Educação Profissional Paulista

Técnico em **Desenvolvimento de Sistemas**

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO

Interfaces com o usuário

Fundamentos de interfaces com o usuário

Componentes de interface de usuário

Código da aula: [SIS]ANO2C1B2S10A3

Mapa da Unidade 3 Componente 1

Interfaces com o usuário

semana 10

Você está aqui!

Fundamentos de interfaces com o usuário

semana 11

Desenvolvimento de interfaces com o usuário

semana 13

Melhoria e otimização de interfaces com o usuário

semana 15

Componentes avançados de interface

semana 16

Acessibilidade e usabilidade

Interfaces com o usuário

Mapa da Unidade 3 Componente 1

Você está aqui!

10

Fundamentos de interfaces com o usuário

Aula 3 – Componentes de interface de usuário

Código da aula: [SIS]ANO2C1B2S10A3

Objetivos da aula

- Identificar componentes de interface de usuário.

Recursos didáticos

- Recurso audiovisual para exibição de vídeos e imagens.
- Lápis e caderno para anotações.

Duração da aula

50 minutos

Habilidades técnicas

- Compreender os conceitos básicos de UI/UX.

Habilidades socioemocionais

- Trabalhar a curiosidade – Explorar diferentes abordagens de UI/UX.

Ponto de **partida**

Imagine que você foi contratado como desenvolvedor mobile por uma empresa de e-commerce que está desenvolvendo um novo aplicativo para gerenciar seus produtos. A equipe de design entregou as telas, mas não explicou como os componentes de interface, como botões, campos de entrada e listas/grids, devem ser implementados para garantir a melhor experiência possível ao usuário. Seu desafio é desenvolver a interface desse aplicativo, certificando-se de que todos os componentes estejam otimizados tanto em termos de performance quanto de usabilidade.

Agora, para iniciar o planejamento da interface, considere as seguintes perguntas:

- Por que é importante garantir a responsividade e o feedback visual ao implementar botões em um aplicativo de e-commerce?
- Quais são os principais cuidados a serem tomados ao implementar campos de entrada em formulários de cadastro de produtos em um sistema de e-commerce?
- Como as listas e as grids podem ser otimizadas para exibir uma grande quantidade de produtos de forma eficiente e responsiva em um aplicativo mobile?

Construindo o **conceito**

Componentes de interface de usuário em programação mobile

No desenvolvimento de aplicações móveis, os componentes de interface do usuário desempenham um papel crucial na experiência final do usuário. Cada componente, seja um botão, um campo de entrada ou uma lista/grid, deve ser implementado com atenção a detalhes técnicos, performance e usabilidade. Nesse conteúdo, vamos focar as aplicações dessas práticas para um sistema de e-commerce de gerenciamento de produtos.

Construindo o **conceito**

Botões (*Buttons*)

Os botões são elementos interativos fundamentais, responsáveis por ações disparadas pelo usuário. No desenvolvimento de interfaces para mobile, o comportamento, a acessibilidade e o desempenho dos botões devem ser otimizados para garantir uma boa experiência.

Construindo o **conceito**

Botões (*Buttons*)

Ciclo de vida

O ciclo de vida de um botão começa desde sua criação na árvore de componentes até sua destruição quando a tela é fechada ou o botão não é mais necessário. A otimização deve levar em conta o uso mínimo de recursos.

Eventos de toque

Os botões reagem a eventos de toque, como onClickListener no Android. É importante tratar esses eventos de forma eficiente para evitar sobrecarga de processamento, especialmente em operações complexas, como adicionar produtos ao carrinho de compras ou processar pagamentos.

Animações e feedback visual

Feedback instantâneo ao toque é essencial em mobile. Animações sutis, como a mudança de cor ou a elevação do botão, podem melhorar a experiência do usuário, porém devem ser leves para não comprometer a performance.

Performance

Para evitar atrasos, é essencial garantir que o código executado no clique seja eficiente. Operações pesadas, como chamadas de API para adicionar um item ao carrinho, devem ser tratadas de forma assíncrona, evitando bloqueios na UI.

Construindo o **conceito**

Exemplo técnico detalhado (Kotlin – Android)

```
Button(
    onClick = {
        // Processamento assíncrono para adicionar o produto ao carrinho
        lifecycleScope.launch {
            val resultado = adicionarAoCarrinhoAsync(produto)
            if (resultado.sucesso) {
                exibirMensagem("Produto adicionado com sucesso!")
            } else {
                exibirMensagem("Erro ao adicionar o produto.")
            }
        }
    },
    modifier = Modifier
        .padding(16.dp)
        .fillMaxWidth(),
    colors = ButtonDefaults.buttonColors(
        backgroundColor = Color.Green,
        contentColor = Color.White
    ),
    shape = RoundedCornerShape(8.dp) // Botão com cantos arredondados
) {
    Text(text = "Adicionar ao Carrinho")
}
```

Construindo
o **conceito**

Exemplo técnico detalhado (Kotlin – Android)

Neste exemplo:

- A operação de adicionar ao carrinho é feita de forma assíncrona, utilizando lifecycleScope.launch, a fim de garantir que a interface não trave durante o processo.
- O botão utiliza cores customizadas e um RoundedCornerShape para melhorar a estética.
- A função adicionarAoCarrinhoAsync() encapsula a lógica de adicionar o item ao carrinho, permitindo que o aplicativo trate de falhas ou sucessos corretamente.

Construindo
o **conceito**

Boas práticas

- **Desabilitar botão durante processamento:** quando o usuário clica em um botão que executa uma ação demorada (como finalização de compra), o botão deve ser temporariamente desativado para evitar cliques múltiplos e processamento duplicado.
- **Tamanho responsivo:** em dispositivos móveis, a área de toque do botão deve ser suficientemente grande para facilitar o toque com o dedo (geralmente, 48x48dp no Android).

Construindo o **conceito**

Campos de entrada (*Text Fields*)

Campos de entrada são essenciais em sistemas de e-commerce, permitindo ao usuário inserir informações, como nome de produto, descrição, preço e quantidade de estoque.

Características técnicas dos campos de entrada:

- **Validação em tempo real:** um dos desafios ao desenvolver campos de entrada em aplicativos móveis é validar os dados enquanto o usuário digita. Por exemplo, no cadastro de um produto, o campo de preço deve aceitar apenas valores numéricos e dentro de um limite especificado.

- **Teclados customizados:** dependendo do tipo de dado esperado, o teclado exibido deve mudar. Campos de preço, por exemplo, devem abrir o teclado numérico, enquanto campos de texto, como nome do produto, devem usar o teclado-padrão.

- **Gestão de foco:** quando o usuário toca em um campo de entrada, o sistema operacional gerencia automaticamente o foco, mas é importante garantir que a navegação entre os campos seja intuitiva, usando teclas de avanço (*Next*) para guiar o usuário.

Construindo
o **conceito**

Exemplo técnico detalhado (Kotlin – Android)

```
TextField(
    value = precoProduto,
    onValueChange = { novoValor ->
        // Validação do valor numérico
        if (novoValor.isNumeric()) {
            precoProduto = novoValor
        } else {
            exibirMensagem("Digite um valor numérico válido.")
        }
    },
    label = { Text("Preço do Produto") },
    placeholder = { Text("Ex: 99.90") },
    modifier = Modifier.fillMaxWidth(),
    keyboardOptions = KeyboardOptions.Default.copy(
        keyboardType = KeyboardType.Number
    ),
    singleLine = true
)
```

Construindo
o **conceito**

Exemplo técnico detalhado (Kotlin – Android)

Neste exemplo:

- O campo de entrada valida dinamicamente se o valor inserido é numérico antes de permitir a atualização da variável precoProduto.
- O KeyboardType.Number exibe um teclado numérico, otimizando a entrada de preços.

Construindo
o **conceito**

Boas práticas

- **Feedback imediato:** ao validar os dados de entrada, o feedback deve ser imediato e claro. Isso melhora a experiência do usuário e minimiza erros.

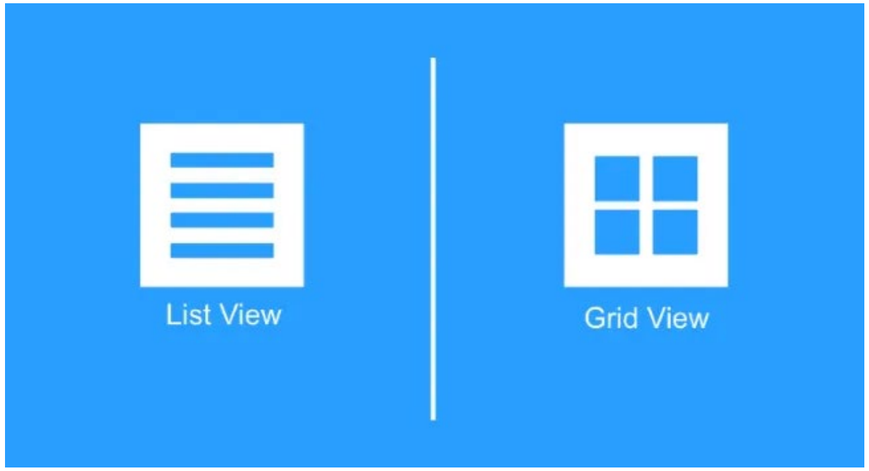
- **Limitação de caracteres:** para evitar entradas excessivas ou incorretas, como no campo de descrição do produto, pode-se limitar o número de caracteres, utilizando maxLength.

Construindo o **conceito**

Listas e grids (*Lists and Grids*)

Listas e grids são componentes fundamentais para exibir grandes quantidades de dados, como produtos em um catálogo de e-commerce.

O desafio é otimizar a renderização de itens para garantir uma navegação suave, mesmo em dispositivos com menos capacidade de processamento.



Reprodução – BABICH, 2016. Disponível em: https://uxplanet.org/mobile-ux-design-list-view-and-grid-view-8f129b56fd5b. Acesso em: 19 nov. 2024.

Construindo o **conceito**

Características técnicas de listas e grids

1

Lazy Loading

Listas longas devem implementar técnicas de carregamento preguiçoso (*lazy loading*) para exibir apenas os itens visíveis e carregar novos conforme o usuário rola a tela. Isso economiza memória e melhora a performance.

2

Renderização de itens

A renderização de cada item em uma lista ou grid deve ser rápida e eficiente. O uso de RecyclerView no Android, por exemplo, é ideal para otimizar o desempenho em listas longas.

3

Interatividade

Cada item em uma lista ou grid pode ter eventos associados, como cliques, para que o usuário possa interagir com produtos diretamente, seja para editar suas informações ou visualizar detalhes.

Construindo o **conceito**

Exemplo técnico detalhado (Kotlin – Android – Lista preguiçosa):

```
LazyColumn {
    items(produtos) { produto ->
        Card(
            modifier = Modifier
                .padding(8.dp)
                .clickable {
                    // Navegar para detalhes do produto
                    navegarParaDetalhesProduto(produto.id)
                }
        ) {
            Column {
                Text(text = produto.nome)
                Text(text = "R$ ${produto.preco}")
            }
        }
    }
}
```

Situação

Desenvolvimento mobile

Você faz parte de uma equipe de desenvolvimento mobile que está criando um sistema de gerenciamento de produtos para um e-commerce. No último ciclo de desenvolvimento, após implementar os botões, os campos de entrada e as listas/grids da interface, você apresentou o trabalho à equipe de design, que forneceu um feedback crítico. Eles mencionaram que a interface não está intuitiva o suficiente e que os usuários podem ter dificuldade ao interagir com alguns componentes, especialmente em dispositivos de telas menores. Você e sua equipe ficaram frustrados, pois acreditavam que estavam seguindo as especificações técnicas de modo correto e que o feedback não era justo.

Como líder técnico, você percebe que a equipe está desmotivada e que a comunicação entre os desenvolvedores e o time de design não está fluindo como deveria. Agora, é sua responsabilidade agir para melhorar o clima da equipe e garantir que a interface seja revisada com sucesso, levando em consideração o feedback.

Situação fictícia produzida pela SEDUC-SP.

Ser
sempre +

Ação

Diante dessa situação, você pode adotar uma abordagem mais colaborativa e reflexiva com sua equipe. Reúna os desenvolvedores e os designers em uma reunião e aborde os seguintes pontos:

- Como você e sua equipe podem lidar de forma construtiva com o feedback recebido do time de design?
- De que forma a empatia pode ajudar a melhorar a colaboração entre desenvolvedores e designers?
- Qual é o papel da responsabilidade compartilhada nesse cenário e como ela pode ser aplicada?

Situação fictícia produzida pela SEDUC-SP.

O que nós **aprendemos hoje?**

Então, ficamos assim...

1. Compreendemos que é importante garantir a responsividade e o feedback visual ao implementar botões em um aplicativo de e-commerce;

2. Entendemos os principais cuidados a serem tomados ao implementar campos de entrada em formulários de cadastro de produtos em um sistema de e-commerce;

3. Descobrimos como as listas e as grids podem ser otimizadas para exibir uma grande quantidade de produtos de forma eficiente e responsiva em um aplicativo mobile.

Saiba mais

Que tal aprendermos mais sobre as ferramentas e o mercado de desenvolvimento Kotlin? Clique no link abaixo e curta o conteúdo.

CÓDIGO FONTE TV. **Chegou a vez do Kotlin em 2024?** (Análise da linguagem ao mercado). Disponível em: https://www.youtube.com/watch?v=0_-6xnOyxSo&t=16s. Acesso em: 19 nov. 2024.

Referências da aula

CÓDIGO FONTE TV. **Chegou a vez do Kotlin em 2024?** (Análise da linguagem ao mercado). Disponível em: https://www.youtube.com/watch?v=0_-6xnOyxSo&t=16s. Acesso em: 19 nov. 2024.

CRONAPP. **Interface gráfica do usuário: quais os tipos e como funciona na prática?**, 31 maio 2022. Disponível em: https://blog.cronapp.io/interface-grafica-do-usuario/. Acesso em: 19 nov. 2024.

GOOGLE FOR DEVELOPERS. **Design para dispositivos móveis**, [s.d.]. Disponível em: https://developer.android.com/design/ui/mobile?hl=pt-br. Acesso em: 19 nov. 2024.

MAZZI, C. DroiDraw: criando interface gráfica para Android. **DevMedia**, 2010. Disponível em: https://www.devmedia.com.br/droidraw-criando-interface-grafica-para-android/17837. Acesso em: 19 nov. 2024.

Identidade visual: imagens © Getty Images.

Orientações ao professor

Slide 6

Orientações:
Professor, a seção **Ponto de partida** aparece no início de cada aula e tem como objetivod ativar o conhecimento prévio dos estudantes sobre o tema e estimular seu pensamento crítico e suas habilidades comunicativas. Por meio de uma situação-problema ou de um exemplo próximo da realidade do estudante, pretende-se sair da abstração conceitual e promover um diálogo dinâmico para explorar hipóteses, soluções e compartilhar eventuais experiências que eles já possam ter com os tópicos a serem abordados na aula. Também é um momento de engajá-los em relação ao tema.

Tempo: 5 minutos

Expectativas de respostas:

Perguntas norteadoras:

1. Por que é importante garantir a responsividade e o feedback visual ao implementar botões em um aplicativo de e-commerce?

Sugestão de resposta: botões são elementos interativos que o usuário utiliza para realizar ações como adicionar um produto ao carrinho ou finalizar uma compra. A responsividade garante que o botão seja suficientemente grande e fácil de clicar, evitando frustrações, enquanto o feedback visual (como mudança de cor ou animações) indica ao usuário que sua ação foi registrada. Sem esses cuidados, a experiência do usuário pode ser prejudicada, resultando em desistências ou cliques múltiplos que sobrecarregam o sistema.

2. Quais são os principais cuidados a serem tomados ao implementar campos de entrada em formulários de cadastro de produtos em um sistema de e-commerce?

Sugestão de resposta: os campos de entrada precisam ser configurados com tipos de dados corretos, como campos de texto para nomes de produto e campos numéricos para preços. A validação em tempo real é fundamental para evitar a inserção de dados incorretos, como valores fora do esperado. Além disso, é importante garantir que o layout do formulário seja intuitivo, com *placeholders* e dicas visuais que orientem o usuário sobre o que deve ser preenchido, além de fornecer feedbacks imediatos em caso de erros.

3. Como as listas e as grids podem ser otimizadas para exibir uma grande quantidade de produtos de forma eficiente e responsiva em um aplicativo mobile?

Sugestão de resposta: para listas e grids que exibem muitos produtos, técnicas, como *lazy loading* (carregamento preguiçoso), são essenciais para garantir que apenas os itens visíveis sejam renderizados, melhorando a performance e evitando o consumo excessivo de memória. Além disso, grids devem ser utilizadas quando se quer exibir mais informações visuais, organizando os itens em colunas, enquanto listas são mais adequadas para uma navegação vertical. Implementar funcionalidades, como *scroll* infinito ou paginação, também contribui para uma melhor experiência do usuário.

Slide 7

Tempo da seção: 25 minutos

Slide 20

Orientações:
A seção **Ser sempre +** tem como objetivo desenvolver e aprimorar as competências socioemocionais dos estudantes, focando especificamente as situações desafiadoras que podem surgir no ambiente profissional.

Tempo: 25 minutos

Gestão de sala de aula:
- Mantenha um ambiente de diálogo aberto e respeitoso.
- Assegure a participação equitativa, promovendo uma discussão inclusiva.
- Reconheça a complexidade do tema e a diversidade de perspectivas que os estudantes podem trazer.
- Forneça feedback construtivo e direcionamento à medida que os estudantes exploram possíveis soluções para o cenário proposto.
- Ajude os estudantes a refinarem suas ideias e a considerarem todas as implicações de suas sugestões.

Expectativas de respostas:

Perguntas para trabalhar a situação:

1. Como você e sua equipe podem lidar de forma construtiva com o feedback recebido do time de design?

Resposta esperada: a equipe deve encarar o feedback como uma oportunidade de melhoria, não como uma crítica destrutiva. É importante adotar uma postura de escuta ativa e entender as preocupações do time de design, buscando soluções que beneficiem o produto final.

2. De que forma a empatia pode ajudar a melhorar a colaboração entre desenvolvedores e designers?

Resposta esperada: a empatia ajuda a equipe de desenvolvimento a entender as necessidades e as preocupações do time de design, que, muitas vezes, está focado na experiência do usuário. Ao colocarem-se no lugar do outro, os desenvolvedores podem ajustar suas implementações para atender melhor a essas necessidades, resultando em um produto mais coerente.

3. Qual é o papel da responsabilidade compartilhada nesse cenário e como ela pode ser aplicada?

Resposta esperada: a responsabilidade compartilhada significa que tanto desenvolvedores quanto designers têm o mesmo objetivo: entregar uma interface funcional e intuitiva. Ao trabalharem juntos e compartilharem a responsabilidade pelo sucesso do projeto, as duas equipes podem colaborar de maneira mais eficaz, revisando a interface e implementando melhorias sugeridas de forma cooperativa.

Slide 22

Orientações:
Professor, a seção **O que nós aprendemos hoje?** tem o objetivo de reforçar e esclarecer os conceitos principais discutidos na aula. Essa revisão pode ser uma ferramenta de avaliação informal do aprendizado dos estudantes, identificando áreas que podem precisar de mais atenção em aulas futuras.

Tempo: 2 minutos

Gestão de sala de aula:
- Mantenha um tom positivo e construtivo, reforçando o aprendizado em vez de focar em correções.
- Seja direto e objetivo nas explicações para manter a atividade dentro do tempo estipulado.
- Engaje os estudantes rapidamente, pedindo confirmações ou reações breves às definições apresentadas.

Condução da dinâmica:
- Explique que esta parte da seção**, Então, ficamos assim...,** é um momento de reflexão e esclarecimento sobre os conceitos abordados na aula.
- Informe que será uma rápida revisão para assegurar que os entendimentos dos estudantes estão alinhados com as definições corretas dos conceitos.
- Apresente o slide com a definição sintética de cada conceito principal discutido na aula, ampliando em forma de frases completas.
- Finalize, resumindo os pontos principais e reiterando a importância de cada conceito e como ele se encaixa no contexto maior da aula.
- Reforce a ideia de que essa revisão ajuda a solidificar o entendimento dos estudantes e a prepara-los para aplicar esses conceitos em situações práticas.

Expectativas de respostas:
- Os estudantes devem sair da aula com um entendimento claro e preciso dos conceitos principais.
- A atividade serve como uma verificação rápida do entendimento dos estudantes e uma oportunidade para corrigir quaisquer mal-entendidos.

Educação Profissional Paulista

Técnico em **Desenvolvimento de Sistemas**

Secretaria da Educação

SÃO PAULO
GOVERNO DO ESTADO