

FUNDAÇÃO ESCOLA DE COMÉRCIO ÁLVARES PENTEADO

**Gustavo Henrique Santos Araujo
Emilly Mickeli Depine Da Silva
Fernando Jose Dos Santos
Renan Teixeira Pinheiro**

**Testes e Qualidade de Software (DevOps)
Atributos de qualidade de software**

**SÃO PAULO
2024**

1. Desempenho (Performance)

Definição: O desempenho refere-se à capacidade do sistema de responder rapidamente às requisições dos usuários, lidando com a carga de trabalho de maneira eficiente. Isso inclui tempo de resposta rápido, uso eficiente dos recursos e a habilidade de suportar cargas altas sem degradação significativa da qualidade.

Como foi aplicado no app:

API no Azure: A API foi configurada para ser altamente otimizada, com o uso de técnicas como cache e paginação de resultados para garantir que o tempo de resposta seja mantido baixo mesmo quando muitos usuários acessam os dados simultaneamente.

Consultas eficientes no MySQL: O banco de dados foi projetado para ter índices nas colunas mais consultadas, o que melhora a velocidade das consultas SQL. Além disso, foram implementadas consultas otimizadas e a normalização de dados para reduzir o tempo de execução das operações no banco.

Uso do Retrofit no Android: O Retrofit foi configurado para realizar requisições de forma eficiente, com a implementação de cache local e compressão de dados para minimizar o uso de banda e garantir respostas mais rápidas.

2. Confiabilidade (Reliability)

Definição: A confiabilidade diz respeito à capacidade do sistema de funcionar corretamente sob condições normais de operação, com o mínimo possível de falhas ou erros. Isso inclui a robustez do sistema diante de entradas inválidas ou condições inesperadas.

Como foi aplicado no app:

Autenticação e Autorização: A API possui um sistema de autenticação robusto, utilizando tokens JWT para garantir que apenas usuários autorizados possam acessar funções como o cadastro de alimentos e a troca de mensagens.

Tratamento de Erros: O app foi projetado para lidar com erros de rede e falhas na comunicação entre o servidor e o cliente de forma eficiente. Quando o usuário tenta acessar a API sem conexão, uma mensagem de erro amigável é apresentada, e o sistema tenta novamente realizar a requisição após um breve intervalo.

Resiliência da API: A API no Azure foi configurada com failover automático e backups periódicos para garantir que os dados estejam sempre disponíveis, mesmo em caso de falhas de servidores ou outros problemas técnicos.

3. Usabilidade (Usability)

Definição: A usabilidade refere-se à facilidade com que os usuários podem interagir com o sistema, incluindo aspectos como interface amigável, fluxo intuitivo e facilidade de aprendizado.

Como foi aplicado no app:

Interface do Usuário: O design do aplicativo foi feito com foco na experiência do usuário, garantindo que as telas sejam simples, claras e intuitivas. A navegação entre as telas foi planejada para que o usuário consiga realizar ações como cadastrar alimentos, fazer doações e pesquisar por alimentos rapidamente.

Feedback Visual e Interativo: O aplicativo fornece feedback visual imediato ao usuário para ações como sucesso no cadastro, erro de login ou ao realizar uma doação. Isso melhora a experiência do usuário, pois ele sempre sabe em que estado se encontra dentro da aplicação.

Acessibilidade: A aplicação foi projetada para ser acessível, com fontes legíveis e um esquema de cores que facilita a visualização mesmo em condições de baixa luminosidade.

4. Escalabilidade (Scalability)

Definição: A escalabilidade é a capacidade do sistema de crescer e suportar aumentos na carga de trabalho, seja através da adição de mais recursos ou pela adaptação eficiente a novas demandas sem uma degradação significativa do desempenho.

Como foi aplicado no app:

Escalabilidade na Azure: A aplicação foi hospedada na Azure, que oferece escalabilidade automática para aumentar ou diminuir a quantidade de recursos conforme a demanda. Se houver picos de tráfego (como quando muitos usuários estão cadastrando alimentos ou buscando doações ao mesmo tempo), o servidor pode automaticamente aumentar sua capacidade para lidar com o tráfego extra.

Banco de Dados Escalável: O banco de dados MySQL foi configurado para suportar uma quantidade crescente de usuários e registros, com a possibilidade de particionamento de dados e o uso de réplicas para garantir a alta disponibilidade.

API com suporte a múltiplos usuários simultâneos: A API foi projetada para ser stateless, o que facilita a escalabilidade horizontal. O sistema pode adicionar mais instâncias de servidores conforme o número de requisições aumenta, sem comprometer a integridade ou a performance.