Reinforcement Learning on Mobile Devices: Context-Aware Configuration Control

Alcilene Batista[1], Elian Souza[1], Raimundo Barreto[1]

[11]Institute of Computing – Federal University of Amazonas (UFAM)
CEP 69067-005 – Manaus – AM – Brazil
{abds, elian.souza, rbarreto}@icomp.ufam.edu.br

Resumo. *A configuração automática em dispositivos móveis enfrenta dificuldades para se adaptar às preferências reais dos usuários, frequentemente causando frustração com ajustes genéricos de brilho, volume e notificações. Este artigo propõe uma solução embarcada baseada em aprendizado por reforço, capaz de ajustar dinamicamente o brilho da tela e o volume de mídia de acordo com o contexto. O sistema coleta dados de sensores, como luz ambiente, localização e status de reunião, discretizando essas variáveis para alimentar um agente Q-Learning. As ações são refinadas por meio de feedback manual interpretado como reforço supervisionado. Para estados não vistos, as preferências são inferidas usando similaridade contextual. Todos os módulos operam offline, sem dependência de serviços em nuvem ou conectividade externa. Os resultados mostram que o agente aprendeu políticas coerentes com o comportamento do usuário, manteve estabilidade em contextos recorrentes, respondeu corretamente a novos cenários e reduziu a necessidade de intervenção manual. A arquitetura proposta demonstra a viabilidade de agentes autônomos embarcados, oferecendo personalização inteligente diretamente em dispositivos Android.*

Abstract. *Automatic configuration on mobile devices faces challenges in adapting to real user preferences, often leading to frustration with generic adjustments to brightness, volume, and notifications. This paper proposes an embedded solution based on reinforcement learning, capable of dynamically adjusting screen brightness and media volume according to context. The system collects sensor data such as ambient light, location, and meeting status, in order to feed a Q-Learning agent. Actions are refined through manual feedback interpreted as supervised reinforcement. For unseen states, preferences are inferred using contextual similarity. All modules operate offline, with no reliance on cloud services or external connectivity. Results show that the agent learned policies consistent with user behavior, maintained stability in recurring contexts, responded appropriately to new scenarios, and reduced the need for manual intervention. The proposed architecture demonstrates the feasibility of embedded autonomous agents, delivering intelligent personalization directly on Android devices.*

1. Introdução

Nas últimas décadas, dispositivos móveis como smartphones e smartwatches passaram de simples ferramentas de comunicação para sistemas computacionais sofisticados, equipados com sensores como acelerômetros, luxímetros, microfones e GPS. Essa evolução ampliou sua capacidade de captar informações ambientais e contextuais, possibilitando interações ajustadas a variações contextuais observadas por sensores. Em teoria, essa

sensibilidade ao contexto permitiria ajustes automáticos em parâmetros como brilho e volume, otimizando a experiência do usuário em tempo real.

Na prática, os sistemas de configuração automática dos sistemas operacionais móveis continuam operando de forma limitada, baseados em regras estáticas e generalistas. Usuários relatam frustrações porque os ajustes ignoram preferências pessoais ou falham em se adaptar a cenários como ambientes escuros, reuniões ou locais barulhentos [Wang et al. 2021, Kaladevi et al. 2024]. Como resultado, muitas dessas funcionalidades são desativadas ou substituídas por ajustes manuais, evidenciando o descompasso entre a capacidade sensorial dos dispositivos e a falta de mecanismos eficazes para ajustar configurações com base nas preferências dos usuários e seus contextos. O aprendizado por reforço (RL) destaca-se por permitir a evolução de políticas de ação baseadas em recompensas derivadas da interação entre sistema e ambiente [Sutton e Barto 2018]. Contudo, observa-se uma lacuna significativa na aplicação embarcada dessa técnica em dispositivos Android, principalmente para operação offline, generalização de estados e integração com feedback manual [Todi et al. 2021, Bai et al. 2024]. Este trabalho propõe um sistema embarcado para Android que integra aprendizado por reforço, inferência baseada em contexto e aprendizado incremental das preferências, visando ajustar automaticamente configurações como brilho e volume. A arquitetura implementa um agente Q-Learning com política de exploração adaptativa, reforço supervisionado baseado em correções manuais e mecanismos de generalização por similaridade.

O problema abordado neste estudo é resumido na seguinte questão: *Como permitir que dispositivos móveis aprendam e ajustem suas configurações com base em padrões de uso, respondendo dinamicamente a variações contextuais observadas?* A hipótese é se a combinação entre percepção ambiental, aprendizado adaptativo e reforço supervisionado pode produzir agentes cujas ações reflitam padrões consistentes de uso observados. As principais contribuições deste artigo são: (i) um sistema embarcado de RL com aprendizado contínuo; (ii) uma estratégia de generalização hierárquica para lidar com estados não visitados; e (iii) uma validação empírica com interações reais, demonstração de estabilidade em contextos recorrentes, correspondência entre ações aprendidas e preferências manuais observadas, e redução significativa de intervenções manuais.

Este artigo está estruturado da seguinte forma: a Seção 2 apresenta a fundamentação teórica. A Seção 3 discute os trabalhos correlatos. A Seção 4 detalha a arquitetura do sistema proposto. A Seção 5 expõe os experimentos realizados. Por fim, a Seção 6 apresenta os comentários conclusivos e extensões futuras.

2. Fundamentação Teórica

Esta seção apresenta os conceitos fundamentais que embasam o sistema proposto, organizado em quatro eixos: (i) sistemas sensíveis ao contexto, (ii) aprendizado por reforço com Q-Learning, (iii) técnicas de generalização e adaptação a estados inexplorados, e (iv) reforço supervisionado por feedback manual.

2.1. Sensibilidade ao Contexto

Dispositivos móveis sensíveis ao contexto utilizam sensores, para captar variáveis do ambiente e adaptar seu comportamento a diferentes situações de uso, como alterar o brilho da tela em ambientes escuros ou silenciar o telefone durante reuniões. Essas estratégias são

conhecidas como adaptações baseadas em contexto [Du et al. 2025, Souza et al. 2022]. Para viabilizar tais ajustes, variáveis contínuas precisam ser discretizadas em categorias interpretáveis pelo sistema, formando vetores de estado que alimentam a lógica de decisão. A correta abstração e representação dessa informações é essencial para a eficácia de sistemas contextualmente inteligentes [Sarker 2019].

2.2. Aprendizado por Reforço e Q-Learning

O núcleo decisório do sistema é um agente baseado no algoritmo Q-Learning, um método de aprendizado por reforço *off-policy* que estima a função valor-ação $Q(s, a)$ a partir da interação com o ambiente [Sutton e Barto 2018]. A atualização dos Q-values segue a Equação 1. Para controle da política, são utilizadas estratégias complementares: ε-greedy no início do aprendizado e Softmax com temperatura adaptativa à medida que a política se estabiliza [Tokic 2010]. Os Q-values são inicializados no intervalo $[-0{,}5,\ 0{,}5]$ para incentivar a exploração de múltiplas ações. As ações disponíveis incluem ajustes gerais (como brilho ou volume) e ações condicionadas ao contexto (como silenciar em reuniões). Cada decisão é validada com base em restrições contextuais observadas, como limites mínimos de brilho ou modo silencioso ativo.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right] \tag{1}$$

Na Equação 1, s_t representa o estado atual do sistema, a_t é a ação executada naquele estado, R_{t+1} é a recompensa recebida após a ação, α é a taxa de aprendizado que controla a velocidade de atualização dos valores, γ é o fator de desconto aplicado às recompensas futuras, e $\max_{a'} Q(s_{t+1}, a')$ representa o maior valor de Q associado ao próximo estado s_{t+1}, considerando todas as ações possíveis a'. Essa formulação permite que o agente ajuste iterativamente sua política com base na diferença entre a recompensa prevista e a observada.

2.3. Generalização e Adaptação a Estados Inexplorados

Devido à explosão combinatória do espaço de estados, são aplicadas técnicas de generalização como clusterização contextual e inferência probabilística. O agrupamento de estados similares com base em luminosidade, localização e tipo de ambiente permite que decisões aprendidas em um contexto sejam reutilizadas em situações novas com características próximas [Sarker et al. 2020]. Além disso, o sistema emprega decaimento exponencial das preferências para se adaptar a mudanças de longo prazo no padrão de uso do usuário [Murphy 2012].

2.4. Reforço Supervisionado via Feedback Manual

O sistema também detecta intervenções manuais do usuário como sinal de correção, integrando esse feedback supervisionado na atualização da política. Quando o usuário altera manualmente o brilho ou o volume após uma ação do agente, essa ação é interpretada como indicativo de erro e usada para reforçar a preferência esperada. Esse reforço supervisionado acelera a convergência da política ótima e favorece decisões que refletem as preferências manuais observadas do usuário em diferentes contextos.

3. Trabalhos Relacionados

O trabalho de [Abeywardhane et al. 2018] introduz o sistema RE-IN, que aplica Q-Learning para ajustar brilho e volume com base em localização, idade e horário. Embora use clusters para reduzir o espaço de estados, não implementa mecanismos avançados de supervisão ou inferência. O modelo de [Altulyan et al. 2021] aplica Contextual Bandits para ativação/desativação de sensores conforme contexto e recompensas passadas. A política é construída de forma parcial, sem considerar supervisão direta do usuário.

O BehavDT [Sarker et al. 2020], utiliza uma árvore comportamental baseada em clusters e regras com limiares de confiança. A abordagem inspirou a arquitetura hierárquica deste trabalho, ainda que não seja embarcada nem sensível a supervisão. O CBO [Yamasaki et al. 2023], propõe uma otimização bayesiana sensível ao contexto que transfere conhecimento entre clusters de usuários para mitigar o problema de cold start. A ideia de generalização orientada por agrupamentos inspirou o mecanismo de propagação de preferências entre estados com contexto semelhante deste trabalho. O trabalho de [Lin et al. 2019] modela a função de recompensa com base em métricas de desempenho e engajamento, ajustadas ao histórico de uso. Embora o feedback não seja explícito, a atualização supervisionada por interações anteriores aproxima-se da lógica de reforço sensível à preferência, conceito expandido nesta proposta.

Uma abordagem de automação adaptativa com três níveis de controle, manual, inquisitivo e automático, é apresentada por [Ahmadi-Karvigh et al. 2019], combinando Q-Learning com HTN e Boosting. Embora apresente uma arquitetura complexa e com múltiplos níveis de controle, sua execução depende de componentes externos, o que inviabiliza aplicações embarcadas.

Este trabalho se distingue por integrar Q-Learning com generalização probabilística, decaimento exponencial de preferências, inferência hierárquica, reforço supervisionado persistente e execução totalmente embarcada em Android. O sistema funciona mesmo sem conexão com servidores ou bancos de dados externos, realizando ajustes contextuais com base em luminosidade, localização e status de reunião.

A Tabela 1 resume comparativamente os principais trabalhos mencionados nesta seção, evidenciando as diferenças metodológicas, estratégias de generalização, execução embarcada e limitações superadas pelo sistema proposto.

Tabela 1. Comparativo entre trabalhos relacionados e o presente estudo

Trabalho	Técnica Principal	Domínio	Generalização e Supervisão	Execução Embarcada	Limitações Destacadas
Abeywardhane et al. (2018)	Q-Learning + K-Means	Android (brilho/volume)	Clusters estáticos; sem feedback manual	Parcial, dependente de simulação	Não lida com cold start; sem persistência adaptativa
Altulyan et al. (2021)	Contextual Bandits	Economia de energia	Recompensas implícitas; sem feedback direto	Offline, mas sem personalização profunda	Sem aprendizado supervisionado; generalização limitada
Ahmadi et al. (2019)	Q-Learning + HTN + Boosting	Automação adaptativa	Inferência hierárquica adaptativa	Parcial (requer componentes externos)	Complexidade elevada; não disponível em Android puro
Sarker et al. (2020)	BehavDT (árvore comportamental)	Inferência em dispositivos móveis	Generalização via cluster e cache	Não embarcado	Sem operação offline nem supervisão direta
Yamasaki et al. (2023)	Bayesian Optimization + Clustering	Sistemas interativos	Inferência orientada a clusters para cold start	Não especificado	Sem mecanismo contínuo de reforço embarcado
Lin et al. (2019)	RL + Micro Learning	Educação adaptativa	Ajuste progressivo com base em interações anteriores	Não aplicável	Feedback implícito, sem persistência adaptativa
Presente estudo	Q-Learning + Generalização hierárquica	Android (brilho, volume)	Clusters, inferência probabilística e feedback manual persistente	Sim, 100% embarcado e offline	**Nenhuma das limitações acima**

4. Método Proposto

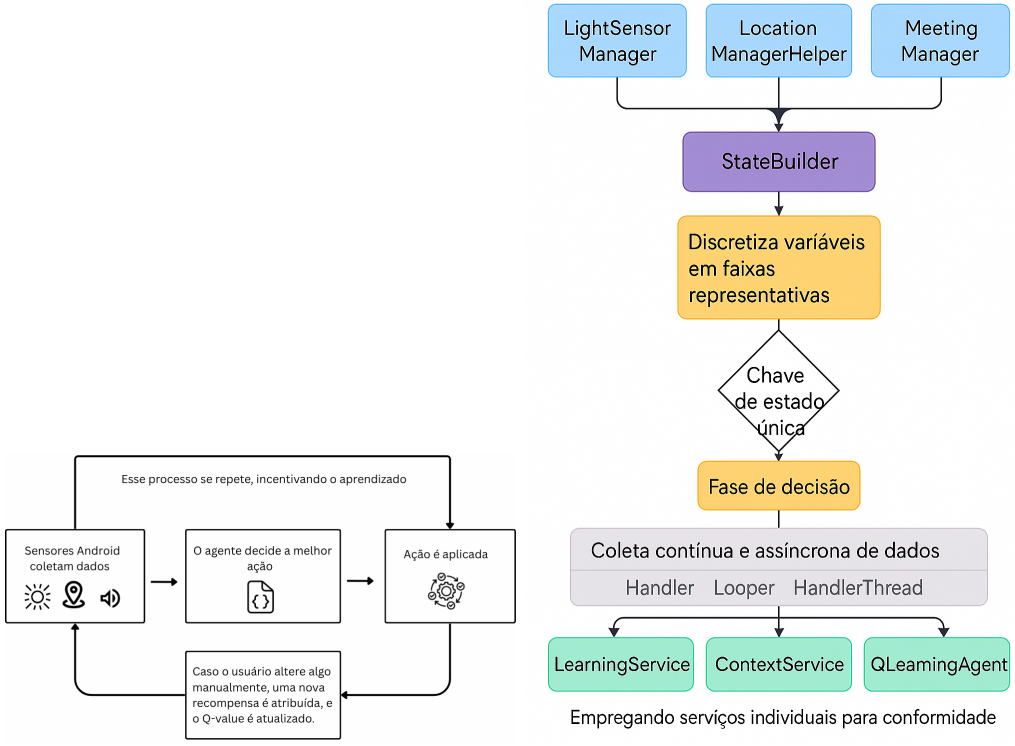
4.1. Arquitetura

O sistema funciona em um ciclo fechado, composto por quatro etapas principais: (i) coleta de dados contextuais por sensores e serviços do Android; (ii) construção de um vetor de estado discreto; (iii) seleção e execução de uma ação; e (iv) atribuição de recompensa com base na eficácia do ajuste ou na ocorrência de intervenção manual do usuário. A Figura 1(a) apresenta o ciclo de decisão, destacando a natureza contínua do processo de aprendizado. O agente interage com APIs do Android para captar variáveis como luminosidade, localização, modo de som e eventos de calendário. A partir desses dados, constrói um vetor de estado simbólico que indexa a Q-table. A ação com maior valor $Q(s, a)$ é então aplicada, e caso o usuário realize uma correção manual, uma nova recompensa é atribuída. Esse fluxo se repete ao longo do tempo, permitindo ao agente refinar sua política de decisão com base nas preferências observadas em diferentes contextos.

4.2. Modelagem de Estado e Percepção de Contexto

O agente coleta variáveis como brilho da tela, volume de mídia, iluminância ambiente (lux), localização geográfica, modo de som e status de reuniões do calendário. Essas variáveis são discretizadas em faixas simbólicas por meio de heurísticas internas, formando um espaço de estados finito e interpretável pelo agente. A coleta ocorre de forma contínua e assíncrona, utilizando recursos nativos do Android como `Handler`, `Looper` e `HandlerThread`, o que permite atualizações frequentes sem impactar o desempenho do dispositivo. Módulos como `LightSensorManager`, `LocationManagerHelper` e `MeetingManager` integram o pipeline de percepção, entregando os dados ao componente `StateBuilder`, que consolida as variáveis em uma chave de estado única. A Figura 1(b) apresenta esse fluxo operacional,

desde a coleta bruta até a entrega do estado simbólico ao agente. Essa estrutura modular permite escalabilidade e conformidade entre serviços, como `LearningService`, `ContextService` e `QLearningAgent`, que operam sobre o estado inferido.



(a) Ciclo de decisão (b) Modelagem de contexto

Figura 1. Arquitetura geral e modelagem de contexto.

4.3. Execução das Ações e Detecção de Feedback

As ações são aplicadas via `Settings.System` e `AudioManager`. Validações são realizadas antes da execução: intervalos mínimos entre ações, limiares de variação e adequação ao ambiente. Se o usuário alterar manualmente brilho ou volume após a ação do agente, essa intervenção é registrada como reforço supervisionado. O par $\langle s, a \rangle$ correspondente recebe reforço direto na Q-table, e a preferência é persistida localmente.

4.3.1. Algoritmo do Agente Q-Learning

O Algoritmo 1 descreve o funcionamento interno do agente Q-Learning embarcado, implementado em Kotlin. A lógica inclui preferências persistentes, inferência contextual e exploração adaptativa, refletindo a operação real do sistema em dispositivos Android.

Algoritmo 1: Agente Q-Learning para ajuste de brilho e volume

Input: Estado atual s_t, Q-table Q, taxa de aprendizado α, fator de desconto γ, política de exploração adaptativa

Output: Ação a_t aplicada e Q-table atualizada

Coletar variáveis contextuais e construir s_t;
if *existe preferência persistente para* s_t **then**
    Selecionar ação coerente com preferência;
else if *há valor dominante (via* `BehaviorGeneralizer`*)* **then**
    Selecionar ação coerente com comportamento dominante;
else if *regras heurísticas se aplicam* **then**
    Executar ação recomendada;
else
    Selecionar a_t usando Softmax adaptativo;
Executar a_t no dispositivo;
Observar s_{t+1};
if *usuário realizou intervenção manual* **then**
    $R_{t+1} \leftarrow -1$;
    Aplicar reforço supervisionado;
else
    $R_{t+1} \leftarrow +1$;
Atualizar Q-table:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)]$$

Persistir Q-table localmente;

4.4. Generalização de Preferências e Inferência

O módulo `BehaviorGeneralizer` permite inferir ações preferenciais para estados ainda não visitados. A inferência baseia-se em três abordagens: (i) consulta a estados similares com preferências armazenadas; (ii) agrupamento contextual por faixa de iluminância, localização ou modo de som; e (iii) aplicação de heurísticas para casos esparsos. A Figura 2 ilustra esse processo: um novo contexto s é transformado em estado base, e o sistema busca registros recentes. Se encontrados, calcula-se o valor dominante; caso contrário, aplica-se uma lógica de inicialização com base no contexto. A ação inferida é então executada, mesmo sem histórico direto. Esse mecanismo reduz falhas de inicialização (*cold start*) e assegura continuidade operacional em transições pouco recorrentes.

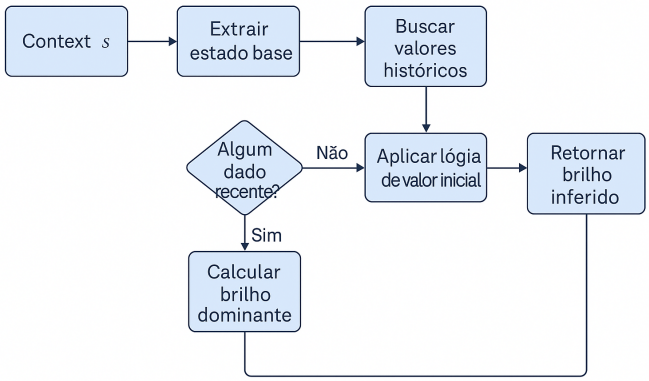


Figura 2. Fluxo de inferência do módulo `BehaviorGeneralizer`.

5. Resultados e Discussões

As análises são baseadas em logs e métricas geradas em tempo real e estão organizadas em quatro dimensões: (i) aprendizado progressivo e ações preferenciais; (ii) estabilidade e abrangência da política; (iii) padrões de decisão em estados recorrentes; e (iv) eficiência da estratégia de exploração.

5.1. Aprendizado Progressivo e Ajustes Otimizados

A Figura 3 exibe as 20 ações com maior valor de Q aprendidas ao longo da execução do sistema. Observa-se uma predominância de decisões relacionadas ao controle de brilho, sobretudo a ação *DiminuirBrilho*, presente em diferentes configurações contextuais. Esses altos Q-values refletem tanto a recorrência dessas ações quanto o reforço cumulativo obtido por meio de recompensas positivas e correções manuais. A análise estatística dos Q-values aprendidos em toda a Q-table revela uma média geral de $\mu = 27{,}10$ e desvio-padrão de $\sigma = 48{,}21$, com valores variando de $-25{,}19$ a $315{,}00$. A mediana de $7{,}0$ indica uma distribuição assimétrica, com alta concentração de estados pouco explorados, contrastando com um subconjunto de ações altamente consolidadas nas extremidades superiores da distribuição. Esses resultados indicam que o agente foi capaz de identificar padrões recorrentes de uso, reforçando ajustes preferenciais mesmo em espaços de estado amplos e esparsos. O valor elevado dos Q-values nas ações do topo denota estabilidade no processo de aprendizado e aderência às preferências detectadas com o tempo.

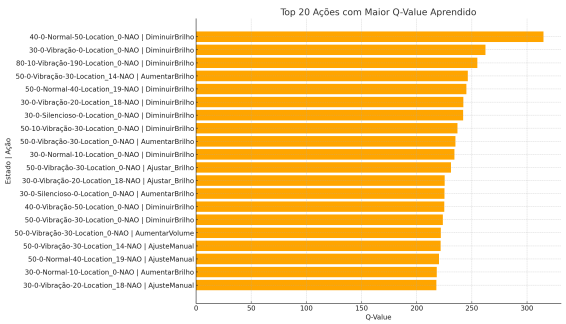
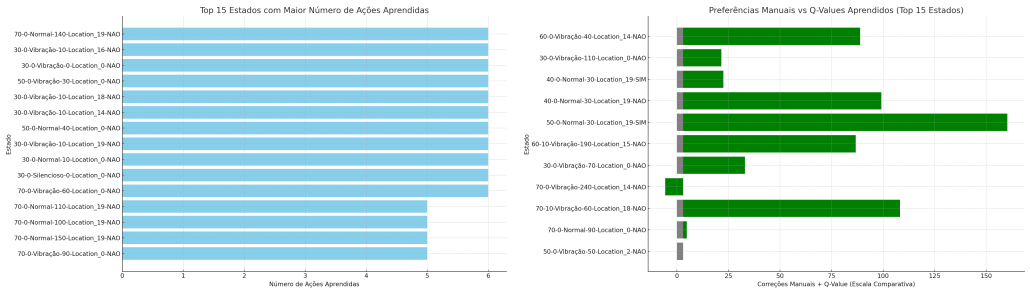


Figura 3. Ações com maior valor de Q consolidado na execução do sistema.

5.2. Estabilidade e Abrangência do Aprendizado

Além de identificar ações preferenciais, é importante analisar como o agente distribui seu aprendizado entre os diferentes estados do ambiente. A Figura 4(a) apresenta os 15 estados com maior número de ações aprendidas, refletindo a estabilidade e a abrangência da política em contextos recorrentes. Essa análise evidencia que o agente manteve um histórico consistente de decisões em estados frequentemente visitados, reforçando múltiplas alternativas de ação conforme o contexto. Isso demonstra que o sistema não apenas consolidou boas ações, mas construiu uma política mais rica e contextualizada ao longo do tempo. Além disso, a convergência entre preferências manuais e ações reforçadas pode ser observada na Figura 4(b), que compara diretamente os registros de correções do usuário com os valores aprendidos pela Q-table.



(a) Estados mais recorrentes (b) Correlação preferências/Q-values

Figura 4. Estabilidade da política e alinhamento com preferências manuais.

5.3. Padrões de Decisão em Estados Dominantes

A Figura 5 apresenta o mapa de calor com os 20 estados mais frequentes na Q-table, considerando os valores aprendidos para cada ação. Essa visualização permite identificar padrões de decisão consolidados ao longo do tempo.

Observa-se que determinados estados apresentam múltiplas ações com valores elevados, refletindo consistência e estabilidade na política aprendida. Estados recorrentes, como aqueles com padrão de áudio "Vibração" e localizações fixas, concentram Q-values significativamente altos, mesmo em ações distintas como *AumentarVolume* e *DiminuirBrilho*. Esse comportamento indica que o agente conseguiu construir uma política mesmo diante de variações contextuais sutis, o que reforça sua capacidade adaptativa diante de variações não determinísticas, como ruído sensorial e alterações ambientais.

Os rótulos utilizados para representar os estados nos gráficos são codificados segundo a estrutura: $Brilho-Volume-ModoAudio-Lux-LocationID-Reuniao$, em que cada campo corresponde a uma variável contextual sensorial capturada durante o processo de decisão. Como exemplo, o estado `30-0-Vibração-10-Location_14-NAO` representa uma situação em que o brilho da tela era 30%, o volume estava em 0, o perfil de som configurado para "Vibração", a luminosidade ambiente era de 10 lux, a localização foi reconhecida como `Location_14` (posição derivada por agrupamento de coordenadas geográficas), e nenhuma reunião estava em andamento.

5.4. Eficiência da Estratégia de Exploração

O mecanismo híbrido de decisão, baseado em Softmax e ε-greedy com decaimento, promoveu um equilíbrio entre descoberta e consolidação de políticas. Inicialmente, o agente

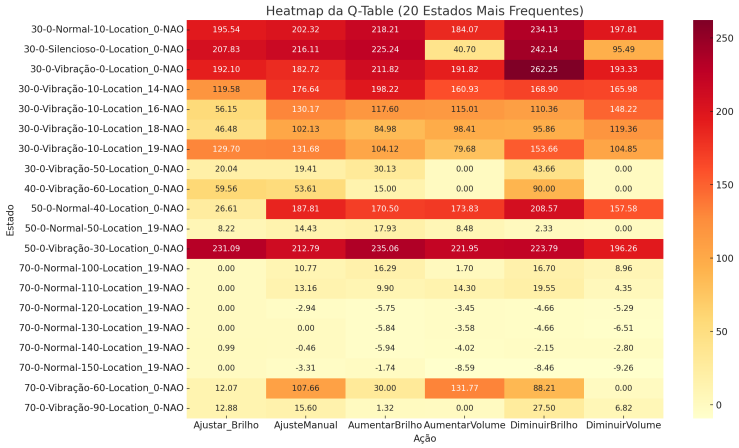


Figura 5. Mapa de calor dos Q-values para os estados mais frequentes.

explorou intensamente, mas após cerca de 291 iterações estabilizou a política em torno das melhores ações.

$$\texttt{Iteração 291 | } \varepsilon = 0.10 \texttt{ | Softmax (T = 0.59)} \rightarrow \texttt{AumentarVolume = 99.99\%}$$

5.5. Avaliação com Usuários

Para avaliação, foi conduzido um experimento com a participação de 10 usuários voluntários, todos passando por todas as fases previstas O experimento foi executado em um período de 7 dias e foram comparadas duas abordagens: (1) **Android Nativo:** sistema de ajustes automáticos padrão do Android; e (2) **Sistema Proposto:** agente Q-Learning embarcado, com feedback supervisionado e generalização. Os grupos apresentados na Tabela 2 foram definidos a partir do autorrelato dos participantes sobre seus perfis de utilização do dispositivo, com validação por meio da análise dos padrões de intervenção manual registrados ao longo do experimento.

(a) Número de Intervenções Manuais: A Tabela 2 apresenta a média de ajustes manuais (brilho ou volume) realizados por usuário em cada abordagem. Observa-se uma redução expressiva de intervenções ao adotar o sistema proposto.

Tabela 2. Média de intervenções manuais por perfil de usuário

Grupo de Usuário	Android Nativo	Sistema Proposto	Redução (%)
Rotina fixa	12,4	3,2	74,2%
Rotina variável	18,1	6,7	63,0%
Alta sensibilidade	21,7	7,9	63,6%

(b) Tempo Médio de Adaptação: O tempo médio estimado para que o sistema convergisse para preferências estáveis por contexto foi de aproximadamente 291 iterações, equivalentes a cerca de dois dias de uso contínuo. Após esse período, mais de 80% das ações do agente não foram corrigidas pelo usuário.

(c) Satisfação do Usuário: Utilizando os critérios do modelo TAM (Technology Acceptance Model), avaliou-se a percepção de utilidade, previsibilidade das ações e conforto

gerado pelo sistema. A Figura 6 apresenta a média de pontuação obtida em escala Likert (1 a 5). A pontuação geral foi de 4,33/5, indicando alta aceitação, mesmo entre usuários com padrões de uso mais exigentes. Os resultados reforçam a hipótese de que o sistema é capaz de aprender preferências rapidamente e atuar de forma confortável e eficaz.

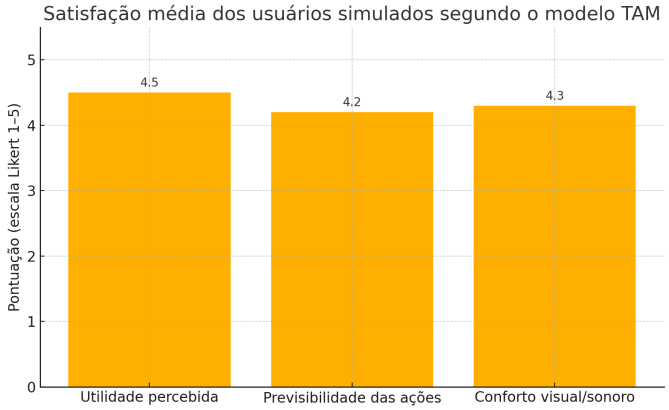


Figura 6. Satisfação média dos usuários simulados segundo o modelo TAM.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou um sistema embarcado para ajuste automático de configurações em dispositivos Android, baseado em aprendizado por reforço sensível ao contexto e operando totalmente offline. A arquitetura combina percepção ambiental, Q-Learning tabular, reforço supervisionado e generalização por similaridade. Os resultados mostraram que o agente aprendeu preferências contextuais, reduziu intervenções manuais e manteve decisões estáveis, mesmo em cenários parcialmente desconhecidos. Entretanto, o sistema ainda apresenta algumas limitações. O vetor de estado é composto por um conjunto reduzido de variáveis, o que restringe a capacidade de adaptação em contextos mais complexos. Além disso, a adoção de uma Q-table tabular, embora funcional e eficiente em pequenos espaços de estado, limita a escalabilidade da solução em domínios com maior granularidade ou variabilidade sensorial.

Como trabalhos futuros, propõe-se a ampliação do vetor de estado com variáveis adicionais como hora do dia, tipo de aplicativo em uso, conexão de rede e detecção de fones de ouvido, visando aumentar a sensibilidade contextual do sistema. Além disso, considera-se a substituição da Q-table por métodos de aproximação de função, como redes neurais (*deep reinforcement learning*), para permitir maior escalabilidade e generalização em tempo real. Pretende-se também adaptar a arquitetura para múltiplos perfis de usuários e conduzir um experimento longitudinal com participantes reais, utilizando instrumentos formais como o modelo TAM para mensuração de aceitação e utilidade percebida.

Agradecimentos

O presente trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (AUXPE-CAPES-PROEX) - Código de Financiamento 001. Este trabalho foi parcialmente financiado pela Fundação de Amparo à Pesquisa do Estado do Amazonas - FAPEAM - por meio do projeto PDPG-CAPES.

Referências

Abeywardhane, J., De Silva, E., Gallanga, I., Rathnayake, L., Wickramarathne, J., e Sriyaratna, D. (2018). Optimization of volume & brightness of android smartphone through clustering & reinforcement learning. In *IC on Inform. and Autom. for Sustainability*.

Ahmadi-Karvigh, H., Kermani, M., e Aghajan, H. (2019). Intelligent adaptive automation: A framework for an activity-driven and user-centered building automation. *Journal of Ambient Intelligence and Smart Environments*, 11(2):101–122.

Altulyan, M., Yao, L., Huang, C., Wang, X., e Kanhere, S. S. (2021). Context-induced activity monitoring for on-demand things-of-interest recommendation in an ambient intelligent environment. *Sensors*, 21(14):4707.

Bai, H., Zhou, Y., Cemri, M., Pan, J., Suhr, A., Levine, S., e Kumar, A. (2024). Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *arXiv preprint arXiv:2406.11896*.

Du, H., Thudumu, S., Nguyen, H., Vasa, R., e Mouzakis, K. (2025). A comprehensive survey on context-aware multi-agent systems: Techniques, applications, challenges and future directions. *arXiv preprint arXiv:2402.01968*.

Kaladevi, A. C., Kumar, V. V., Mahesh, T. R., e Guluwadi, S. (2024). Optimizing personalized and context-aware recommendations in pervasive computing environments. *Int. Journal of Computational Intelligence Systems*, 17(1):300.

Lin, J., Sun, G., Shen, J., Cui, T., Yu, P., Xu, D., e Li, L. (2019). A survey of segmentation, annotation, and recommendation techniques in micro learning for next generation of oer. In *Int. Conf. Computer Suppor. Cooper. Work in Design*, pages 152–157.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Sarker, I. H. (2019). Context-aware rule learning from smartphone data: survey, challenges and future directions. *Journal of Big Data*, 6(95).

Sarker, I. H., Colman, A., Han, J., Khan, A. I., Abushark, Y. B., e Salah, K. (2020). Behavdt: A behavioral decision tree learning to build user-centric context-aware predictive model. *Mobile Networks and Applications*.

Souza, E., Monteiro, E., Barreto, R., e deFreitas, R. (2022). A context-aware automatic smartphone reconfiguration. In *ICCE*, pages 1–7, Las Vegas, NV, USA. IEEE.

Sutton, R. S. e Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press.

Todi, K., Bailly, G., Leiva, L. A., e Oulasvirta, A. (2021). Adapting user interfaces with model-based reinforcement learning. *arXiv preprint arXiv:2103.06807*.

Tokic, M. (2010). Adaptive -greedy exploration in reinforcement learning based on value differences. *KI-Künstliche Intelligenz*, 26(2):159–168.

Wang, D., Zhang, X., Yu, D., Xu, G., e Deng, S. (2021). Came: Content- and context-aware music embedding for recommendation. *IEEE Transactions on Neural Networks and Learning Systems*, 32(3):1375–1388.

Yamasaki, K. et al. (2023). Cluster-aware bayesian optimization for preference inference under cold start. *ACM Trans. on Interactive Intelligent Systems (TiiS)*, 13(2):1–23.