|  |
| --- |
| *TRABAJADOR* |
| ATRIBUTOS  - tipo: \*character  - legajo: \*integrer  - nombre: \*string  - paga: \*integrer  - alta: \*boolean  - sueldo: \*integrer |
| MÉTODOS  //Constructor por defecto  + Trabajador();  //Constructor con parametros  + Trabajador(char nuevo\_tipo,int nuevo\_legajo, string nuevo\_nombre, int nueva\_paga, bool nueva\_alta);  //Setters  + Asignar\_alta(bool nueva\_alta);  + Asignar\_sueldo\_liquidado(int nuevo\_sueldo\_liquidado)  + Liquidar\_sueldo()  + A\_cadena()  //Getters  + Obtener\_tipo()  + Obtener\_alta()  + Obtener\_legajo()  + Obtener\_sueldo\_liquidado()  + Obtener\_nombre()  + Obtener\_paga()  //Destructor  + virtual ~Trabajador(); |

|  |
| --- |
| *NODO* |
| ATRIBUTOS  - Elemento: Trabajador\*  - Psig: Nodo\* |
| MÉTODOS  //Constructor con un parametro  + Nodo(Trabajador\* d);  //Constructor con dos parámetros  + Nodo(Trabajador\* d, Nodo\* psig);  + Asignar\_elemento(Trabajador\* d)  + Asignar\_siguiente(Nodo\* psig);  + Obtener\_elemento()  + Obtener\_siguiente()  //Destructor  + ~Nodo(); |

|  |
| --- |
| *LISTA* |
| ATRIBUTOS  - primero: Nodo\*  - tamanio: integrer |
| MÉTODOS  //Constructor  + Lista();  - cambiar\_nodo(Nodo\* nodo,int pos)  - retirar(int pos)  + obtener\_tamanio()  + agregar(Trabajador\* d, int pos)  + eliminar(int pos)  + es\_vacia()  + consultar(int pos)  + obtener\_nodo(int pos)  + swap(int pos1, int pos2)  + ordenar()  //Destructor  + ~Lista(); |

|  |
| --- |
| *EMPLEADO* |
| ATRIBUTOS  - llegadas\_tarde: int \*  - ausencias: int \* |
| MÉTODOS  //Constructor por defecto  + Empleado();  //Constructor con parámetros  + Empleado(char nuevo\_tipo,int nuevo\_legajo, string nuevo\_nombre, int nueva\_paga, bool nueva\_alta, int nueva\_llegadas\_tarde, int nueva\_ausencias);  + obtener\_llegadas\_tarde()  + obtener\_ausencias()  + liquidar\_sueldo()  //Destructor  + ~Empleado() |

|  |
| --- |
| *CONSULTOR* |
| ATRIBUTOS  - horas\_catedra: int \*  - horas\_descontadas: int \* |
| MÉTODOS  //Constructor por defecto  + Consultor()  //Constructor con parámetros  + Consultor(char nuevo\_tipo,int nuevo\_legajo, string nuevo\_nombre, int nueva\_paga, bool nueva\_alta, int nueva\_horas\_catedra, int nueva\_horas\_descontadas)  + obtener\_horas\_catedra()  + obtener\_horas\_descontadas()  + liquidar\_sueldo()  //Destructor  + ~Consultor() |

|  |
| --- |
| *JORNALERO* |
| ATRIBUTOS  - dias\_trabajados: int \* |
| MÉTODOS  //Constructor por defecto  + Jornalero();  //Constructor con parámetros  + Jornalero(char nuevo\_tipo,int nuevo\_legajo, string nuevo\_nombre, int nueva\_paga, bool nueva\_alta, int nueva\_dias\_trabajados)  + obtener\_dias\_trabajados()  + liquidar\_sueldo()  //Destructor  + ~Jornalero() |

ACLARACIONES:

- LAS RELACIONES ENTRE CLASES ESTÁN DEFINIDAS EN EL ARCHIVO “Diagrama de relación de clases” DONDE SE VE CON MAYOR PROLIJIDAD Y MAS ACOTADAMENTE CADA CLASE.

- EL ARCHIVO “Funciones.h” ESTÁ PROGRAMADO ESTRUCTURADO, ASIQUE NO ESTÁ INCLUIDO EN EL UML PERO SÍ EN EL DIAGRAMA PARA REFLEJAR COMO ES EL MANEJO DEL PROGRAMA POR PARTE DE CADA CLASE