Universal Date and Time Calculator (UDTC)

Manual de utilização

Afonso Silva

Octávio Maia

# Introdução

Este trabalho prático tem como base a criação de uma Calculadora de Tempos Universal. Como indicado pelo enunciado foram implementadas diversas funcionalidades de modo a contemplar a matéria lecionada durante o semestre.

Esta aplicação contém 6 funcionalidades distintas:

* Age calculator
* Calendar
* Chronometer
* Date calculator
* Notebook
* Travels
* World Clock

Sendo estas funcionalidades exploradas mais adiante.

De modo a tirar total partido das funcionalidades que o Java 8 incorpora, foi dado um enorme ênfase à utilização de Streams, Optional, expressões Lambda, etc. Mais à frente no manual irá ser explorado em detalhe as decisões tomadas ao longo do projeto.

# Utilização

#### Obtenção do programa

O executável deste programa pode ser descarregado a partir desta ligação:

<https://github.com/ajcerejeira/udtc/raw/master/udtc.jar>

Também pode descarregar o todo o conteudo do projeto através do git:

git clone https://github.com/ajcerejeira/udtc  
cd udtc/

#### Execução e compilação

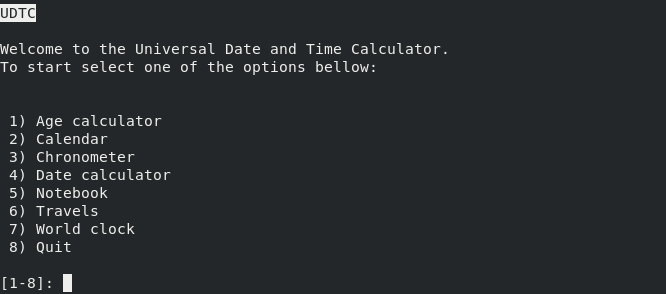
Na pasta raíz do projecto encontra-se o ficheiro udtc.jar pronto a ser executado. Deste modo, para iniciar o programa basta correr o seguinte comando na raíz do projeto:

$ java -jar udtc.jar

Pode também compilar o programa se assim o desejar. O seu código-fonte encontra-se na pasta src e pode ser compilado recorrendo à ferramenta javac ou então, através de um ambiente de desenvolvimento integrado que suporte JAVA.

Após o programa se encontrar compilado, e encontrando-se na directoria onde se encontram os resultados da compilação, basta correr o seguinte comando para iniciar a aplicação:

$ java Main



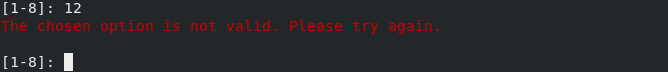
Menu inicial do programa

#### Navegação pela interface

Tal como sugerido no enunciado deste trabalho prático, o modo de utilização da aplicação passa pela navegação de menus puramente textuais. Cada um destes menus apresenta um título (e alguns também um subtítulo) bastante visíveis, indicando o modo atual em utilização da calculadora. De seguida é apresentada uma lista numerada com todas as opções disponíveis no modo atual. O utilizador deve então introduzir a opção pretendida no *prompt* e pressionar a tecla <ENTER>, sendo conduzido para o menu seguinte.

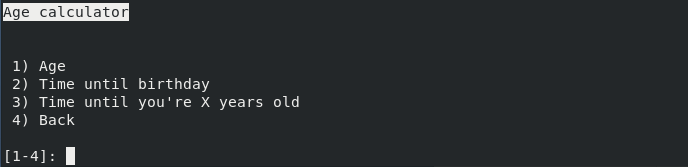
#### Prompt

O *prompt* (pequeno texto que aparece antes do local de introdução do utilizador) indica para cada menu qual o intervalo de opções disponíveis. Regra geral a última opção é a de saída ou retorno ao menu anterior. Quando o utilizador introduz um número inválido, o programa pede para repetir a sua escolha novamente.



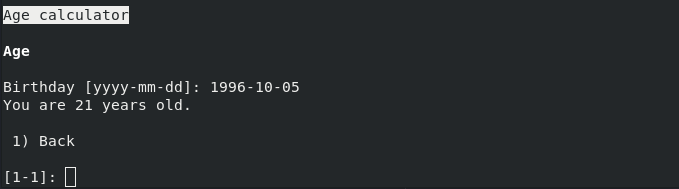
Aviso de introdução de opção incorrecta

## Calculadora de idades

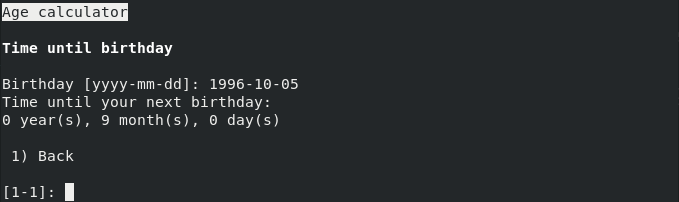


Menu inicial da calculadora de idades

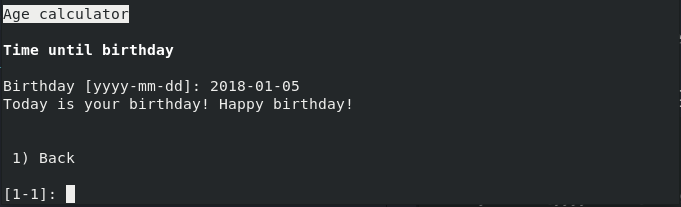
A calculadora de idades permite efetuar algumas operações relacionadas com datas de aniversário, permitindo calcular uma determinada idade, quanto tempo até ao próximo aniversário e quanto tempo demorará a atingir uma determinada idade.



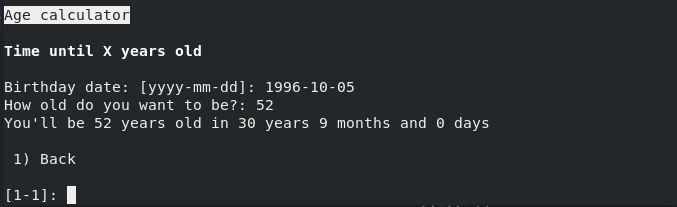
Menu do cálculo da idade



Menu do cálculo do tempo até ao próximo aniversário



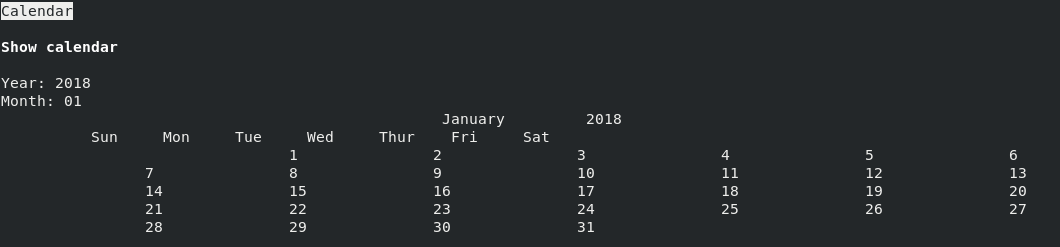
Quando a data de aniversário corresponde ao dia de hoje, o utilizador recebe uma mensagem de parabéns



Menu do cálculo do tempo até se atingir uma determinada idade

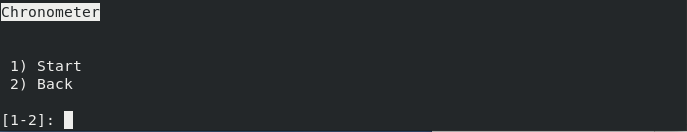
## Calendário

A função de calendário permite simplesmente imprimir em formato tabular um determinado mês de um determinado ano.



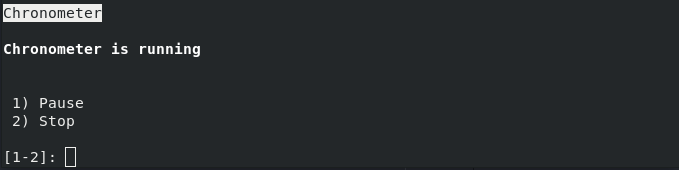
Menu inicial do calendário

## Cronómetero

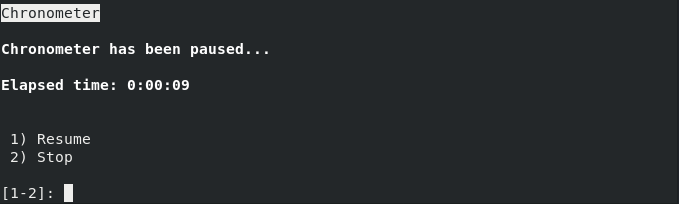


Menu inicial do cronómetro

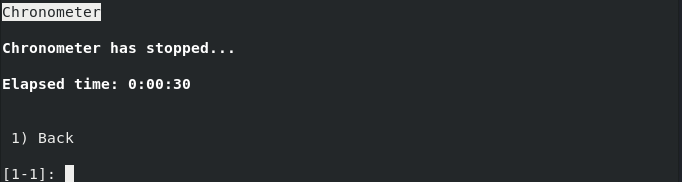
O cronómetro permite contar o tempo desde a sua execução, permitindo efetuar pausas e retomar a contagem. Os resultados da contagem são mostrados sempre que o cronómetro é pausado ou parado. Infelizmente, por se tratar de uma interface textual é bastante complicado imprimir o valor da contagem em tempo real, pois tal envolveria mecanismos de limpeza e escrita de ecrã que não são bem suportados em muitas plafatormas. Em todo o caso, a interface mantém-se simples e de fácil utilização.



Menu de início da execução do cronómetro

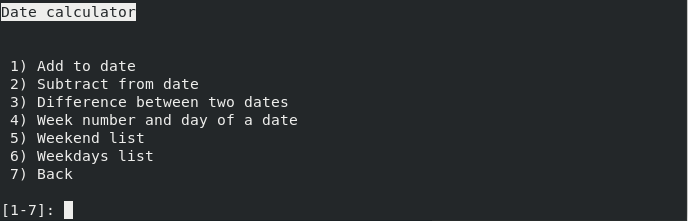


Menu de pausa do cronómetro



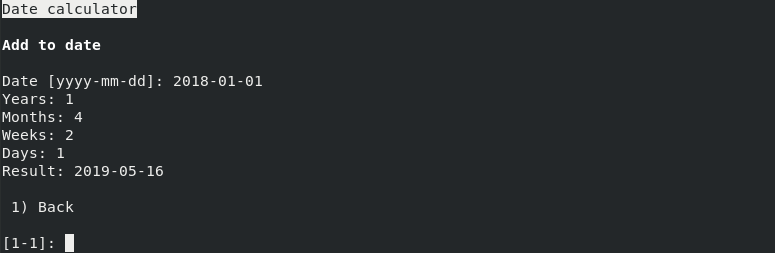
Menu de paragem do cronómetro

## Calculadora de datas

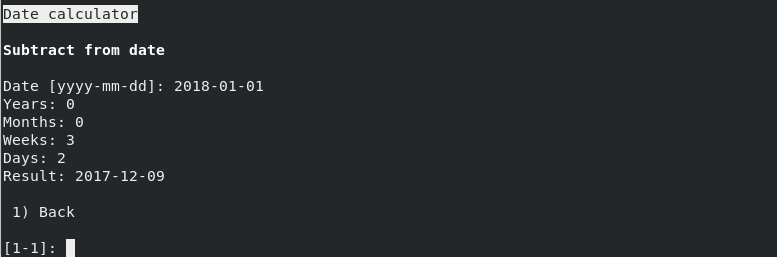


Menu inicial da calculadora de datas

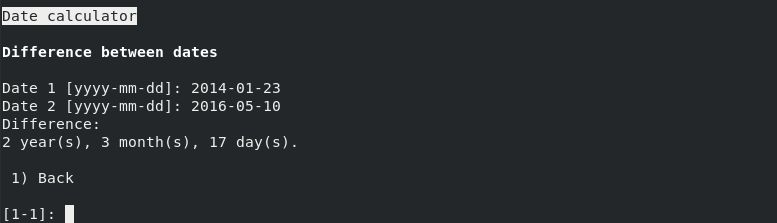
A calculadora de datas é um dos modos mais extensos e mais completos deste programa. Ela permite efetuar cálculos com datas, entre os quais a adição e subtração de datas, diferença entre duas datas. Tem também algumas funções de calendário, já que permite listar os dias e fins de semana entre duas datas, bem como descobrir o número de uma semana e o dia da semana de qualquer data.



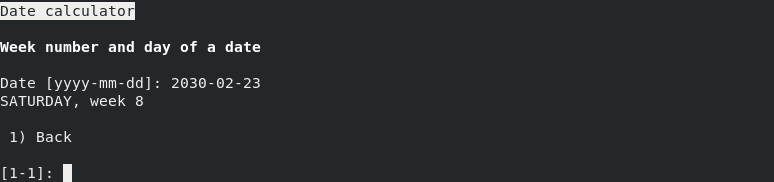
Menu de adição temporal a uma data



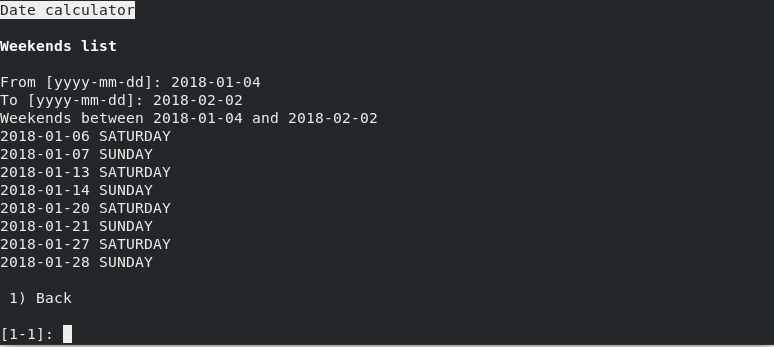
Menu de subtração temporal de uma data



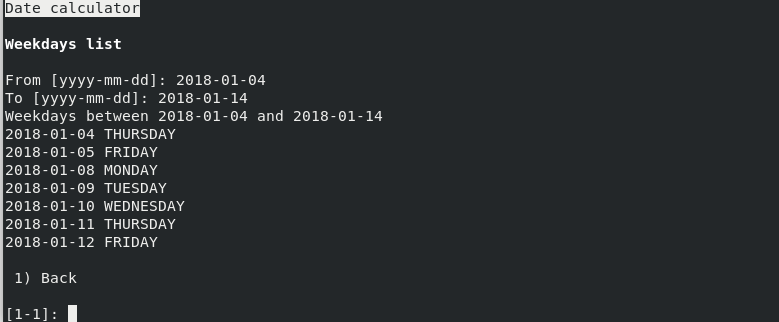
Menu de diferença entre duas datas



Dia da semana e número da semana de uma data

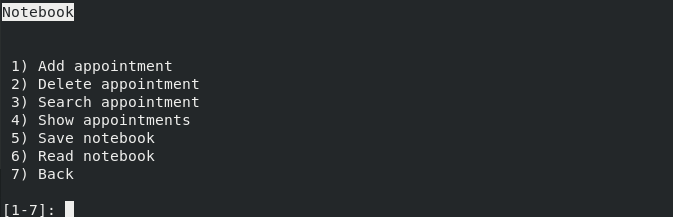


Lista de fins de semana entre duas datas



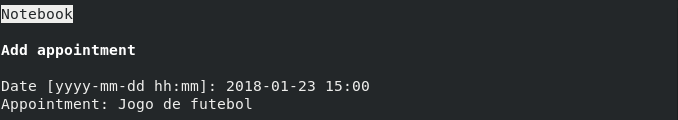
Lista de dias da semana entre duas datas

## Agenda

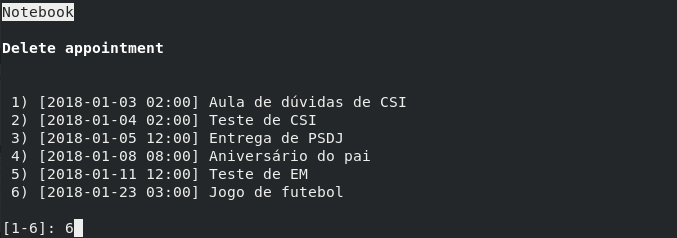


Menu inicial da agenda

A agenda consiste na agregação de diversos eventos, que são simplesmente compostos por uma data de ocorrência e uma breva descrição. Para inserir e remover eventos utilizam-se as opções 1) Add appointment e 2) Delete appointment respetivamete.

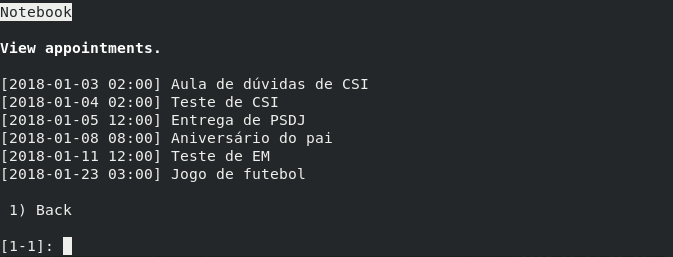


Menu de inserção de eventos

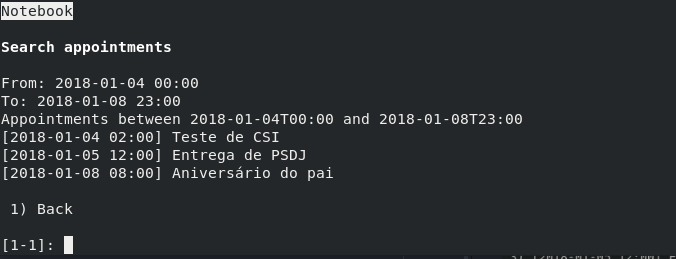


Menu de remoção de eventos

Obviamente que é possível visualizar todos os eventos da agenda, como também filtrá-los por data e hora. Nestas listagens os eventos aparecem sempre ordenados por data



Visualização de todos os eventos



Filtragem de eventos consoante a data

#### Serialização

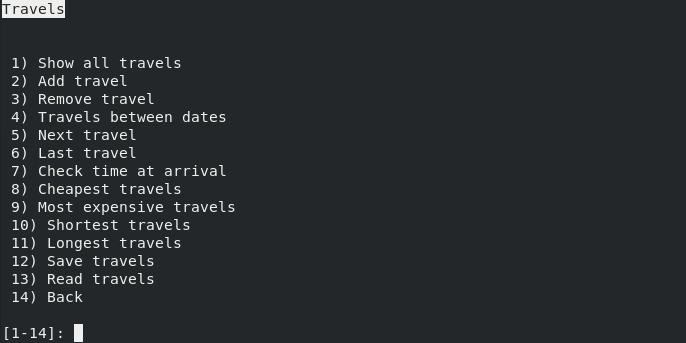
É possível guardar e carregar o estado de uma agenda para um formato textual. Para tal utilizam-se os comandos 4) Save notebook para guardar num ficheiro os eventos que se encontram na agenda, e 5) Read notebook para ler de um ficheiro esses mesmos eventos.

Abaixo se ilustra um exemplo de um ficheiro de dados gerado pela agenda:

[2018-01-03 02:00] Aula de dúvidas de CSI  
[2018-01-04 02:00] Teste de CSI  
[2018-01-05 12:00] Entrega de PSDJ  
[2018-01-08 08:00] Aniversário do pai  
[2018-01-11 12:00] Teste de EM

Ao extrair a agenda para um formato textual torna-se então possível ao utilizador editar a própria agenda mesmo fora da aplicação. Para tal basta utilizar um editor de texto para inserir ou remover novos eventos neste ficheiro, e posteriormente relê-lo com a aplicação. Esta funcionalidade também permite a fácil partilha de agendas entre vários utilizadores.

## Viagens

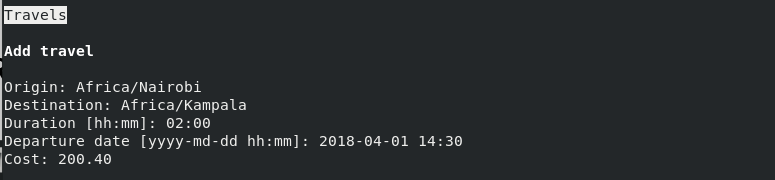


Menu inicial das viagens

O modo de gestão de viagens inclui várias ferramentas de conveniência para o cálculo de estimativas de duração de viagens, cálculo de tempos de chegada, etc...



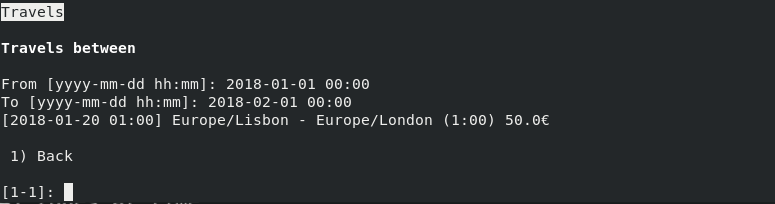
Lista de viagens



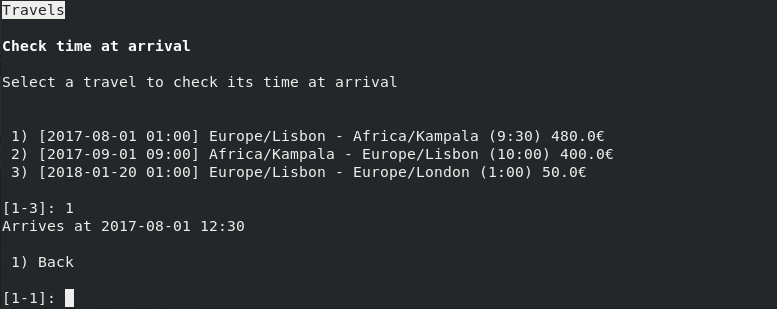
Inserção de uma viagem



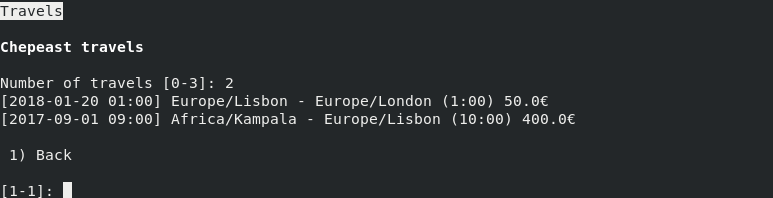
Remoção de uma viagem



Lista de viagens entre duas datas



Tempo de chegada de uma viagem



Viagens mais baratas



Viagens mais longas

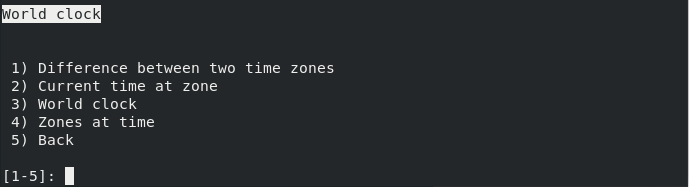
#### Serialização

Tal como a agenda, o modo de gestão de viagens também permite guardar e ler a lista das viagens para um formato textual.

Exemplo de ficheiro de viagens:

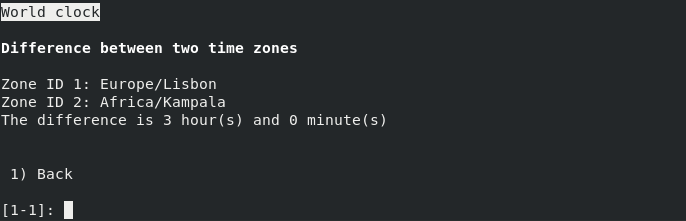
[2017-08-01 01:00] Europe/Lisbon - Africa/Kampala (9:30) 480.0€  
[2017-09-01 09:00] Africa/Kampala - Europe/Lisbon (10:00) 400.0€  
[2018-01-20 13:00] Europe/Lisbon - Europe/London (01:00) 50.0€

## Relógio universal

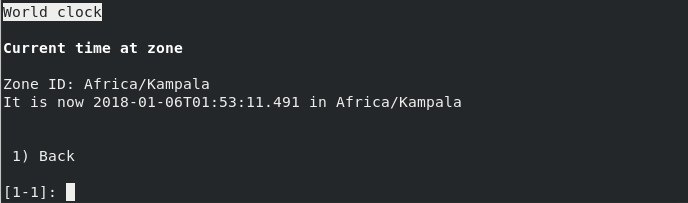


Menu inicial do relógio universal

O relógio universal contém funcionalidades úteis para manipulação de diversos fusos horários. Permite o cálculo das diferenças de tempo entre duas zonas do globo, verficar qual a hora atual numa determinada zona, mostrar que zonas se encontram atualmente numa determinada hora e imprimir a data e hora atuais de todas as regiões do mundo.



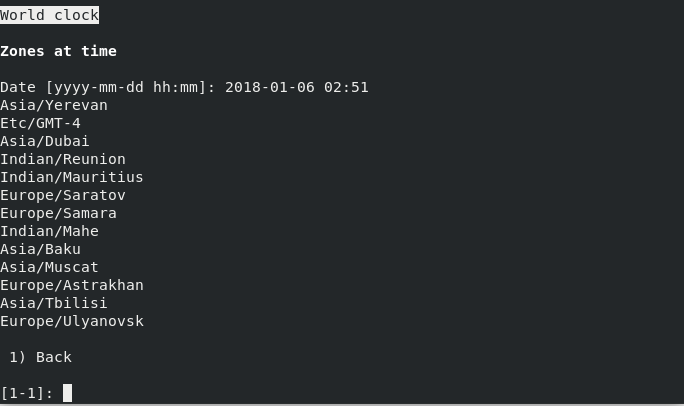
Diferença horária entre duas zonas do globo do mundo



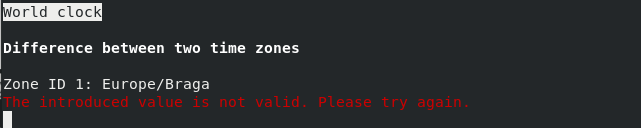
Data e hora em todas as regiões do mundo



Data e hora numa determinada zona do mundo



Lista de zonas que se encontram atualmente numa determinada date e hora



Validação da zona introduzida. Só as TimeZones incluídas na biblioteca do JAVA que são válidas

# Implementação

## Arquitectura

Para garantir uma separação de componentes baseada na sua função, recorreu-se ao padrão de arquitetura de *software* [MVC](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller).

Deste modo o projeto dividiu-se nos três *packages* que compõem a arquitetura MVC:

* Model - núcleo do projeto, definia interfaces e suas implementações para os diversos modos da aplicação. Neste projeto insistiu-se numa primeira fase na definição de Interfaces de JAVA que ditavam os métodos que determinado modo deveria seguir. Aliás, garantiu-se que os outros *packages* só acediam a métodos definidos nas interfaces. Estando as interfaces já definidas começou-se a codificação das classes que as implementavam.
* View - que para cada interface definida no *package* Model codificava a interação textual com o utilizador.
* Controller - que instanciava objetos que implementavam as interfaces definidas no *package* Model e os passava para a View.

Com esta arquitectura o processo de desenvolvimento do projeto decorreu com sucesso. Reforça-se ainda a ideia bem sucedida de definir primeiro as interfaces e só depois as implementar, o que facilitou a divisão das tarefas: era assim possível implementar as *view* para essa interface ao mesmo tempo que a sua implementação.

Foi ainda criado um outro *package* denominado Utils que continha código utilizado por todos os outros packages de utilidade geral. Este era composto pelas seguintes classes:

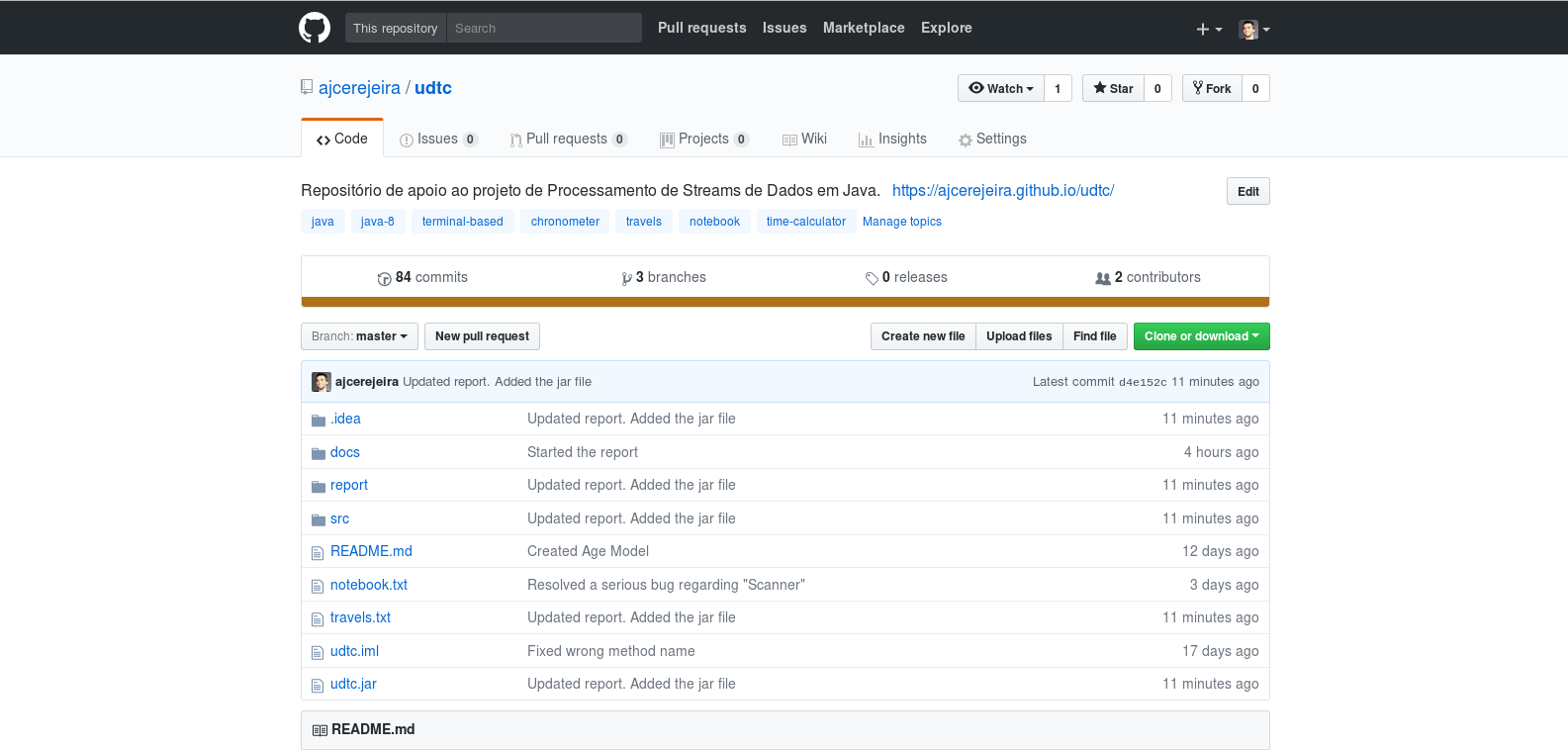
* UI - classe composta por métodos estáticos que facilitam a renderização da interface textual. Inclui o desenho de títulos, subtítulos, menus e introdução textual do utilizador, e impressão de listas de objetos. Com recurso às novas funcionalidades do JAVA 8 (lambdas, referências a métodos, Optional) é muito mais simples e eficaz a criação de novas interfaces.
* Option - classe auxiliar utilizada simplesmente para o desenho de menus. Estes são compostos por várias Option, sendo que cada uma contém um determinado título e a ação a executar quando esta opção é escolhida. Esta ação não é mais do que um objecto que implemente a interface Runnable.
* Parsers - classe auxiliar que inclui *parsers* de datas, tempos, durações, zonas e números. Cada parser apresenta a seguinte assinatura: static Optional<T> parse(String input), onde T é o tipo de dados que se pretende obter com o resultado do *parsing*. Repare-se na utilização da nova classe do JAVA 8, Optional para indicar se o *parsing* foi ou não bem sucedido.

## Controlo de versões

Tal como ditam as regras do bom desenvolvimento de *software*, utilizou-se a ferramenta git como método de controlo de versões durante o desenvolvimento do programa. Esta ferramenta foi bastante útil para a coordenação do projeto entre os dois membros do grupo: cada um utilizou um branch próprio onde ia desenvolvendo as suas soluções para os problemas, evitando assim conflitos no código. No final os branches foram agregados resultando num projeto sustentável e modular.

Todo o código do projeto (este relatório inclusive) foi hospedado no GitHub no seguinte repositório:

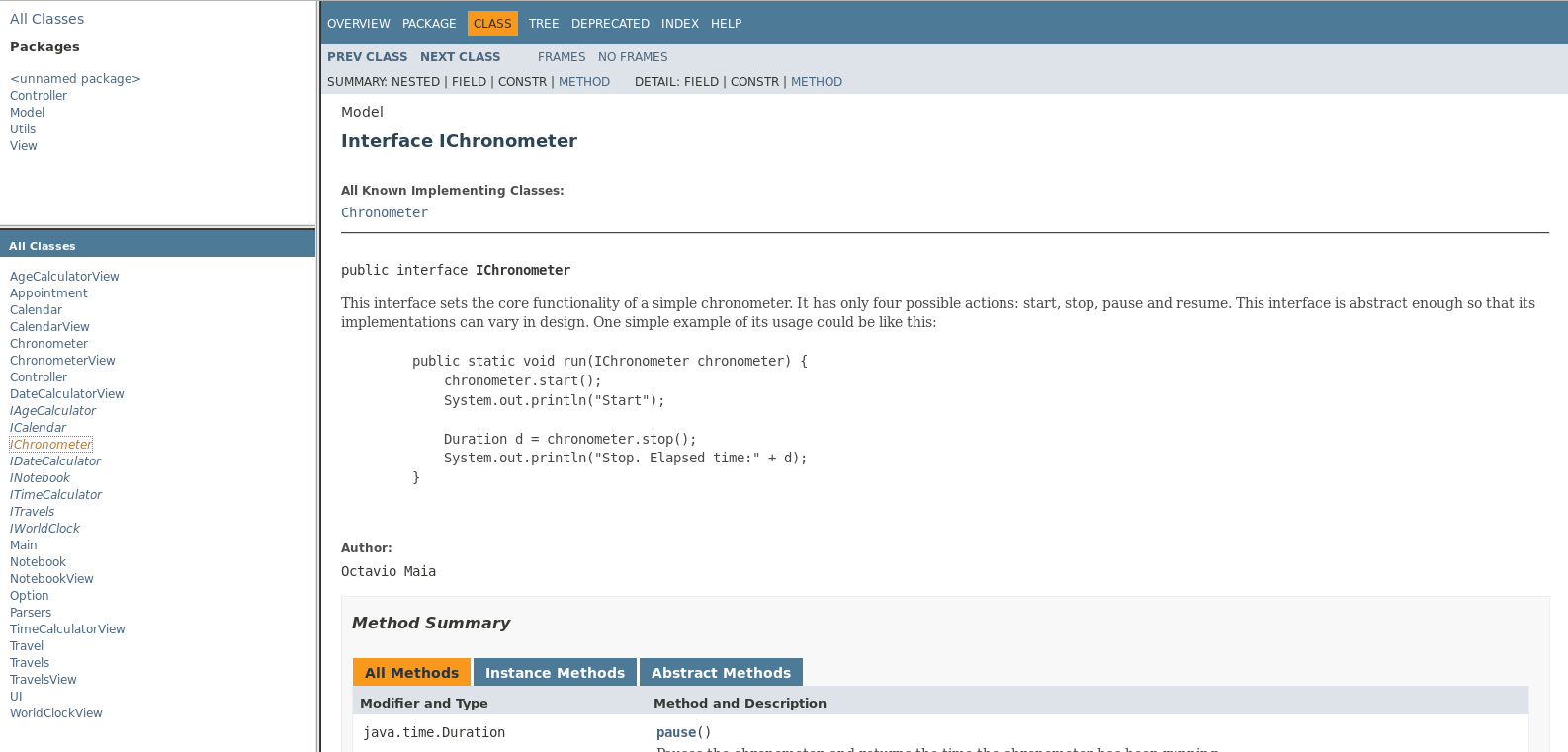
<https://github.com/ajcerejeira/udtc>



Repositório do projeto no GitHub

## Documentação

Toda a documentação do código fonte foi efetuada recorrendo à ferramenta JAVADOC e hospedada como um site do GitHub pages. Esta pode ser consultada diretamente em <https://ajcerejeira.github.io/udtc/>. Lá encontram-se documentadas as classes criadas propositadamente para este programa, encontrando-se informações sobre o seu propósito, implementação e modo de utilização.



Exemplo de página da documentação do projeto

# Conclusão

Termina-se assim o manual relativo à Calculadora de Tempos Universal. Embora o trabalho esteja bastante completo, consideramos que existem pontos que poderiam ser melhorados. Entre eles estariam a implementação dos seguintes métodos:

* Pagamentos - Módulo responsável por pagamentos periódicos, sendo exemplos do mesmo a anuidade de um seguro automóvel, propinas de alunos, etc. A implementação do mesmo seria relativamente fácil, sendo apenas necessário perguntar ao utilizador a periocidade do pagamento, bem como a quantia do mesmo e a data da 1º prestação. Seria também útil exportar esta informação para um ficheiro de texto para futura leitura.
* Cronómetro - O módulo Cronómetro apenas permite iniciar, pausar/continuar e paragem do mesmo. Seria interessante a implementação de um sistema responsável por guardar o ponto atual do cronómetro para futuro carregamento no sistema. Além disso poderiamos criar um sistema responsável por guardar o *runtime* do cronómetro atual quando fosse pressionada uma tecla, de modo a guardar vários *timestamps*.
* Calendário - O módulo Calendário apenas permite mostrar o calendário de um dado mês e ano em *pretty-printing*, sendo que uma possivel adição a nível de funcionalidades seria a sincronização com o módulo responsável pelos Appointments e Travels. Assim sendo, no *pretty-printing* do calendário seria adicionada informação relativa aquele mês, como por exemplo: viagens/appointments que ocorrem naquele mês, etc.

A implementação das funcionalidades em cima descritas seria trivial, visto que o projeto foi desenvolvido de forma a ter uma fácil expansão de modos de funcionamento.

Relativamente ao Java 8, podemos afirmar que se tratou de uma agradável surpresa, sendo que a programação baseada no paradigma funcional simplificou a escrita de código. Além disso, a utilização de *Streams*, além de oferecer um ganho de performance, simplifica muito a escrita de código comparativamente com versões anteriores do Java.