



Universidade do Minho  
Mestrado em Engenharia Informática  
Engenharia de Linguagens  
Projecto Integrado - Grupo 1  
**1ª Avaliação Intermédia**  
Ano Letivo de 2012/2013

pg22820 - **António Silva**  
pg22781 - **Rui Brito**

12 de Dezembro de 2012

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Planeamento</b>	<b>3</b>
<b>3</b>	<b>Modelação</b>	<b>4</b>
3.1	Diagrama de Classes . . . . .	4
3.2	Use Cases . . . . .	4
3.3	Base de Dados . . . . .	4
<b>4</b>	<b>Linguagem formal para Identificação e Formação</b>	<b>5</b>
4.1	Gramática . . . . .	5
4.2	Processador . . . . .	6
4.3	Exemplo de Input . . . . .	6
<b>5</b>	<b>Linguagem de anotação para descrição das Actividades</b>	<b>7</b>
<b>6</b>	<b>Conclusão</b>	<b>7</b>

## 1 Introdução

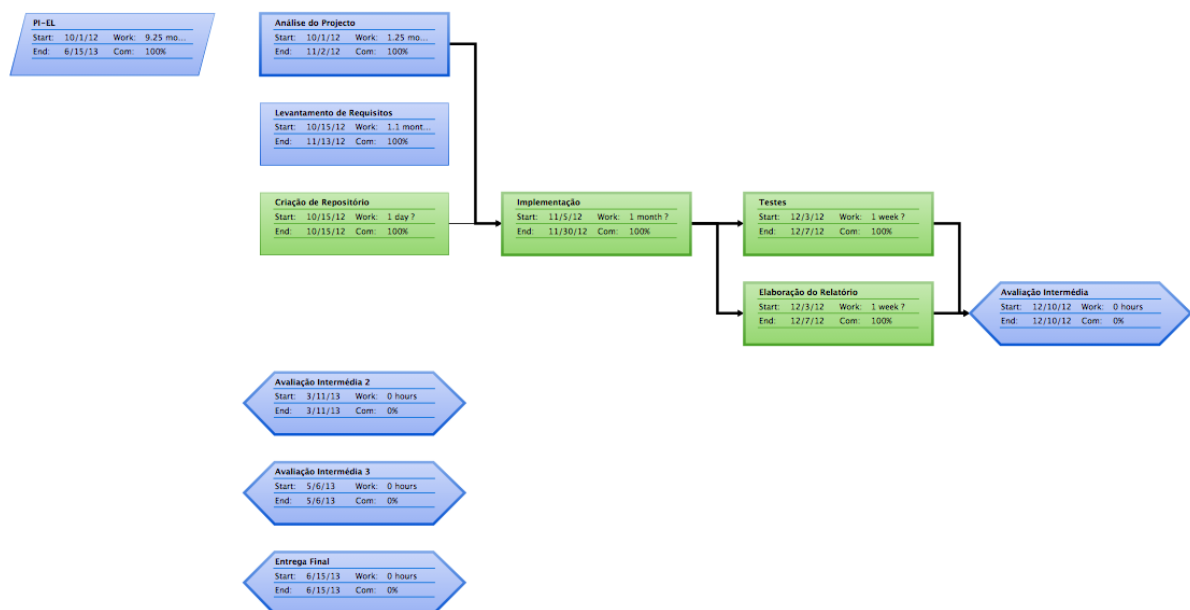
O projecto consiste no desenvolvimento de um sistema de informação que permita gerir a informação curriculares de um docente universitário.

Essa informação a recolher, armazenar e publicar inclui, além da identificação completa do docente, dados sobre a formação, as várias actividades académicas desenvolvidas e resultados atingidos.

Nesta primeira fase foram-nos pedidos o planeamento das actividades a desenvolver, a modelação do problema em mãos (Diagrama de classes, Esquema de Base de Dados, Use Cases...), uma gramática e respectivo processador para uma linguagem de informação e formação Adicionalmente, foi pedido ainda um esquema de uma linguagem de anotação para as actividades desenvolvidas.

## 2 Planeamento

A figura abaixo apresenta o planeamento do projecto.

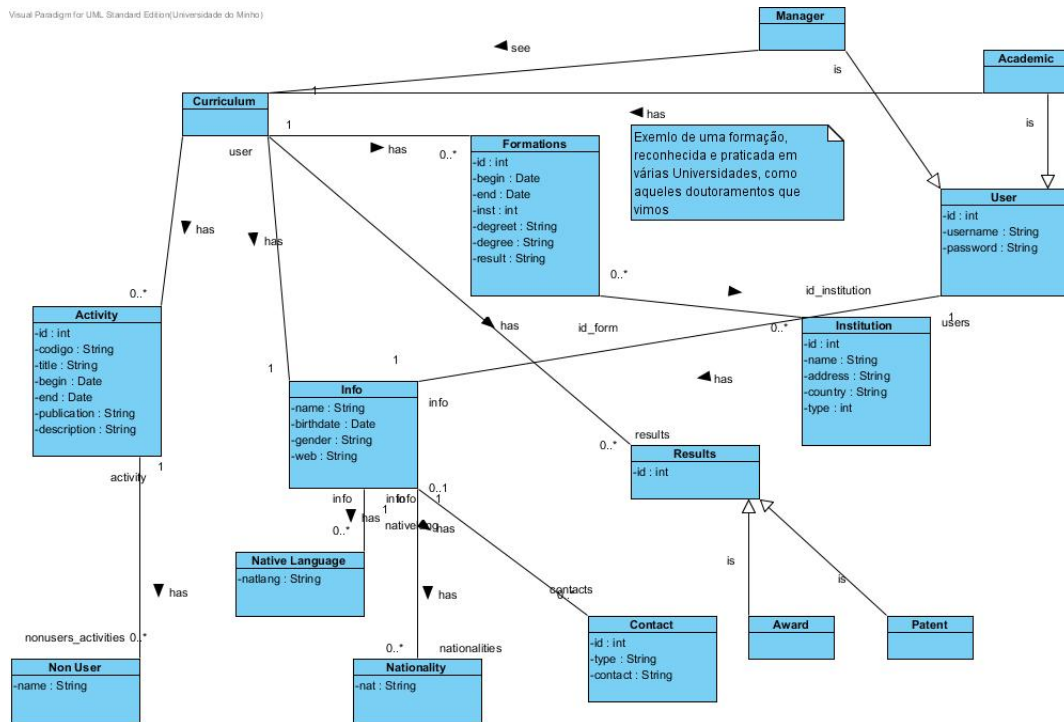


**Figura 1:** Planeamento do projecto

Nota: a figura apresenta um "net plan" pois não era possível por motivos de espaço inserir uma organização por semanas

## 3 Modelação

### 3.1 Diagrama de Classes



**Figura 2:** Diagrama de Classes

O Diagrama de Classes, presente na figura 2 inicialmente desenvolvido estava consideravelmente mais pobre e foi enriquecido também à medida que fomos avançando no projecto. Foi também um enorme ponto de partida para a criação da Base de Dados. A única parte ainda bastante subdesenvolvida é a dos resultados pelo facto de ainda não termos avançado muito nessa questão e ter ficado somente aquilo que retirámos das primeiras leituras, quer do enunciado, quer de exemplos facultados ou encontrados.

### 3.2 Use Cases

Os *Uses Cases* na figura 3 referem-se essencialmente a tarefas possíveis de serem feitas, quer pelo Gestor, quer pelo Académico (na maioria dos casos o académico será o docente). Cada *Use Case* possui uma descrição textual que nos permitiu já ponderar um pouco sobre como irá o sistema responder ao utilizador e interagir com outros sistemas. Podemos ver dois exemplos da descrição textual dos uses cases na figura 4 e 5

### 3.3 Base de Dados

A estrutura da base de dados (figura 6) foi pensada de forma a permitir armazenar, como é óbvio, os dados que foram analisados por exemplo no diagrama de classes. Os seus relacionamentos foram facilmente idealizados.

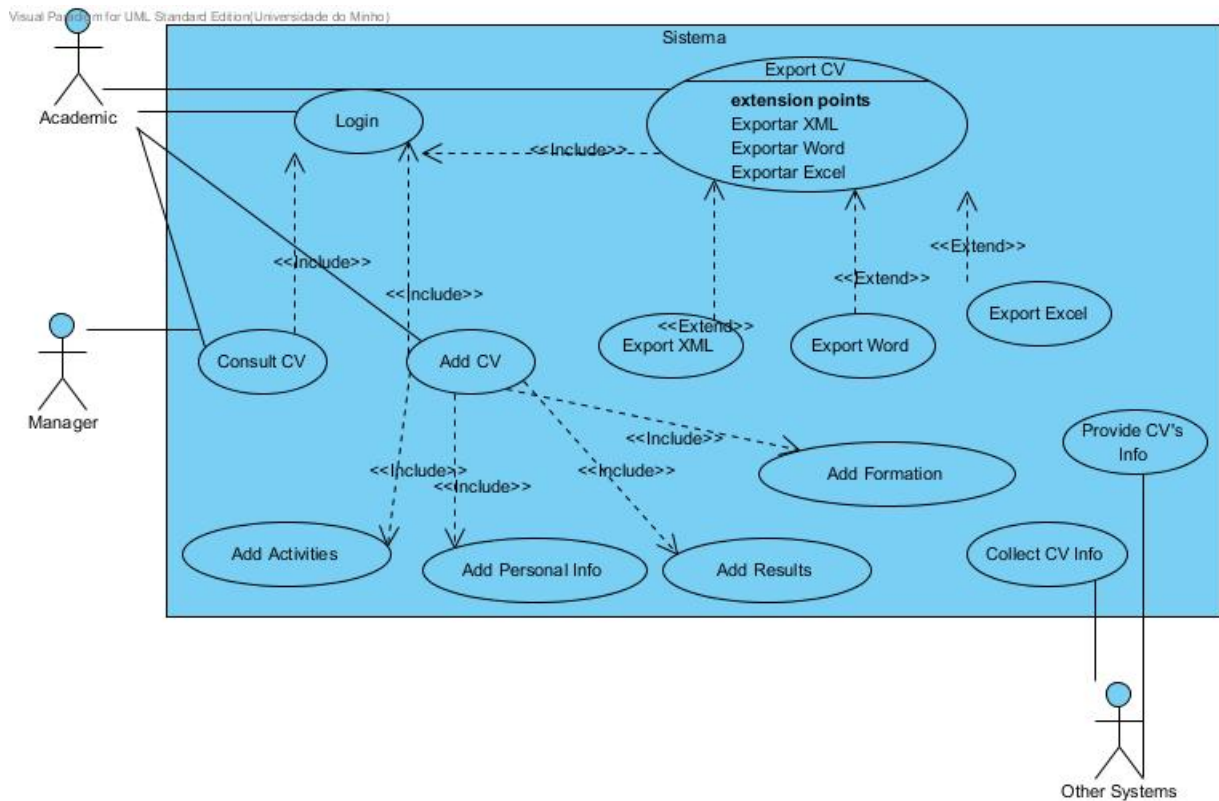


Figura 3: Use Cases

## 4 Linguagem formal para Identificação e Formação

### 4.1 Gramática

A Gramática para a linguagem de identificação e formação foi facilmente criada, recorrendo ao que já tínhamos analisado para o diagrama de classes. No entanto, permitiu-nos também enriquecer mais o diagrama de classes, pois ao irmos escrevendo a gramática lembramo-nos de coisas que nos poderiam fazer falta.

Apesar de tudo não refinámos ainda muito certos campos como o email e o url porque são coisas definidas por normas externas, que queríamos tentar seguir e adaptar à gramática desenvolvida no *AntLR*.

Também aqui decidimos fazer algo que permita futuras actualizações sem implicar uma completa reestruturação da gramática. Uma delas foi aquilo a que nos chamamos *Special ID* (SPID), por exemplo para valores como o País. Isto porque o País é um campo cujos valores podem ser normalizados de modo a que não existam dois países iguais com nomes diferentes, e poderá permitir mais para a frente se acharmos conveniente criar mais uma relação na Base de Dados, de modo a reduzir o espaço ocupado, por exemplo pelas nacionalidades.

SPID

```
: ('A'..'Z') ('a'..'z')* ( ' ' ('A'..'Z') ('a'..'z')*)*
;
```

## 4.2 Processador

Quando discutimos o nosso processador, decidimos desde uma fase embrionária que queríamos evitar a repetição de código. Assim sendo tentamos passar grande parte da responsabilidade para o ficheiro *info\_import.php*, que seria um *template*. Assim grande parte das coisas geradas pelo Parser seriam simplesmente valores etiquetados que ele saberia onde colocar.

Apesar de parcialmente implementada, a detecção de erros semânticos não está completa por falta de tempo. Erros como a data de início ser superior à de fim são apanhados, mas apenas é mostrada uma mensagem e o processamento continua.

Também neste momento o formulário para importação do documento é bastante reduzido e serve unicamente para indicar o ficheiro em questão.

```
<html>
<head></head>
<body>
<form method=\"post\" action=\"info_import.php\" enctype=\"multipart/form-data\">
    <fieldset>
        <legend>Info e Form:</legend>
<input type='file' name='ficheiro' />
    </fieldset>
    <input type=\"submit\" name=\"Enviar\" />
</form>
</body>
</html>
```

O código de execução do *parser* e leitura dos resultados também não é muito complexo. No entanto permite que estejam vários utilizadores simultâneos a executar a aplicação web, sem existir nenhum tipo de conflitos já que o *stdout* é redireccionado para a leitura do php através de um *handler*.

```
$f = (popen('java -jar AntLRParser.jar "'.$_FILES['ficheiro']['tmp_name'].'"', "r"));

$valor = "";
while (!feof($f)) {$valor .= fread($f, 60);}
```

Depois as inserções são feitas na Base de Dados MySQL recorrendo à classe PDO do php.

## 4.3 Exemplo de Input

Aqui está um exemplo de *input* válido para a gramática desenvolvida.

```
@info {
Name: "xpto"
Nationalities: [Portuguese,Canadian]
PersonalContacts: [
Phone: "252000000" ,
Email: xpto@st.pt
]
Birthdate: 13/05/2001
```

```
Gender: M
NativeLang: [Portuguese,English]
Web: http://xpto.com
}
@form {
Begin: 01/01/2008
End: 01/02/2012
Institution:
Name: "Universidade do Minho"
Address: "Gualtar Braga"
Country: Portugal
Type: Public University
Degree: BSc "Engenharia Informática"
Result: 20
}
```

## 5 Linguagem de anotação para descrição das Actividades

O *Schema* (na figura 7) desenvolvido para a descrição de actividades, teve também por modelo o que já tínhamos definido para a Base de Dados, para tentar equilibrar os dados que poderiam ser guardados e os que seriam enviados.

Um facto bastante relevante é permitir que uma actividade esteja relacionada com mais que um utilizador, podendo descrevê-lo como utilizador do sistema, ou não utilizador (e indicando somente o seu nome).

## 6 Conclusão

O objectivo desta primeira avaliação do Projecto Integrado, é garantir que o mesmo segue já a um bom ritmo, servindo também já como um suporte para o desenvolvimento futuro, na medida em que será uma base sobre a qual podemos continuar a construir já mais cientes dos caminhos correctos que escolhemos e daqueles que não estavam assim tão correctos, ao ser mostrado à equipa docente os resultados obtidos até à data.

Flow of Events		Actor Input	System Response
	1		O sistema pergunta se quer ver o seu currículo ou outro.
	2	O utilizador indica que pretende ver o seu currículo	
	3		O sistema guarda o id do utilizador
	4		O sistema carrega a informação pessoal da Base de Dados baseada no id
	5		O sistema carrega os dados sobre a formação da Base de Dados baseada no id
	6		O sistema carrega dados sobre as actividades constantes na BD baseada no id
	7		O sistema carrega dados sobre os resultados obtidos presentes na BD baseado no id
	8		O sistema sintetiza toda a informação obtida
	9		O sistema apresenta a informação sintetizada ao utilizador
2 - <u>Alternativa</u>		Actor Input	System Response
	1	O utilizador indica que pretende ver outro currículo	
	2		O sistema pede os critérios de procura ao utilizador
	3	O utilizador indica os critérios de procura do currículo que pretende ver	
	4		O sistema apresenta uma lista com os currículos que satisfazem os critérios
	5	O utilizador escolhe o registo que pretende ver.	
	6		O sistema guarda o id desse registo
	7		O sistema regressa ao ponto 4
2.5 - <u>Alternativa</u>		Actor Input	System Response
	1	O utilizador não escolhe nenhum dos registos apresentados	
	2	O utilizador cancela a pesquisa	
	3		O sistema redirecciona o utilizador para a sua página principal
2.5.2 - <u>Alternativa</u>		Actor Input	System Response
	1	O utilizador altera os critérios de procura	
	2		O sistema regressa ao ponto 2.4

Figura 4: Use Case - Consult CV



Flow of Events		Actor Input	System Response
	1	O serviço indica qual o serviço que pretende utilizar	
	2		O sistema verifica que esse serviço existe e está registado
	3	O serviço indica quais os dados que pretende	
	4		O sistema verifica que o utilizador permite a partilha desses dados
	5		O sistema codifica a informação para um formato interoperável
	6		O sistema responde ao serviço com a informação pretendida
2 - Excepção [serviço não existe/não registado]		Actor Input	System Response
	1		O sistema verifica que o serviço não existe ou não está registado
	2		O sistema responde ao serviço com um código de erro
4 - Excepção [sem permissões]		Actor Input	System Response
	1		O sistema verifica que o utilizador não permite a partilha desses dados
	2		O sistema responde ao serviço com um código de erro

Figura 5: Use Case - Provide CV's info

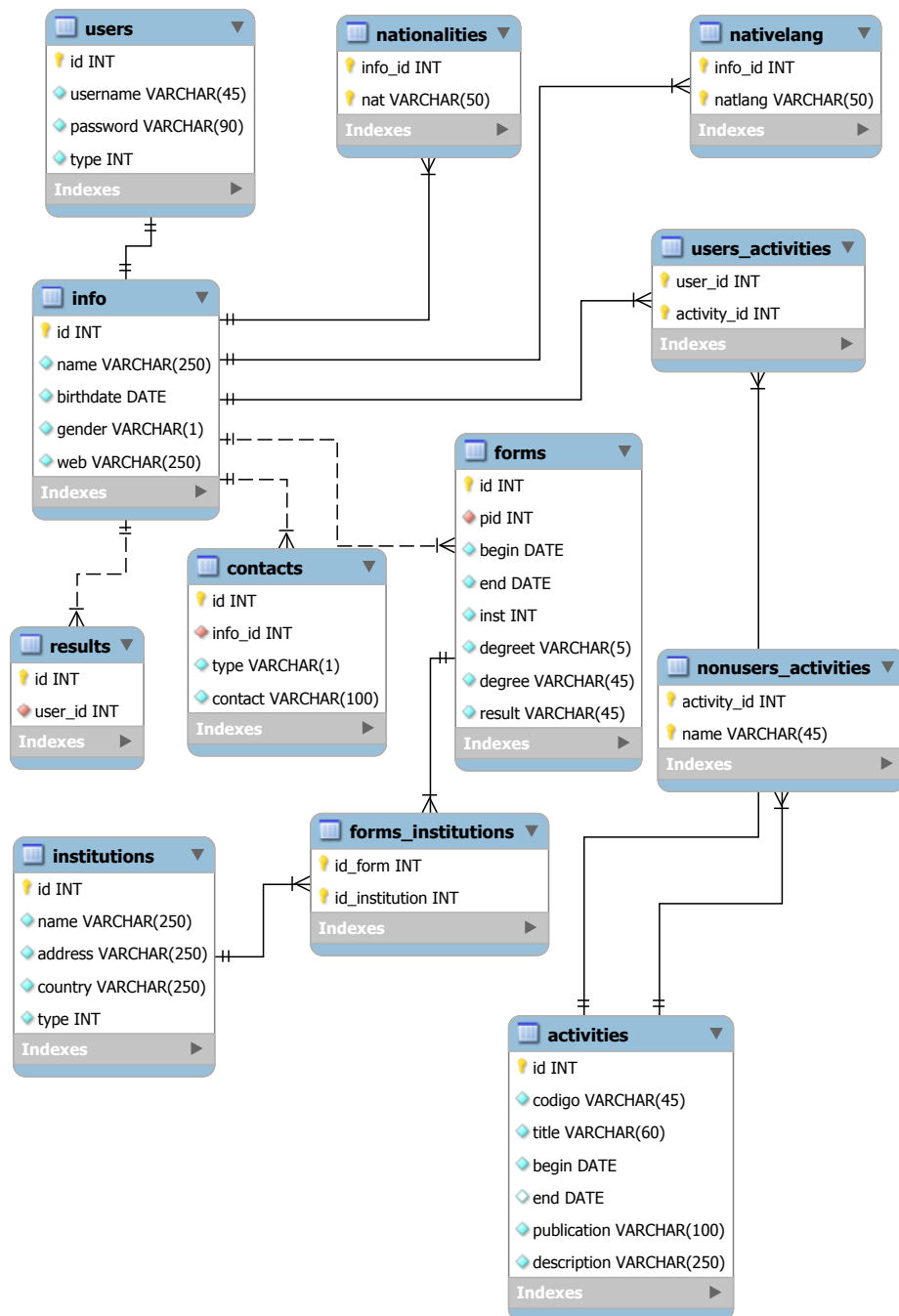


Figura 6: Base de Dados

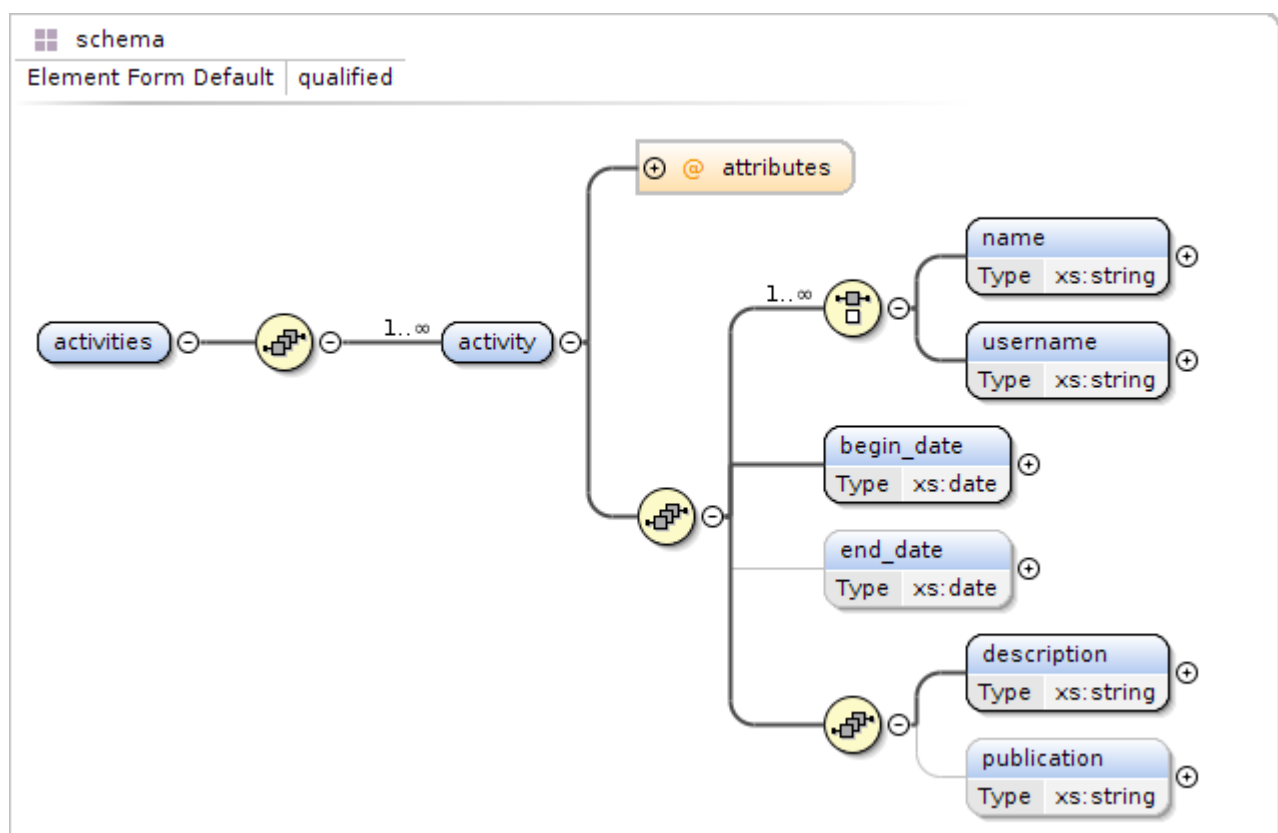


Figura 7: Schema da linguagem