

Local-First Shopping List Application

SHOPPING LISTS ON THE CLOUD

Ana Rita de Oliveira Carneiro

Jorge Costa

José Cunha

DESCRIÇÃO DO PROBLEMA

Este projeto aborda a criação de uma aplicação de lista de compras que enfrenta desafios específicos em termos de escalabilidade, colaboração em tempo real e persistência de dados.

A aplicação adota uma abordagem "local-first", garantindo que os dados sejam persistentes localmente nos dispositivos dos usuários. A integração com a nuvem é essencial para compartilhamento de dados entre usuários e fornecimento de armazenamento de backup

Cada lista de compras é identificada por um ID exclusivo, facilitando o compartilhamento e a colaboração entre usuários. Usuários que possuem esse ID têm permissão contínua para adicionar e excluir itens na lista, promovendo uma experiência colaborativa eficaz.

SOLUÇÃO TÉCNICA

Lado do Cliente

A aplicação adota uma abordagem "local-first", assegurando que os dados permaneçam persistentes localmente nos dispositivos dos usuários. Essa abordagem proporciona um acesso ágil e uma experiência responsiva, mesmo em situações de conectividade limitada.

Início da Aplicação

Ao iniciar a aplicação, é criada uma storage para guardar listas e é associado um novo User ID ao cliente. Na interface aparecem duas opções que são a criação de uma nova lista ou o acesso a uma lista existente.

```
User ID: c64d747f-bb23-4f8b-9c72-6972d5cbe790
Options:
1. Create a new shopping list
2. Enter an existing shopping list
Enter your choice (1/2): _
```

SOLUÇÃO TÉCNICA

Criação de Listas

A aplicação verifica se há conexão com o servidor usando um poller. Caso exista, a lista de compras e o respectivo ID são criados online e o conteúdo é devolvido ao cliente. Se não houver conexão com o servidor em 2 segundos, a lista será adicionada ao armazenamento local e será atualizada no servidor quando a conexão for restabelecida.

```
User ID: 4dac70d5-1e23-4c3e-ba0e-2383f88263d2
Options:
1. Create a new shopping list
2. Enter an existing shopping list
Enter your choice (1/2): 1
Enter the name for the new shopping list: NewList
Created new shopping list with ID: cc8475a5-343b-406d-8827-63cd056c3c92
```

Criação de lista online

```
User ID: c64d747f-bb23-4f8b-9c72-6972d5cbe790
Options:
1. Create a new shopping list
2. Enter an existing shopping list
Enter your choice (1/2): 1
Enter the name for the new shopping list: NewList
Server is unreachable. Creating shopping list locally.
Created new shopping list with ID: 80dbe725-b980-484f-8b3c-28b1530fd455
```

Criação de lista offline

SOLUÇÃO TÉCNICA

Acesso a uma Lista existente

É pedido que seja inserido o ID da lista que se pretende aceder. A aplicação verifica se essa lista existe localmente e acede à mesma. Caso a lista não exista localmente, a aplicação verifica se a lista existe na cloud, e caso existir é adicionada ao armazenamento local e é acedida.

```
User ID: 162acb82-1297-43f3-90ea-612490daa21e
Options:
1. Create a new shopping list
2. Enter an existing shopping list
Enter your choice (1/2): 2
Enter the ID of the existing shopping list: 204bdcce-4396-4f3f-8fed-07f21b27639d

List Name: NewList

Options for the shopping list:
1. Add an item
2. Delete an item
3. Back to main menu
4. Refresh
Enter your choice (1/2/3/4):
```

Lista existe localmente

```
User ID: 387c9664-cd26-4772-9297-ee3f1c6a4fdb
Options:
1. Create a new shopping list
2. Enter an existing shopping list
Enter your choice (1/2): 2
Enter the ID of the existing shopping list: 204bdcce-4396-4f3f-8fed-07f21b27639d
List does not exist locally. Checking server storage...
Connected to server 5557
List NewList found.

List Name: NewList

Options for the shopping list:
1. Add an item
2. Delete an item
3. Back to main menu
4. Refresh
Enter your choice (1/2/3/4):
```

Lista não existe localmente e é procurada no server

SOLUÇÃO TÉCNICA

Adição e Remoção de Items

O cliente adiciona e exclui itens com e sem conexão com o servidor. Cada entrada de item contém informações de nome do item, quantidade, timestamp e User ID de quem fez a modificação.

```
List Name: NewList

Options for the shopping list:
1. Add an item
2. Delete an item
3. Back to main menu
4. Refresh
Enter your choice (1/2/3/4): 1
Enter the name of the item to add: rice
Item does not exist. Adding item.

List Name: NewList
Item Name: rice, Quantity: 1, Time: 1702324580, Client ID: 162acb82-1297-43f3-90ea-612490daa21e
```


SOLUÇÃO TÉCNICA

Sincronização com o servidor e outros clientes

A cada 2 segundos a aplicação verifica se há conexão com o servidor. Se houver, as listas de compras do cliente são enviadas ao servidor e ocorre um merge de CRDT's no servidor para sincronizar os dados. Uma resposta é enviada pelo servidor com as listas atualizadas e ocorre também um merge de CRDT's no lado do cliente.

```
List Name: NewList
Options for the shopping list:
1. Add an item
2. Delete an item
3. Back to main menu
4. Refresh
Enter your choice (1/2/3/4): 4

List Name: NewList
Item Name: rice, Quantity: 1, Time: 1702324580, Client ID: 162acb82-1297-43f3-90ea-612490daa21e
```

Cliente faz refresh da lista

```
Enter the name of the item to add: rice
Item exists. Incremented quantity.

List Name: NewList
Item Name: rice, Quantity: 3, Time: 1702324764, Client ID: 162acb82-1297-43f3-90ea-612490daa21e

Options for the shopping list:
1. Add an item
2. Delete an item
3. Back to main menu
4. Refresh
Enter your choice (1/2/3/4): 1
Enter the name of the item to add: potatoes
Item does not exist. Adding item.

List Name: NewList
Item Name: rice, Quantity: 3, Time: 1702324764, Client ID: 162acb82-1297-43f3-90ea-612490daa21e
Item Name: potatoes, Quantity: 1, Time: 1702324776, Client ID: 162acb82-1297-43f3-90ea-612490daa21e
```

Cliente 1 - edição mais antiga

```
Enter the name of the item to add: corn
Item does not exist. Adding item.

List Name: NewList
Item Name: rice, Quantity: 1, Time: 1702324580, Client ID: 162acb82-1297-43f3-90ea-612490daa21e
Item Name: corn, Quantity: 1, Time: 1702324807, Client ID: 387c9664-cd26-4772-9297-ee3f1c6a4fdb

Options for the shopping list:
1. Add an item
2. Delete an item
3. Back to main menu
4. Refresh
Enter your choice (1/2/3/4): 1
Enter the name of the item to add: rice
Item exists. Incremented quantity.

List Name: NewList
Item Name: rice, Quantity: 2, Time: 1702324815, Client ID: 387c9664-cd26-4772-9297-ee3f1c6a4fdb
Item Name: corn, Quantity: 1, Time: 1702324807, Client ID: 387c9664-cd26-4772-9297-ee3f1c6a4fdb
```

Cliente 2 - edição mais recente

SOLUÇÃO TÉCNICA

```
List Name: NewList
Item Name: rice, Quantity: 3, Time: 1702324764, Client ID: 162acb82-1297-43f3-90ea-612490daa21e
Item Name: potatos, Quantity: 1, Time: 1702324776, Client ID: 162acb82-1297-43f3-90ea-612490daa21e

Options for the shopping list:
1. Add an item
2. Delete an item
3. Back to main menu
4. Refresh
Enter your choice (1/2/3/4): 4

List Name: NewList
Item Name: rice, Quantity: 2, Time: 1702324815, Client ID: 387c9664-cd26-4772-9297-ee3f1c6a4fdb
Item Name: potatos, Quantity: 1, Time: 1702324776, Client ID: 162acb82-1297-43f3-90ea-612490daa21e
Item Name: corn, Quantity: 1, Time: 1702324807, Client ID: 387c9664-cd26-4772-9297-ee3f1c6a4fdb
```

Cliente 1 após merge

```
List Name: NewList
Item Name: rice, Quantity: 2, Time: 1702324815, Client ID: 387c9664-cd26-4772-9297-ee3f1c6a4fdb
Item Name: corn, Quantity: 1, Time: 1702324807, Client ID: 387c9664-cd26-4772-9297-ee3f1c6a4fdb

Options for the shopping list:
1. Add an item
2. Delete an item
3. Back to main menu
4. Refresh
Enter your choice (1/2/3/4): 4

List Name: NewList
Item Name: rice, Quantity: 2, Time: 1702324815, Client ID: 387c9664-cd26-4772-9297-ee3f1c6a4fdb
Item Name: corn, Quantity: 1, Time: 1702324807, Client ID: 387c9664-cd26-4772-9297-ee3f1c6a4fdb
Item Name: potatos, Quantity: 1, Time: 1702324776, Client ID: 162acb82-1297-43f3-90ea-612490daa21e
```

Cliente 2 após merge

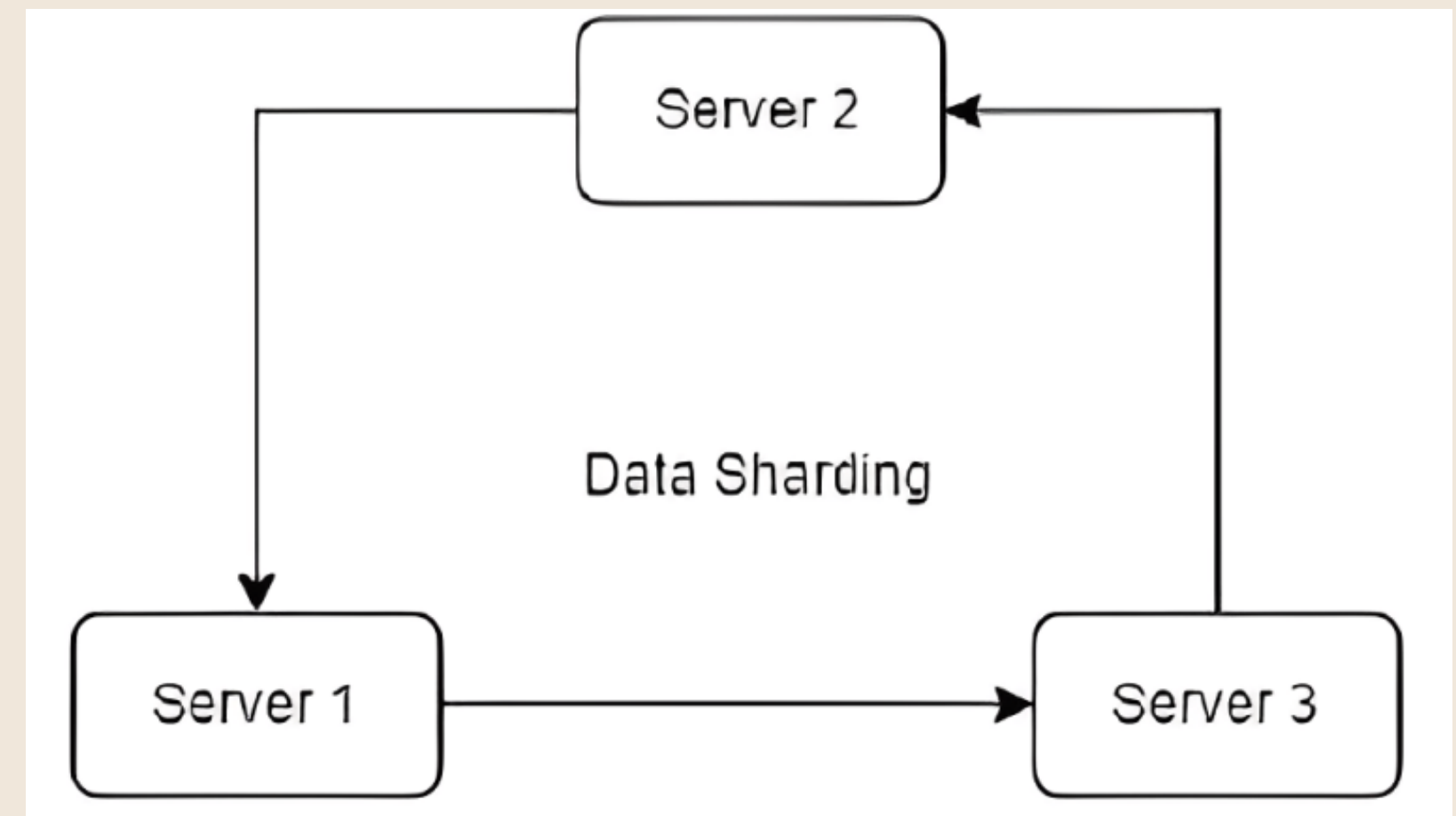
Durante o merge é aplicado um processo Last Writer Wins, que significa que a entrada com maior processamento de timestamp nos dados procede.

SOLUÇÃO TÉCNICA

Servidores na Nuvem

Implementamos estratégias para lidar com mudanças concorrentes nas listas, assegurando alta disponibilidade e consistência em todo o sistema.

Utilizamos sharding para garantir a independência de cada lista. Isso permite a escalabilidade horizontal, possibilitando o gerenciamento eficiente de vários usuários.

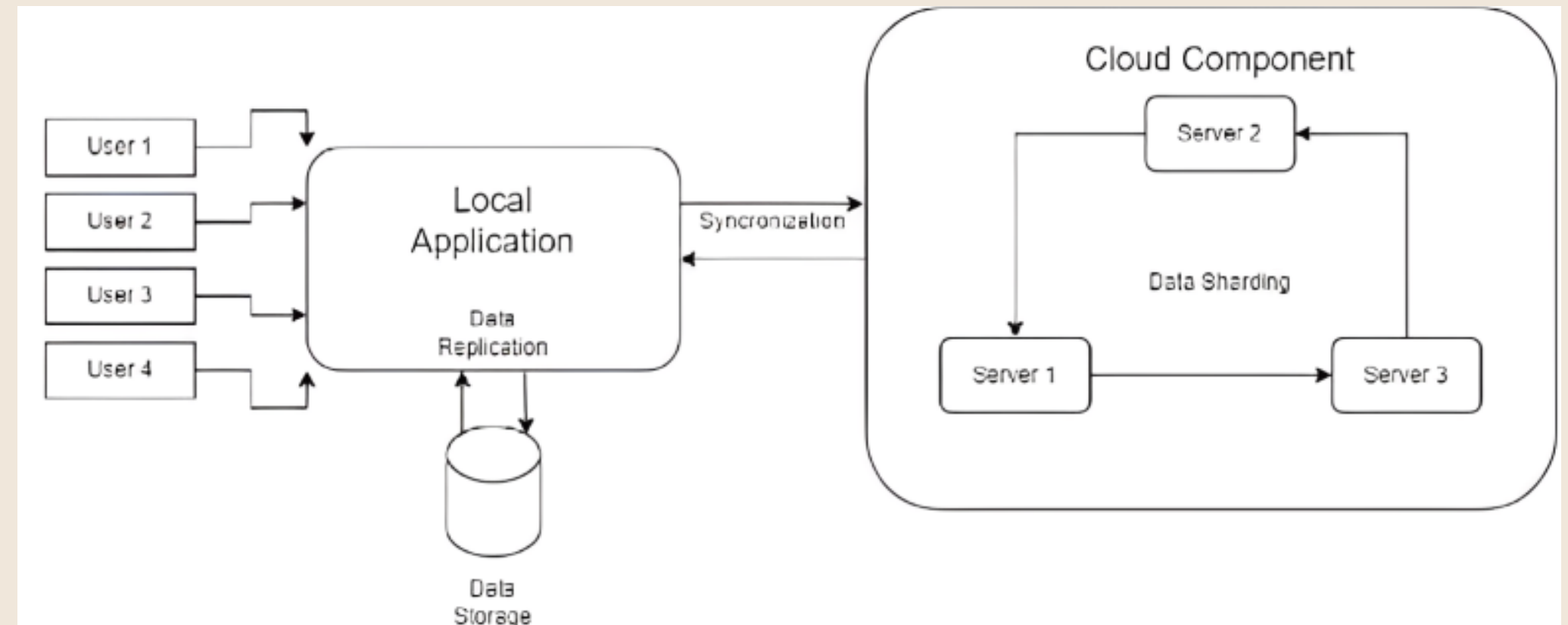


Solução Técnica

A aplicação local opera no dispositivo do utilizador, estabelecendo uma base descentralizada.

A sincronização entre utilizadores é coordenada pela componente na nuvem, composta por três servidores estáticos.

O processo de sharding consiste na escolha de qual servidor irá ser responsável por cada lista, com base no ID da mesma.



AVALIACÃO DA SOLUÇÃO

A aplicação contém algumas limitações:

- Não foi implementado um armazenamento caso a aplicação seja encerrada, e sim foi desenvolvida uma simulação de local-first, sendo que o armazenamento existe durante a execução da aplicação. Seria necessário escolher uma estrutura de dados e desenvolver um parser para leitura e escrita.
- Não foram implementados mecanismos de login, sendo que é criado um novo User ID sempre que a aplicação é executada.
- Não foram implementados mecanismos complexos de segurança para prevenir acesso indevido às listas.
- No caso de criação de listas offline, a aplicação ao criar um ID para a lista localmente pode criar conflito caso esse ID já exista no servidor.
- Caso um item seja removido e a quantidade seja zero, o mesmo não desaparece da lista.
- Não foi implementado load balancer.
- Não foi implementado proxy, o processo de sharding é efetuado no lado do cliente.
- A aplicação do lado do cliente apenas sabe que existem 3 servidores estáticos, o que limita caso sejam criados mais servidores.
- Não foram implementados sistemas de backup de dados entre os servidores.