

FIAP – Faculdade de Informática e Administração Paulista
Curso: Sistemas de Informação

Arthur Dias da Silva Bianchi – RM 99162
Eduardo Costa Nascimento dos Anjos – RM 552519
Enzo Puerta Meschini – RM 550807

DOCUMENTAÇÃO DO PROJETO: WORKMIND
Global Solution 2º semestre

São Paulo – 2025

1. Visão Geral e Tecnologias

O WorkMind é uma solução móvel nativa desenvolvida para apoiar o bem-estar e a produtividade no ambiente de trabalho moderno. O projeto utiliza tecnologias de ponta para garantir performance e escalabilidade.

Tecnologias Adotadas e Benefícios:

- React Native (com Expo): Escolhido pela capacidade de desenvolvimento multiplataforma (Android/iOS) com uma única base de código em JavaScript. O Expo acelera o ciclo de desenvolvimento e facilita testes em dispositivos reais.
- Oracle Database (PL/SQL): Utilizado para a persistência robusta de dados e processamento de regras de negócio complexas (como gamificação) diretamente no banco através de Procedures.
- JavaScript (ES6+): Linguagem versátil e amplamente suportada, permitindo o uso de funcionalidades modernas de programação assíncrona e manipulação de arrays.

2. Descrição de Funções, Métodos e Componentes

A aplicação está estruturada em componentes funcionais, separando responsabilidades de interface e lógica:

App.js (Controlador Principal):

Gerencia o estado global de autenticação (usuarioLogado).

Implementa a navegação condicional entre a tela de Login e as telas internas (Dashboard).

Gerencia o menu de abas inferior (Tab Bar) persistente.

screens/FocusScreen.js (Produtividade):

Método iniciarPomodoro(): Aciona o timer decrescente utilizando setInterval.

Animação de Respiração: Utiliza a API Animated do React Native para criar o efeito visual de expansão/contração pulmonar (ciclo 4-2-6).

Integração com Hardware: Utiliza a API Vibration para feedback tático ao final dos ciclos.

screens/AuthScreen.js (Segurança):

Método handleCadastro(): Realiza validações de entrada (campos vazios, senha fraca, e-mail duplicado) antes de submeter os dados.

Gerencia a alternância entre formulários de Login e Cadastro.

screens/MissoesScreen.js (Gamificação):

Calcula o nível do usuário com base na soma de XP (Experiência) acumulado.

Renderiza listas de tarefas interativas.

screens/BemEstarScreen.js (Saúde Mental):

Registra métricas diárias de Humor, Energia e Estresse.

Utiliza FlatList para exibição eficiente do histórico de registros.

3. Justificativas Técnicas

A. Gerenciamento de Memória e Eficiência

Para garantir que o aplicativo rode suavemente em dispositivos móveis com recursos limitados, foram aplicadas as seguintes boas práticas:

1. Limpeza de Processos (useEffect Cleanup): Na tela de Foco, todos os intervals (timers) e listeners de animação possuem funções de limpeza (return no useEffect). Isso impede *memory leaks* (vazamento de memória) quando o usuário troca de tela enquanto um timer está rodando.
2. Renderização Otimizada de Listas: Substituímos o uso de ScrollView por FlatList nas telas de histórico e missões. O FlatList renderiza apenas os itens visíveis na tela (lazy rendering), economizando drasticamente memória RAM em listas longas.
3. Estado Localizado: O uso estratégico de useState em componentes menores evita que a aplicação inteira seja re-renderizada quando apenas um detalhe (como o timer) é atualizado.

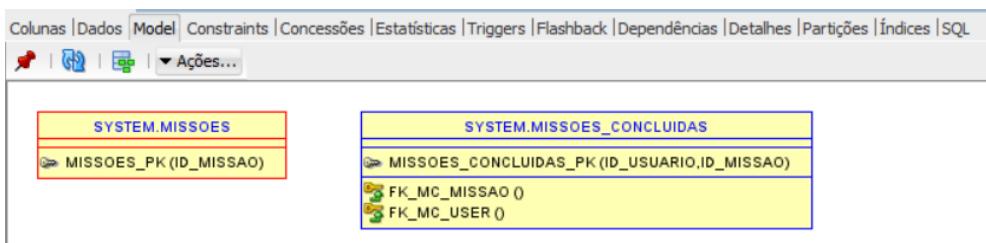
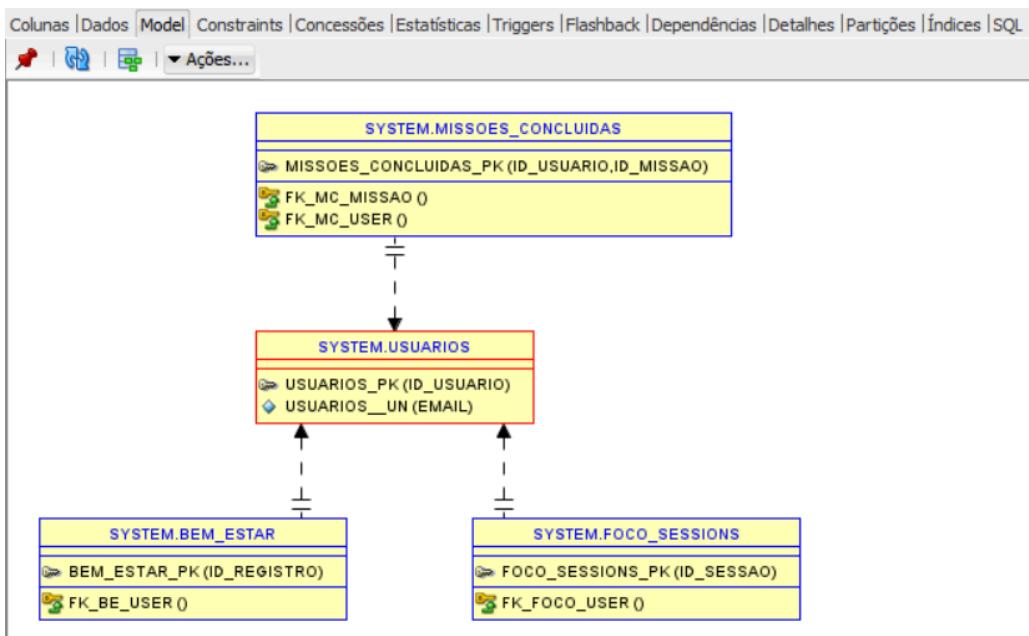
B. Interface e Experiência do Usuário (UX)

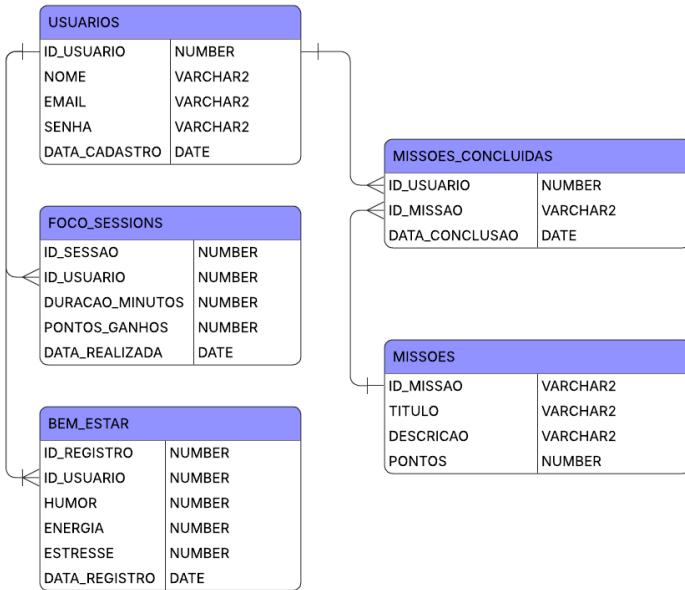
A interface foi desenhada para ser acessível e reduzir a ansiedade:

1. Design System "Dark Mode": O uso de tons escuros (#0f172a) reduz a emissão de luz azul, minimizando a fadiga visual do usuário durante o uso prolongado.
2. Feedback Multissensorial: O aplicativo não depende apenas da visão. O uso de vibrações tátteis durante os exercícios de respiração permite que o usuário realize as atividades de olhos fechados, aumentando a imersão e relaxamento.
3. Navegação Simplificada: A barra de navegação inferior fixa permite que qualquer funcionalidade principal seja acessada com apenas um toque, respeitando as heurísticas de usabilidade móvel.

4. Modelagem de Dados e Automação

A. Diagrama





B. Automação com PL/SQL

Foi implementada uma Stored Procedure no Banco de Dados Oracle para automatizar o cálculo de desempenho do usuário. Em vez de processar regras de negócio pesadas no celular do usuário, o banco de dados processa o engajamento e define o nível do usuário automaticamente.

```

create or replace NONEDITIONABLE PROCEDURE SP_CALCULA_METRICAS_USUARIO (
    P_ID_USUARIO IN NUMBER,
    P_TOTAL_PONTOS OUT NUMBER,
    P_NIVEL OUT VARCHAR2
) AS
    V_PONTOS_MISSOES NUMBER := 0;
    V_PONTOS_FOCO NUMBER := 0;
BEGIN
    -- 1. Soma pontos das missões concluídas
    SELECT NVL(SUM(m.PONTOS), 0)
    INTO V_PONTOS_MISSOES
    FROM MISSOES_CONCLUIDAS mc
    JOIN MISSOES m ON mc.ID_MISSAO = m.ID_MISSAO
    WHERE mc.ID_USUARIO = P_ID_USUARIO;

    -- 2. Soma pontos das sessões de foco
    SELECT NVL(SUM(PONTOS_GANHOS), 0)
    INTO V_PONTOS_FOCO
    FROM FOCO_SESSIONS
    WHERE ID_USUARIO = P_ID_USUARIO;

    -- 3. Totaliza
    P_TOTAL_PONTOS := V_PONTOS_MISSOES + V_PONTOS_FOCO;

    -- 4. Define o Nível (Lógica igual ao seu JS)
    IF P_TOTAL_PONTOS < 30 THEN
        P_NIVEL := 'Iniciante';
    ELSIF P_TOTAL_PONTOS < 60 THEN
        P_NIVEL := 'Em evolução';
    ELSE
        P_NIVEL := 'Protagonista do futuro';
    END IF;

    -- Opcional: Gravar log no console do Oracle (Dbms Output)
    DBMS_OUTPUT.PUT_LINE('Usuário ID: ' || P_ID_USUARIO || ' | Pontos: ' || P_TOTAL_PONTOS || ' | Nível: ' || P_NIVEL);
END;
    
```

```

CREATE OR REPLACE PROCEDURE SP_CALCULA_METRICAS_USUARIO (
    P_ID_USUARIO IN NUMBER,
    P_TOTAL_PONTOS OUT NUMBER,
    P_NIVEL OUT VARCHAR2
) AS

    V_PONTOS_MISSOES NUMBER := 0;
    V_PONTOS_FOCO NUMBER := 0;

BEGIN

    -- 1. Automatiza a soma de pontos por Missões

    SELECT NVL(SUM(m.PONTOS), 0) INTO V_PONTOS_MISSOES
    FROM MISSOES_CONCLUIDAS mc
    JOIN MISSOES m ON mc.ID_MISSAO = m.ID_MISSAO
    WHERE mc.ID_USUARIO = P_ID_USUARIO;

    -- 2. Automatiza a soma de pontos por Foco/Produtividade

    SELECT NVL(SUM(PONTOS_GANHOS), 0) INTO V_PONTOS_FOCO
    FROM FOCO_SESSIONS WHERE ID_USUARIO = P_ID_USUARIO;

    -- 3. Consolida o resultado

    P_TOTAL_PONTOS := V_PONTOS_MISSOES + V_PONTOS_FOCO;

    -- 4. Regra de Nivelamento (Lógica de Negócio Centralizada)

    IF P_TOTAL_PONTOS < 30 THEN P_NIVEL := 'Iniciante';
    ELSIF P_TOTAL_PONTOS < 60 THEN P_NIVEL := 'Em evolução';
    ELSE P_NIVEL := 'Protagonista do futuro';
    END IF;

END;
/

```