

- **Argumentación de la solución propuesta**

En el diagrama de clases podemos observar que hay 3 clases de las cuales 2 heredan de Empleado: Administrador y Jornalero. Esto, con el objetivo de reutilizar código y hacer el programa mucho mas eficiente.

En la clase Administrador, se agregó 2 atributos, HorasExtras y porcentajeCompensacion para aumentar el salario del administrador con base a sus horas trabajadas. También, se creó un constructor que contiene 2 atributos que fueron heredados de la clase Empleados, y 1 atributos propio de la clase el cual fue HorasExtras, para facilitar diferentes operaciones que requerían este parámetro. Y, por último, agregamos 2 métodos, calcularSalario, para poder tener el salario total de cada administrador y toString, para poder imprimir los diferentes parámetros heredados y no heredados de Empleado).

En la clase Jornalero, se agrego 1 atributo el cual fue HorasExtras que nos sirvió para calcular el salario total. Al igual que en la clase Administrador, se creó un constructor que contiene 2 atributos heredados de la clase Empleados, y 1 atributo propio de la clase el cual fue HorasExtras. Para concluir, tenemos los métodos, los cuales fueron calcular salario para conocer el salario total de cada Jornalero, y toString, para imprimir los parámetros heredados y no heredados de Empleado

Ventajas	Desventajas
Es mucho más fácil de programar	Mucho código depende de otro, así que si se llegara a perder la clase Empleado, se tendría que reescribir las otras 2 clases
Se reutiliza mucho código	Si tenemos demasiadas clases en una jerarquía, puede llegar a ser complejo de entender para el programador
Es mas legible, ya que los documentos no están llenos de código	Es más difícil detectar errores
Facilita el mantenimiento	