

Universidade do Minho - Escola de Engenharia  
Mestrado Integrado em Engenharia Informática  
Processamento de Dados com Streams de Java

---

# Calculadora Universal

---

*Autores :*  
André Vieira (A78322)  
Daniel Maia (A77531)  
Gonçalo Camaz (A76861)

Versão 1.0  
8 de Dezembro de 2018

# 1 Introdução

O objetivo deste trabalho prático é o da implementação de um programa capaz de calcular datas e tempos, tanto locais como com fusos, e proporcionar a funcionalidade de uma agenda de eventos. Este programa, denominado de **Calculadora Universal**, recorrerá predominantemente à *Date-Time API* e será construída de acordo com o modelo MVC.

Adicionalmente, será elaborada uma interface gráfica através da qual se poderá interagir com o programa. Por fim, será escrito um manual de utilizador, em *Markdown* no qual será detalhada a funcionalidade implementada e instruções claras relativas à sua utilização.

No final, será feito um balanço do trabalho efetuado, dos resultados obtidos e do possível trabalho futuro com a Calculadora Universal.

## 2 Estrutura

Esta aplicação foi estruturada tendo em conta o conceito MVC (Model View e Controller). No Model é feita a manipulação dos dados da aplicação, no Controller, a camada de controlo entre a View e o Model, e na View temos a camada de interação com o utilizador.

### 2.1 Model

No Model foram definidos as seguintes classes:

- User.java - Classe relativa do utilizador;  
Variáveis de instância:
  - Nome - Nome do utilizador
  - Email - Email do utilizador
  - Password - Password do utilizador
  - Agenda - Agenda do utilizador
- Users.java - Classe com todos os utilizadores registados  
Variáveis de instância:
  - usersMap - HashMap com todos os utilizadores
- Agenda.java - Agenda relativa a um utilizador  
Variáveis de instância:
  - Slots - Slots das reuniões do utilizador
  - Contactos - Contactos do utilizador
- Slot.java - Slot de tempo  
Variáveis de instância:
  - NomeSlot - Nome do slot
  - Local - Local onde vai decorrer a reunião
  - Data - Data da reunião
  - Inicio - Hora de início
  - Fim - Hora de fim
- Contacto.java - Contactos de um utilizador  
Variáveis de instância:

- Nome - Nome do contacto
- Email - Email do contacto
- Telemóvel - Telemóvel do contacto
- UniversalCalculatorData.java - Classe para guardar os dados da aplicação
- Model.java - Classe principal a que o Controller tem acesso.

Variáveis de instância:

- Users - Todos os utilizadores
- CurrentUser - Utilizador com sessão iniciada

Para guardar os dados da aplicação decidimos utilizar uma API de java para o efeito. Assim sendo, quando é necessário guardar os dados da aplicação (sempre que um utilizador é registado; adicionar/remover contactos/slots de um utilizador;). Quando a aplicação é iniciada são carregados todos os dados armazenados em ficheiro. Estamos então a guardar todos os utilizadores que estão contidos no *Model*.

## 2.2 View

Para maximizar a compreensibilidade do programa, recorreu-se a uma interface gráfica derivada de Java Swing, da qual se extraiu o esqueleto para a implementação da View, uma vez que o Java Swing não obedece à estrutura do modelo MVC. Tendo gerado e adaptado adequadamente os formulários necessários, estes foram integrados no programa de modo a providenciar um meio pelo qual o utilizador poderá interagir com o programa.

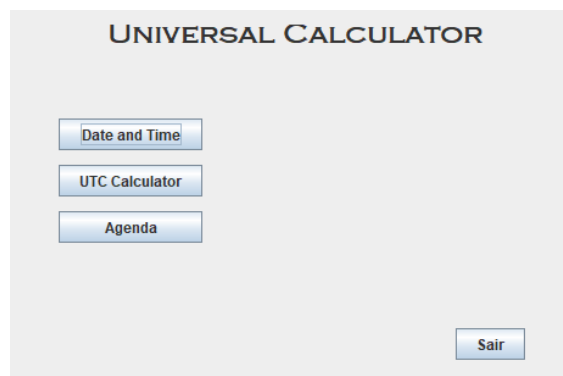


Figura 1: Menu Principal.

## UNIVERSAL CALCULATOR

### DATE & TIME

TIME BETWEEN DATES

Anos

Calcular

DATE & TIME CALCULATOR

+

▼

ANOS

0

MESES

0

SEMANAS

0

DIAS

0

Calcular

Regressar

Figura 2: Menu de cálculo de datas e tempos locais.

## UNIVERSAL CALCULATOR

### TIME ZONE

TIME  (24 HOURS, EXAMPLE 15:00:00)

FROM

-12

▼

☐ USE LOCAL UTC

TO

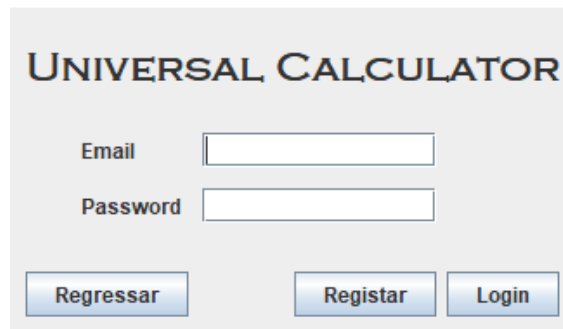
-12

▼

Calcular

Regressar

Figura 3: Menu de cálculo de tempos e datas com fusos.



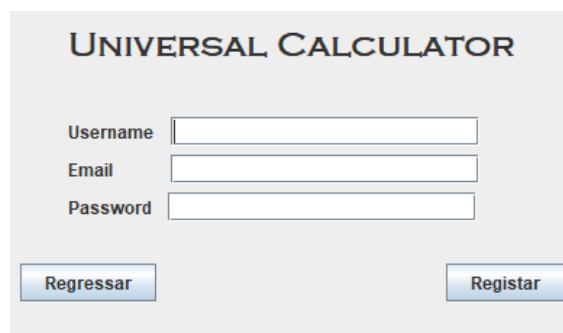
**UNIVERSAL CALCULATOR**

Email

Password

[Regressar](#) [Registar](#) [Login](#)

Figura 4: Menu de Login da agenda.



**UNIVERSAL CALCULATOR**

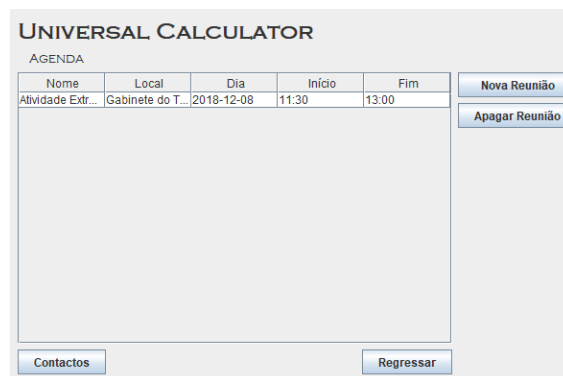
Username

Email

Password

[Regressar](#) [Registar](#)

Figura 5: Menu de registo de utilizador da agenda.



**UNIVERSAL CALCULATOR**

AGENDA

Nome	Local	Dia	Início	Fim
Atividade Extr...	Gabinete do T...	2018-12-08	11:30	13:00

[Nova Reunião](#)

[Apagar Reunião](#)

[Contactos](#) [Regressar](#)

Figura 6: Menu principal da agenda de reuniões.

**REUNIÃO**

Nome

Local

Data  ?

Início  (Exemplo: 12:00:00)

Slot Size  (Minutos)

Number of Slots

**Aceitar** **Cancelar**

Figura 7: Menu de inserção de uma nova reunião na agenda.

**UNIVERSAL CALCULATOR**

AGENDA

Nome	Telefone	Email
Paulo Matos	915647320	paulo.matos@compusev...

**Novo Contacto**  
**Apagar Contacto**

**Reuniões** **Regressar**

Figura 8: Menu dos contactos da agenda.

**CONTACTO**

Nome

Telefone

Email

**Aceitar** **Cancelar**

Figura 9: Menu de inserção de um novo contacto.

## 2.3 Controller

O Controller é responsável por direcionar o fluxo de dados entre o Model e a View. É este que instrui o Model para que execute a lógica do programa e direciona dados relevantes para a View. Serve assim como o elo de ligação entre a interface e os algoritmos do programa. Caso seja necessária a implementação de uma nova View, é necessário simplesmente adicionar um método inicie o respetivo fluxo para a mesma e o conjunto de métodos que invocam a API do Model necessária para o seu funcionamento.

### 3 Date-Time API

Para executar os cálculos de datas e horas essenciais ao bom funcionamento da Calculadora Universal, recorreu-se ao conjunto de classes e interfaces disponibilizada pelo Java 8, conhecida como a **Date-Time API**. Criada com o intuito de substituir as classes `java.util.Date` e `java.util.Calendar`, o *package* `java.time` proporciona uma variedade de métodos que permitem uma maior simplicidade na criação de algoritmos baseadas no cálculo de datas e horas.

Através desta API, é possível facilmente executar aritméticas de soma e subtração graças aos métodos `plus` e `minus`, o que provou ser vital na elaboração da calculadora de tempos locais e de fusos horários. Pode-se também calcular incrementos de tempo através dos métodos `next` e `previous` e queries que nos permitem reconhecer a posição temporal de uma dada data e hora em relação a outra, o que simplificou e agilizou a construção da agenda de reuniões.

### 4 Conclusão

O modelo MVC proporciona uma estrutura altamente modificável e extensível, na qual as funcionalidades podem ser facilmente alteradas ou adicionadas.

Por outro lado, a Date-Time API permite a criação de métodos mais consistentes e legíveis, graças à sua rica biblioteca de métodos.

Concluindo, a implementação da Calculadora Universal provou-se produtiva na aquisição de conhecimentos relativos ao funcionamento da estruturação e utilização do modelo MVC e da API proporcionada pelo Java 8. Desta forma, obteve-se uma base sólida que permite proceder de forma consciente para possíveis próximas fases de extensão ao programa existente, por exemplo, na modularização do componente Controller, separando-o em subcomponentes especializados.