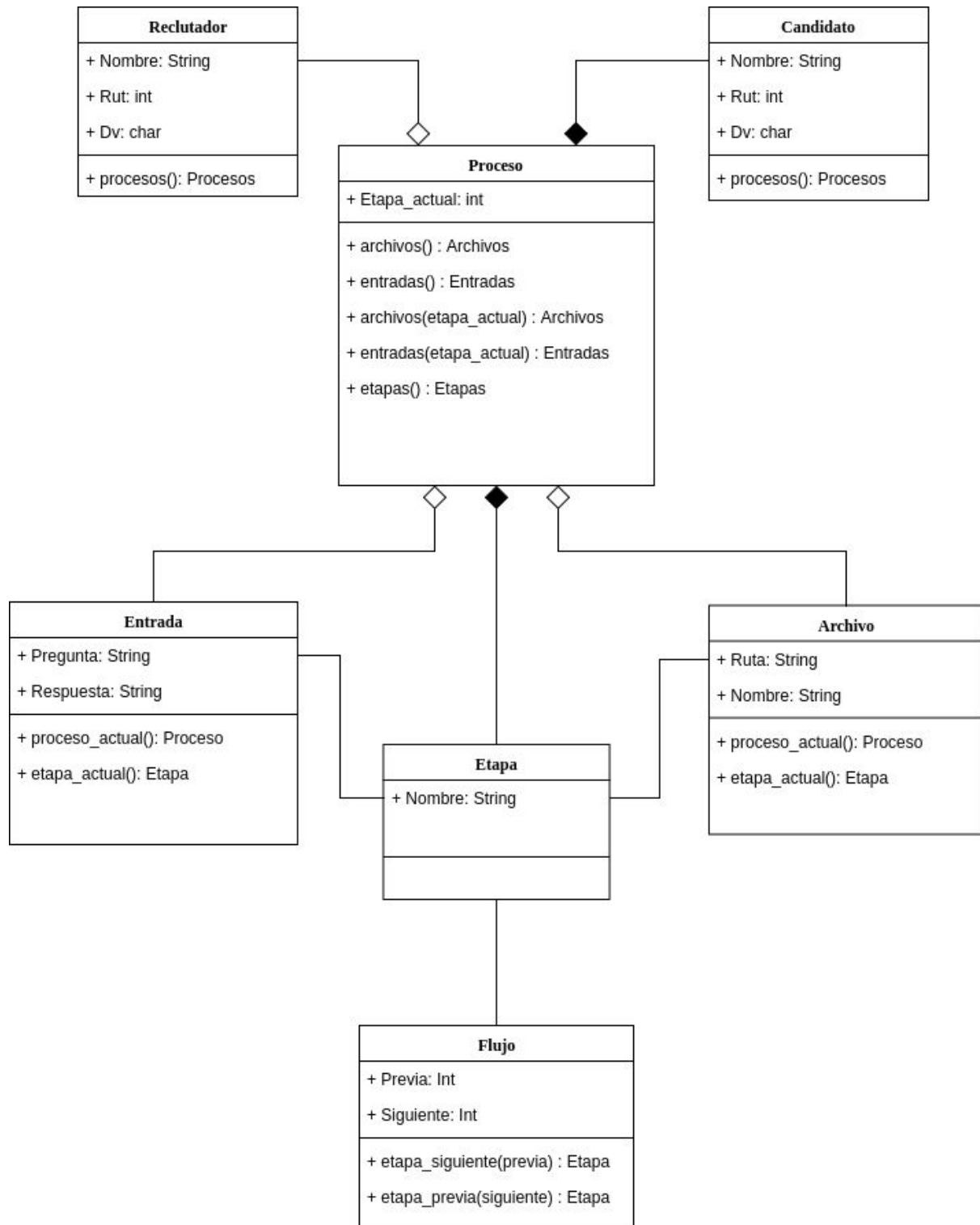


Álvaro Martínez Toledo

Test 4Talent

1-



Explicación: Siguiendo el caso ejemplo, las 5 etapas descritas se pueden guardar como **Etapas**, y sus flujos estarán definidos en **Flujo**. **Flujo** guarda tanto Etapa Previa como Siguiente, por lo que es posible construir una secuencia configurable.

Álvaro Martínez Toledo

Test 4Talent

Proceso guardará la candidatura de un **Candidato**, y tendrá registradas todas las **Etapas** por la cual ha pasado. Además, el **Proceso** tendrá **Entradas** y **Archivos**.

Entradas y **Archivos** se corresponden a su vez con **Etapas**, por lo que al final será posible obtener las **Entradas** y **Archivos** que una **Postulación** contiene en una determinada **Etapas**.

Entradas contiene el atributo tipo string 'preguntas', dado que las entrevistas pueden contener preguntas libres.

2.-

```
1. USERS = [  
2.   {name: "Juan", country: "Chile", age: 30},  
3.   {name: "Pedro", country: "Argentina", age: 15},  
4.   {name: "Manuel", country: "Perú", age: 26},  
5.   {name: "Jose", country: "Ecuador", age: 22},  
6. ]
```

```
1. def sort_by_country  
2.   USERS.sort_by{|hash| hash[:country]}  
3. end  
4.  
5. def get_youngest  
6.   USERS.sort_by{|hash| hash[:age]}.first[:name]  
7. end
```

3.-

https://github.com/amartineztoledo/weather_app

<http://weather-4talent.herokuapp.com/>

Se han desactivado rutas y acciones para Cities (new, create, delete, destroy, edit, update), todo esto tiene sus respectivas pruebas escritas.

El botón Borrar Temperaturas (solo en heroku) se realizó para borrar los datos con facilidad.

Álvaro Martínez Toledo

Test 4Talent

4.-

1. `scope :private_benefits, -> {joins(postulations: [{benefit: [:company]}]).where("companies.type = ? and benefits.amount < ?", "private", 10000000)}`
2. `# Se utiliza scope para obtener datos usando queries.`

Nota: En rails, la columna type se utiliza para herencia de tablas (Single Table Inheritance), por lo que este código fallará a no ser que se creen las clases Private y Public como hijas de Company. Por último, es necesario que en type se almacene el nombre de la clase escrita con una mayúscula en su inicio ("Private", "Public"), ya que las clases no pueden comenzar con minúscula.

Claro está, que se puede desactivar que se use STI con la columna type.

El código de ejemplo modificado quedaría así:

1. `class Company < ActiveRecord::Base`
 `TYPES = ["Private", "Public"]`
 `has_many :benefits`
 `validates :type, inclusion: { in: TYPES }`
2. `end`
- 3.
4. `class Private < Company`
5. `end`
- 6.
7. `class Public < Company`
8. `end`

Y el scope:

3. `scope :private_benefits, -> {joins(postulations: [{benefit: [:company]}]).where("companies.type = ? and benefits.amount < ?", "Private", 10000000)}`