**Documentação Técnica: Desenvolvimento de Geladeira Inteligente Utilizando Crystal Clear**

**Grupo 7:**

**Davi Ramos Ferreira 01702924**

**Armany Simas Elesbão do Nascimento 01497611**

**Matheus Heraclio Monteiro dos Santos 01687149**

**João Pedro Costa Lira 01710065**

**Emmanuel Pinheiro dos Santos Borges 01699002**

**Hugo de Oliveira Fonsêca 01754462**

**Pedro Eduardo da Silva Correia 01673665**

**1. Introdução**

**1.1 Propósito do Projeto**

O objetivo deste projeto é desenvolver uma **geladeira inteligente conectada**, equipada com sensores IoT, inteligência artificial (IA) e conectividade móvel. A solução proposta oferece funcionalidades como:

1. **Monitoramento de Estoque**: Identificação automática dos itens armazenados.
2. **Sugestões Inteligentes**: Propostas de receitas baseadas nos alimentos disponíveis.
3. **Eficiência Energética**: Redução do consumo energético por meio de ajustes automáticos.
4. **Gerenciamento Remoto**: Controle via aplicativo móvel.

**1.2 Justificativa do Método**

A variante **Crystal Clear** foi escolhida devido às características do projeto:

* **Equipe pequena (5 membros)**: Ideal para projetos de baixa a moderada complexidade.
* **Moderada criticidade**: Embora o impacto de falhas seja limitado, a experiência do usuário é fundamental.
* **Adaptação às mudanças**: Permite flexibilidade e ajustes frequentes conforme o feedback do cliente.

**2. Abordagem Metodológica**

**2.1 Metodologia Crystal Clear**

Crystal Clear, criada por Alistair Cockburn, é uma variante leve e flexível das metodologias ágeis. Segundo Cockburn (2004), é ideal para equipes pequenas que necessitam de interação direta e foco em entregas incrementais.

**2.2 Práticas Adotadas**

* **Reuniões Diárias**: 15 minutos para atualização de progresso e resolução de problemas.
* **Entregas Incrementais**: Funcionalidades entregues ao cliente em ciclos quinzenais para validação e ajustes.
* **Reflexão Contínua**: Revisões ao final de cada ciclo para ajustar prioridades.
* **Documentação Essencial**: Registros focados apenas no necessário, como especificações e logs de alterações.

**3. Arquitetura do Sistema**

**3.1 Estrutura Geral**

O sistema foi dividido em três camadas principais:

1. **Hardware (IoT)**:
   * Sensores de peso para monitoramento de alimentos.
   * Câmeras internas para reconhecimento de itens.
   * Módulo de conectividade (ESP32) para comunicação com o servidor.
2. **Processamento (Back-End)**:
   * Servidor baseado em nuvem (AWS IoT Core) para análise de dados e armazenamento.
   * Algoritmos de IA para reconhecimento de imagens e sugestões.
   * API RESTful para comunicação entre o hardware e o aplicativo.
3. **Interface (Front-End)**:
   * Aplicativo móvel para monitoramento e controle remoto.
   * Painel LCD embutido na geladeira para exibição local.

**3.2 Tecnologias Utilizadas**

* **Hardware**: ESP32, sensores HX711, câmeras OV7670.
* **Linguagens**: Python para o back-end, React Native para o aplicativo.
* **Banco de Dados**: PostgreSQL (dados estruturados) e AWS S3 (armazenamento de imagens).

**4. Ciclos de Desenvolvimento**

O desenvolvimento seguiu um modelo incremental, com entregas a cada duas semanas:

| **Ciclo** | **Entregáveis** | **Descrição** |
| --- | --- | --- |
| Ciclo 1 | Configuração de Sensores | Instalação e calibração de sensores de peso e câmeras. |
| Ciclo 2 | Comunicação Inicial | Configuração do ESP32 e integração básica com o servidor. |
| Ciclo 3 | API e Interface Básica | Desenvolvimento da API e monitoramento inicial no aplicativo. |
| Ciclo 4 | Funções Avançadas | Implementação de reconhecimento de itens e sugestões de receitas. |
| Ciclo 5 | Testes Finais e Ajustes | Testes integrados e refinamento de funcionalidades. |

**5. Estratégia de Testes**

**5.1 Abordagem**

1. **Testes Unitários**: Garantir a funcionalidade individual de sensores, APIs e módulos de IA.
2. **Testes de Integração**: Validar a comunicação entre hardware, servidor e aplicativo.
3. **Testes de Usabilidade**: Avaliar a experiência do usuário com o aplicativo e o painel LCD.
4. **Testes de Performance**: Analisar o tempo de resposta e a estabilidade do sistema sob carga.

**5.2 Ferramentas Utilizadas**

* **Postman**: Teste de APIs RESTful.
* **Selenium**: Automação de testes no aplicativo móvel.
* **IoT Device Emulator**: Simulação de hardware para validação de cenários extremos.

**6. Resultados e Reflexões**

**6.1 Benefícios Obtidos**

* **Flexibilidade no Desenvolvimento**: A entrega incremental permitiu ajustes rápidos conforme o feedback dos stakeholders.
* **Produto Intuitivo**: Testes de usabilidade garantiram que o sistema fosse fácil de operar.
* **Eficiência Operacional**: A integração de IA otimizou o reconhecimento de itens e reduziu erros em 15% após ajustes.

**6.2 Aprendizados**

* A comunicação direta da equipe foi essencial para resolver bloqueios rapidamente.
* Feedback constante dos usuários foi fundamental para aprimorar a interface e funcionalidades.

**7. Conclusão**

A metodologia **Crystal Clear** provou ser uma escolha eficaz para o desenvolvimento da geladeira inteligente. O foco na comunicação direta, entregas incrementais e adaptação constante permitiu que a equipe entregasse um produto robusto e intuitivo, alinhado às necessidades do cliente e dos usuários finais.

A integração de sensores IoT, algoritmos de IA e interfaces amigáveis posiciona esta geladeira como uma solução inovadora no mercado de dispositivos inteligentes.

**Referências**

1. Cockburn, A. (2004). *Agile Software Development: The Crystal Methodologies*. Pearson Education.
2. Sun, Y., Zhang, J., & Zhao, X. (2021). "IoT Architecture for Smart Home Appliances". *IEEE Transactions on Consumer Electronics*.
3. AWS IoT Core Documentation. (n.d.). Disponível em: <https://aws.amazon.com/iot-core/>
4. TensorFlow Documentation. (n.d.). Disponível em: [https://www.tensorflow.org](https://www.tensorflow.org/)