


Prueba Java #02

Caso

Se necesita tener **registros** de todos los programadores para mantener un *control sobre ellos en la empresa* y obtener **métricas** de este listado.

Formulario

Cada **Trabajador** tiene los siguientes datos:

- Rut
- Nombre
- Estado civil (*solo puede ser una*)
 - Casado(a)
 - Separado(a) judicialmente
 - Divorciado(a)
 - Viudo(a)
 - Soltero
- Sexo (*solo puede ser una*)
 - Femenino
 - Masculino
- Especialidad (*pueden ser varias*)
 - Java
 - C#
 - SQL
 - PHP
 - CSS
 - JavaScript
 - MongoDB
 - Redis
 - Spring
 - Django
- **Ciudad** (ref 1) 
- **Sueldo bruto**

💻 Estos datos deben poder ingresar en un **formulario** y mostrarse en una **tabla**

🔥 **Donde** 🔥:

Debe existir un botón que agrega estos datos a una tabla

- Agrega exitosamente a una persona a la tabla con el botón (5 pts 💎)
- Luego de agregar una persona **Limpia los campos** y deja listo el formulario para agregar otra (5 pts 💎)
 - **nota:** la limpieza de los campos deben ser en todos los componentes del formulario

nota : los datos que se **mostrarán** (tabla) no serán los mismos del **formulario**

nota 2: ver las referencias 🔗 en el apartado *referencias y consideraciones* para obtener mas información sobre su creación y comportamiento

Muestra de datos (Tabla)

Estos datos del **trabajador** deben se deben mostrar en una **tabla** la cual tiene el siguiente formato

JTable

Rut	Nombre	Estado Civil	Sexo	Especialidad	Ciudad	Sueldo Liquidado
111	Crist	S	M	full kchero	owo	1231231

🔥 **Donde** 🔥:

- Rut
 - Muestra el rut de la persona (1 pts 💎)
- Nombre
 - Muestra el nombre de la persona (1 pts 💎)
 - Muestra solo los primeros 5 caracteres de su nombre completo
 - Ej: Cristian Estay -> Crist (2 pts 💎)
- Estado civil
 - Muestra el estado civil de la persona (1 pts 💎)
 - Solo se muestra la letra inicial del estado civil (1 pts 💎)
- Sexo
 - Muestra el sexo de la persona (1 pts 💎)
 - Solo muestra **F** o **M** correspondientemente (1 pts 💎)
- Especialidad
 - Muestra las especialidades de la persona (1 pts 💎)

- Las muestra separadas por el carácter `|` (pipe) (1 pts ◆)
- Si las especialidades son mas de 4 mostrará `full kchero` (2 pts ◆)
- Ciudad
 - Muestra la ciudad de la persona (1 pts ◆)
 - Si la ciudad comienza con 'R' o 'r' mostrara `pueblo paleta` (2 pts ◆)
- Sueldo líquido
 - El **sueldo líquido** se calcula en función de `sueldo bruto * 0.9` (2 pts ◆)
 - A el resultado matemático anterior se le agregaran **\$100.000 adicionales por especialidad** (2 pts ◆)
 - Si tiene **más de 5 especialidades** solo se considerara una **unica bonificación de \$550.000** (2 pts ◆)

Muestra de datos (Resumen)

La muestra de **datos resumen** será un conteo de los valores anteriores los cuales se considerará

🔥 Donde 🔥

Se muestra la:

- Cantidad de trabajadores agregados al sistema (2 pts ◆)
- Cantidad total de mujeres (2 pts ◆)
- Cantidad total de hombre (2 pts ◆)
- Cantidad total de sueldos líquidos de todos los trabajadores (2 pts ◆)

nota: todos estos datos se mostrarán en varios **JLabels** en el mismo formulario

Referencias y consideraciones 🔗

🔥 Donde 🔥

ref 1:

- La ciudad es un **JComboBox** que en primera instancia estará vacío (3 pts ◆)
- En este **JComboBox** podrá agregarse registros desde el formulario **Meta-data** (3 pts ◆)
 - **Meta-data** es un formulario que está en otro panel no visible para el formulario principal (usar JTabbedPane) (1 pts ◆)
- No puede agregar una persona con una ciudad vacía (3 pts ◆)