* **Datos de Alumno:**

Regueiro, Ignacio Hernán

División 1°C

Entiendo que la consigna no aplica para el recuperatorio del segundo parcial; No agregamos ninguna estructura a algo previo con 2 estructuras ya existentes, como indica la consigna.

* **Enunciado de informes Anexados:**

Los informes agregados son:

**LinkedList\* Regueiro\_Project\_FilterByContractor(LinkedList\* pArrayListProject, LinkedList\* pArrayListTeam, int saveLL);**

Informa la lista de proyectos filtradas por un contractor (En este caso, fixeado al contractor A)

**void Regueiro\_Project\_FilterByWorktime(LinkedList\* pArrayListProject, LinkedList\* pArrayListTeam);**

Informa la lista de proyectos filtradas por el tiempo estimado de trabajo (En este caso, fixeado a un tiempo mayor a 60h)

**void Regueiro\_Project\_CountExcludingContractorB(LinkedList\* pArrayListProject, LinkedList\* pArrayListTeam);**

Informa la cantidad de proyectos existentes descartando los que contengan al Contractor B.

**void Regueiro\_Project\_CountByPayment(LinkedList\* pArrayListProject, LinkedList\* pArrayListTeam);**

Informa la cantidad de proyectos existentes que superen los 1800 en pago.

* **Documentación y prototipo de funciones núcleo.**

**int Menu();**

Despliega el menú de opciones y devuelve la opción seleccionada.

**Project\* project\_new();**

**Project\* project\_newParametros(char\* idProjectStr, char\* idTeamStr, char\* nameStr,char\***

**estimatedWorktimeStr, char\* contractorStr, char\* paymentStr);**

**Team\* team\_newParametros(char\* idTeamStr, char\* nameStr, char\* membersStr);**

**Team\* team\_new();**

Constructores con y sin parámetros de las estructuras Project y Team.

**void project\_showProject(Project\* project, Team\* team);**

**void project\_showProjectByIndex(LinkedList\* pArrayListProject, LinkedList\* pArrayListTeam, int index);**

ShowProject recibe un proyecto junto con el equipo correspondiente y los muestra.  
ShowProjectByIndex recibe ambas LL, y el índice del proyecto, y llama a ShowProject para mostrar el proyecto.

**int project\_filterByWorktime(void\* pElement);**

**int project\_filterByContractor(void\* pElement);**

FilterByWorktime y FilterByContractor Se utilizan en conjunto de ll\_filter. Reciben ambas un Proyecto, casteado en la función, y comparan contra un Worktime / Contractor fixeado. Devuelven 1/0 dependiendo de si se debe agregar o no a la LL.

**int project\_countProjectsByPayment(void\* pElement);**

**int project\_countProjectsExcludingCB(void\* pElement);**

countProjectsByPayment y countProjectsExcludingCB Se utilizan en conjunto con ll\_count. Reciben ambas un proyecto, casteado en la función. La primera compara contra un Payment fixeado (1800) y cuenta si cumple la condición. La segunda compara contra un contractor fixeado (Contractor B) y cuenta si no se cumple la igualdad. Ambas devuelven el contador.

**Team\* team\_getById(LinkedList\* pArrayListTeam, int searchId);**

Obtiene una estructura de Team en base al ID de búsqueda. Devuelve NULL si no se encuentra el ID específicado.

**int parser\_ProjectFromText(char\* path , LinkedList\* pArrayListProject);**

**int parser\_TeamFromText(char\* path , LinkedList\* pArrayListTeam);**

Parsers de las estructuras Project y Team. Reciben los datos de los .csv, realizan el parsing llamando a las funciones constructoras, crean las LinkedList y agregan los elementos correspondientes.

**int Regueiro\_Project\_SaveFilterAsText(char\* path , LinkedList\* pArrayListProject, LinkedList\* pArrayListTeam)**

Función de tipo controller. Se encarga de guardar en un nuevo archivo la LinkedList filtrada creada por Regueiro\_Project\_FilterByContractor.

* **Link al video explicativo.**

[**https://drive.google.com/file/d/1ylrh5ojja8\_dJL4pqJebfaf0sr8mVD\_m/view?usp=sharing**](https://drive.google.com/file/d/1ylrh5ojja8_dJL4pqJebfaf0sr8mVD_m/view?usp=sharing)