



## Ciclo 1 Fundamentos de programación

### Reto 1

**Descripción del problema:** Una empresa desea saber el total a pagar un empleado durante 1 día laborado, teniendo en cuenta que la empresa paga a sus empleados según las horas laborales trabajadas y horas extras las cuales se deben calcular automáticamente si pasan de 8 horas, la parametrización del sistema dependerá de lo siguiente:

Horas laborales 8 si el trabajador excede el mínimo las horas se tomaran como extras.

Valor hora laboral: 3.785

Valor hora extra: 4.731

Una vez calculado el valor a pagar según las horas trabajas se procederá a pagarle al usuario las siguientes prestaciones de ley:

(+)Auxilio de transporte mensual: 106.454.

(+)Bonificaciones: Ingresadas por el usuario.

(-) salud y pensión mensual: 258927.

Escriba una función qué reciba cómo parámetro: una variable de tipo diccionario con el nombre del trabajador, horas trabajadas, bonificación, que permita calcular el valor a pagar al empleado incluyendo las prestaciones de ley durante un día, es importante redondear a dos decimales el valor total a pagar, además debe mostrar un mensaje con la siguiente estructura: “El empleado {nombre} total a pagar {total}” .

### Ejemplo:

nombre	horas	boni	return
Ricardo	10	200000	El empleado Ricardo total a pagar 234659.57



### Entradas Diccionario:

```
{  
  "nombre": "Ricardo", # Nombre del empleado  
  "horas": 10, # Horas laboradas por un día  
  "boni": 200000 # Bonificación realizadas en el día  
}
```

### Salida:

Tipo del retorno	Descripción
str	“El empleado {nombre} total a pagar {total}”.

### Esqueleto:

```
'''  
Empleado Diccionario  
empleado = {  
    "nombre": "Ricardo", # Nombre del empleado  
    "horas": 10, # Horas laboradas por un día  
    "boni": 200000 # Bonificación realizadas en el día  
}  
'''  
def calcularPago(empleado : dict)-> str:  
  
    pass
```