**OBJETIVO**

Desenvolver e lançar um site/aplicativo interativo que ensina programação por meio de desafios simples e progressivos, visando alcançar usuários iniciantes em programação.

Os jogadores enfrentarão uma variedade de desafios de programação, como resolver algoritmos, corrigir bugs de código, e criar funções para avançar nos desafios.

Os jogadores ganharão pontos e recompensas por completar os desafios com eficiência e precisão.

**ETAPAS**

- **Definição do Propósito**

Nosso propósito com esse projeto é trazer maior acessibilidade às pessoas no mundo da programação. Explicando e treinando os conhecimentos aprendidos de forma interativa, fácil e prática.

- **Escopo do Projeto**

Funcionalidades principais: Aprender a programar de uma forma fácil e prático.

Plataformas suportadas: Web, Android, ios, linux, ws, mac.

Limitações e Exclusões:

Requisitos técnicos: Usaremos linguagem HTML, CSS, JavaScript, Unity com C# ou outra plataforma equivalente.

Desing e Interface do Usuário: Uma interface intuitiva e atraente, com elementos visuais claros com instruções detalhadas sobre o desafio.

Teste e Garantia de Qualidade: Faremos vários testes do básico até uma função específica e sempre revisando o conteúdo, monitorando o desempenho e fazendo uma manutenção proativa.

- **Recursos Necessários**

Recursos Humanos: Desenvolvedores, Gerente de Projeto, Designer de Interface do Usuário (UI, Designer Gráfico, Especialista em Experiência do Usuário (UX), Testadores/QA.

Recursos Tecnológicos: Hardware, Software, Plataforma de Hospedagem.

Recursos Financeiros: Orçamento para Desenvolvimento, Reserva para Contingências.

Recursos de Conhecimento: Documentação, Acesso a Bibliotecas e Frameworks.

Recursos de Comunicação: Ferramentas de Comunicação, Plataforma de Colaboração.

Recursos de Marketing : Material de Marketing, Estratégia de Marketing.

- **Cronograma Preliminar**

Lista de atividades: Uma lista inicial das tarefas que devem ser realizadas durante o projeto.

Duração estimada: Uma estimativa do tempo necessário para completar cada atividade.

Sequência de atividades: Uma ordem aproximada em que as atividades devem ser realizadas, mostrando suas dependências e inter-relações.

Milestones (marcos): Pontos de verificação ou eventos significativos ao longo do projeto que ajudam a medir o progresso.

Recursos preliminares: Uma indicação inicial dos recursos necessários para executar as atividades do projeto.

- **Riscos Iniciais**

Escopo Mal Definido: Se os requisitos do jogo não forem claramente definidos desde o início, pode levar a mudanças frequentes no projeto, aumentando os custos e o tempo de desenvolvimento.

Problemas de Tecnologia: Dificuldades técnicas, como a escolha de uma plataforma inadequada, problemas de desempenho ou compatibilidade com dispositivos, podem atrasar o progresso do jogo.

Equipe Inexperiente: Se a equipe de desenvolvimento não tiver experiência suficiente com o tipo de jogo que estão criando, isso pode resultar em dificuldades na implementação de recursos ou na resolução de problemas técnicos.

Orçamento Limitado: Restrições financeiras podem limitar a capacidade da equipe de adquirir recursos necessários, como software, ferramentas ou assets, o que pode afetar a qualidade final do jogo.

Concorrência de Mercado: Se o jogo estiver entrando em um mercado saturado ou altamente competitivo, pode ser difícil atrair a atenção dos jogadores e alcançar o sucesso comercial.

Gestão de Tempo: Subestimar o tempo necessário para desenvolver o jogo ou não ter um plano de projeto sólido pode resultar em atrasos significativos no lançamento.

Problemas de Escala: Se o jogo for bem-sucedido e a base de jogadores crescer rapidamente, podem surgir desafios relacionados à escalabilidade do servidor, suporte ao cliente e manutenção contínua do jogo.

- **Critérios de Sucesso**

Mediremos o sucesso do projeto analisando os dados analíticos referente a sessão de cada usuário, verificando seu engajamento, retenção e avaliações.

Priorizaremos os seguintes aspectos:

Engajamento do Usuário: Manter uma interação cada vez maior com o usuário.

Facilidade de Uso: Usar uma interface simples e amigável.

Aprendizado Efetivo: Ensinar de forma prática e direta programação.

Diversão e Entretenimento: Juntar a linha de aprendizado com um jogo, tornando essa jornada mais fácil e prática.

Feedback Positivo: Receber retornos positivos de quem está usando o aplicativo.

Retenção de Usuários: Elevar o tempo de sessão da pessoa no app.

Avaliações e Classificações: Receber avaliações e corrigir os problemas detectados, melhorando cada vez mais.

Conclusão de Metas e Desafios: Identificar possíveis falhas ou problemas e corrigi-los.

Compatibilidade e Desempenho: Manter uma alta compatibilidade do app, tanto para computadores via web quanto para dispositivos mobile.

Comunidade Engajada: Criar e manter uma comunidade de pessoas que utilizam o app, aumentando o aprendizado.

- **Distribuição e Comunicação**

Um dos grandes desafios, ao criar um aplicativo, é conseguir usuários. Portanto, para alcançar a maior parte do público alvo, investiremos em tráfego orgânico e tráfego pago.

No tráfego orgânico, vamos apostar na divulgação por parte de grandes influencers da área como Léo Andrade, Fiasco, Curso em Vídeo, Código Fonte TV e Lucas Montato. Todos são extremamente conhecidos e referência no mundo da programação.

Já no tráfego pago, focaremos em anúncios do Google Ads com o público alvo sendo do sexo masculino, de 15 a 35 anos cujo os interesses de pesquisa envolvem o campo da informática, cybersegurança e desenvolvimento web e mobile.

**REQUISITOS**

Página de Boas-Vindas: Uma página inicial acolhedora que apresenta a proposta do site e orienta os visitantes sobre como começar. Com um bom contraste de cores para facilitar a leitura através de ferramentas de identificação de contraste.

Separação Visual Prática: Seções dedicadas a diferentes linguagens de programação (como Python, JavaScript, HTML/CSS, etc.), com conteúdo estruturado em lições e módulos.

Exemplos Práticos: Fornecer exemplos de código prático que os usuários possam executar e experimentar diretamente no navegador ou aplicativo.

Desempenho: Garantir que o site/app carregue rapidamente, mesmo em conexões de internet mais lentas, para uma experiência de usuário otimizada. Testar em diferentes conexões (Wi-Fi, Cabo Ethernet, 3G, 4G e 5G).

Segurança: Implementar medidas de segurança robustas para proteger os dados dos usuários, como criptografia SSL e práticas de segurança recomendadas.

Escalabilidade: Projetar o site e o aplicativo de forma que possa lidar com um grande número de usuários simultâneos sem comprometer o desempenho. Testar em diferentes dispositivos e plataformas.

Compatibilidade com Navegadores: Certificar-se de que o site seja compatível com os principais navegadores da web, como Google Chrome, Mozilla Firefox, Safari e Microsoft Edge.

Conformidade com Padrões Web: Seguir os padrões e diretrizes recomendados da web, como HTML5, CSS3 e JavaScript ECMAScript.

Conteúdo Multimídia: Incorporar vídeos, imagens e gráficos para tornar o aprendizado mais envolvente e compreensível.

Responsividade: Garantir que o site seja totalmente acessível e funcione bem em dispositivos móveis, tablets e computadores desktop. Testar em diferentes equipamentos, tanto celulares como computadores.

Busca Avançada: Uma função de busca que permita aos usuários encontrar rapidamente conteúdo específico dentro do site.

Acompanhamento do Progresso: Permitir que os usuários registrem seu progresso e vejam quais lições concluíram ou estão em andamento.