

ED 1 – Desenvolvimento para Dispositivos Móveis

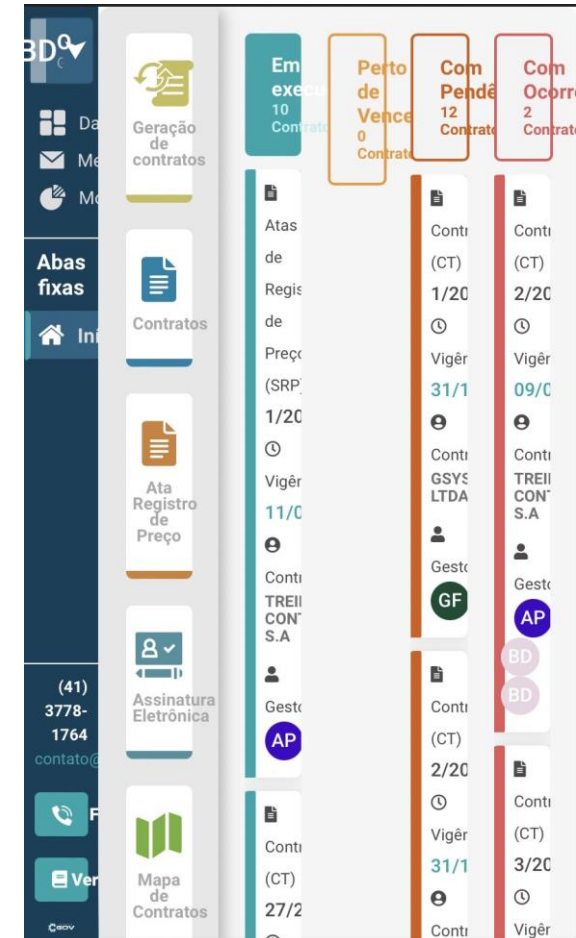
Brenda de Sobral Jesus

1 – Encontre exemplos de aplicativos com:

Interfaces mal projetadas para mudanças de orientação: Contratos GOV



Interface mal projetadas para telas pequenas: Contratos GOV



2 – Encontre exemplos de aplicativos que:

Realizam interações com outros dispositivos:

Google Home: Controla dispositivos domésticos inteligentes, como lâmpadas, termostatos, câmeras e alto-falantes. O Google Home permite que os usuários configurem, gerenciem e controlem dispositivos inteligentes conectados em sua casa usando comandos de voz via Google Assistant ou diretamente pelo aplicativo.



Realizam interações com outros aplicativos:

Spotify: : Spotify pode ser integrado com aplicativos como Waze e Google Maps, permitindo que os usuários controlem a reprodução de música sem sair do aplicativo de navegação.



3. Pesquise maneiras de como resolver ou evitar os seguintes problemas de hardware:

Alto consumo de energia:

- Evite Loops Infinitos ou Desnecessários: Verifique se não há loops que continuam a rodar sem necessidade. Isso pode consumir muitos recursos.
- Use APIs de Forma Eficiente: Certifique-se de usar APIs e bibliotecas que são otimizadas para consumo de energia.
- Reduza as Solicitações de Rede: Minimizar as solicitações de rede desnecessárias ou agrupar solicitações pode reduzir o consumo de energia.
- Sincronização Inteligente: Use técnicas como a sincronização adaptativa, onde os dados são sincronizados apenas quando necessário.

Alto consumo de memória:

- Evite Estruturas de Dados Desnecessárias: Limite o uso de coleções grandes, como listas ou mapas, que podem ocupar muita memória. Use tipos de dados primitivos quando possível.

Limpe Coleções Regularmente: Garanta que listas, mapas e outras coleções sejam limpos de objetos obsoletos ou não mais necessários.

- Evite Processos de Longa Duração: Serviços ou processos em segundo plano que mantêm objetos em memória por longos períodos podem causar problemas de memória. Use IntentService ou JobIntentService, que são projetados para gerenciar recursos de forma mais eficiente.
- Termine Processos Apropriadamente: Certifique-se de que serviços e processos em segundo plano são encerrados corretamente para liberar recursos.

3. Pesquise maneiras de como resolver ou evitar os seguintes problemas de hardware:

Alto consumo de internet móvel:

- Minimize o número de solicitações feitas ao servidor, combinando várias requisições em uma única sempre que possível. Utilize técnicas de batching para agrupar dados e reduzir a frequência de envio.
- Armazene dados que não precisam ser atualizados com frequência no armazenamento local do dispositivo, reduzindo a necessidade de solicitar os mesmos dados repetidamente.
- Comprima imagens, vídeos e outros recursos antes de enviá-los pela rede. Use formatos modernos de compressão, como WebP para imagens e H.265 para vídeos, que oferecem boa qualidade com tamanhos de arquivo menores.
- Compacte os dados que são enviados e recebidos, utilizando algoritmos como Gzip ou Brotli para reduzir o tamanho das transferências.

Problemas de conectividade com a internet:

- Verifique a Conexão: Antes de fazer solicitações de rede, verifique se o dispositivo está conectado à internet. Utilize `ConnectivityManager` no Android e `Reachability` no iOS.
- Tipo de Conexão: Identifique se a conexão é Wi-Fi, 4G/5G ou mais lenta, e ajuste o comportamento do aplicativo conforme necessário.
- Timeouts Apropriados: Configure tempos de espera para evitar que o aplicativo fique travado esperando uma resposta.
- Feedback ao Usuário: Informe o usuário sobre atrasos e ofereça opções para tentar novamente ou cancelar.