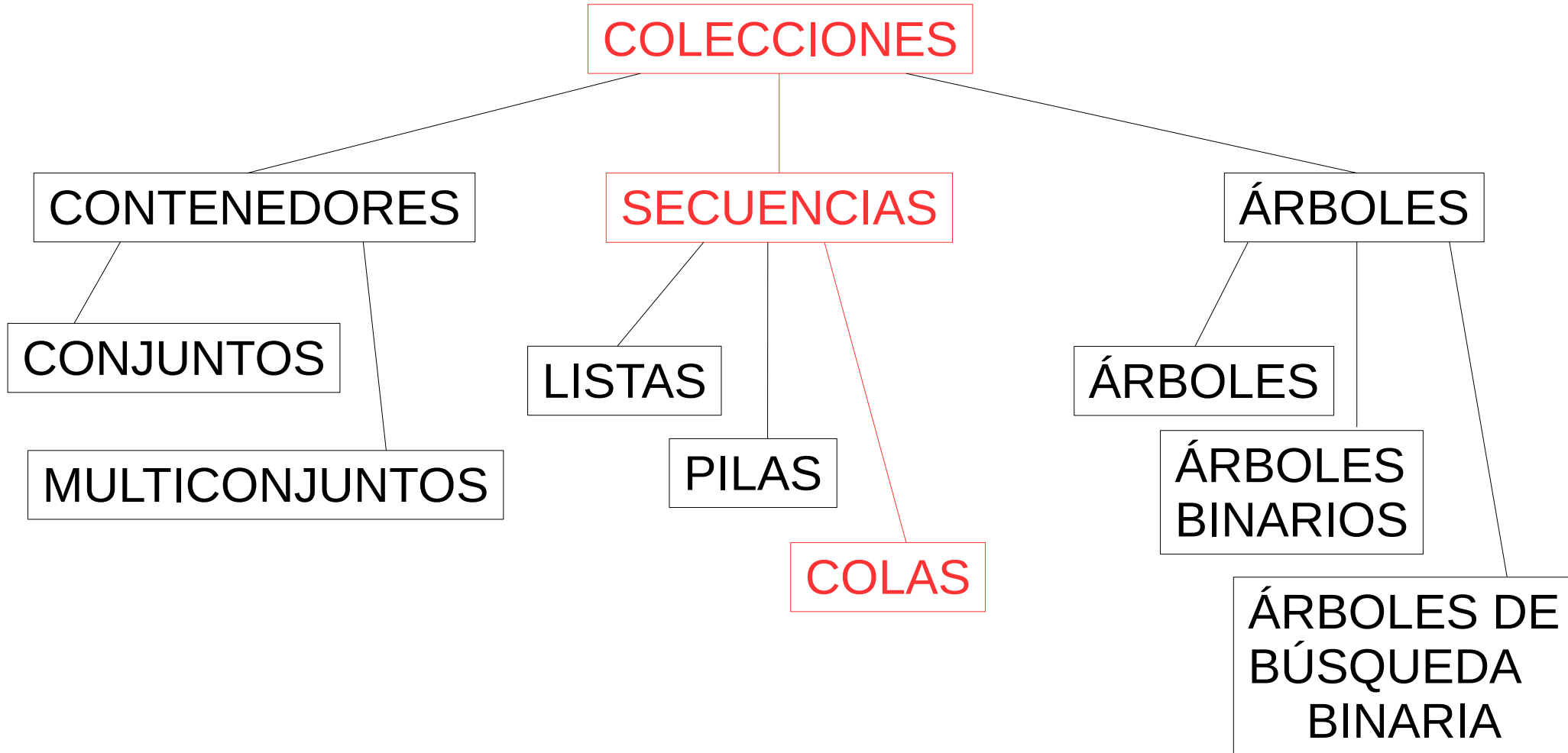
```
/* Incluye un elemento en la cima de la pila. Modifica el *  
 * tamaño de la misma. *  
 * @param elem el elemento que se quiere añadir en la cima*/  
public void push (E elem);  
  
/* Elimina la cima de la pila. Modifica el tamaño de la *  
 * pila. *  
 * @Pre !isEmpty (); */  
public void pop ();  
}
```

SECUENCIAS: COLAS

Tipos Abstractos de Datos estudiados en esta asignatura



Tipos Abstractos de Datos estudiados en esta asignatura



ESTRUCTURAS DE DATOS

COLAS

- Acceso por dos puntos → inserción y borrado
- Se organizan directamente a su inserción
 - Política FIFO (First In First Out)
- No se puede modificar sus elementos accesibles
 - Copiar y destruir TODO el contenido
- Recorrido destructivo
- Operaciones:

ESTRUCTURAS DE DATOS

COLAS

- Acceso por dos puntos → inserción y borrado
- Se organizan directamente a su inserción
 - Política FIFO (First In First Out)
- No se puede modificar sus elementos accesibles
 - Copiar y destruir TODO el contenido
- Recorrido destructivo
- Operaciones:
 - Encolar, desencolar y consultar primero

ESTRUCTURAS DE DATOS

COLAS

```
/* Representa una cola de elementos. Una cola es una          *
 * especialización de una secuencia, que mantiene el orden    *
 * de almacenamiento de sus elementos y una política de        *
 * acceso First In First Out (FIFO)                             */
public interface QueueIF<E> extends SequenceIF<E> {

    /* Devuelve el primer elemento de la cola.                  *
     * @Pre !isEmpty()                                           *
     * @return la cabeza de la cola (su primer elemento).      */
    public E getFirst ();

    /* Incluye un elemento al final de la cola. Modifica el    *
     * tamaño de la misma.                                       *
     * @param elem el elemento que debe encolar (añadir).      */
    public void enqueue (E elem);
```

...

ESTRUCTURAS DE DATOS

COLAS

```
/* Elimina el primer elemento de la cola. Modifica la      *  
 * tamaño de la misma.                                     *  
 * @Pre !isEmpty();                                         */  
public void dequeue ();  
}
```


ESTRATEGIAS DE PROGRAMACIÓN Y ESTRUCTURAS DE DATOS

Estructuras de Datos Básicas (II)

Secuencias: Listas, Pilas y Colas