

**Programação Avançada em Java**

**Relatório Projecto nº 3**

**JavaServer Faces (JSFs) & AJAX**

**Actualização da calculadora (Projeto2) utilizando AJAX e JSFs**

Rafaela Lourenço

Ricardo Quirino

## Conteúdo

Conteúdo2

Descrição da aplicação3

Calculadora4

*Chat*6

*Filtros*7

Melhoramentos no código inicial8

Pré-desenvolvimento9

Testes unitários (*JUnit* e *Mockito*)12

## Descrição da aplicação

A aplicação desenvolvida neste terceiro projecto consiste numa actualização da Calculadora Web desenvolvida no segundo projecto da cadeira de Programação Avançada em Java.

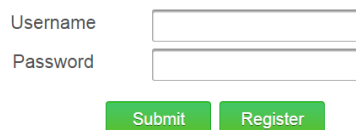
As ferramentas a utilizar neste projecto são os JSFs (*Java Servers Faces*) e o AJAX (*Asynchronous Javascript and XML*). Estas ferramentas permitem que apenas algumas partes da página web seja recarregada de cada vez que existe uma acção dos botões.

O *link* do projecto é: , onde os asteriscos são o número da porta do servidor local.

A aplicação consiste numa primeira página onde se efectua o Login de um utilizador existente ou o registo de um novo utilizador. Existem na base de dados dois utilizadores registados:

**Username:** ricardo, **password:** 123

**Username:** rafaella, **password:** 456



A login form with two input fields. The first field is labeled 'Username' and the second is labeled 'Password'. Below the fields are two green buttons: 'Submit' and 'Register'.

Caso não haja uma correspondência correcta entre *username* e *password* são mostrados diferentes avisos:

- Se o utilizador existir, mas a palavra passe não corresponder, é mostrado o aviso “*Password Inválida*”
- Se o utilizador existir e tiver iniciado sessão que se encontra ainda activa, apresentará o aviso “Utilizador com sessão iniciada e activa!”
- Caso o *username* inserido não esteja registado a mensagem apresentada é “*Username inexistente*”

O registo de utilizador é possível, desde que o *username* escolhido não existe na lista de utilizadores já registados.

Quer o processo de *login* quer o de registo de novo utilizador requiere a introdução obrigatória dos dados, caso contrário os campos aparecem marcados devidamente com a cor vermelha.

Username

Password

Quando o *login* é efectuado com sucesso, o utilizador é redireccionado para a página da calculadora que possui os seguintes elementos:

Na barra superior:

Mensagem de saudação ao utilizador “logado”, data e hora do servidor, botão de “Logout” e botões que permitem a troca entre a calculadora normal e a calculadora científica

No centro:

A calculadora e a sala de *chat*.

Em baixo:

Os botões de acesso ao histórico da sessão e à estatística do servidor.

## Calculadora

Bem-vindo, **rafaela**

14/05/2015 11:28

LOGOUT

Normal

Científica

AC +/- % /

7 8 9 x

4 5 6 +

1 2 3 -

0 ← . =

HISTÓRICO

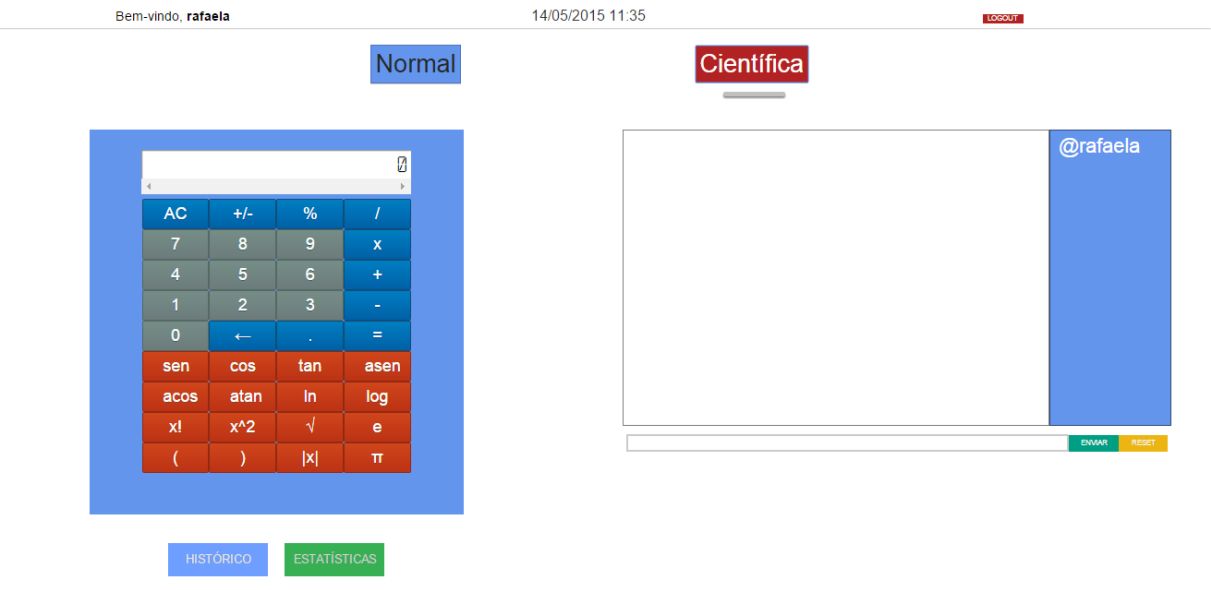
ESTATÍSTICAS

@rafaela

ENVIAR RESET

Versão normal da calculadora

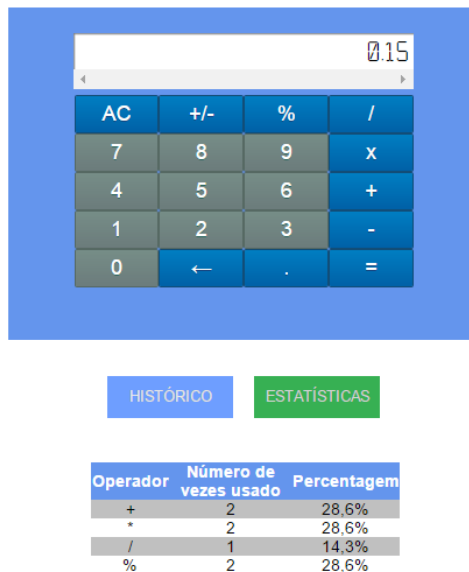
Quando se muda para o modo científico, os botões das funções científicas da calculadora ficam visíveis.



Versão científica da calculadora



Vista do histórico da sessão, com o tempo médio de cada operação realizada.



Vista das estatísticas do servidor local.

## Chat

O *chat* é constituído, fundamentalmente por três secções:

- Janela principal das mensagens
- Secção de utilizadores *online*
- Secção para escrever e enviar mensagens

A sala de *chat* tem como objectivo permitir a trocar de mensagens entre todos os utilizadores registados e que façam login de forma válida.

Estes utilizadores aparecerão, na forma do seu *username* ou nome de utilizador na secção à direita, permitindo assim perceber se existem utilizadores e quais esses mesmos utilizadores.

A maior secção é a janela de mensagens. É nesta secção que aparecem as mensagens dos utilizadores, sendo que cada mensagem inclui: Data da mensagem, nome do utilizador e a mensagem, propriamente dita.

Quando um utilizador entra ou sai da sala de *chat*, aparecerá uma notificação na forma de mensagem a avisar desse evento. Se todos os utilizadores saírem da sala, o histórico de mensagens será eliminado.

A actualização da janela de mensagens é necessária, para que mensagens recentes possam ser visualizadas pelos utilizadores, assim como a própria lista de utilizadores *online*. Este processo é obtido utilizando uma *framework* do JSF chamada *PrimeFaces* através da ferramenta **p:poll**:

```
<p:poll interval="1" update="onlineUsers janela" />
```

“Interval” define o intervalo de tempo de actualização ou “refresh”, que no nosso caso é de apenas 1 segundo. Em “update” são indicadas as secções abrangidas pela actualização.

Utilizámos ainda uma outra ferramenta desta *framework*: **<p:idleMonitor>**, que, caso o utilizador não tenha interagido com a página, permite terminar a sessão ao fim de um determinado tempo, estabelecido por nós nesta própria ferramenta.

## **Filtros**

Um dos requisitos do trabalho era privar o acesso à calculadora e sala de *chat* a utilizadores anónimos (sem registo).

Para tal, foi implementado um filtro de autenticação. Este filtro pode ser explicado em três pontos:

- Criação de uma classe “*Authentication.java*” que implementa a interface “*Filter*”, e que serve para configurar a acção do Filtro. Neste caso, se uma variável definida na classe “*LoggedUser.java*”, declarada como “*AUTH\_KEY*” e que é instanciada quando o utilizador faz *login*, com o valor do *username* e colocada no *SessionMap* (semelhante a um “*Map*” e que serve para lidar com atributos da sessão *HTTP*), for nula (i.e. o utilizador que faz o “*GET*” à calculadora - um *request*, portanto - não tiver um valor para “*username*” e atribuído à “*AUTH\_KEY*”) então o utilizador é redireccionado para uma página de erro, definida nesta própria classe “*Authentication*”.
- Como foi referido, sempre que é feito um login válido, o *username* é enviado para o *SessionMap* como “*value*” com relação com a “*key*”: “*AUTH\_KEY*”.
- Finalmente no “*web.xml*” é definido o filtro: o seu nome, a classe respectiva, bem como o “*url-pattern*” para a pasta que contém os ficheiros a que um utilizador com login válido terá acesso (onde se inclui o *xhtml* da calculadora).

O segundo filtro que implementamos é opcional para este projecto mas consideramos que faz sentido a sua implementação.

Este segundo filtro tem como função redireccionar um utilizador com sessão iniciada e válida para a página da calculadora e sala de *chat* caso mude temporariamente para uma página externa, isto é, não faria sentido, depois de fazer o login e ter acedido à página da calculadora e eventualmente ter saído para aceder a um *site* externo, ao regressar ter que fazer login novamente.

Este filtro é implementado de forma semelhante ao anterior, pois verifica a mesma variável, diferenciando-se apenas na sua acção: neste caso, ao verificar que o “*value*” da “*key*” *AUTH\_KEY* não é nulo, depois de enviado um “*request*” para obter a página inicial do projecto, o utilizador é redireccionado para a página da calculadora e não para a página “*default*”: a de login.

Este filtro serve também como medida de segurança, pois se o utilizador fizer *login*, nesse instante, mais ninguém conseguirá entrar com o mesmo nome de utilizador e palavra-passe. Esta situação poderia levar à situação de o utilizador fazer *login*, aceder à calculadora e num momento sair para um endereço externo, e ao regressar não conseguiria voltar a aceder à calculadora. Na verdade, este problema está precavido por uma outra via, que é a expiração de sessão definida no *web.xml* da seguinte forma:

```
<session-config>

    <session-timeout>60</session-timeout>

</session-config>
```

## Melhoramentos no código inicial

Foram feitas algumas alterações ao código original da calculadora, como a unificação das calculadoras normal e científica numa só página “*.xhtml*”, a substituição de algumas funções *JavaScript* para *XHTML/JSF* e algumas alterações ao nível do design de modo a permitir a inclusão da sala de *chat* na mesma página.

## Testes unitários (*JUnit* e *Mockito*)

Foram desenvolvidos vários testes unitários com as ferramentas *JUnit*, *Mockito* e *Hamcrest*.

Os principais testes incidiram sobre as funções desenvolvidas por nós (cálculo de factorial e de percentagens).

Além destas funções foram testadas todas as operações da calculadora normal e da calculadora científica.

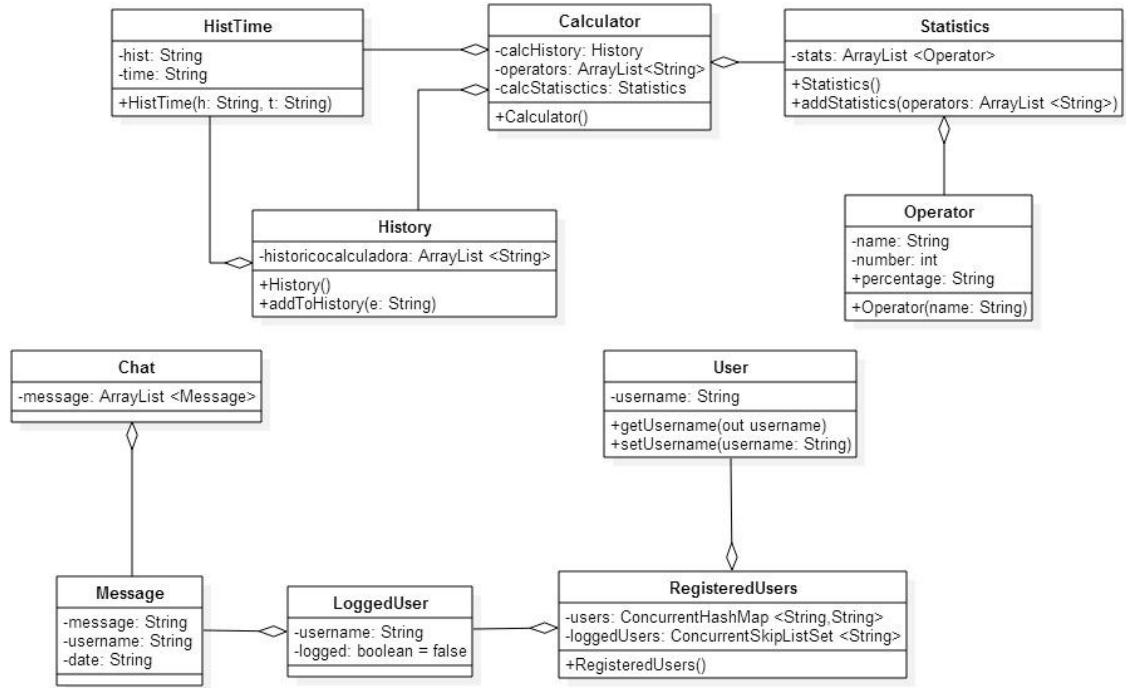
Além de testar a capacidade de cálculo da máquina foram testadas as diferentes mensagens de erro que podem ocorrer no login dos utilizadores (utilizador já com sessão activa, *username* já existente, *password* errada, entre outros).

É importante referir, como nota final, a utilização em alguns testes da *annotation* *@Spy*, em detrimento de *@Mock*, como forma de resolver o problema de necessitar de aceder a métodos reais de alguns *Mocks*. Nestes casos quando inicialmente utilizámos *@Mock*, quando no teste era necessário recorrer a um método da dependência, esse método não era executado (por exemplo, quando era necessário adicionar um elemento a uma lista, apesar dessa lista estar presente, a adição não era executada), e para que tal fosse possível recorremos ao *@Spy*.



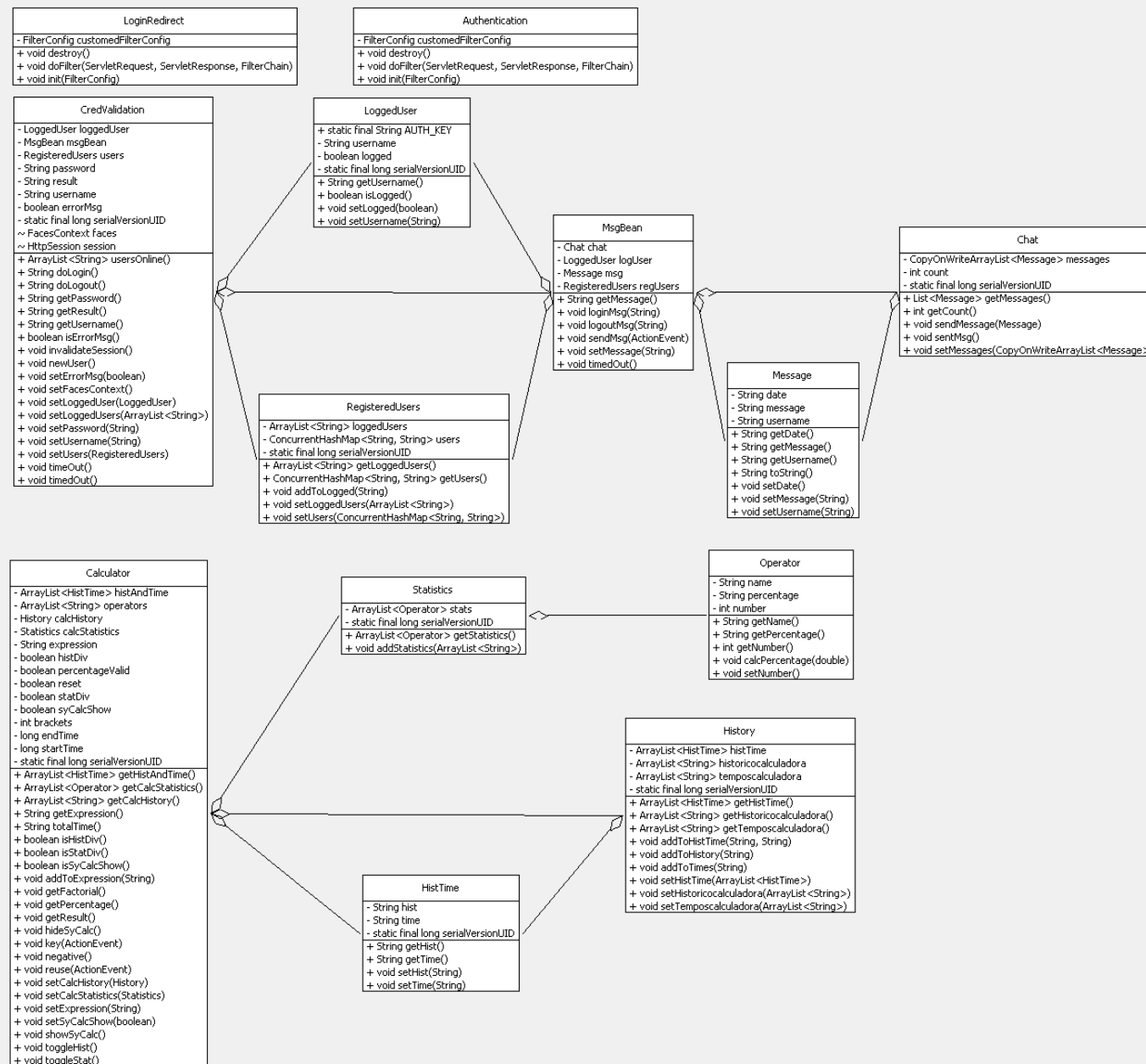
## Pré-desenvolvimento

### Diagrama UML inicial



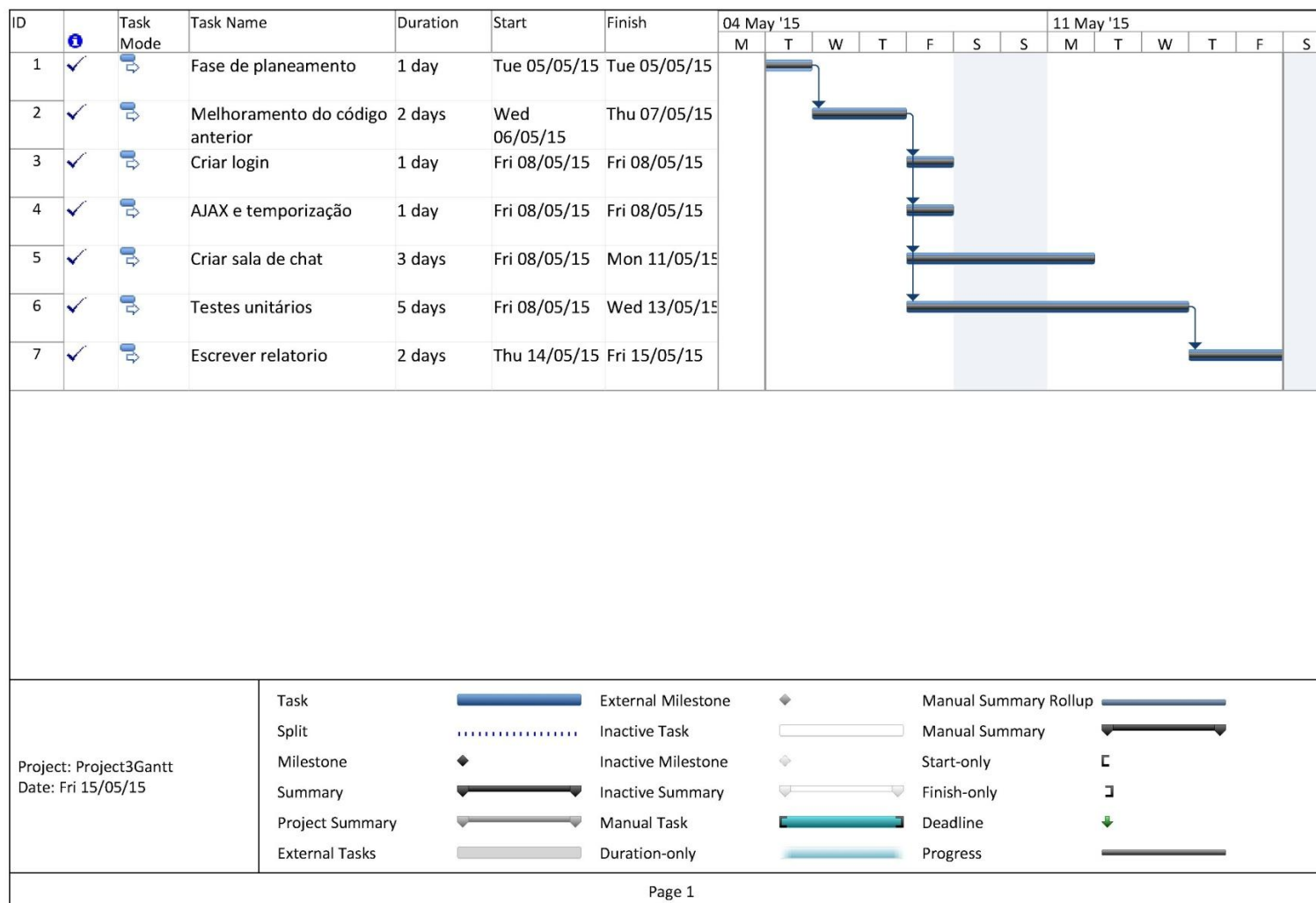
(elaborado no programa StarUML)

## Diagrama UML final



(elaborado na aplicação Code2UML)

## Gráfico Gantt



(elaborado com recurso à aplicação "Microsft Project 2010)

## Testes unitários (*JUnit* e *Mockito*)

Foram desenvolvidos vários testes unitários com as ferramentas *JUnit*, *Mockito* e *Hamcrest*.

Os principais testes incidiram sobre as funções desenvolvidas por nós (cálculo de factorial e de percentagens).

Além destas funções foram testadas todas as operações da calculadora normal e da calculadora científica.

Alem de testar a capacidade de cálculo da máquina foram testadas as diferentes mensagens de erro que podem ocorrer no login dos utilizadores (utilizador já com sessão activa, *username* já existente, *password* errada, entre outros).