

Resumo Descritivo Técnico

#08 – Desenvolvimento de Aplicativos Móveis.

Versão 2024/2025

Sumário

Escopo e Ocupação	3
Padrões e Pesos de Competência.....	3
Formato do Projeto Teste	3
Estratégia de Avaliação e Pontuação.....	4
Tecnologias, Stack e Restrições	4
Exigências por Módulo	4

Escopo e Ocupação

Um desenvolvedor mobile profissional deve ser capaz de planejar, projetar, implementar, testar e otimizar apps para dispositivos móveis, conciliando design/UX, arquitetura, implementação (incluindo APIs, sensores e recursos nativos) e testes (automatizados e de desempenho). Há ênfase em comunicação com o cliente, gestão do trabalho e boas práticas (ética, segurança, confidencialidade).

Padrões e Pesos de Competência

A matriz de competência distribui a importância relativa em 7 frentes, somando 100%. Esse mapa deve orientar o peso do seu treinamento e a rubrica interna:

- (1) Organização e gestão do trabalho – 8%
- (2) Comunicação e interpessoais – 7%
- (3) Prática sustentável – 5%
- (4) Planejamento inicial, design e estrutura de teste – 25%
- (5) Arquitetura de sistema – 15%
- (6) Implementação e desenvolvimento do produto – 30%
- (7) Testes finais, troubleshooting e otimização – 10%

Formato do Projeto Teste

O Projeto Teste é composto por 6 provas (módulos A–D) com carga prática de 3h por prova (totalizando 18 a 22 horas no evento):

Módulo A – Design (Figma): 2 provas (6h) → protótipos navegáveis de alta fidelidade (tablet + celular), seguindo manual de marca.

Módulo B – Funcionalidades (Emulador): 2 provas (6h) → consumo de API REST, regras de negócio, conformidade.

Módulo C – Implementações (Emulador): 1 prova (3h) → funções nativas (gestos, geolocalização, giroscópio, acelerômetro, etc.).

Módulo D – Testes (Emulador): 1 prova (3h) → desenvolver a partir de casos de teste + testes automatizados.

Todas as provas devem compor um contexto único, porém independentes (cada prova traz insumos/arquivos próprios).

Estratégia de Avaliação e Pontuação

Dois tipos de avaliação: medição (objetiva) e julgamento (subjetiva); tudo é lançado no CIS (sistema informatizado).

A avaliação é estruturada por Critérios, Subcritérios e Aspectos; cada Aspecto é avaliado isoladamente (objetivo ou subjetivo).

Cada aspecto tem pontuação máxima de 2 e descreve uma escala subjetiva 0–3 para julgamento (observa-se uma inconsistência textual: fala em “escala de 10 pontos”, mas lista 0–3; recomendo confirmar com a coordenação como será parametrizado no CIS local).

Três avaliadores julgam cada aspecto, havendo um “coringa” para evitar conflito (avaliador do mesmo competidor).

Tecnologias, Stack e Restrições

Plataforma principal: Flutter

Ambiente/hardware: Windows, VS Code, Android Studio, Flutter SDK, JDK/JRE, Git, Gogs, Postman, Figma, Office; i7 10ª gen, 16 GB RAM, SSD 256 GB, monitor 27".

Pacotes permitidos: somente os oficiais do publisher flutter.dev no pub.dev.

Internet restrita: consulta apenas à documentação oficial listada (Android/Flutter/Google, SQLite, Material, bancos de ícones especificados, Google Translate). Outros sites são proibidos.

****treinar sob restrição real (sem StackOverflow/GitHub) e validar previamente que soluções usam APIs/pacotes oficiais. ****

Exigências por Módulo

Design (Figma):

Seguir guia de marca e especificações iOS/Android (layout, tipografia, grid, estados, acessibilidade).

Entregar protótipos navegáveis (phone + tablet).

Documentar UX e hand-off (componentes, tokens, espaçamentos).

Funcionalidades (REST):

Consumo de API RESTful, tratamento de erros/estados (loading/empty/failure), regras de negócio e persistência local quando aplicável.

Conformidade com REST (verbo/rota/status/body), organização de camadas e boas práticas (logging, retries controlados).

Implementações nativas

Integração com sensores e recursos do dispositivo (gestos, GPS, giroscópio/acelerômetro, etc.), animações 2D/3D e renderização eficiente.

Testes (TDD/automatizados)

Construir a partir de casos de teste, implementar testes automatizados (unidade/widget/instrumentação conforme cabível), registrar resultados e depurar.