ARQUITETURA DE APLICAÇÕES MÓVEIS

EQUIPE:

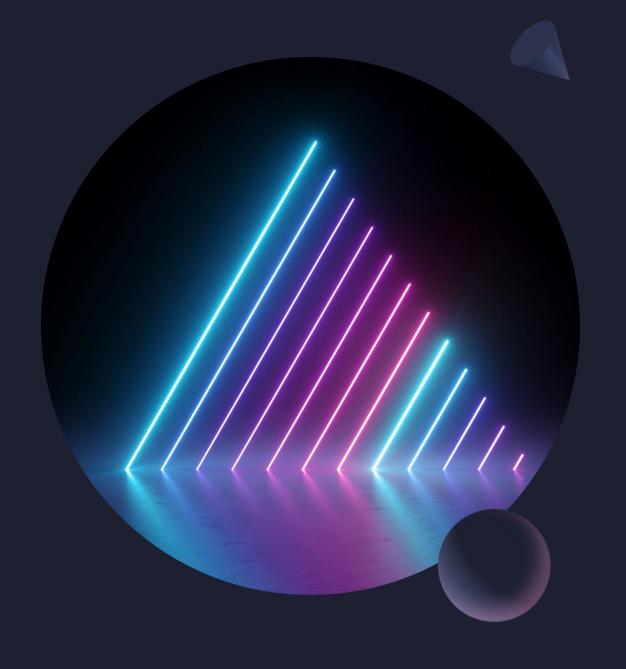
BRUNO HOLANDA REBOUCAS (427774)

GUILHERME SANTANA DE ARAUJO (427584)

JAZIEL LOUREIRO DE SOUSA (496044)

JOAO VICTOR GONCALVES SILVA (501128)

MAHOUVI EMMANUEL AGONDANOU (485668)



TÓPICOS A SEREM ABORDADOS NA APRESENTAÇÃO

- DEFINIÇÃO, CONCEITO E EXEMPLOS DE ARQUITETURA DE APLICAÇÕES MÓVEIS
- EXISTÊNCIA DE PADRÕES ARQUITETURAIS E ARQUITETURAS DE REFERÊNCIAS
- ABORDAGENS DE ÁPOIO AO DESIGN ÁRQUITETURAL (PROCESSOS, FERRAMENTAS E MÉTODOS)
- DESAFIOS NA ÁRQUITETURA DE SOFTWARE E ENGENHARIA DE SOFTWARE

DEFINIÇÃO

ARQUITETURA ESTRUTURA QUE DESCREVE A ORGANIZAÇÃO E O DESIGN DE UM APLICATIVO MÓVEL. ELA DEFINE COMO AS DIFERENTES PARTES DO APLICATIVO SE RELACIONAM E INTERAGEM ENTRE SI, E COMO OS DADOS SÃO PROCESSADOS E ARMAZENADOS.

ELA INCLUI COMPONENTES COMO A CAMADA DE INTERFACE DO USUÁRIO, QUE FORNECE A INTERFACE DO APLICATIVO, A CAMADA DE LÓGICA DE NEGÓCIOS, QUE PROCESSA DADOS E LÓGICA DO APLICATIVO, E A CAMADA DE ARMAZENAMENTO DE DADOS, QUE GERENCIA O ARMAZENAMENTO E ACESSO A DADOS.

NELE O OBJETIVO DA ARQUITETURA DE UM APLICATIVO MÓVEL É FORNECER UMA ESTRUTURA SÓLIDA PARA O DESENVOLVIMENTO DO APLICATIVO, GARANTINDO QUE O CÓDIGO SEJA ORGANIZADO, FÁCIL DE MANTER E ESCALÁVEL. ISSO TAMBÉM AJUDA A MELHORAR A EFICIÊNCIA DO DESENVOLVIMENTO E REDUZIR O TEMPO DE DESENVOLVIMENTO.

Para desenvolver uma arquitetura de aplicação móvel, é necessário considerar requisitos de funcionalidade, desempenho, escalabilidade, segurança, usabilidade e manutenção.

CONCEITO

A ARQUITETURA DE APLICAÇÕES MÓVEIS É A ESTRUTURA BÁSICA QUE DEFINE COMO OS DIFERENTES COMPONENTES DE UM APLICATIVO MÓVEL SE RELACIONAM ENTRE SI E COMO FUNCIONAM EM CONJUNTO PARA FORNECER FUNCIONALIDADES AOS USUÁRIOS FINAIS.

ELA E UMA PARTE CRÍTICA DO DESENVOLVIMENTO DE APLICATIVOS MÓVEIS, POIS AJUDA A GARANTIR QUE O APLICATIVO SEJA ESCALÁVEL, FÁCIL DE MANTER E DE MODIFICAR, E CAPAZ DE LIDAR COM AS DEMANDAS DE UM GRANDE NÚMERO DE USUÁRIOS.

AS PRINCIPAIS PREOCUPAÇÕES EM RELAÇÃO À ARQUITETURA DE APLICATIVOS MÓVEIS INCLUEM O DESEMPENHO, A SEGURANÇA, A CAPACIDADE DE INTEGRAÇÃO COM OUTROS SISTEMAS, A PORTABILIDADE ENTRE PLATAFORMAS MÓVEIS E A FACILIDADE DE USO.

EXEMPLOS

ARQUITETURA CLIENTE-SERVIDOR:

 NESSA ARQUITETURA, A APLICAÇÃO MÓVEL É DIVIDIDA EM DUAS PARTES PRINCIPAIS: O CLIENTE, QUE É EXECUTADO NO DISPOSITIVO MÓVEL, E O SERVIDOR, QUE EXECUTA A LÓGICA DE NEGÓCIOS E ARMAZENA OS DADOS DO APLICATIVO.
 OS DISPOSITIVOS MÓVEIS SE COMUNICAM COM O SERVIDOR PARA OBTER OU ENVIAR DADOS.

ÅRQUITETURA BASEADA EM MICROSERVIÇOS:

ESSA ARQUITETURA ENVOLVE DIVIDIR O
 APLICATIVO EM VÁRIOS SERVIÇOS
 PEQUENOS E INDEPENDENTES, CADA UM
 EXECUTANDO UMA TAREFA ESPECÍFICA.
 CADA SERVIÇO É IMPLANTADO
 SEPARADAMENTE E SE COMUNICA COM
 OUTROS SERVIÇOS POR MEIO DE APIS.

ARQUITETURA DE CAMADAS:

NESSA ARQUITETURA, A APLICAÇÃO É
 DIVIDIDA EM CAMADAS, COM CADA CAMADA
 EXECUTANDO UMA TAREFA ESPECÍFICA. ÁS
 CAMADAS COMUNS INCLUEM A CAMADA DE
 APRESENTAÇÃO, A CAMADA DE LÓGICA DE
 NEGÓCIOS E A CAMADA DE ACESSO A
 DADOS.

ARQUITETURA ORIENTADA A EVENTOS:

• NESSA ARQUITETURA, A APLICAÇÃO É
PROJETADA PARA RESPONDER A EVENTOS
ESPECÍFICOS, COMO TOQUES NA TELA OU
ALTERAÇÕES DE DADOS. O APLICATIVO É
DIVIDIDO EM COMPONENTES QUE OUVEM
EVENTOS E RESPONDEM A ELES DE
MANEIRA APROPRIADA.

A ESCOLHA DA ARQUITETURA ADEQUADA DEPENDE DOS REQUISITOS ESPECÍFICOS DO
APLICATIVO E DA EQUIPE DE DESENVOLVIMENTO. É IMPORTANTE ESCOLHER UMA
ARQUITETURA QUE SEJA ESCALÁVEL, FÁCIL DE MANTER E DE MODIFICAR, E CAPAZ DE LIDAR
COM AS DEMANDAS DE UM GRANDE NÚMERO DE USUÁRIOS.

EXISTÊNCIA DE PADRÕES ARQUITETURAIS E ARQUITETURAS DE REFERÊNCIAS

MVC (MODEL-VIEW-CONTROLLER):

ESSE É UM DOS PADRÕES MAIS
 POPULARES NA ARQUITETURA DE
 APLICATIVOS MÓVEIS. ELE SEPARA A
 APLICAÇÃO EM TRÊS CAMADAS PRINCIPAIS:
 O MODELO (QUE REPRESENTA A LÓGICA DE
 NEGÓCIOS), A VISÃO (QUE REPRESENTA A
 INTERFACE DO USUÁRIO) E O
 CONTROLADOR (QUE ATUA COMO
 INTERMEDIÁRIO ENTRE A VISÃO E O
 MODELO).

MVVM (MODEL-VIEW-VIEWMODEL):

 ESTE PADRÃO É SEMELHANTE AO MVC, MAS ADICIONA UMA CAMADA DE VINCULAÇÃO DE DADOS QUE PERMITE QUE AS INFORMAÇÕES SEJAM COMPARTILHADAS DE FORMA MAIS EFICIENTE ENTRE A VISÃO E O MODELO.

CLEAN ARCHITECTURE:

• ESSA ABORDAGEM ENFATIZA A SEPARAÇÃO CLARA ENTRE AS DIFERENTES CAMADAS DO APLICATIVO, PARA QUE AS MUDANÇAS EM UMA CAMADA NÃO AFETEM AS OUTRAS CAMADAS. ELA UTILIZA CONCEITOS COMO DEPENDENCY INVERSION, SEPARATION OF CONCERNS, ENTRE OUTROS.

HEXAGONAL ARCHITECTURE:

• TAMBÉM CONHECIDA COMO ARQUITETURA DE PORTAS E ÁDAPTADORES, ESSA ABORDAGEM COLOCA A LÓGICA DE NEGÓCIOS NO CENTRO DO DESIGN E A ISOLA DO MUNDO EXTERIOR POR MEIO DE PORTAS DE ENTRADA E SAÍDA. ISSO TORNA O APLICATIVO MAIS FLEXÍVEL E FACILMENTE MODIFICÁVEL.

NATIVE APP ARCHITECTURE:

DESENVOLVIMENTO DE APLICATIVOS
MÓVEIS TEM SUA PRÓPRIA ARQUITETURA
DE REFERÊNCIA, QUE É PROJETADA PARA
AJUDAR OS DESENVOLVEDORES A CRIAR
APLICATIVOS NATIVOS DE FORMA MAIS
EFICIENTE. ALGUNS EXEMPLOS INCLUEM A
ARQUITETURA DE COMPONENTES DO
ANDROID E A ARQUITETURA DE DESIGN
DO IOS.

ABORDAGENS DE APOIO AO DESIGN ARQUITETURAL (PROCESSOS, FERRAMENTAS E MÉTODOS)

PROCESSOS:

REQUISITOS E ANÁLISE DE NECESSIDADES:

 O PRIMEIRO PASSO NO PROCESSO DE DESIGN ARQUITETURAL É ENTENDER AS NECESSIDADES DO APLICATIVO E DOS USUÁRIOS, BEM COMO OS REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS.

DEFINIÇÃO DA ARQUITETURA:

• COM BASE NA ANÁLISE DE REQUISITOS, OS ARQUITETOS DE SOFTWARE DEFINEM A ARQUITETURA DO APLICATIVO MÓVEL, IDENTIFICANDO AS PRINCIPAIS CAMADAS, COMPONENTES E SERVIÇOS NECESSÁRIOS PARA ATENDER AOS REQUISITOS.

PROTOTIPAGEM E DESIGN DA INTERFACE DO USUÁRIO:

A INTERFACE DO USUÁRIO É UM
 COMPONENTE CRÍTICO DA EXPERIÊNCIA DO
 USUÁRIO DO APLICATIVO MÓVEL. OS
 ARQUITETOS DE SOFTWARE GERALMENTE
 CRIAM PROTÓTIPOS E DESIGNS DE
 INTERFACE DO USUÁRIO PARA GARANTIR
 QUE O APLICATIVO ATENDA ÀS
 EXPECTATIVAS DOS USUÁRIOS FINAIS.

IMPLEMENTAÇÃO:

• OS ARQUITETOS DE SOFTWARE
TRABALHAM COM A EQUIPE DE
DESENVOLVIMENTO PARA IMPLEMENTAR A
ARQUITETURA DEFINIDA. ISSO ENVOLVE A
CODIFICAÇÃO DE COMPONENTES,
SERVIÇOS E OUTRAS FUNCIONALIDADES
DO APLICATIVO..

TESTE E VERIFICAÇÃO:

• O APLICATIVO MÓVEL É TESTADO EM
DIFERENTES DISPOSITIVOS, EMULADORES E
AMBIENTES PARA GARANTIR QUE
FUNCIONE CORRETAMENTE E ATENDA AOS
REQUISITOS DEFINIDOS. TESTES DE
UNIDADE, INTEGRAÇÃO, ACEITAÇÃO E
USABILIDADE SÃO COMUNS NESTA FASE.

IMPLANTAÇÃO E MANUTENÇÃO:

DEPOIS QUE O APLICATIVO MÓVEL É
 DESENVOLVIDO E TESTADO COM SUCESSO,
 ELE É IMPLANTADO NO AMBIENTE DE
 PRODUÇÃO E LANÇADO PARA OS
 USUÁRIOS FINAIS. A MANUTENÇÃO
 CONTÍNUA É REALIZADA PARA CORRIGIR
 PROBLEMAS E GARANTIR QUE O
 APLICATIVO CONTINUE A FUNCIONAR
 CORRETAMENTE.

• OS PROCESSOS CITADOS ANTERIORMENTE PODEM SER ADAPTADOS POR ARQUITETOS DE SOFTWARE PARA ATENDER ÀS NECESSIDADES E REQUISITOS ESPECÍFICOS DE CADA PROJETO.

FERRAMENTAS:

PADRÕES DE ARQUITETURA:

• SÃO MODELOS DE SOLUÇÕES PARA PROBLEMAS COMUNS DE DESIGN ARQUITETURAL, QUE PODEM SER APLICADOS EM DIFERENTES CONTEXTOS. EXEMPLOS INCLUEM MVC (MODEL-VIEW-CONTROLLER), MVVM (MODEL-VIEW-VIEWMODEL) E CLEAN ARCHITECTURE.

DIAGRAMAS UML:

• A LINGUAGEM DE MODELAGEM UNIFICADA (UML) É UMA LINGUAGEM VISUAL QUE PODE SER USADA PARA REPRESENTAR AS DIFERENTES CAMADAS, COMPONENTES E FLUXOS DE DADOS DO APLICATIVO.

PADRÕES DE DESIGN:

 SÃO SOLUÇÕES COMPROVADAS PARA PROBLEMAS ESPECÍFICOS DE DESIGN, COMO O PADRÃO SINGLETON, QUE GARANTE QUE APENAS UMA INSTÂNCIA DE UMA CLASSE SEJA CRIADA.

FERRAMENTAS DE DESIGN DE INTERFACE DO USUÁRIO:

 SÃO FERRAMENTAS QUE PERMITEM CRIAR E VISUALIZAR PROTÓTIPOS DE INTERFACE DO USUÁRIO PARA O APLICATIVO, PERMITINDO TESTAR DIFERENTES LAYOUTS E FLUXOS DE NAVEGAÇÃO.

FERRAMENTAS DE CONTROLE DE VERSÃO:

• SÃO FERRAMENTAS USADAS PARA
RASTREAR E GERENCIAR AS ALTERAÇÕES
NO CÓDIGO-FONTE DO APLICATIVO.
EXEMPLOS INCLUEM GIT E SVN.

FERRAMENTAS DE GERENCIAMENTO DE PROJETOS:

• SÃO FERRAMENTAS USADAS PARA GERENCIAR O DESENVOLVIMENTO DO PROJETO, INCLUINDO PLANEJAMENTO, MONITORAMENTO E CONTROLE DO PROGRESSO. EXEMPLOS INCLUEM TRELLO, ASANA E JIRA.

• ASSIM COMO OS "PROCESSOS" CITADOS ANTERIORMENTE, OS ARQUITETOS DE SOFTWARE TAMBÉM PODEM ADAPTAR ESSAS FERRAMENTAS E ABORDAGENS PARA ATENDER ÀS NECESSIDADES E REQUISITOS ESPECÍFICOS DE CADA PROJETO.

MÉTODOS:

ANÁLISE DE REQUISITOS:

• O PRIMEIRO PASSO NA ARQUITETURA DE APLICATIVOS MÓVEIS É ENTENDER OS REQUISITOS DO APLICATIVO, INCLUINDO OS RECURSOS QUE ELE PRECISA OFERECER E O PÚBLICO-ALVO PARA O QUAL ELE É DIRECIONADO. ISSO ENVOLVE A COLETA DE INFORMAÇÕES DE VÁRIAS FONTES, INCLUINDO OS STAKEHOLDERS E OS USUÁRIOS FINAIS.

DEFINIÇÃO DA ARQUITETURA:

• COM BASE NOS REQUISITOS, OS

ARQUITETOS DE SOFTWARE DEFINEM A

ARQUITETURA DO APLICATIVO MÓVEL, QUE
ENVOLVE A IDENTIFICAÇÃO DAS PRINCIPAIS
CAMADAS, COMPONENTES E SERVIÇOS
QUE O APLICATIVO PRECISA PARA ATENDER
AOS REQUISITOS.

SELEÇÃO DE TECNOLOGIAS:

UMA VEZ QUE A ARQUITETURA É DEFINIDA,
 OS ARQUITETOS DE SOFTWARE ESCOLHEM
 AS TECNOLOGIAS QUE SERÃO USADAS
 PARA IMPLEMENTAR O APLICATIVO MÓVEL.
 ISSO INCLUI A ESCOLHA DE LINGUAGENS
 DE PROGRAMAÇÃO, FRAMEWORKS,
 BANCOS DE DADOS E OUTRAS
 FERRAMENTAS NECESSÁRIAS PARA
 DESENVOLVER E IMPLANTAR O APLICATIVO.

DESIGN DA INTERFACE DO USUÁRIO:

• A INTERFACE DO USUÁRIO É UM
COMPONENTE FUNDAMENTAL DO
APLICATIVO MÓVEL E, PORTANTO, PRECISA
SER CUIDADOSAMENTE PROJETADA PARA
ATENDER ÀS NECESSIDADES DOS
USUÁRIOS. ISSO ENVOLVE A DEFINIÇÃO DA
ESTRUTURA DA INTERFACE DO USUÁRIO, A
CRIAÇÃO DE WIREFRAMES E A
ELABORAÇÃO DE UM DESIGN VISUAL
ATRAENTE.

DESENVOLVIMENTO E TESTE:

O DESENVOLVIMENTO DO APLICATIVO
 MÓVEL É REALIZADO USANDO AS
 TECNOLOGIAS SELECIONADAS E SEGUINDO
 AS DIRETRIZES DE DESIGN DA INTERFACE
 DO USUÁRIO. O APLICATIVO É TESTADO DE
 FORMA RIGOROSA PARA GARANTIR QUE
 ELE FUNCIONE CORRETAMENTE E ATENDA
 AOS REQUISITOS.

IMPLANTAÇÃO E MANUTENÇÃO:

UMA VEZ QUE O APLICATIVO MÓVEL É
 DESENVOLVIDO E TESTADO COM SUCESSO,
 ELE É IMPLANTADO NO AMBIENTE DE
 PRODUÇÃO E LANÇADO PARA OS
 USUÁRIOS FINAIS. A MANUTENÇÃO
 CONTÍNUA É REALIZADA PARA CORRIGIR
 PROBLEMAS E GARANTIR QUE O
 APLICATIVO CONTINUE A FUNCIONAR
 CORRETAMENTE.

DESAFIOS NA ÁRQUITETURA DE SOFTWARE E ENGENHARIA DE SOFTWARE:

GERENCIAMENTO DE REQUISITOS:

 ÎDENTIFICAR, DOCUMENTAR E GERENCIAR OS REQUISITOS DO SISTEMA É UM DESAFIO IMPORTANTE, JÁ QUE ESSES REQUISITOS PODEM SER COMPLEXOS E EVOLUIR COM O TEMPO.

COMPATIBILIDADE:

• GARANTIR QUE O SOFTWARE SEJA
COMPATÍVEL COM DIFERENTES
PLATAFORMAS, SISTEMAS OPERACIONAIS,
DISPOSITIVOS E NAVEGADORES É UM
DESAFIO CONSTANTE, ESPECIALMENTE EM
APLICAÇÕES MÓVEIS.

SEGURANÇA:

GARANTIR A SEGURANÇA DO SOFTWARE É
 UM DESAFIO IMPORTANTE, ESPECIALMENTE
 EM APLICAÇÕES QUE LIDAM COM
 INFORMAÇÕES SENSÍVEIS, COMO DADOS
 PESSOAIS E FINANCEIROS.

MANUTENÇÃO:

 GARANTIR QUE O SOFTWARE POSSA SER FACILMENTE MANTIDO E ATUALIZADO É UM DESAFIO IMPORTANTE, ESPECIALMENTE EM APLICAÇÕES MÓVEIS QUE PODEM TER ATUALIZAÇÕES FREQUENTES.

ESCALABILIDADE:

 GARANTIR QUE O SOFTWARE POSSA LIDAR COM UMA GRANDE QUANTIDADE DE USUÁRIOS E DADOS É UM DESAFIO IMPORTANTE, ESPECIALMENTE EM APLICAÇÕES MÓVEIS QUE PODEM TER UM GRANDE NÚMERO DE USUÁRIOS SIMULTÂNEOS

PERFORMANCE:

 GARANTIR QUE O SOFTWARE TENHA UM DESEMPENHO ADEQUADO É UM DESAFIO IMPORTANTE, ESPECIALMENTE EM APLICAÇÕES MÓVEIS QUE PODEM TER RECURSOS LIMITADOS, COMO BATERIA E CONEXÃO DE INTERNET.

ARQUITETURA:

 ESCOLHER A ARQUITETURA ADEQUADA PARA O SOFTWARE É UM DESAFIO IMPORTANTE, POIS UMA ARQUITETURA MAL ESCOLHIDA PODE LEVAR A PROBLEMAS DE ESCALABILIDADE, MANUTENÇÃO E PERFORMANCE.

DOCUMENTAÇÃO:

GARANTIR QUE O SOFTWARE SEJA
 ADEQUADAMENTE DOCUMENTADO É UM
 DESAFIO IMPORTANTE, POIS A
 DOCUMENTAÇÃO É NECESSÁRIA PARA O
 ENTENDIMENTO E MANUTENÇÃO DO
 SOFTWARE.

TESTES:

GARANTIR QUE O SOFTWARE SEJA
 TESTADO ADEQUADAMENTE É UM
 DESAFIO IMPORTANTE, POIS OS TESTES
 DEVEM SER REALIZADOS EM
 DIFERENTES CENÁRIOS E SITUAÇÕES
 PARA GARANTIR A QUALIDADE DO
 SOFTWARE.

COLABORAÇÃO:

GARANTIR QUE AS EQUIPES DE
 DESENVOLVIMENTO TRABALHEM DE
 FORMA COLABORATIVA E EFICIENTE É
 UM DESAFIO IMPORTANTE,
 ESPECIALMENTE EM PROJETOS DE
 GRANDE PORTE OU QUE ENVOLVEM
 EQUIPES DISTRIBUÍDAS.

FIM!

