

# **RELATÓRIO DE ANÁLISE DE DESENVOLVIMENTO**

**PROJETO E-COMMERCE -AVALIAÇÃO AP004**

**TIME 3**

**DATA 24/05/2024**



**EQUIPE E PAPÉIS**

Gabriel dos Santos Teles - Líder Técnico

Alan Prates dos Santos - Desenvolvedor

Daniel Andrade Penedo Santos - Desenvolvedor

Giuseppe Mota Araújo - Desenvolvedor

Luann Firme Bomfim – Desenvolvedor

**SUGESTÃO DE ALTERAÇÕES EM MÓDULOS**

1. Melhorias de performance
2. Melhorias de segurança
3. Melhorias de escalabilidade

**JUSTIFICATIVA DE ALTERAÇÕES EM MÓDULOS**

**1. Quanto à performance:**

- A utilização de caching reduz a carga do servidor e melhora o tempo de resposta. Implementar cache para armazenar dados frequentemente acessados pode reduzir significativamente a quantidade de requisições ao banco de dados.

- O uso do Refit melhora o consumo de APIs, facilitando a comunicação com serviços externos e tornando o código mais limpo e fácil de manter. O Refit permite a criação de clientes de API de forma declarativa, reduzindo a quantidade de código boilerplate e aumentando a eficiência.

**2. Quanto à segurança:**

- Adicionar detalhes específicos sobre melhorias de segurança, como a implementação de autenticação OAuth, criptografia de dados sensíveis, ou adoção de práticas de desenvolvimento seguro.

**3. Quanto à escalabilidade:**

- O uso de Arquitetura de Micro serviços permite que a aplicação evolua de forma mais efetiva, possibilitando a escalabilidade independente de componentes do sistema. Isso facilita a adição de novas funcionalidades e a adaptação às demandas crescentes sem impactar negativamente outras partes do sistema.

- Adicionar processos de Failover para recuperação automática de falhas a fim de evitar uma parada completa na aplicação, sendo eles balanceamento de carga (distribuição da carga de trabalho entre várias instâncias a fim de não sobrecarregar nenhuma), adição do Kubernetes ou Docker Swarm( possuem recursos de autocorreção que gerenciam o ciclo de vida da aplicação) e interrupção de circuito junto com políticas de tentativas/tempo limite( fornecem uma forma de não sobrecarregar e aliviar o tráfego na aplicação)

- Implementar a Arquitetura de Camadas evita acoplamento e facilita a criação de testes unitários eficientes, visando uma melhor manutenção do código, também possibilitando mais liberdade ao desenvolver o Front, já que a apresentação pode ser feita em diversas linguagens, para múltiplas plataformas(Ex: mobile, web, etc... ).

- Implementação de html via framework a fim de obter uma identidade visual mais agradável e de fácil modificação

**RELATÓRIO DE ATIVIDADES**

|  |  |  |  |
| --- | --- | --- | --- |
| **TAREFA** | **RESPONSÁVEL** | **TEMPO PREVISTO** | **TEMPO EXECUTADO** |
| ATUALIZAR PACOTES | Alan Prates, Gabriel Teles | 8 horas | 8 horas |
| SEPARAR CAMADAS | Daniel Penedo, Giuseppe Mota | 12 horas | 12 horas |
| REFATORAR ENTIDADES | Alan Prates,  Luann Firme | 10 horas | 10 horas |
| REFATORAR CONTROLLERS | Giuseppe Mota, Luann Firme | 14 horas | 14 horas |
| ADICIONAR TESTES UNITÁRIOS | Gabriel Teles | 8 horas | 8 horas |
| ADICIONAR FRONT( Angular 17) | Luann Firme, Gabriel Teles | 8 horas | 8 horas |