

# TP2-POO: Formulário Web

Este trabalho prático incide sobre os conceitos de herança, polimorfismo, genéricos e coleções, estudados no âmbito do paradigma de programação orientada a objetos. Para exercitar os conceitos aprendidos, vamos desenvolver um projeto.

## 1-Descrição do projeto

Os formulários são peças chave no processo de aquisição de dados na Web. Várias etapas são necessárias para realizar o processo nomeadamente, a criação do formulário em HTML, a introdução dos dados pelo utilizador, a validação e o armazenamento dos dados, pelo servidor Web.

Neste trabalho prático vamos criar a classe `Form`, que representa um modelo neste domínio, para auxiliar na criação e validação de dados em formulários para a Web. Claro é, que é necessário aplicar os conceitos e as técnicas aprendidas em POO.

Formulário Web

Username: tia

Email: tia@gmail.com

Age: 16

Um formulário é composto por vários campos. Cada campo tem uma *label* associada, a qual permite identificar o campo. Cada campo pode ser do tipo `StringField` ou `NumberField`, de acordo com o tipo de dados de entrada. É ainda necessário validar dos dados e no caso de um campo ser inválido é adicionado o erro de validação à lista de erros. Para validar cada campo são criados validadores, por exemplo `Length`, `Required` ou `NumberRange`. Adicionalmente o formulário permite a renderização em HTML e a apresentação dos dados inseridos no formato JSON.

Note que a classe `Form` estende um `HashMap`.  
Um caso de uso, da classe `Form` é a seguinte:

### Exemplo 1 - formulário válido

#### Cliente 1

```
class UsernameForm extends Form{
    UsernameForm(){
        super();
        this.put("username",
            new StringField("Username",new Validator[]{new Length(3)}));
        this.put("email",
            new StringField("Email", new Validator[]{new Required()}));
        this.put("age", new NumberField("Age",
            new Validator[]{new NumberRange(16,99)}));
    }
}

class Client1 {
    public static void main(String[] args) {
        UsernameForm form = new UsernameForm();

        form.get("username").setData("tia");
        form.get("email").setData("tia@gmail.com");
        form.get("age").setData(16);
        form.validate();

        for(String err : form.errors)
            System.out.println(err);

        System.out.println(form.content());
        System.out.println(form.json());
    }
}
```

A tarefa deste projeto é desenvolver um programa que permita resolver as especificações dadas na classe `Client`, utilizando o paradigma de POO. Note que deve identificar as classes, as suas responsabilidades e estruturar o programa de modo a implementar a funcionalidade descrita, utilizando também heranças, polimorfismo, genéricos e coleções.

## 2-Tarefa

A tarefa consiste em implementar a classe `Form` e todas as classes auxiliares, para estruturar a aplicação e garantir, adicionar futuras funcionalidades

facilmente.

Para o `Client` apresentado anteriormente (*Cliente 1*), é obtido o seguinte output.

### Exemplo 1 - formulário válido

#### Output 1

```
<form>
    <label for='email'>email:</label>
    <input name='email' type='text' value='tia@gmail.com' /><br>
    <label for='age'>age:</label>
    <input name='age' type='number' value='16' /><br>
    <label for='username'>username:</label>
    <input name='username' type='text' value='tia' /><br>
</form>
{'email':'tia@gmail.com','age':16,'username':'tia'}
```

### Exemplo 2 - formulário inválido

Para o seguinte `Client` é obtido o seguinte output.

#### Cliente 2

```
class UsernameForm extends Form{
    UsernameForm(){
        super();
        this.put("username",
            new StringField("Username",new Validator[]{new Length(3)}));
        this.put("email",
            new StringField("Email", new Validator[]{new Required()}));
        this.put("age", new NumberField("Age",
            new Validator[]{new NumberRange(16,99)}));
    }
}

class Client2 {
    public static void main(String[] args) {
        UsernameForm form = new UsernameForm();

        form.get("username").setData("ti");
        form.get("email").setData("");
        form.get("age").setData(13);
        form.validate();

        for(String err : form.errors)
```

```

        System.out.println(err);

        System.out.println(form.content());
        System.out.println(form.json());
    }
}

```

## Output 2

```

Error: value empty
Error: value not in range
Error: less than min
<form>
    <label for='email'>email:</label>
    <input name='email' type='text' value='' /><br>
    <label for='age'>age:</label>
    <input name='age' type='number' value='13' /><br>
    <label for='username'>username:</label>
    <input name='username' type='text' value='ti' /><br>
</form>
{'email':'','age':13,'username':'ti'}

```

## 3-Requisitos

- (a) Realize o planejamento através do desenho do diagrama de classes que identifica as classes do problema e da solução.
- (b) Utilize os princípios e técnicas estudadas sobre programação orientada a objetos, identificando e implementando classes coesas.
- (c) Siga uma metodologia de desenvolvimento baseado em testes (*TDD - Test Driven Development*).
- (d) Considere a utilização de exceções.
- (e) Considere os conceitos lecionados em POO nomeadamente, herança, polimorfismo, genéricos e coleções.

## 4-Relatório

O TP2 incluiu o desenvolvimento de um relatório, que deve conter:

- (a) Uma descrição do problema e a abordagem do grupo para a sua resolução.
- (b) O desenho de um diagrama de classes, com a análise realizada.

- (c) Os testes unitários e testes de cenário desenvolvidos.
- (d) Conclusão.

## **5-Entrega**

O trabalho prático deve ser entregue em formato digital, incluindo o código do projeto e o relatório, com indicação dos nomes dos elementos do grupo (máximo 2 elementos), para o email *tiago.candeias@ismat.pt*, até ao dia 09-01-2023.

## **6-Avaliação**

Esta parte do trabalho vale 50% da nota da componente prática da unidade curricular. Os critérios de avaliação são: clareza do texto, modelação adequada do problema, clareza das explicações, qualidade do código e comentários no código.