Práctica V

En esta práctica vamos a aplicar polimorfismo sobre el ejercicio de la práctica IV. A partir de la solución propuesta en dicha práctica (la tenéis disponible en el canal de slack #prácticas) implementar las modificaciones indicadas en los siguientes apartados.

Apartado 1 [1 punto]

Modificar la clase Trabajador: añadir método virtual puro std::string *WhoAmI*() const;

Modificar la clase Empleado, Becario y Responsable redefiniendo el método virtual puro de su clase base de manera que devuelva los strings “empleado”, “becario” y “responsable” respectivamente.

Apartado 2 [4 puntos]

Sobrecargar el operador << para mostrar por terminal toda la información encapsulada en un trabajador cualquiera (Empleado, Becario o Responsable), utilizando la siguiente declaración:

```
std::ostream& operator <<(std::ostream &os, std::shared_ptr<Trabajador> const &trab);
```

Recuerda que deberás aplicar downcasting para poder mostrar la información contenida en las clases derivadas de la clase Trabajador. Ya que sin este mecanismo no podrás acceder desde la clase base (Trabajador) a las funcionalidades que expandan sus clases derivadas (Empleado, Becario, Responsable).

Apartado 3 [5 puntos]

Modificar la función principal de manera que tengamos un único contenedor de punteros inteligentes a trabajador std::vector<shared_ptr<Trabajador>> Plantilla; donde almacenaremos todos los puestos creados durante la ejecución del programa (Becarios, Empleados y Responsables).

Recuerda que deberás aplicar downcasting en los cases del switch que consideres necesario.