

Escopo do Projeto - FlexDesk

Equipe formada por:

Guilherme Elizeu Krupinski

Kelvin França

Gustavo Ehrat Meinert

Carlos Eduardo Gonçalves

Partes Interessadas: Locadores e Locatários de Coworking

Escopo do Projeto:

Identificar e Construir uma solução para sanar os desafios significativos na busca por espaços de coworking confiáveis e acessíveis. Dessa forma, conectar as empresas e trabalhadores autônomos aos espaços disponíveis em sua região.

Portanto, oferecer uma solução simplificada para a crescente demanda, fornecendo às empresas e autônomos acesso a ambientes colaborativos e flexíveis, ideais para impulsionar a produtividade e a inovação.

Produto:

Aplicativo de locação de espaços de Coworking. A *FlexDesk* tem como missão promover espaços de coworking, oferecendo escritórios e auditórios para outras empresas e grupos realizarem suas atividades, proporcionando um ambiente colaborativo e funcional que estimula a produtividade e a criatividade.

Metodologia Adotada

A metodologia Kanban é um sistema visual para gerenciamento de fluxo de trabalho, originário da indústria japonesa e popularizado pela Toyota. Ele se baseia em cartões (ou "kanbans") que representam tarefas ou itens em diferentes etapas

do processo. Esses cartões são movidos através de um quadro, que geralmente é dividido em colunas representando as diferentes fases do processo, desde o backlog até a conclusão.

O método Kanban oferece uma abordagem flexível e adaptável para gerenciar o trabalho, destacando o fluxo contínuo de atividades e minimizando o trabalho em progresso. Ele é especialmente adequado para projetos como o desenvolvimento de software ou aplicativos, onde há uma variedade de tarefas em andamento simultaneamente e a necessidade de priorizar e responder rapidamente às mudanças no ambiente.

Ao adotar o Kanban para o projeto de um aplicativo de coworking, obtemos os seguintes benefícios:

1. Visibilidade do processo* Com o quadro Kanban, todos os membros da equipe têm uma visão clara do estado atual das tarefas e do progresso do projeto.

2. Flexibilidade: O Kanban permite que a equipe ajuste rapidamente as prioridades e responda às mudanças nas demandas do mercado ou nas necessidades dos usuários.

3. Redução de desperdícios: Ao limitar o trabalho em progresso e identificar gargalos no processo, o Kanban ajuda a minimizar o tempo e os recursos desperdiçados.

4. Melhoria contínua: Através da análise do fluxo de trabalho e do feedback dos clientes, a equipe pode identificar oportunidades de melhorias incrementais e iterativas no processo.

5. Colaboração: O Kanban promove a colaboração entre os membros da equipe, uma vez que todos têm visibilidade das tarefas e podem trabalhar juntos para resolver problemas e atingir objetivos.

Em resumo, o Kanban é uma escolha sólida para o desenvolvimento de um aplicativo de coworking devido à sua flexibilidade, foco no fluxo de trabalho e capacidade de adaptação às mudanças constantes no ambiente de desenvolvimento de software. Ele permite uma abordagem ágil e colaborativa para a entrega de valor aos usuários finais.

Requisitos Funcionais

RF01	O usuário deve se cadastrar no sistema
RF02	O usuário deve ser capaz de fazer login no sistema
RF03	Deve haver diferenciação entre usuário PF e PJ
RF04	O usuário deve ser capaz de recuperar a senha através de confirmação de código enviado via e-mail cadastrado
RF05	O cadastro do usuário deve ser preenchido com Nome, CPF/CNPJ, E-mail, Senha e Endereço completo
RF06	O usuário deve ser capaz de cadastrar um imóvel, com imagens, descrição, comodidades, valor
RF07	Telefone/WhatsApp e Endereços adicionais podem ser preenchido de forma opcional no cadastro do usuário
RF08	O usuário deve ser capaz de editar as informações do imóvel como imagens, descrição, comodidades, valor, tipo, localização
RF09	O usuário deve ser capaz de pesquisar um imóvel por comodidades, valor, localização, tamanho e tipo
RF10	O usuário deve ser capaz de cadastrar mais de um imóvel
RF11	O usuário deve ser capaz de pagar a locação diretamente pelo aplicativo (pix, cartão de crédito)
RF12	O usuário deve ser capaz de avaliar o imóvel alugado (estrelas 1-5 e

	comentários)
RF13	O usuário deve ser capaz de avaliar o locatário que utilizou o seu imóvel (estrelas 1-5 e comentários)
RF14	A locação do imóvel pode ser realizada por dia, semana, mês ou ano
RF15	A locação do imóvel precisa da confirmação do locador para ser efetivada
RF16	A página principal do sistema deve contar com imóveis recomendados para locação
RF17	A seção de busca deve conter filtros não obrigatórios como comodidades, intervalo de valor, localização, tamanho, tipo e datas
RF18	O usuário poderá ver, alterar, cancelar e salvar reservas de interesse
RF19	O perfil do usuário deve ter informações como dados pessoais, informações de pagamento, imóveis publicados, avaliação geral
RF20	O usuário deve ser capaz de alterar a senha através da seção “perfil do usuário”
RF21	A seção “Explorar” deve contar com imóveis populares para locação
RF22	O sistema deve notificar eventos chave como: confirmação de locação, confirmação de pagamento e chamadas ao usuário
RF23	O sistema deve ter uma seção de suporte ao cliente
RF24	O suporte ao cliente deve ser dividido em 2 níveis de atendimento: 1º chatbot, 2º time de suporte

Requisitos Não Funcionais

RNF01	O aplicativo deve ser integrado com um sistema de pagamentos seguro para pagamento e recebimento das reservas
RNF02	O aplicativo deve ter uma interface intuitiva e amigável para que seja fácil de usar para uma ampla gama de usuários
RNF03	O aplicativo deve ser responsivo para que funcione em diversos tipos de plataformas
RNF04	Deve-se projetar o aplicativo para lidar com o aumento de usuários e reservas sem que afete o desempenho
RNF05	Respeitar os dados dos usuários e cumprir os regulamentos da LGPD
RNF06	Os usuários podem avaliar e dar FeedBack para a plataforma
RNF07	O sistema deve ter integração com o calendário do Google Agenda
RNF08	O sistema deve continuar operando mesmo quando ocorrem falhas em componentes individuais
RNF09	O sistema deve ter uma maneira de fazer backup e recuperação de dados para garantir a integridade dos dados de usuários
RNF10	O sistema deve se recuperar de falhas e restaurar o sistema a um estado operacional sem perda de dados significativa
RNF11	O sistema deve ser mantido e modificado ao longo do tempo com facilidade para atender às necessidades em evolução
RNF12	O sistema deve levar um tempo breve para responder a uma solicitação do usuário
RNF13	Garantir que o aplicativo esteja sempre disponível, reduzindo ao máximo períodos de inatividades planejados ou não

Protótipo

Como proposta de construção do produto final, desenvolvemos um protótipo do aplicativo mobile de coworking a fim de demonstrar as funcionalidades esperadas e caminhos a serem seguidos. Além disso, a prototipação foi utilizada para refinar os requisitos funcionais do projeto, ao verificar as necessidades que surgiram ou mudaram a partir da visualização do produto.

Para a prototipação do projeto, utilizamos a ferramenta Figma.

Link do Protótipo:

<https://www.figma.com/design/mWqBylCDO6NxOxllbnxpMi/FlexDex---App?node-id=0-1&t=OKt5cxGzOopn5LZ2-0>

Testes

Para realizar os testes no aplicativo móvel desenvolvido, escolhemos a Kobiton. Kobiton acelera a entrega de aplicativos móveis por meio de testes manuais, automatizados e sem código em dispositivos reais. A plataforma de testes móveis com IA aumentada da Kobiton oferece integração e testes contínuos com uma exclusividade de uma hora.

A plataforma inclui automação orientada por IA, com recursos como scripts de autocorreção e fluxos de trabalho de teste sem script. Além disso, proporciona uma execução mais rápida de scripts, resultando em ciclos de teste mais ágeis. Os usuários recebem resultados detalhados que auxiliam na tomada de decisões e contam com opções de implantação flexíveis para diversas necessidades de testes. Os dispositivos reais da Kobiton garantem baixa latência e altas taxas de quadros, contribuindo para testes rápidos e precisos.

Características

- Facilidade de uso: Centraliza o histórico de testes e registros de dados para aumentar a colaboração e simplificar a experiência do usuário, agilizando as

sessões de teste. Também disponibiliza os dispositivos e configurações reais mais recentes baseados em nuvem.

- Recursos de teste: Inclui testes de desempenho, Test Cloud personalizável, Agile Test Enabler, Rich Test Logs e eficiência otimizada.

- Implantação segura: Oferece uma opção de implantação local, proporcionando maior controle sobre a segurança durante a implantação.

- Gestos naturais: Permite realizar todos os gestos necessários para testar dispositivos, como rolar, deslizar e tocar.

- Linguagens de programação suportadas Suporta linguagens como C#, Java, Ruby, NodeJS, PHP e Python.

- Frameworks suportados Compatível com diversos frameworks como React Native, Ionic, Electron, Native Script e Xamarin.

- Tipos de testes Suporta testes de desempenho, automação, manuais, funcionais, entre outros.

- Integrações Integra-se perfeitamente com Travis CI, Jenkins, Azure DevOps, XebiaLabs, TeamCity, CircleCI, Jira e mais.

- Outras características: Oferece funcionalidade de gravação e reprodução, navegação cruzada, automação sem código, depuração remota e teste em dispositivos reais.

- Plataformas suportadas: Compatível com dispositivos iOS e Android.

Ao utilizar a Kobiton, garantimos uma abordagem robusta e eficiente para o teste de nosso aplicativo móvel, assegurando uma entrega de alta qualidade.

Análise de Riscos

Ausência de recursos para o desenvolvimento do projeto - Financiamento:

Como uma iniciativa para prevenir este risco teremos que buscar financiamento externo de partes interessadas, assim evitando problemas como falta de escalabilidade e qualidade no produto final

Falta de priorização de ordem de desenvolvimento e implantação dos módulos do projeto - Priorização: Mesmo este sendo um risco de baixo impacto devemos organizar as prioridades de entrega do projeto

Atrasar os prazos de entrega do projeto - Estimativa: Estar preparado para lidar com dificuldades e problemas no andamento do projeto, visto que este risco, caso venha a se tornar um problema pode gerar aumento do custo final do projeto e reduzir a credibilidade da empresa

Falta de efetividade na comunicação da equipe - Comunicação: Usar uma ferramenta de gestão de atividades e projetos, e adotar a prática de Dailys para evitar desperdício de recursos da equipe

Descontinuação das ferramentas adotadas para o desenvolvimento - Tecnologia: Adotar ferramentas e tecnologias consolidadas no mercado prevenindo assim a inutilização da aplicação desenvolvida

Acesso inesperado de usuários acima do previsto na aplicação - Desempenho: Investir em um número de acessos acima do esperado para não comprometer o desempenho e funcionamento da aplicação

Quantidade baixa de usuários na aplicação - Aquisição: Contratar uma empresa especializada em divulgação e marketing para assim não ter problemas de falta de retorno financeiro para manutenção e evolução da aplicação

Não atender às expectativas do público alvo: Levantando as necessidades específicas do público alvo para evitar ter que manter os usuários fidelizados à ferramenta