

<div style="text-align: center;"> <b>Universidad Tecnológica Nacional</b>  <b>Facultad Regional Avellaneda</b> </div> <div style="text-align: right;">  </div>					
Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos					
Materia: Laboratorio de computación I					
Apellido:			Fecha:		
Nombre:			Docente <sup>(2)</sup> :		
División:			Nota <sup>(2)</sup> :		
Legajo:			Firma <sup>(2)</sup> :		
Instancia <sup>(1)</sup> :	<b>PP</b>		<b>RPP</b>		<b>SP</b> <b>X</b> <b>RSP</b> <b>FIN</b>

(1) Las instancias validas son: 1<sup>er</sup> Parcial (**PP**), Recuperatorio 1<sup>er</sup> Parcial (**RPP**), 2<sup>do</sup> Parcial (**SP**), Recuperatorio 2<sup>do</sup> Parcial (**RSP**), Final (**FIN**). Marque con una cruz.

(2) Campos a ser completados por el docente.

**En Lenguaje ANSI C:** Desarrollar la biblioteca ArrayList y un programa que la utilice.

**Estructura:**

```
typedef struct {
    int reservedSize; // Tamaño reservado
    int size; // Tamaño de la Lista
    void ** pElements; //Puntero a los elementos de la Lista
    void (*deleteArrayList)();
    void (*add)();
    void (*remove)();
    void (*set)();
    void* (*get)();
    int (*indexOf)();
    int (*len)();
    void (*clear)();
    void (*push)();
    void* (*pop)();
    int (*contains)();
    int (*isEmpty)();
    // extras
    int (* containsAll)();
    struct ArrayList* (* clone)();
    struct ArrayList* (*subList)();
}ArrayList;
```

**Funciones requeridas:**

**ArrayList\* al\_newArrayList(void)** -> Crea y retorna un nuevo ArrayList. Es el constructor, ya que en él daremos valores iniciales a las variables y asignaremos las funciones a sus punteros.

**void al\_deleteArrayList(ArrayList\*)** -> Elimina el ArrayList

**void al\_add(ArrayList\* self, void\* element)** -> Agrega un elemento al final de ArrayList..

**void al\_remove(ArrayList\* self, int index)** -> Elimina un elemento en ArrayList, en el índice especificado.

**void al\_set(ArrayList\* self, int index, void\* element)** -> Inserta un elemento en ArrayList, en el índice especificado.

**void\* al\_get(ArrayList\* self, int index)** -> Retorna un puntero al elemento que se encuentra en el índice especificado.

**int al\_indexOf(ArrayList\* self, void\* element)** -> Retorna el índice de la primera aparición de un valor en el ArrayList.

**int al\_len(ArrayList\* self)** -> Retorna el tamaño del ArrayList.

**void al\_clear(ArrayList\* self)** -> Borra todos los elementos de ArrayList.

**void al\_push(ArrayList\* self, int index, void\* element)** -> Desplaza los elementos e inserta en la posición index

**void\* al\_pop(ArrayList\* self, int index)** -> Retorna un puntero al elemento que se encuentra en el índice especificado y luego lo elimina de la lista.

**int al\_contains(ArrayList\* self, void\* element)** -> Comprueba si existe del elemento que se le pasa como parámetro y devuelve 0 o 1.

**int al\_isEmpty(ArrayList\* self)** -> Retorna cero si contiene elementos y uno si no los tiene.

**Funciones extras:**

*int al\_containsAll(ArrayList\* self, ArrayList\* array)* -> Comprueba si los elementos pasados son contenidos por el ArrayList.

*ArrayList\* al\_clone(ArrayList\* self)* -> Retorna un nuevo ArrayList copia del ArrayList original

*ArrayList\* al\_subList(ArrayList\* self, int from, int to)* -> Retorna un nuevo ArrayList con el subconjunto de elementos.