

Atividade de avaliação individual 2

Professor Eng. Dr. Gerson Penha

Introdução

Uma interface gráfica de usuário (GUI, do inglês Graphical User Interface) é a forma de interação entre o ser humano e o computador baseada em elementos visuais, como janelas, ícones, menus e botões, que podem ser manipulados por dispositivos de entrada como o mouse ou a tela sensível ao toque. Diferente das interfaces de linha de comando, onde o usuário precisa digitar instruções em texto, a GUI permite que ações sejam realizadas de forma mais intuitiva, usando representações gráficas para comandos e informações. Essa abordagem revolucionou a computação pessoal, tornando-a mais acessível para pessoas sem conhecimento técnico avançado.

A história da primeira interface gráfica começa no final da década de 1960. Em 1968, Douglas Engelbart apresentou a famosa "Mother of All Demos", uma demonstração que já incluía conceitos como o uso do mouse, janelas na tela, hiperlinks e edição colaborativa de texto. Embora não fosse ainda uma GUI como conhecemos, esse evento foi a base para o desenvolvimento futuro de interfaces gráficas completas.

Foi no início dos anos 1970, no centro de pesquisa Xerox PARC (Palo Alto Research Center), que a primeira interface gráfica funcional de fato foi criada. Em 1973, os pesquisadores desenvolveram o Xerox Alto, um computador experimental que introduziu a interação por janelas sobrepostas, ícones clicáveis e menus, tudo controlado por um mouse. O Xerox Alto não foi vendido comercialmente em larga escala, mas influenciou profundamente a indústria.

Em 1981, a Xerox lançou o Xerox Star, o primeiro computador a comercializar uma GUI. Apesar de trazer conceitos inovadores, como a "área de trabalho" (desktop), pastas e metáforas visuais para documentos, seu preço elevado limitou a adoção. No entanto, suas ideias inspiraram empresas como a Apple, que, em 1983, lançou o Apple Lisa e, no ano seguinte, o Macintosh, popularizando definitivamente a interface gráfica entre consumidores domésticos e empresas.

Assim, o marco histórico da primeira GUI funcional ocorreu em 1973, com o Xerox Alto, enquanto a primeira GUI comercial surgiu em 1981, com o Xerox Star. A partir dessas criações, a interface gráfica evoluiu continuamente, tornando-se um padrão indispensável na computação moderna.

Com a difusão da internet o conceito de GUI ganhou outro aspecto. Os profissionais passaram a desenvolver interfaces para serem executadas por softwares conhecidos como navegadores da internet (web browser). A partir daí o profissional que trabalha no desenvolvimento de interfaces para internet foi denominado de web designer.

O web designer é o profissional responsável por planejar, criar e organizar a parte visual e estrutural de um site, unindo estética e funcionalidade para oferecer uma boa experiência ao usuário. Seu trabalho envolve escolher cores, tipografias, layouts, imagens e elementos interativos, sempre levando em consideração a identidade da marca e a usabilidade da página. Em essência, o web designer atua como o "arquiteto visual" da internet, transformando ideias em interfaces atrativas e funcionais.

A profissão nasceu nos anos 1990, acompanhando o crescimento da World Wide Web. No início, os sites eram simples e compostos principalmente por textos e links, com pouca preocupação estética. Ainda assim, havia a necessidade de alguém que estruturasse a informação e aplicasse os recursos visuais disponíveis da época, como tabelas, imagens em baixa resolução e cores básicas. O web designer precisava conhecer HyperText Markup Language (HTML) e, posteriormente, Cascading Style Sheets (CSS), para transformar o design em páginas navegáveis.

Na virada para os anos 2000, a internet se popularizou rapidamente, e os sites passaram a exigir mais atratividade e interatividade. Nesse período, o web designer assumia quase todo o processo criativo e técnico: fazia o layout, codificava o HTML, aplicava estilos com CSS e, muitas vezes, incluía scripts em JavaScript para funções interativas. Ferramentas como o Macromedia Flash também se tornaram comuns, permitindo animações e efeitos visuais elaborados. A profissão exigia versatilidade e domínio tanto da parte gráfica quanto do desenvolvimento básico.

A partir da década de 2010, com o crescimento dos dispositivos móveis e a necessidade de sites responsivos, o papel do web designer começou a se transformar. Surgiram especializações como UI designer (design de interface) e UX designer (design de experiência do usuário), que dividiram responsabilidades. Ao mesmo tempo, ferramentas como WordPress, Wix e Webflow permitiram que pessoas sem conhecimento avançado de código criassem sites prontos, o que reduziu a demanda por web designers em trabalhos simples. No entanto, aumentou a procura por profissionais capazes de criar experiências únicas, personalizadas e alinhadas a estratégias de marca.

Hoje, o web designer moderno precisa dominar não apenas os fundamentos do design visual, mas também compreender conceitos de usabilidade, acessibilidade, design responsivo e até Search Engine Optimization (SEO). Em muitos casos, é esperado que saiba trabalhar com softwares de prototipagem, como Figma ou Adobe XD, e colaborar diretamente com desenvolvedores front-end. Assim, a profissão permanece viva, mais integrada a um ecossistema multidisciplinar, onde criatividade e conhecimento técnico caminham lado a lado para atender às exigências de um público.

Contextualização

Criar uma interface gráfica para um sistema web hoje é um processo que vai muito além de simplesmente “desenhar telas bonitas”. Envolve pesquisa, planejamento, testes e colaboração entre diferentes profissionais. O objetivo é criar uma experiência visual que seja agradável, funcional e acessível, facilitando ao máximo a interação do usuário com o sistema. Esse processo pode variar conforme o tipo e a complexidade do projeto, mas há um conjunto de etapas e boas práticas amplamente adotadas no mercado.

O primeiro passo é a definição de requisitos. Antes de abrir qualquer software de design, é fundamental entender o que o sistema precisa fazer e quem vai utilizá-lo. Isso inclui reuniões com o cliente ou a equipe de produto para mapear funcionalidades, compreender os objetivos de negócio e identificar o perfil dos usuários. Nesta fase, também são levantadas informações sobre concorrentes, tendências de mercado e restrições técnicas, como frameworks e bibliotecas que serão utilizadas.

Em seguida, vem a etapa de pesquisa e prototipagem inicial. Aqui, o designer começa criando wireframes — representações simples e sem cores que mostram a estrutura e a organização dos elementos na tela. O objetivo é testar a lógica da interface antes de investir tempo no detalhamento visual. Essa fase também pode incluir testes de usabilidade rápidos com usuários reais para validar se a navegação e a disposição das informações fazem sentido.

Depois, parte-se para o design visual propriamente dito. É o momento de aplicar cores, tipografias, ícones, imagens e estilos que transmitam a identidade da marca e reforcem a mensagem do sistema. Nesta fase, ferramentas como Figma, Adobe XD ou Sketch são amplamente utilizadas para criar protótipos interativos, permitindo que a equipe simule a navegação antes da codificação. Boas práticas importantes incluem garantir contraste suficiente para acessibilidade, manter consistência nos elementos visuais e respeitar princípios de design responsivo, para que a interface funcione bem em diferentes tamanhos de tela.

Com o design aprovado, inicia-se a implementação em código, geralmente feita por desenvolvedores front-end. Aqui, o trabalho do web designer ou do UI designer é fundamental para fornecer especificações detalhadas, como medidas, espaçamentos, estilos e estados de interação (por exemplo, como um botão muda ao ser clicado). O uso de design systems e bibliotecas de componentes reutilizáveis ajuda a manter a consistência e acelerar o desenvolvimento.

Por fim, ocorre a fase de testes e ajustes. A interface é avaliada em diferentes navegadores e dispositivos para garantir que tudo funcione como esperado. Testes de usabilidade mais aprofundados podem revelar problemas que passaram despercebidos na prototipagem. Além disso, o sistema é monitorado após o lançamento para identificar oportunidades de melhoria, já que a interface não é um produto estático, mas sim algo que deve evoluir com as necessidades do usuário e do negócio.

Seguir esse fluxo — da pesquisa ao teste final — não é uma fórmula rígida, mas uma sequência lógica de boas práticas que aumenta a chance de criar interfaces web eficientes, agradáveis e duradouras. Hoje, mais do que nunca, o design de interfaces exige uma abordagem centrada no usuário, alinhada com a estratégia do produto e sustentada por colaboração entre designers, desenvolvedores e demais consumidores ou donos do produto.

Atividade:

A primeira versão do sistema para gestão da produção de aeronaves foi um sucesso. O sistema foi comercializado e atualmente está em uso em duas fabricas de aviões de pequeno porte, com sedes na Ásia e Europa. Com a entrega desses dois clientes na carteira de clientes da empresa houve um aumento exponencial de receita, que irá permitir investimentos, melhorias e contratação de mais funcionários. Sua empresa está crescendo rápido. Os clientes e usuários mais empolgados com o sistema foram os engenheiros de produção e engenheiros aeronáuticos, que são os principais funcionários responsáveis pela gestão da produção das aeronaves.

O novo objetivo é expandir e oferecer o sistema para grandes empresas como Boeing, Airbus, Embraer e a Bombardier. Além delas, você também tem interesse em empresas

focadas na produção de aeronaves de defesa e outras especialidades, empresas como a Lockheed Martin, a Dassault Aviation, a BAE Systems e a Gulfstream. Contudo, há um problema com o sistema, ele é do tipo CLI.

Você percebeu que GUIs baseados na web, combinados com arquiteturas centralizadas e gerenciadas por nuvem, estão ultrapassando a CLI como interface preferencial. Na verdade, as GUIs estão começando a ficar tão boas que a gestão por CLI pode parecer antiquada. Também, com as plataformas de rede modernas de hoje, a GUI não fica mais em segundo plano com relação à CLI em muitos casos. Assim, por parte das empresas que fazem software, mais tempo e esforço têm sido gastos em facilidade de uso, a capacidade de lançar centralizadamente atualizações de configuração para vários dispositivos e criar uma experiência de administração de usuário que fornece um equilíbrio entre simplicidade e profundidade de recursos de configuração.

A GUI tende a ser mais fácil de usar. Você recebeu uma opinião de um dos engenheiros que passou a utilizar o sistema da Aerocode e percebeu que com linhas de comando, há uma curva de aprendizado mais longa antes de um funcionário se tornar proficiente e começar a utilizar o sistema diariamente. Com uma GUI bem projetada, no entanto, essa mesma curva de aprendizado pode ser drasticamente reduzida.

Dado as informações recebidas e sua percepção quanto ao uso de uma GUI, você decidiu contratar uma consultoria especializada, uma empresa de marketing e desenvolvimento de marcas e identidade visual. Após uma análise da situação atual da Aerocode, de seus clientes e de futuros clientes, a empresa de marketing lhe entregou um relatório com algumas sugestões para o desenvolvimento da primeira GUI do sistema de gestão da produção de aeronaves. A primeira informação que lhe despertou interesse foi sobre uma forma de desenvolvimento de GUI chamada de aplicação de página única.

A empresa de marketing lhe explicou que, atualmente, é possível afirmar que uma grande parte das aplicações web modernas é desenvolvida sob o conceito de aplicação de página única, ou SPA (Single Page Application), embora não seja a maioria absoluta. Esse modelo se popularizou especialmente em sistemas que demandam interatividade constante e respostas rápidas, como plataformas SaaS, dashboards administrativos, redes sociais e aplicativos de e-commerce. No entanto, aplicações tradicionais baseadas em múltiplas páginas (MPA – Multi Page Application) ainda são amplamente utilizadas, principalmente em sites institucionais, portais de notícias e sistemas que priorizam SEO ou carregamento rápido de conteúdo estático.

Além disso, com o avanço de frameworks como React, Vue.js e Angular, as SPAs se tornaram extremamente populares, especialmente em sistemas que exigem interação rápida e dinâmica.

Também foi explicado que uma aplicação de página única não segue exatamente o modelo tradicional MVC (Model–View–Controller) que era predominante em frameworks de back-end como Ruby on Rails, Laravel ou ASP.NET MVC. Em vez disso, as SPAs tendem a adotar padrões mais adequados ao funcionamento no navegador. Alguns exemplos são o MVVM (Model–View–ViewModel), muito comum no Angular e no Vue.js, e padrões de fluxo de dados unidirecional, como Flux ou Redux, usados em React. Outra característica importante é a arquitetura baseada em componentes, em que cada

parte da interface encapsula sua lógica, estilo e apresentação. Nesse cenário, o backend geralmente atua como fornecedor de dados por meio de APIs REST ou GraphQL, deixando toda a lógica de apresentação e navegação sob responsabilidade do frontend.

As aplicações de página única apresentam diversas vantagens. Uma das principais é a experiência de uso mais fluida, já que a navegação não exige recarregar a página inteira: apenas partes específicas da interface são atualizadas, proporcionando maior velocidade e responsividade. Além disso, após o carregamento inicial, o tráfego com o servidor se limita a troca de dados (normalmente em formato JSON), reduzindo a sobrecarga e aumentando a eficiência. Outro benefício é a reutilização de componentes, que facilita a manutenção e acelera o desenvolvimento de novas funcionalidades. Por fim, SPAs se integram facilmente a conceitos como PWA (Progressive Web Apps), permitindo que aplicações web tenham comportamento próximo ao de aplicativos nativos, com recursos como funcionamento offline e notificações push.

Juntamente com o relatório e explicações sobre o desenvolvimento de GUI como uma SPA, a empresa marketing lhe enviou um orçamento para o desenvolvimento da interface do sistema de gestão da produção de aeronaves - o desenvolvimento seria feito por um de seus parceiros comerciais. Mas, você percebeu um detalhe muito importante, este parceiro comercial também desenvolve sistemas para seus concorrentes e, inclusive, já esteve envolvido em escândalos sobre transferência de informações confidenciais e espionagem. Por isso, você não fechou o negócio.

Com medo de ser vítima de espionagem e pensando na importância que este novo passo representa para as futuras negociações da Aerocode, você decidiu desenvolver a primeira GUI do sistema sozinho. Afinal, você detém todo o conhecimento necessário para o desenvolvimento e é o principal interessado que o trabalho seja feito da melhor forma possível.

O primeiro passo a ser feito é a construção do wireframe do projeto, da GUI. Neste passo, você irá elaborar todas as imagens que compõe o wireframe da GUI. Você deve lembrar que o objetivo principal do wireframe é servir como um guia para a organização do conteúdo e das funcionalidades, ajudando a alinhar expectativas entre designers, desenvolvedores e stakeholders antes da fase de design detalhado e implementação.

Você irá escrever um relatório, um documento PDF, com as imagens que você elaborou, a descrição dos requisitos e entendimento do público-alvo e dos objetivos do projeto. Além disso, você irá descrever o fluxo de navegação e a hierarquia de informações, determinando quais elementos devem aparecer e em que ordem. Seu relatório deve conter imagens como os exemplos das Figura 1 e Figura 2.

A Figura 1 apresenta um exemplo de um wireframe de baixa qualidade. Um wireframe de baixa qualidade (ou low-fidelity wireframe) é uma versão simples e pouco detalhada de um esboço de interface, usada para representar a estrutura e a organização básica de um layout sem se preocupar com aparência visual final. Ele geralmente é composto apenas por formas geométricas (caixas, linhas, círculos), texto genérico (Lorem Ipsum ou rótulos simples) e indicações básicas de elementos como botões, menus e imagens. Não há uso de cores complexas, tipografias elaboradas ou imagens reais — o objetivo é focar na funcionalidade e disposição dos elementos, não no design estético.

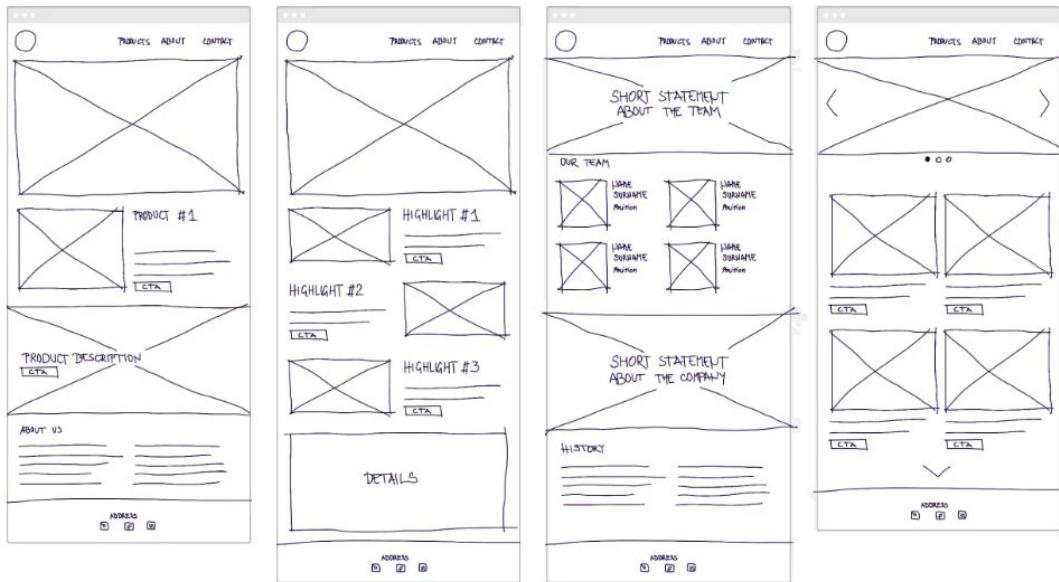


Figura 1. Wireframe de baixa qualidade

A Figura 2 apresenta um exemplo de wireframe de fluxo de usuário (user flow). Um wireframe do tipo user flow é um tipo de representação visual que combina dois conceitos: o wireframe (esboço estrutural de telas) e o fluxo do usuário (user flow, ou o caminho que a pessoa percorre dentro de um sistema para realizar uma tarefa).

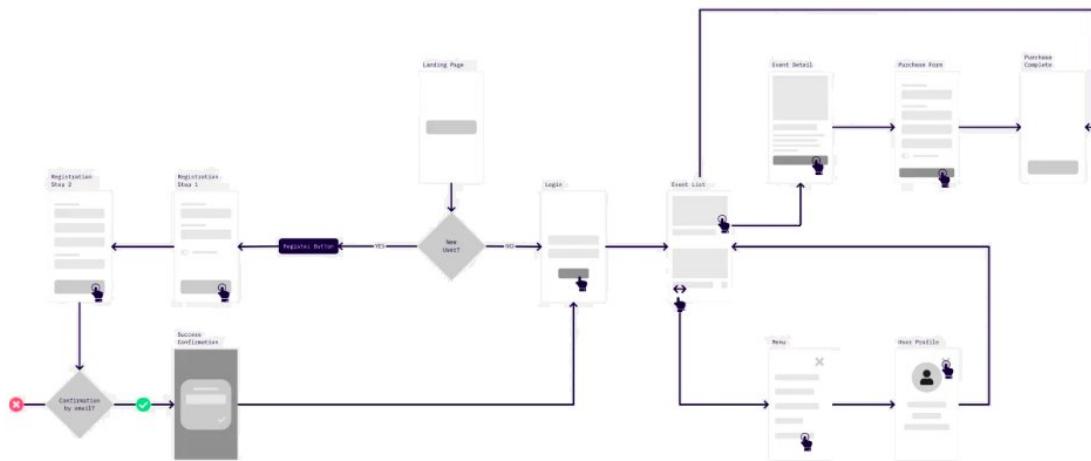


Figura 2. Wireframe fluxo de usuário

Enquanto um wireframe comum mostra a aparência e disposição de elementos em uma tela, o wireframe de user flow também evidencia a sequência e as interações entre essas telas. Ele é usado para mapear passo a passo como o usuário navega pela aplicação, desde o ponto de entrada até o objetivo final.

Em um wireframe de user flow, cada tela ou etapa é representada por um rascunho (em baixa ou média fidelidade), e as conexões entre elas são indicadas por setas ou linhas que mostram as ações do usuário ou transições do sistema. Isso permite visualizar não

apenas como cada página será, mas também como elas se conectam e em que ordem são acessadas.

Após a etapa de construção do wireframe e relatório seu próximo passo será o desenvolvimento da aplicação SPA. Para esta etapa o você deve utilizar como framework principal o React.

O React é um framework (na verdade, mais precisamente uma biblioteca JavaScript) focado na criação de interfaces de usuário, especialmente para aplicações web e mobile. Ele foi criado pelo engenheiro Jordan Walke, do Facebook, e lançado oficialmente em 2013 como um projeto de código aberto. A motivação para o seu desenvolvimento surgiu da necessidade de tornar mais eficiente a atualização e renderização de elementos na interface quando os dados mudam, algo que nas soluções tradicionais da época era custoso e pouco escalável.

O React introduziu conceitos inovadores, como o DOM Virtual (Virtual DOM), que permite atualizar apenas os elementos que realmente mudaram na página, em vez de recarregar toda a estrutura. Isso resulta em melhor desempenho e experiência para o usuário. Além disso, ele trouxe a ideia de componentes reutilizáveis, pequenas unidades de interface independentes que podem ser combinadas para formar interfaces complexas. Essa abordagem modular facilita a manutenção do código e promove a reutilização, acelerando o desenvolvimento.

A popularidade do React cresceu rapidamente por vários motivos. Primeiro, por ter sido adotado e mantido por uma grande empresa como o Facebook (agora Meta), o que garantiu confiança e suporte contínuo. Segundo, por sua comunidade ativa e o vasto ecossistema de bibliotecas e ferramentas complementares. Outro fator decisivo foi a sua flexibilidade, pois o React pode ser usado tanto em projetos pequenos quanto em grandes sistemas corporativos, integrando-se facilmente com outras tecnologias. Além disso, o fato de poder ser usado também para aplicações móveis por meio do React Native ampliou ainda mais seu alcance.

Hoje, o React é uma das tecnologias mais utilizadas no desenvolvimento front-end e continua evoluindo com recursos modernos, como hooks e server components, mantendo-se relevante e competitivo no mercado. Ele é escolhido por equipes de desenvolvimento no mundo todo não apenas pela performance, mas também pela clareza e escalabilidade que proporciona ao criar interfaces ricas e interativas.

Por fim, nesta segunda etapa, somente deve ser desenvolvida a aplicação front-end, sem um back-end, como um protótipo navegável e que deve funcionar em servidores das seguintes plataformas: Windows 10 ou superior, Linux Ubuntu 24.04.03 ou superior e distribuições Linux derivadas do Ubuntu.