Experiência do usuário - DSM - Professora Lucineide - 29/11/25

Nome: Andressa Stéphane Toledo da Silva

Tema: Segurança Digital – Experiência do Usuário em Aplicativos Seguros

1 – Descoberta, Mapa de Expectativas e Priorização (Aulas 1 a 3)

Uma equipe está criando um aplicativo de segurança digital para autenticação multifatorial (MFA) destinado ao público geral.

Durante entrevistas, usuários relataram:

> "Não entendo o que preciso fazer quando recebo um código."
>
> "Tenho medo de aprovar algo errado."
>
> "Quero mais segurança, mas sem complicações."

Explique como você conduziria:

a) Mapa de Expectativas,

Sentimentos Desejados: Confiança, Controle, Rapidez, Tranquilidade e Segurança.

Funcionalidades Essenciais: Notificação segura clara e objetiva, Orientação para usuários leigos, Aprovação de dois fatores quando solicitado.

Frustrações a Evitar: Não entender o código/solicitação, medo de clicar errado, complexidade no fluxo.

Indicadores de Sucesso: Alta taxa de aprovação correta, baixa taxa de erros acidentais, tempo de interação baixo, garantindo um sistema eficiente e confiável.

Requisitos de UX: Texto legível e claro, ícones intuitivos de aprovar/negar, feedback visual imediato de sucesso/falha/outros.

b) Priorização MoSCoW

Must Have: Notificação (push) que transmita confiabilidade, contendo detalhes do acesso; Aprovação de dois fatores para garantir segurança ao usuário.

Should Have: Orientação ao usuário para prevenir falhas de entendimento.

Could Have: Modo noturno.

Won't Have: Configuração de alertas personalizados.

c) e como esses dois artefatos influenciariam o início do projeto.

Ambos os artefatos são essenciais para a fase de **Descoberta e Definição** de um projeto, segundo as práticas de UX, etapa em que ocorre o levantamento do problema, a análise de experiências anteriores e a compreensão das necessidades e dores dos usuários.

O **Mapa de Expectativas** é especialmente importante no início do desenvolvimento de um aplicativo de segurança digital para autenticação multifatorial (MFA). A partir dele, é possível identificar o que o usuário espera do sistema, por exemplo, que seja confiável, claro, seguro e simples de utilizar. Também permite mapear dores relevantes, como receios durante aprovações ou frustrações que precisam ser evitadas. Esse entendimento garante que o produto proporcione uma experiência satisfatória, aumentando as chances de o usuário continuar utilizando o aplicativo, recomendá-lo a outras pessoas e fornecer feedbacks valiosos para evolução contínua do projeto.

O **Método MoSCoW** complementa o Mapa de Expectativas, pois, com base nesse levantamento inicial, torna-se possível definir as prioridades do projeto: o que é o **core** do sistema (o que deve ser desenvolvido primeiro), quais funcionalidades são importantes como refinamentos e o que seria apenas um diferencial, sem impacto direto na usabilidade. Assim, o MoSCoW ajuda a estruturar um desenvolvimento mais estratégico e eficiente, garantindo que o sistema seja confiável, completo e alinhado aos objetivos e expectativas do usuário.

2 – Personas e Jornada do Usuário (Aulas 2 e 3)

Considere estes dois perfis reais de usuários:

- Usuário A: estudante, utiliza redes sociais intensamente, nunca configurou segurança avançada.
- Usuário B: profissional de finanças, possui preocupação elevada com fraudes.

a) Crie as personas completas (objetivos, dores, barreiras e comportamentos).

Usuário A: Pedro Henrique, 22 anos

Objetivo: Acessar contas rapidamente em vários dispositivos e ter praticidade no uso diário

Frustração: Frustração quando precisa lembrar múltiplas senhas ou passar por várias etapas.

Tecnologia: Celular moderno, internet boa.

Citação: "Só quero que funcione rápido, não tenho paciência para coisas muito burocráticas."

Usuário B: Thiago, 45 anos

Objetivo: Segurança para evitar golpes/fraudes e clareza em qual conta está sendo notificada para evitar confusão e falha.

Frustração: Informação incompleta/ambígua no alerta, e realizar aprovação de forma errônea.

Tecnologia: Celular e notebook de alta qualidade e internet excelente.

Citação: "Preciso de certeza absoluta antes de aprovar qualquer coisa. O risco é alto."

b) Construa a Jornada do Usuário para a tarefa “Aprovar login seguro em outro dispositivo”.

Etapa: Antes do alerta

Ações do usuário: Inicia um login em um novo dispositivo e aguarda a confirmação pelo aplicativo.

Sentimentos: Expectativa e desejo de rapidez e atenção e necessidade de controle sobre o processo.

Pontos de dor: Incômodo com demora na notificação e insegurança caso o usuário não entenda por que a verificação é necessária.

Oportunidades de melhoria: Enviar notificações rápidas e claras assim que a tentativa de login for detectada
e exibir uma explicação breve sobre o motivo da verificação.

Etapa: Durante o alerta

Ações do usuário: Recebe a notificação → Abre o aplicativo → Visualiza dispositivo solicitante, localização, data e hora → Decide entre “Aprovar” ou “Negar”.

Sentimentos:

Pedro: Impaciente.
Thiago: Cauteloso.

Pontos de dor: Informações confusas ou insuficientes no alerta; Interface poluída ou com muitos passos; Falta de clareza sobre qual conta ou dispositivo está pedindo acesso.

Oportunidades de melhoria: Exibir informações essenciais de forma organizada e fácil de entender; Utilizar um design limpo e direto, adaptado para diferentes níveis de experiência.

Etapa: Depois do alerta

Ações do usuário: Recebe o feedback da ação tomada → Segue com o login caso tenha aprovado → Analisa atividades recentes caso tenha negado

Sentimentos:

Pedro: Alívio.

Thiago: Seguro.

Pontos de dor: Feedback demorado ou pouco claro; Falta de orientação em caso de tentativa suspeita;Incerteza sobre próximos passos após negar uma solicitação.

Oportunidades de melhoria: Exibir mensagens objetivas e imediatas (“Login aprovado”, “Acesso bloqueado”); e disponibilizar histórico de acessos para revisão rápida.

c) Explique como a experiência para A e B deve ser distinta e por quê.

A experiência deve ser diferente porque cada usuário tem necessidades distintas. **Usuário A** busca rapidez e simplicidade, então o fluxo precisa ser direto, com poucas informações e botões claros para aprovar rapidamente. Já o **Usuário B** prioriza segurança e precisão, por isso precisa de mais detalhes visíveis na tela, confirmação adicional e informações completas para tomar uma decisão segura.

3 – Fluxos de Usuário em Segurança Digital (Aulas 4 e 5)

O aplicativo precisa permitir que o usuário:

“Aprove um acesso à sua conta através de uma notificação segura no celular.”

Ele deve funcionar mesmo com:

- rede lenta
- mudanças de dispositivo
- múltiplas contas cadastradas

Descreva um fluxo de usuário completo, incluindo:

- fluxo principal,
- fluxos alternativos,
- cenários de erro,
- e como garantir clareza e minimização de riscos ao usuário.

Fluxo principal:

[Início]

↓

[Receber notificação de tentativa de login]

↓

[Abrir o aplicativo de segurança]

↓

[Visualizar detalhes do dispositivo solicitante]

↓

[Clicar em “Aprovar”]

↓

[Receber confirmação de sucesso]

↓

[Fim]

Fluxo alternativo:

Cenário: Usuário com múltiplas contas cadastradas

[Notificação recebida]

↓

[Abrir o app]

↓

(Decisão: Qual conta está pedindo acesso?)

→ [Seleciona conta A]

→ [Seleciona conta B]

→ [Seleciona outra conta]

↓

[Visualizar detalhes]

↓

[Aprovar]

↓

[Receber confirmação de sucesso]

↓

[Fim]

Cenário: Mudança de dispositivo (novo celular)

[Usuário tenta aprovar]

↓

[Solicita verificação adicional para garantir identidade]

↓

[Usuário confirma via PIN, biometria ou e-mail]

↓

[Acessa o alerta]

↓

[Aprova]

↓

[Receber confirmação de sucesso]

↓

[Fim]

Cenário: Rede lenta

[Usuário abre o alerta]

↓

[App carrega informações essenciais primeiro]

↓

[Mostrar “carregando detalhes…”]

↓

[Usuário aprova após os dados forem carregados]

↓

[Receber confirmação de sucesso]

↓

[Fim]

Cenário: Erro

Caso 1: Detalhes não carregam (rede muito lenta)

- Aplicativo mostra mensagem: “Não foi possível carregar todas as informações. “
- Opções para usuário: Tentar novamente / Negar acesso por segurança.

Caso 2: Notificação expirada

- Aplicativo mostra mensagem: “Esse pedido de acesso expirou. Por favor, tente fazer login novamente.”

Caso 3: Dispositivo não reconhecido

- Aplicativo emite alerta: “Solicitação suspeita detectada.”
- Opções: Negar imediatamente e analisar histórico com localização.

Caso 4: Conflito entre contas

- Usuário possui várias contas e não consegue identificar qual conta pediu acesso.
- Aplicativo mostra mensagem: “Mais de uma conta foi identificada, selecione a conta correta antes de aprovar a solicitação.”

Cenário: Como garantir clareza e minimizar riscos

- Exibir informações essenciais de forma imediata: dispositivo, localização aproximada, horário e conta solicitante.
- Botões de ação bem destacados: “Aprovar” em verde, “Negar” em vermelho.
- Detalhes adicionais acessíveis: para usuários mais cautelosos, um botão “Ver mais detalhes”.

- Feedback rápido e direto: “Acesso aprovado” ou “Tentativa bloqueada”.
- Segunda camada de verificação em situações de risco: novo dispositivo, país diferente, horário incomum.
- Histórico de tentativas visível: aumenta a confiança e permite revisão.
- Mensagens claras em rede lenta: sempre priorizar carregamento das informações mínimas.

4 – Prototipação e Heurísticas em Segurança (Aulas 5, 6 e 7)

Durante testes com três versões da tela de “Autorização de Acesso”, usuários relataram:

- falta de clareza sobre qual dispositivo está pedindo acesso;
- medo de confirmar algo indevido;
- dúvida entre botão “Negar” e “Bloquear”.

Explique:

a) como cada nível de prototipação ajuda a identificar esses problemas;

Caso 1: Baixa Fidelidade (Wireframe)

- Ajuda a identificar problemas de entendimento básico, como a falta de clareza sobre qual dispositivo está pedindo acesso.
- Usuários conseguem dizer rapidamente: “Não está claro o que é isso”, antes mesmo de analisar a estética do aplicativo.
- Foca exclusivamente no conteúdo, hierarquia visual e lógica da interface.

Caso 2: Média Fidelidade

- Permite testar fluxos e rotas de interação, identificando dúvidas entre botões como “Negar” e “Bloquear”.
- Mostra se o usuário entende a diferença entre ações sem que o design visual distraia sua interpretação.
- Ajuda a revelar confusão sem interferência do estilo final (cores, ícones, animações).

Caso 3: Alta Fidelidade

- Evidencia problemas relacionados ao medo de confirmar algo de forma errônea, pois simula uma experiência muito próxima ao que será o aplicativo.

- O usuário reage emocionalmente ao alerta, expondo inseguranças e riscos de erro.

- Mostra se o design final reforça segurança, confiança e clareza nos detalhes apresentados.

b) quais heurísticas foram violadas;

Visibilidade do status do sistema
→ A tela não deixa claro qual dispositivo está solicitando o acesso.

Correspondência entre o sistema e o mundo real
→ Termos como "Negar" e "Bloquear" não refletem claramente ações distintas.

Prevenção de erros
→ O usuário sente medo de confirmar algo indevido, indicando risco alto de erro crítico.

Ajuda ao usuário no reconhecimento e tomada de decisão
→ Falta contexto suficiente para decidir com segurança.

c) como corrigir cada violação;

Caso: Falta de clareza sobre o dispositivo

- Mostrar informações essenciais logo no topo: nome do dispositivo, localização aproximada, tipo de navegador e data/hora.
- Usar layout que destaque essas informações.

Caso: Medo de confirmar algo indevido

- Adicionar elementos de confiança: ícone de segurança, mensagem "Confirme apenas se você reconhece este dispositivo" e botão "Ver mais detalhes"

- Opcional: segunda etapa de confirmação em cenários de risco.

Confusão entre "Negar" e "Bloquear"

- Redefinir rótulos para ações claras: "Negar acesso" (recusar apenas esta tentativa) e "Bloquear dispositivo" (impedir futuros acessos)

- Adicionar descrições curtas abaixo de cada botão.

Falta de consistência e prevenção de erros

- Padronizar textos, ícones e ações em todas as telas.

- Garantir que ações de risco sempre tenham confirmação adicional.

d) como registrar as evidências para o relatório final.

Prints e capturas dos testes, ou seja, salvar imagens das telas usadas em protótipos e das interações dos usuários; Transcrições curtas de falas observadas em segredo; Métricas observadas, por exemplo: tempo para tomada de decisão, número de erros ou hesitações e quantidade/identificação de cliques equivocados; Listar os erros, documentar as soluções que foram aplicadas após os testes.

5 – Testes A/B, Prototipagem Rápida e Iteração (Aulas 7 e 8)

Duas abordagens de notificação foram testadas:

- Versão A: mensagem curta e dois botões (“Permitir”, “Negar”).
- Versão B: mensagem com detalhes do dispositivo → localização → horário → botão “Entendi” antes de ação.

Resultados:

- A: 90% aprovação, 40% erros por “aprovação acidental”
- B: 65% aprovação, 3% erros, maior tempo de interação

a) Interprete os resultados.

Versão A tem alta aprovação, mas também alto risco: pois 40% dos cliques foram acidentais, indicando baixa clareza e decisão impulsiva.

Versão B reduz drasticamente os erros, mostrando melhor compreensão e maior segurança, porém aumenta o tempo de interação e reduz aprovações.

Ou seja, a Versão A é rápida mas insegura, e a B é segura mas lenta.

b) Explique quais decisões de design devem ser tomadas.

Manter o nível de detalhes e clareza da Versão B, pois reduz erros críticos. Além de também reduzir os bloqueios desnecessários, removendo etapas que não agregam à decisão. E ter o suficiente de informação para evitar erros, mas menos passos que na versão B, para ter uma eficiência similar ou melhor que a Versão A.

c) Descreva como criar uma versão C usando prototipagem rápida.

- Criar wireframe que demonstre o fluxo, como por exemplo:

[Notificação recebida]

↓

[Usuário abre a notificação]

↓

[Tela de resumo do acesso]

Mostra em uma única linha:

• Dispositivo solicitando acesso

• Localização aproximada

• Horário da tentativa

↓

[Exibir detalhes compactos]

“Ver mais detalhes” (opcional)

↓

(Ação final com botões claros e seguros)

[Botão 1: "Aprovar este dispositivo"]

[Botão 2: "Recusar – não reconheço"]

↓

[Confirmação da ação]

↓

[Fim]

- Criar modelo clicável no Figma ou similar em menos de 30 min.
- Testar com 3–5 usuários para validar clareza e velocidade.
- Implementar Versão C.

d) Como documentar o ciclo de iteração completo para o relatório EU.03?

- Objetivo da iteração – o que estava sendo testado (ex.: reduzir erros de aprovação acidental).
- Versões testadas – descrição breve das telas A, B e C.
- Métricas coletadas – taxas de aprovação, erros, tempo de interação.
- Resultados e aprendizados – o que funcionou, o que não funcionou e por quê.
- Decisão de design – qual versão será adotada e justificativa.

- Próximos passos – ajustes, novos testes ou refinamentos.