**Especificação Funcional - lmb-hub-websocket**

**Projeto:** lmb-hub-websocket

**Data:** 2025-05-26

**Versão: 1.0**

**Responsável:** [Seu nome ou equipe]

Outra forma para fazer:

|  |  |  |  |
| --- | --- | --- | --- |
| **Projeto** | **Data** | **Versão** | **Responsável** |
| [*lmb-hub-websocket*] | 26/05/2025 | 1.0 | [Seu nome ou equipe] |
|  |  |  |  |

# 1. Objetivo

Descrever de forma clara e objetiva o comportamento do serviço ‘*lmb-hub-websocket*’, que provê comunicação em tempo real entre usuários conectados por WebSocket, usando recursos *serverless* da AWS.

# 2. Visão Geral da Solução

A arquitetura utiliza:

* AWS API Gateway WebSocket para gerenciar conexões;
* Funções Lambda para tratar eventos (*$connect*, *$disconnect*, *sendMessage*);
* DynamoDB para armazenar os ‘*connection\_id*’ vinculados a um ‘*session\_id*’.

Permite comunicação em tempo real entre múltiplos usuários, ideal para dashboards, chats e sistemas de notificação.

# 3. Requisitos Funcionais

* RF1: Registrar conexões ativas no DynamoDB na rota ‘*$connect*’.
* RF2: Remover conexões do DynamoDB na rota ‘*$disconnect*’.
* RF3: Enviar mensagens para os ‘*connection\_id*’ vinculados a ‘*session\_id*’.
* RF4: Validar requisições e tratar erros de forma adequada.

# 4. Requisitos Não Funcionais

* RNF1: Escalabilidade automática via infraestrutura serverless
* RNF2: Log de falhas via CloudWatch
* RNF3: Latência média inferior a 500ms

# 5. Casos de Uso

## 5.1. Conexão ($connect)

* Ator: Navegador
* Descrição: Registra o ‘*connection\_id*’ e o ‘*session\_id*’ no DynamoDB ao conectar.
* Resposta Esperada: 200 OK

## 5.2. Desconexão (*$disconnect*)

* Ator: API Gateway
* Descrição: Remover o ‘*connection\_id*’ do banco.
* Resposta Esperada: 200 OK

## 5.3. Envio de Mensagem (*sendMessage*)

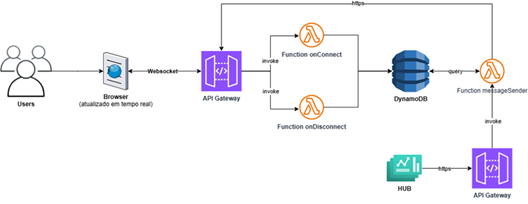
* Ator: Função Lambda chamada pelo HUB
* Descrição: Receber uma lista de ‘*sessions*’ e envia mensagens para todos os ‘*connection\_id*’ vinculados.
* Resposta Esperada: 200 OK

# 6. Estrutura de Dados

|  |  |  |
| --- | --- | --- |
| **Atributo** | **Tipo** | **Descrição** |
| session\_id | S | Identificador da sessão de usuário |
| connection\_id | S | ID da conexão WebSocket |

# 7. Fluxo Arquitetural

O diagrama abaixo representa o fluxo da solução:



# 8. Validações e Tratamento de Erros

* Mensagem sem ‘*sessions*’ → HTTP 400: “*No sessions provided*”
* Conexão inválida → Remove o ‘*connection\_id*’ do banco
* Erro interno → HTTP 500 + log detalhado no CloudWatch

# 9. Segurança e Autenticação

Atualmente o WebSocket não possui autenticação, mas está preparado para futura integração com tokens ou controle de acesso.

# 10. Considerações Finais

Este serviço é modular, escalável e adaptável para múltiplos cenários de comunicação em tempo real, como sistemas de notificação, painéis de monitoramento, chats, entre outros.