

Builder pattern

Necesitamos una clase Persona, con nombre y edad (los dos obligatorios) ...

```
private String nombre;  
private int edad;  
  
public Persona(String nombre, int edad) {  
    this.nombre = nombre;  
    this.edad = edad;  
}
```


... también tendremos el municipio, pero este es opcional ...

```
public class Persona {  
  
    private String nombre;  
    private int edad;  
    private String municipio;  
  
    public Persona(String nombre, int edad) {  
        this.nombre = nombre;  
        this.edad = edad;  
    }  
  
    public Persona(String nombre, int edad, String municipio) {  
        this.nombre = nombre;  
        this.edad = edad;  
        this.municipio = municipio;  
    }  
}
```


... si es menor de edad, es obligatorio el colegio, si es mayor de edad el lugar de trabajo...

```
public Persona(String nombre, int edad) {  
}  
  
public Persona(String nombre, int edad, String municipio) {  
}  
  
public Persona(String nombre, int edad, String colegio) {  
}  
  
public Persona(String nombre, int edad, String lugarTrabajo) {  
}  
  
public Persona(String nombre, int edad, String municipio, String colegio) {  
}  
  
public Persona(String nombre, int edad, String municipio, String lugarTrabajo) {  
}
```

ya ni compila ... y aunque compilase ...

“

Separate the construction of a complex object from its representation so that the same construction process can create different representations.

— *GoF*

”

Constructor de Persona private. Acceso solo mediante el "Builder"

```
Persona madre = new Persona.Builder("Maria")  
    .setMunicipio("Selva")  
    .setMayor(37, "Google")  
    .build();
```

```
Persona hijo = new Persona.Builder("Pedro")  
    .setMenor(4, "Colegio de Selva")  
    .build();
```

el nombre es obligatorio
municipio opcional

métodos específicos al tipo
de persona

method chaining


```
public static class Builder {  
    private Persona persona;
```

```
    public Builder(String nombre) {  
        persona = new Persona();  
        persona.nombre = nombre;  
    }
```

← obligatorios comunes

```
    public Builder setMunicipio (String municipio) {  
        persona.municipio = municipio;  
        return this;  
    }
```

← opcionales

method
chaining →

...

```
    public Persona build() {  
        return persona;  
    }  
}
```


También el Builder

```
public Builder setMenor(int edad, String colegio) {  
    if (edad >= 18) throw new IllegalArgumentException("es mayor de edad " + edad);  
    persona.edad = edad;  
    persona.colegio = colegio;  
    persona.lugarTrabajo = null;  
    return this;  
}
```


Nuevos requisitos:

- tanto colegio como lugar de trabajo son opcionales
- puede haber más propiedades exclusivas a mayores y menores
- queremos que el compilador nos "guie" mostrando solo las correctas

```
Persona madre = new Persona.Builder("Maria")  
    .setMunicipio("Selva")  
    .setMayor(37)  
    .setLugarTrabajo("Google")  
    .build();
```

```
Persona hijo = new Persona.Builder("Pedro")  
    .setMenor(4)  
    .setColegio("Colegio de Selva")  
    .build();
```

```
Persona mal = new Persona.Builder("Luisa")  
    .setMayor(20)  
    .setColegio("Colegio de Villa Arriba")  
    .build();
```

← ¡No compila!


```
public BuilderMayor setMayor(int edad) {  
    if (edad < 18) throw new IllegalArgumentException("es menor de edad " + edad);  
    persona.edad = edad;  
    return new Persona.BuilderMayor(persona);  
}
```

```
public Persona build() {  
    return persona;  
}
```

```
public static class BuilderMayor {  
  
    private Persona persona;  
  
    public BuilderMayor(Persona persona) {  
        this.persona = persona;  
    }  
  
    public BuilderMayor setLugarTrabajo (String trabajo) {  
        persona.lugarTrabajo = trabajo;  
        return this;  
    }  
  
    public Persona build() {  
        return persona;  
    }  
}
```


Builder pattern