

## Proposta do Aplicativo

**Descrição Detalhada:** O "Missões Diárias" é um aplicativo voltado para usuários que desejam tornar seu dia a dia mais dinâmico e interessante por meio de pequenas missões diárias. Ele visa ajudar as pessoas a realizar pequenas tarefas que incentivam hábitos positivos ou estimulam a criatividade, como "cumprimente três pessoas hoje" ou "tente aprender uma palavra nova". O público-alvo são pessoas que buscam motivação para completar pequenas metas diárias, especialmente jovens e adultos que gostam de gamificação.

**Objetivo:** Ajudar os usuários a adicionar pequenos desafios em suas rotinas diárias, incentivando ações positivas e criativas.

**Público-Alvo:** Adultos e jovens que querem motivação diária e gostam de realizar pequenos desafios no seu dia a dia.

**Problema Solucionado:** Auxilia pessoas que precisam de uma motivação adicional para realizar pequenas tarefas, como se organizar melhor, interagir socialmente ou aprender algo novo.

### Funcionalidades Principais

1. **Tela Inicial com Missão do Dia:** Exibe uma missão diária aleatória. A missão muda automaticamente a cada 24 horas.
2. **Botão para Aceitar ou Pular Missão:** O usuário pode aceitar a missão ou pular para outra.
3. **Contagem de Missões Completadas:** Registro das missões realizadas e acompanhamento de "dias consecutivos" de missões completadas.
4. **Gamificação Leve:** Conquistas visuais desbloqueadas após completar um número específico de missões consecutivas (ex: após completar 5, 10, 15 dias consecutivos).
5. **Persistência de Dados:** Dados simples como as missões realizadas e contagem de dias consecutivos são armazenados no dispositivo usando SharedPreferences ou arquivos internos.

### Design e Usabilidade

- **Interface Atraente:** O app deve ter uma interface amigável, com cores leves e animações suaves ao navegar entre as missões e realizar ações.
- **Navegação Intuitiva:** Simples navegação entre telas. A principal ação (aceitar ou pular missões) deve ser clara e acessível.

- **Feedback Visual e Sonoro:** Pequenas animações ou sons podem ser utilizados quando uma missão é completada.

## **Estrutura do App**

### **1. Tela Inicial (MainActivity)**

- Exibe a missão do dia e botões para "Aceitar" ou "Pular Missão".
- Usa um TextView para exibir a missão atual.
- Botões para aceitar ou pular a missão.
- Animação simples quando a missão é atualizada ou aceita.

### **2. Tela de Conquistas (AchievementsActivity)**

- Exibe conquistas desbloqueadas (em um RecyclerView ou GridView).
- Exibe missões completas e dias consecutivos.

### **3. Tratamento de Missões**

- As missões podem ser armazenadas em um array de strings.
- Ao abrir o app, uma missão é selecionada aleatoriamente.
- Se a missão for completada ou ignorada, uma nova missão é exibida.

## **Implementação Técnica**

### **1. Persistência de Dados**

- Use SharedPreferences para armazenar o número de missões completadas e os dias consecutivos.
- Salve a última missão completada e a data para garantir que uma nova missão seja exibida apenas após 24 horas.

### **2. Interface do Usuário**

- Crie uma interface leve e moderna, com animações simples ao completar uma missão (ex: uso de Lottie Animation ou transições de imagem).
- Use componentes do Android SDK, como ConstraintLayout para organizar os elementos visuais.

### **3. Tratamento de Eventos**

- Implementar os cliques nos botões "Aceitar" e "Pular", e atualizar a missão com um evento simples.
- Usar Toast ou Snackbar para fornecer feedback ao usuário.

## **Código Exemplo (Estrutura Básica)**

Aqui está um exemplo básico da lógica por trás da escolha da missão e contagem de missões completadas:

```
class MainActivity : AppCompatActivity() {
```

```
private lateinit var missionTextView: TextView
private lateinit var acceptButton: Button
private lateinit var skipButton: Button

private val missions = arrayOf(
    "Cumprimente 3 pessoas hoje",
    "Aprenda uma palavra nova",
    "Faça uma caminhada de 15 minutos",
    "Escreva sobre o seu dia",
    "Ajude alguém com algo"
)
private var currentMissionIndex = 0
private var consecutiveDays = 0

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    missionTextView = findViewById(R.id.missionTextView)
    acceptButton = findViewById(R.id.acceptButton)
    skipButton = findViewById(R.id.skipButton)

    // Carregar missão
    loadMission()

    acceptButton.setOnClickListener {
        acceptMission()
    }

    skipButton.setOnClickListener {
        skipMission()
    }
}
```

```

private fun loadMission() {
    val sharedPreferences = getSharedPreferences("mission_prefs",
Context.MODE_PRIVATE)
    currentMissionIndex = sharedPreferences.getInt("mission_index", 0)
    consecutiveDays = sharedPreferences.getInt("consecutive_days", 0)

    missionTextView.text = missions[currentMissionIndex]
}

```

```

private fun acceptMission() {
    // Salvar progresso
    val sharedPreferences = getSharedPreferences("mission_prefs",
Context.MODE_PRIVATE)
    val editor = sharedPreferences.edit()
    editor.putInt("consecutive_days", consecutiveDays + 1)
    editor.apply()

    Toast.makeText(this, "Missão completada!", Toast.LENGTH_SHORT).show()

    // Atualizar missão
    loadNewMission()
}

```

```

private fun skipMission() {
    // Simplesmente carregar uma nova missão
    loadNewMission()
}

```

```

private fun loadNewMission() {
    currentMissionIndex = (currentMissionIndex + 1) % missions.size
    missionTextView.text = missions[currentMissionIndex]

    // Salvar nova missão

```

```
        val sharedPreferences = getSharedPreferences("mission_prefs",  
Context.MODE_PRIVATE)  
        val editor = sharedPreferences.edit()  
        editor.putInt("mission_index", currentMissionIndex)  
        editor.apply()  
    }  
}
```