| ***UTN – UA Mar del Plata – TSP***  **Laboratorio de Computación 2**  **Segundo Parcial** Noviembre 2019 | Nombre y Apellido: | Nota: |
| --- | --- | --- |

Decidimos que queremos organizar todas las películas que vimos, de forma ordenada, para tener orden en nuestra vida cinéfila. Por lo tanto, empezamos diseñando estas estructuras:

| **árbol** | **Película** |
| --- | --- |
| typedef struct nodoArbolPeli{  pelicula dato;  struct nodoArbolPeli \* izq;  struct nodoArbolPeli \* der;  } nodoArbolPeli; | typedef struct  {  int id;  char nombre[20];  int duracion;  char direccion[30];  char genero[20];  int anio;  } pelicula; |

Luego, esto es lo que debemos organizar a continuación:

| Obtenido | Valor | Inciso |
| --- | --- | --- |
|  | 10 | 1 - Diseñar una estructura compuesta de arreglo de árboles, donde cada celda contenga un árbol con todas las películas de un director, y desarrollar la función que permita inicializar el arreglo. |
|  | 10 | 2 - Agregar un nuevo director o directora. |
|  | 10 | 3 - Buscar la celda de un director o directora de forma recursiva y devuelva su posición. |
|  | 10 | 4 - Insertar un nuevo elemento (Tenga en cuenta de que si la celda no existe debe ser creada). |
|  | 15 | 5 - Pasar todos los elementos desde el origen de datos(archivo), al arreglo de árboles. |
|  | 10 | 6 - Mostrar todos los elementos del arreglo, listando directores con sus películas. |
|  | 15 | 7 - Buscar una película por dirección y nombre. |
|  | 20 | 8 - Hacer una función que busque a una película por director y año, dentro del árbol.  (Tenga en cuenta que están ordenadas por nombre) |
|  | - | 9 - Realizar un main que llame a las funciones anteriores y demuestre el correcto funcionamiento del programa. |

**Tabla de puntuación:**

| **Obtenido** | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nota** | 1 | 2 | 4 | 5 | 6 | 6 | 7 | 8 | 9 | 10 |
| **Estado** | **Desaprobado** | | | | | **Aprobado** | | | | |