

Universal Date and Time Calculator (UDTC)

Manual de utilização

Afonso Silva a70387

Octávio Maia a71369

Conteúdo

1	Introdução	3
2	Utilização	4
	Calculadora de idades	5
	Calendário	6
	Cronómetro	6
	Calculadora de datas	9
	Agenda	9
	Viagens	13
	Relógio universal	16
3	Implementação	20
	Arquitectura	20
	Controlo de versões	21
	Documentação	21
4	Conclusão	23

1 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.

2 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 conteúdo do projeto através do `git`:

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

Execução e compilação

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

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.

```
UDTC

Welcome to the Universal Date and Time Calculator.
To start select one of the options bellow:

1) Age calculator
2) Calendar
3) Chronometer
4) Date calculator
5) Notebook
6) Travels
7) World clock
8) Quit

[1-8]:
```

Figura 2.1: Menu inicial do programa

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.

```
[1-8]: 12
The chosen option is not valid. Please try again.

[1-8]:
```

Figura 2.2: Aviso de introdução de opção incorrecta

Calculadora de idades

```
Age calculator

1) Age
2) Time until birthday
3) Time until you're X years old
4) Back

[1-4]:
```

Figura 2.3: 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.

```
Age calculator

Age

Birthday [yyyy-mm-dd]: 1996-10-05
You are 21 years old.

1) Back

[1-1]: 
```

Figura 2.4: Menu do cálculo da idade

```
Age calculator

Time until birthday

Birthday [yyyy-mm-dd]: 1996-10-05
Time until your next birthday:
0 year(s), 9 month(s), 0 day(s)

1) Back

[1-1]: 
```

Figura 2.5: Menu do cálculo do tempo até ao próximo aniversário

Calendário

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

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 plataformas. Em todo o caso, a interface mantém-se simples e de fácil utilização.

```
Age calculator

Time until birthday

Birthday [yyyy-mm-dd]: 2018-01-05
Today is your birthday! Happy birthday!

1) Back

[1-1]: █
```

Figura 2.6: Quando a data de aniversário corresponde ao dia de hoje, o utilizador recebe uma mensagem de parabéns

```
Age calculator

Time until X years old

Birthday date: [yyyy-mm-dd]: 1996-10-05
How old do you want to be?: 52
You'll be 52 years old in 30 years 9 months and 0 days

1) Back

[1-1]: █
```

Figura 2.7: Menu do cálculo do tempo até se atingir uma determinada idade

```
Calendar

Show calendar

Year: 2018
Month: 01

          Sun      Mon      Tue      Wed      Thur      January
                                Fri      Sat      2018
          1          2          3          4          5          6
          7          8          9          10         11         12         13
          14         15         16         17         18         19         20
          21         22         23         24         25         26         27
          28         29         30         31
```

Figura 2.8: Menu inicial do calendário

```
Chronometer

1) Start
2) Back

[1-2]: █
```

Figura 2.9: Menu inicial do cronómetro

```
Chronometer
Chronometer is running

1) Pause
2) Stop

[1-2]: 
```

Figura 2.10: Menu de início da execução do cronómetro

```
Chronometer
Chronometer has been paused...

Elapsed time: 0:00:09

1) Resume
2) Stop

[1-2]: 
```

Figura 2.11: Menu de pausa do cronómetro

```
Chronometer
Chronometer has stopped...

Elapsed time: 0:00:30

1) Back

[1-1]: 
```

Figura 2.12: Menu de paragem do cronómetro

Calculadora de datas

```
Date calculator

1) Add to date
2) Subtract from date
3) Difference between two dates
4) Week number and day of a date
5) Weekend list
6) Weekdays list
7) Back

[1-7]:
```

Figura 2.13: 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.

```
Date calculator

Add to date

Date [yyyy-mm-dd]: 2018-01-01
Years: 1
Months: 4
Weeks: 2
Days: 1
Result: 2019-05-16

1) Back

[1-1]:
```

Figura 2.14: Menu de adição temporal a uma data

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** respetivamente.

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

```
Date calculator

Subtract from date

Date [yyyy-mm-dd]: 2018-01-01
Years: 0
Months: 0
Weeks: 3
Days: 2
Result: 2017-12-09

1) Back

[1-1]:
```

Figura 2.15: Menu de subtração temporal de uma data

```
Date calculator

Difference between dates

Date 1 [yyyy-mm-dd]: 2014-01-23
Date 2 [yyyy-mm-dd]: 2016-05-10
Difference:
2 year(s), 3 month(s), 17 day(s).

1) Back

[1-1]:
```

Figura 2.16: Menu de diferença entre duas datas

```
Date calculator

Week number and day of a date

Date [yyyy-mm-dd]: 2030-02-23
SATURDAY, week 8

1) Back

[1-1]:
```

Figura 2.17: Dia da semana e número da semana de uma data

```

Date calculator

Weekends list

From [yyyy-mm-dd]: 2018-01-04
To [yyyy-mm-dd]: 2018-02-02
Weekends between 2018-01-04 and 2018-02-02
2018-01-06 SATURDAY
2018-01-07 SUNDAY
2018-01-13 SATURDAY
2018-01-14 SUNDAY
2018-01-20 SATURDAY
2018-01-21 SUNDAY
2018-01-27 SATURDAY
2018-01-28 SUNDAY

1) Back

[1-1]: 

```

Figura 2.18: Lista de fins de semana entre duas datas

```

Date calculator

Weekdays list

From [yyyy-mm-dd]: 2018-01-04
To [yyyy-mm-dd]: 2018-01-14
Weekdays between 2018-01-04 and 2018-01-14
2018-01-04 THURSDAY
2018-01-05 FRIDAY
2018-01-08 MONDAY
2018-01-09 TUESDAY
2018-01-10 WEDNESDAY
2018-01-11 THURSDAY
2018-01-12 FRIDAY

1) Back

[1-1]: 

```

Figura 2.19: Lista de dias da semana entre duas datas

```

Notebook

1) Add appointment
2) Delete appointment
3) Search appointment
4) Show appointments
5) Save notebook
6) Read notebook
7) Back

[1-7]: 

```

Figura 2.20: Menu inicial da agenda

```
Notebook

Add appointment

Date [yyyy-mm-dd hh:mm]: 2018-01-23 15:00
Appointment: Jogo de futebol
```

Figura 2.21: Menu de inserção de eventos

```
Notebook

Delete appointment

1) [2018-01-03 02:00] Aula de dúvidas de CSI
2) [2018-01-04 02:00] Teste de CSI
3) [2018-01-05 12:00] Entrega de PSDJ
4) [2018-01-08 08:00] Aniversário do pai
5) [2018-01-11 12:00] Teste de EM
6) [2018-01-23 03:00] Jogo de futebol

[1-6]: 6
```

Figura 2.22: Menu de remoção de eventos

```
Notebook

View appointments.

[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
[2018-01-23 03:00] Jogo de futebol

1) Back

[1-1]:
```

Figura 2.23: Visualização de todos os eventos

```
Notebook
Search appointments

From: 2018-01-04 00:00
To: 2018-01-08 23:00
Appointments between 2018-01-04T00:00 and 2018-01-08T23:00
[2018-01-04 02:00] Teste de CSI
[2018-01-05 12:00] Entrega de PSDJ
[2018-01-08 08:00] Aniversário do pai

1) Back

[1-1]: █
```

Figura 2.24: 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

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. . .

```
Travels

1) Show all travels
2) Add travel
3) Remove travel
4) Travels between dates
5) Next travel
6) Last travel
7) Check time at arrival
8) Cheapest travels
9) Most expensive travels
10) Shortest travels
11) Longest travels
12) Save travels
13) Read travels
14) Back

[1-14]:
```

Figura 2.25: Menu inicial das viagens

```
Travels

Show travels

[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 01:00] Europe/Lisbon - Europe/London (1:00) 50.0€

1) Back

[1-1]:
```

Figura 2.26: Lista de viagens

```
Travels

Add travel

Origin: Africa/Nairobi
Destination: Africa/Kampala
Duration [hh:mm]: 02:00
Departure date [yyyy-md-dd hh:mm]: 2018-04-01 14:30
Cost: 200.40
```

Figura 2.27: Inserção de uma viagem

```

Travels

Remove travel

1) [2017-08-01 01:00] Europe/Lisbon - Africa/Kampala (9:30) 480.0€
2) [2017-09-01 09:00] Africa/Kampala - Europe/Lisbon (10:00) 400.0€
3) [2018-01-20 01:00] Europe/Lisbon - Europe/London (1:00) 50.0€
4) [2018-04-01 02:30] Africa/Nairobi - Africa/Kampala (2:00) 200.4€

[1-4]: 4

```

Figura 2.28: Remoção de uma viagem

```

Travels

Travels between

From [yyyy-mm-dd hh:mm]: 2018-01-01 00:00
To [yyyy-mm-dd hh:mm]: 2018-02-01 00:00
[2018-01-20 01:00] Europe/Lisbon - Europe/London (1:00) 50.0€

1) Back

[1-1]:

```

Figura 2.29: Lista de viagens entre duas datas

```

Travels

Check time at arrival

Select a travel to check its time at arrival

1) [2017-08-01 01:00] Europe/Lisbon - Africa/Kampala (9:30) 480.0€
2) [2017-09-01 09:00] Africa/Kampala - Europe/Lisbon (10:00) 400.0€
3) [2018-01-20 01:00] Europe/Lisbon - Europe/London (1:00) 50.0€

[1-3]: 1
Arrives at 2017-08-01 12:30

1) Back

[1-1]:

```

Figura 2.30: Tempo de chegada de uma viagem

```
Travels

Cheapest travels

Number of travels [0-3]: 2
[2018-01-20 01:00] Europe/Lisbon - Europe/London (1:00) 50.0€
[2017-09-01 09:00] Africa/Kampala - Europe/Lisbon (10:00) 400.0€

1) Back

[1-1]:
```

Figura 2.31: Viagens mais baratas

```
Travels

Longest travels

Number of travels [0-3]: 1
[2017-09-01 09:00] Africa/Kampala - Europe/Lisbon (10:00) 400.0€

1) Back

[1-1]:
```

Figura 2.32: 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

O relógio universal contém funcionalidades úteis para manipulação de diversos fusos horários. Permite o cálculo das diferenças de tempo entre duas zonas do globo, verificar 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.


```
World clock

1) Difference between two time zones
2) Current time at zone
3) World clock
4) Zones at time
5) Back

[1-5]:
```

Figura 2.33: Menu inicial do relógio universal

```
World clock

Difference between two time zones

Zone ID 1: Europe/Lisbon
Zone ID 2: Africa/Kampala
The difference is 3 hour(s) and 0 minute(s)

1) Back

[1-1]:
```

Figura 2.34: Diferença horária entre duas zonas do globo do mundo

```
World clock

Current time at zone

Zone ID: Africa/Kampala
It is now 2018-01-06T01:53:11.491 in Africa/Kampala

1) Back

[1-1]:
```

Figura 2.35: Data e hora em todas as regiões do mundo

```
World clock
World clock
2018-01-05 10:00 [Etc/GMT+12]
2018-01-05 11:00 [Pacific/Pago Pago, Pacific/Samoa, Pacific/Niue, US/Samoa, Etc/GMT+11, Pacific/Midway]
2018-01-05 12:00 [Pacific/Honolulu, Pacific/Parotonga, Pacific/Tahiti, Pacific/Johnston, America/Atka, US/Hawaii, SystemV/HST10, America/Adak, US/Aleutian, Etc/GMT+10]
2018-01-05 01:00 [Etc/GMT+9, US/Alaska, America/Juneau, America/Motlatkila, Pacific/Gambier, America/Yakutat, America/Sitka, Pacific/Marquesas, SystemV/YST9, America/Anchorage,]
2018-01-05 02:00 [Etc/GMT+8, Canada/Yukon, US/Pacific-New, Canada/Pacific, PST8PDT, Pacific/Pitcairn, America/Dawson, Mexico/BajaNorte, America/Tijuana, SystemV/PST8, America/S]
2018-01-05 03:00 [Etc/GMT+7, US/Arizona, Mexico/BajaSur, America/Dawson_Creek, America/Denver, America/Yellowknife, America/Inuvik, America/Mazatlan, SystemV/MST7, America/Bois]
2018-01-05 04:00 [America/El Salvador, America/Guatemala, America/Belize, America/Managua, America/Chicago, America/Tegucigalpa, Etc/GMT+6, America/Regina, Mexico/General, Amer]
2018-01-05 05:00 [America/Panama, America/Indiana/Petersburg, America/Europe, Cuba, Etc/GMT+5, Pacific/Easter, America/Fort Wayne, America/Havana, America/Porto Acre, US/Mich]
2018-01-05 06:00 [America/Marijot, Canada/Atlantic, America/Grand Turk, Etc/GMT+4, America/Manaus, America/St Thomas, America/Anguilla, America/Barbados, America/Curacao, Ameri]
2018-01-05 07:00 [America/Cuiaba, Chile/Continental, America/Miquelon, America/Argentina/Catamarca, America/Argentina/Cordoba, America/Araguaina, America/Argentina/Salta, Etc/G]
2018-01-05 08:00 [Etc/GMT+2, Brazil/East, America/Sao Paulo, America/Noronha, Brazil/DeNoronha, Atlantic/South Georgia]
2018-01-05 09:00 [Etc/GMT+1, Atlantic/Cape Verde, Atlantic/Azores, America/Scoresbysund]
2018-01-05 10:00 [Europe/London, GMT, Etc/GMT-0, Europe/Jersey, Atlantic/St Helena, Europe/Guernsey, Europe/Isle of Man, Etc/GMT+0, Africa/Banjul, Etc/GMT, Africa/Freetown, GB-]
2018-01-05 11:00 [Europe/Brussels, Europe/Warsaw, CET, Etc/GMT-1, Europe/Luxembourg, Africa/Tunis, Europe/Malta, Europe/Busingen, Africa/Malabo, Europe/Skopje, Europe/Sarajevo,]
2018-01-06 12:00 [Africa/Cairo, Africa/Mbabane, Etc/GMT-2, Europe/Zaporozhye, Libya, Africa/Kigali, Africa/Tripoli, Israel, Europe/Kaliningrad, Africa/Windhoek, Europe/Buchares,]
2018-01-06 01:00 [Asia/Aden, Africa/Nairobi, Europe/Istanbul, Etc/GMT-3, Indian/Comoro, Antarctica/Syowa, Africa/Mogadishu, Africa/Asmera, Asia/Istanbul, Europe/Moscow, Africa/]
2018-01-06 02:00 [Asia/Yerevan, Etc/GMT-4, Asia/Dubai, Indian/Reunion, Indian/Mauritius, Europe/Saratov, Europe/Samara, Indian/Mahe, Asia/Baku, Asia/Muscat, Iran, Asia/Tehran,]
2018-01-06 03:00 [Asia/Aqtau, Etc/GMT-5, Asia/Samarkand, Asia/Karachi, Asia/Yekaterinburg, Asia/Dushanbe, Indian/Maldives, Asia/Oral, Asia/Tashkent, Antarctica/Mawson, Asia/Aqt]
2018-01-06 04:00 [Asia/Kashgar, Etc/GMT-6, Asia/Almaty, Asia/Dacca, Asia/Omsk, Asia/Dhaka, Indian/Chagos, Asia/Quyzylorda, Asia/Kolkata, Asia/Bishkek, Antarctica/Vostok, Asia/Kal]
2018-01-06 05:00 [Asia/Pontianak, Etc/GMT-7, Asia/Phnom Penh, Asia/Novosibirsk, Antarctica/Davis, Asia/Tomsk, Asia/Jakarta, Asia/Barnaul, Indian/Christmas, Asia/Ho Chi Minh, As]
2018-01-06 06:00 [Asia/Kuching, Asia/Chungking, Etc/GMT-8, Australia/Perth, Asia/Macao, Asia/Macau, Asia/Choibalsan, Asia/Shanghai, Asia/Ulan Bator, Asia/Chongqing, Asia/Ulaanb]
2018-01-06 07:00 [Etc/GMT-9, Pacific/Palau, Asia/Chito, Asia/Dili, Asia/Jayapura, Australia/Eucila, Asia/Yakutsk, Asia/Pyeongyang, ROK, Asia/Seoul, Asia/Khandyga, Japan, Asia/Tok]
2018-01-06 08:00 [Pacific/Yap, Pacific/Port Moresby, Australia/North, Pacific/Chuuk, Australia/Queensland, Pacific/Guam, Pacific/Truk, Asia/Vladivostok, Pacific/Saipan, Antarc]
2018-01-06 09:00 [Australia/Hobart, Australia/Tasmania, Australia/ACT, Pacific/Ponape, Pacific/Bougainville, Australia/Yancowinna, Australia/Victoria, Australia/Adelaide, Antar]
2018-01-06 10:00 [Pacific/Kwajalein, Pacific/Wallis, Pacific/Funafuti, Pacific/Nauru, Kwajalein, Pacific/Make, Pacific/Tarawa, Asia/Kamchatka, Etc/GMT-12, Asia/Anadyr, Pacific/]
2018-01-06 11:00 [Pacific/Fakaofo, Antarctica/McMurdo, Pacific/Fiji, Pacific/Enderbury, NZ, Antarctica/South Pole, Pacific/Auckland, Pacific/Tongatapu, Etc/GMT-13]
2018-01-06 12:00 [NZ-CHAT, Pacific/Chatham, Pacific/Apia, Pacific/Kiritimati, Etc/GMT-14]

1) Back
[1-1]:
```

Figura 2.36: Data e hora numa determinada zona do mundo

```
World clock
Zones at time
Date [yyyy-mm-dd hh:mm]: 2018-01-06 02:51
Asia/Yerevan
Etc/GMT-4
Asia/Dubai
Indian/Reunion
Indian/Mauritius
Europe/Saratov
Europe/Samara
Indian/Mahe
Asia/Baku
Asia/Muscat
Europe/Astrakhan
Asia/Tbilisi
Europe/Ulyanovsk

1) Back
[1-1]:
```

Figura 2.37: Lista de zonas que se encontram atualmente numa determinada date e hora

```
World clock

Difference between two time zones

Zone ID 1: Europe/Braga
The introduced value is not valid. Please try again.
```

Figura 2.38: Validação da zona introduzida. Só as TimeZones incluídas na biblioteca do JAVA que são válidas

3 Implementação

Arquitetura

Para garantir uma separação de componentes baseada na sua função, recorreu-se ao padrão de arquitetura de *software* MVC.

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>

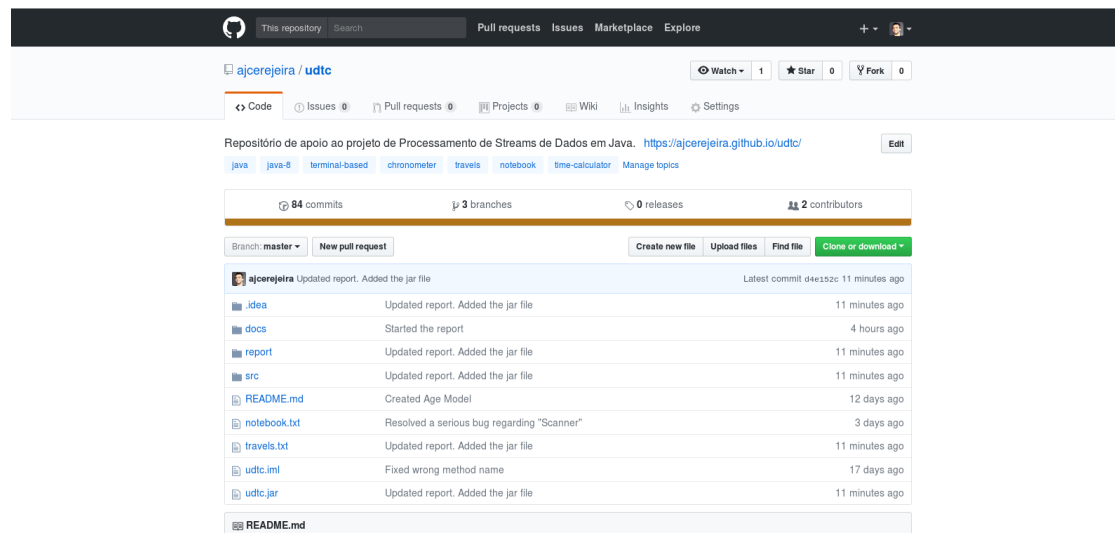


Figura 3.1: Repositório do projeto no GitHub

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.

All Classes

Packages

<unnamed package>
Controller
Model
Utils
View

All Classes

AgeCalculatorView
Appointment
Calendar
CalendarView
Chronometer
ChronometerView
Controller
DateCalculatorView
IAgeCalculator
ICalendar
IChronometer
IDateCalculator
INotebook
ITimeCalculator
ITravel
IWorldClock
Main
Notebook
NotebookView
Option
Parsers
TimeCalculatorView
Travel
Travels
TravelsView
UI
WorldClockView

OVERVIEW
PACKAGE
CLASS
TREE
DEPRECATED
INDEX
HELP

PREV CLASS
NEXT CLASS
FRAMES
NO FRAMES

SUMMARY
NESTED
FIELD
CONSTR
METHOD
DETAIL
FIELD
CONSTR
METHOD

Model

Interface IChronometer

All Known Implementing Classes:
Chronometer

public interface **IChronometer**

This interface sets the core functionality of a simple chronometer. It has only four possible actions: start, stop, pause and resume. This interface is abstract enough so that its implementations can vary in design. One simple example of its usage could be like this:

```

public static void run(IChronometer chronometer) {
    chronometer.start();
    System.out.println("Start");

    Duration d = chronometer.stop();
    System.out.println("Stop. Elapsed time:" + d);
}

```

Author:
Octavio Maia

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type	Method and Description	
java.time.Duration	pause()	

Figura 3.2: Exemplo de página da documentação do projeto

4 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 periodicidade do pagamento, bem como a quantia do mesmo e a data da 1ª 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 poderíamos 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 possível 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.