

**Instituto Tecnológico de Tijuana**

**Subdirección Académica**

**Departamento de Sistemas y Computación**

**Semestre agosto-diciembre 2018**

**Carrera:** Ing. En Sistemas Computacionales Serie SC1A

**Materia:** Estructura de Datos

**Unidad 3 – Listas Enlazadas**

**Alumno:**15211883 - Angeles Valadez Jonathan

**Profesor:** Ray Brunett Parra Galaviz

**Listas**

La lista es un mecanismo de colección potente, pero simple, que proporciona al programador una amplia variedad de operaciones. Sin embargo, no todos los lenguajes de programación incluyen una colección de listas. En estos casos, la noción de una lista debe ser implementada por el programador.

Una **lista** es una colección de ítems donde cada ítem mantiene una posición relativa con respecto a los otros. Más específicamente, nos referiremos a este tipo de lista como una lista no ordenada. Podemos considerar que la lista tiene un primer ítem, un segundo ítem, un tercer ítem, y así sucesivamente. También podemos referirnos al inicio de la lista (el primer ítem) o al final de la lista (el último). Por simplicidad, asumiremos que las listas no pueden contener ítems duplicados.

Por ejemplo, la colección de números enteros 54, 26, 93, 17, 77 y 31 puede representar una lista desordenada simple de calificaciones de exámenes. Tenga en cuenta que los hemos escrito como valores delimitados por comas, una forma común de mostrar la estructura de la lista. Por supuesto, Python mostraría esta lista como [54, 26, 93, 17, 77,31].

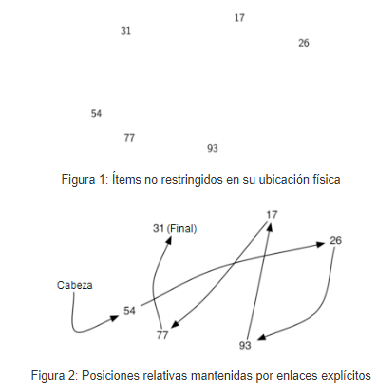
**El tipo abstracto de datos Lista No Ordenada**

La estructura de una lista no ordenada, como se ha descrito anteriormente, es una colección de ítems en la que cada ítem mantiene una posición relativa con respecto a los demás. A continuación se indican algunas posibles operaciones para las listas no ordenadas.

* Lista() crea una nueva lista que está vacía. No necesita parámetros y devuelve una lista vacía.
* agregar(item) agrega un nuevo ítem a la lista. Necesita el ítem y no devuelve nada. Asume que el ítem aún no está en la lista.
* remover(item) elimina el ítem de la lista. Necesita el ítem y modifica la lista. Asume que el ítem está presente en la lista.
* buscar(item) busca el ítem en la lista. Necesita el ítem y devuelve un valor booleano.
* estaVacia() comprueba si la lista está vacía. No necesita parámetros y devuelve un valor booleano.
* tamano() devuelve el número de ítems en la lista. No necesita parámetros y devuelve un entero.
* anexar(item) agrega un nuevo ítem al final de la lista, convirtiéndolo en el último ítem de la colección. Necesita el ítem y no devuelve nada. Asume que el ítem aún no está en la lista.
* indice(item) devuelve la posición del ítem en la lista. Necesita el ítem y devuelve el índice. Asume que el ítem está en la lista.
* insertar(pos,item) agrega un nuevo ítem a la lista en la posición pos. Necesita el ítem y no devuelve nada. Asume que el ítem aún no está en la lista y que hay suficientes elementos existentes para tener la posición pos.
* extraer() elimina y devuelve el último ítem de la lista. No necesita nada y devuelve un ítem. Asume que la lista tiene al menos un ítem.
* extraer(pos) elimina y devuelve el ítem en la posición pos. Necesita la posición y devuelve el ítem. Asume que el ítem está en la lista.

**Implementación de una lista no ordenada: Listas enlazadas**

Con el fin de implementar una lista no ordenada, vamos a construir lo que comúnmente se conoce como una lista enlazada (encadenada o ligada). Recordemos que tenemos que estar seguros de que podemos mantener el posicionamiento relativo de los ítems. Sin embargo, no existe ningún requisito que mantenga ese posicionamiento en memoria contigua. Por ejemplo, considere la colección de elementos mostrados en la Figura 1. Parece que estos valores se han colocado al azar. Si podemos mantener alguna información explícita en cada ítem, es decir, la ubicación del ítem siguiente (véase la Figura 2), entonces la posición relativa de cada ítem puede expresarse simplemente siguiendo el enlace de un ítem al siguiente.



**Bibliografía**

[En Línea]: <http://interactivepython.org/runestone/static/pythoned/BasicDS/Listas.html>

[En Línea]:

<http://interactivepython.org/runestone/static/pythoned/BasicDS/ElTipoAbstractoDeDatosListaNoOrdenada.html>