

Tolerância a Faltas

Trabalho Prático

-

Mestrado em Engenharia Informática
Universidade do Minho

Grupo nº 8

PG41080	João Ribeiro Imperadeiro
PG41081	José Alberto Martins Boticas
PG41091	Nelson José Dias Teixeira

21 de maio de 2020



Conteúdo

1	Introdução	3
2	Descrição do problema e requisitos	4
3	Implementação	5
3.1	<i>Middleware</i> genérico	5
3.1.1	<i>ServerConnection</i>	5
3.1.1.1	Replicação	5
3.1.1.2	Temporizadores	5
3.1.1.3	Comunicação entre servidores	5
3.1.1.4	Comunicação com clientes	5
3.1.2	<i>ClientConnection</i>	5
3.2	Servidor - <i>Supermarket</i>	5
3.2.1	Funcionamento	5
3.2.2	<i>CartSkeleton</i>	5
3.2.3	<i>CatalogSkeleton</i>	5
3.3	Cliente	5
3.3.1	Funcionamento	5
3.3.2	<i>CartStub</i>	5
3.3.3	<i>CatalogStub</i>	5
4	Valorizações	6
5	Conclusão	7
A	Observações	8

Lista de Figuras

Capítulo 1

Introdução

Toda a gente, de uma forma ou de outra, já esteve em contacto com uma loja *online*. Muitos podem mesmo dizer que já dependem deste tipo de serviços para efetuar as suas compras. Por não estarem diretamente relacionadas com uma localização física, estas lojas estão disponíveis para todos, independentemente de onde se encontre fisicamente no mundo.

Assim, levantam-se alguns problemas relacionados com a implementação destes tipos de serviços, como por exemplo a oferta de um serviço com bom desempenho para todos os clientes, independentemente da sua localização física. Isto leva à necessidade de não depender de apenas um servidor central, distribuindo a disponibilização do serviço por diversos servidores. Ora, isto leva a que seja necessário ter cuidados extra nas interações com os clientes, como a manutenção da consistência entre os servidores, para que, no caso em que um destes servidores falhe, o cliente não seja afetado negativamente.

Posto isto, é-nos proposto a implementação, em *Java*, de uma versão simplificada de um supermercado *online* distribuído por vários servidores, que por sua vez seja o mais tolerante a faltas possível. Para isso, entre outras ferramentas, será utilizado o protocolo *Spread* para comunicação em grupo, à semelhança do que se sucedeu ao longo das aulas da componente prática desta unidade curricular.

Capítulo 2

Descrição do problema e requisitos

Capítulo 3

Implementação

3.1 *Middleware* genérico

3.1.1 *ServerConnection*

3.1.1.1 Replicação

3.1.1.2 Temporizadores

3.1.1.3 Comunicação entre servidores

3.1.1.4 Comunicação com clientes

3.1.2 *ClientConnection*

3.2 Servidor - *Supermarket*

3.2.1 Funcionamento

3.2.2 *CartSkeleton*

3.2.3 *CatalogSkeleton*

3.3 Cliente

3.3.1 Funcionamento

3.3.2 *CartStub*

3.3.3 *CatalogStub*

Capítulo 4

Valorizações

Capítulo 5

Conclusão

Apêndice A

Observações

- Documentação *Java* 8:
`https://docs.oracle.com/javase/8/docs/api/`
- *Maven*:
`https://maven.apache.org/`
- *Spread toolkit*:
`http://www.spread.org/index.html`
- *Atomix*:
`https://atomix.io/`
- *HSQLDB*:
`http://hsqldb.org/`