

Documento de Descrição de Domínio

Mermã, a Música!

Versão do Projeto: 0.0.1
Versão do Documento: 1.0

Empresa Cliente: Projeto pessoal feito por GABRIEL ADERALDO
Empresa Responsável pelo Documento: GABRIEL VIEIRA SORIANO ADERALDO

Especialista de Domínio: GABRIEL VIEIRA SORIANO ADERALDO
Quality Assurance Lead: GABRIEL VIEIRA SORIANO ADERALDO
Product Manager: GABRIEL VIEIRA SORIANO ADERALDO
Tech Lead: GABRIEL VIEIRA SORIANO ADERALDO
Product Owner (P.O): GABRIEL VIEIRA SORIANO ADERALDO

Data: March 5, 2025

Contents

1	Personas do sistema	1
1.1	Persona 1 - O Competidor Hardcore	1
1.1.1	Informações Gerais	1
1.1.2	Objetivos e Necessidades	1
1.1.3	Desafios e Frustrações	1
1.1.4	Comportamento e Hábitos	1
1.1.5	Frase Representativa	1
1.2	Persona 2 - A Fã de Música Personalizada	1
1.2.1	Informações Gerais	2
1.2.2	Objetivos e Necessidades	2
1.2.3	Desafios e Frustrações	2
1.2.4	Comportamento e Hábitos	2
1.2.5	Frase Representativa	2
1.3	Persona 3 - O Streamer/Influencer	2
1.3.1	Informações Gerais	2
1.3.2	Objetivos e Necessidades	3
1.3.3	Desafios e Frustrações	3
1.3.4	Comportamento e Hábitos	3
1.3.5	Frase Representativa	3
1.4	Persona 4 - O Nostálgico	3
1.4.1	Informações Gerais	3
1.4.2	Objetivos e Necessidades	3
1.4.3	Desafios e Frustrações	3
1.4.4	Frase Representativa	3
1.5	Persona 5 - A Jogadora Social	4
1.5.1	Informações Gerais	4
1.5.2	Frase Representativa	4
2	Contexto do Sistema	1
2.1	Mermã, a Música! - Descrição do Projeto	1
2.2	Arquitetura do Sistema	1
2.3	CORE DOMAIN - Sistema de Partidas	2
2.3.1	Bounded Context	2
2.3.2	Gerenciamento de Partidas	2
2.3.3	Responsabilidades	2
2.3.4	entidades	3
2.3.5	Objetos de Valor	3
2.3.6	Relações com outros Bounded Context	3
2.3.7	Caso de Uso 1: Criar uma Partida	4
2.3.8	Caso de Uso 2: Entrar em uma Sala	6
2.3.9	Caso de Uso 3: Iniciar Partida	7
2.3.10	Caso de Uso 4: Responder a uma Pergunta	7
2.3.11	Caso de Uso 5: Finalizar Rodada	9
2.3.12	Caso de Uso 6: Finalizar Partida	9
2.3.13	MAPA DE CONTEXTO	11

1 Personas do sistema

1.1 Persona 1 - O Competidor Hardcore

Jogador altamente competitivo, interessado em estatísticas, rankings e melhorar sua performance dentro do jogo.

1.1.1 Informações Gerais

Nome: Lucas Oliveira

Idade: 22 anos

Profissão: Universitário de Engenharia da Computação

Localização: Rio de Janeiro, Brasil

Tecnologia: Usa computador e dois monitores para jogar

Experiência: Experiente em jogos de quiz, como AMQ e Kahoot

1.1.2 Objetivos e Necessidades

- Melhorar suas habilidades e ficar no topo do ranking.
- Buscar modos personalizados e estratégias para vencer.
- Ter estatísticas detalhadas de desempenho.

1.1.3 Desafios e Frustrações

- Se irrita com interfaces confusas e lentidão no jogo.
- Quer mais ferramentas para treinar antes das partidas.

1.1.4 Comportamento e Hábitos

- Prefere jogar no computador com várias abas abertas.
- Gosta de participar de torneios e eventos competitivos.
- Discute estratégias e dicas em fóruns e Discord.

1.1.5 Frase Representativa

"Se não tem ranking e estatísticas, eu nem perco meu tempo!"

1.2 Persona 2 - A Fã de Música Personalizada

Jogadora casual que adora criar playlists e compartilhar suas descobertas musicais com amigos.

1.2.1 Informações Gerais

Nome: Ana Martins

Idade: 25 anos

Profissão: Social Media

Localização: São Paulo, Brasil

Tecnologia: Celular e tablet, raramente usa PC

Experiência: Conhece jogos casuais e apps de música

1.2.2 Objetivos e Necessidades

- Jogar sem complicação e por diversão.
- Criar salas com suas playlists personalizadas.
- Compartilhar os resultados com os amigos nas redes sociais.

1.2.3 Desafios e Frustrações

- Não gosta de interfaces complexas ou confusas.
- Se frustra quando não pode compartilhar facilmente seu progresso.

1.2.4 Comportamento e Hábitos

- Usa o celular para jogar casualmente.
- Cria e compartilha playlists em plataformas como Spotify.

1.2.5 Frase Representativa

"Eu só quero me divertir e ouvir boas músicas com meus amigos!"

1.3 Persona 3 - O Streamer/Influencer

Criador de conteúdo que busca formas de interagir com sua audiência através do jogo.

1.3.1 Informações Gerais

Nome: Rafa Souza

Idade: 28 anos

Profissão: Criador de conteúdo (YouTube e Twitch)

Localização: Curitiba, Brasil

Tecnologia: PC gamer, microfone e setup profissional de streaming

Experiência: Experiente em jogos interativos e de quiz

1.3.2 Objetivos e Necessidades

- Criar conteúdo interativo para seu público.
- Ter modos de jogo que permitam interação com os espectadores.
- Compartilhar highlights e placares.

1.3.3 Desafios e Frustrações

- Falta de ferramentas para capturar e compartilhar momentos do jogo.
- Interfaces que não são otimizadas para streaming.

1.3.4 Comportamento e Hábitos

- Faz transmissões ao vivo de jogos interativos.
- Testa novos jogos para atrair audiência.

1.3.5 Frase Representativa

"Esse jogo precisa ser fácil de streamar e interagir com o chat!"

1.4 Persona 4 - O Nostálgico

Jogador que sente saudades do AMQ e quer uma experiência parecida, mas com mais variedade de músicas.

1.4.1 Informações Gerais

Nome: Pedro Lima

Idade: 30 anos

Profissão: Profissional de TI

Localização: Belo Horizonte, Brasil

Tecnologia: PC e ocasionalmente celular

Experiência: Já jogou muito AMQ

1.4.2 Objetivos e Necessidades

- Reviver a experiência do AMQ com novos estilos musicais.
- Criar desafios temáticos com amigos.

1.4.3 Desafios e Frustrações

- Jogos que limitam demais a personalização das salas.

1.4.4 Frase Representativa

"Eu só queria um AMQ, mas com mais músicas além de anime."

1.5 Persona 5 - A Jogadora Social

Joga apenas para se divertir com os amigos e não se importa com competição.

1.5.1 Informações Gerais

Nome: Júlia Costa

Idade: 20 anos

Profissão: Estudante de Artes

Localização: Salvador, Brasil

Tecnologia: Celular

Experiência: Baixa, joga só por diversão

1.5.2 Frase Representativa

"Quero só jogar de boa enquanto converso com meus amigos."

2 Contexto do Sistema

2.1 Mermã, a Música! - Descrição do Projeto

Mermã, a Música! é um jogo de quiz musical rápido, leve e multiplayer, desenvolvido para rodar diretamente no navegador. Inspirado no conceito de jogos como Anime Music Quiz (AMQ), o diferencial do Mermã, a Música! é permitir que os jogadores joguem com músicas normais, utilizando suas próprias playlists de serviços de streaming como Spotify.

O objetivo do jogo é simples: adivinhar o nome da música ou do artista antes que o tempo acabe. Os jogadores podem participar de partidas solo para treino ou desafiar amigos em salas multiplayer com diferentes modos de jogo, incluindo um sistema ranqueado com pontuação baseada no tempo de resposta e acertos. O sistema de ELO, inspirado em jogos competitivos como Valorant e LoL, garante que os melhores jogadores subam no ranking de forma justa.

O jogo é open-source, permitindo contribuições da comunidade para expandir funcionalidades e melhorar a experiência. Sua arquitetura segue princípios de Domain-Driven Design (DDD) e Clean Architecture, garantindo um código modular e organizado. O backend é desenvolvido em Deno com TypeScript, utilizando REST API para operações gerais e WebSockets para sincronização em tempo real. A persistência dos dados combina PostgreSQL (para jogadores, salas e ranking), MongoDB (para playlists e histórico de partidas) e Redis (para cache e tempo real).

A aplicação é hospedada em uma VPS na MagaluCloud, utilizando Docker Compose para orquestrar os serviços, garantindo flexibilidade e escalabilidade. Inicialmente, o jogo será um monólito modular, mas poderá evoluir para microsserviços conforme a necessidade.

Com um design pensado para ser acessível, ágil e competitivo, Mermã, a Música! se propõe a ser o quiz musical definitivo para quem quer testar seus conhecimentos e se divertir com amigos de forma dinâmica e envolvente.

Slogan: "Mermã, tu conhece mesmo essas músicas?"

2.2 Arquitetura do Sistema

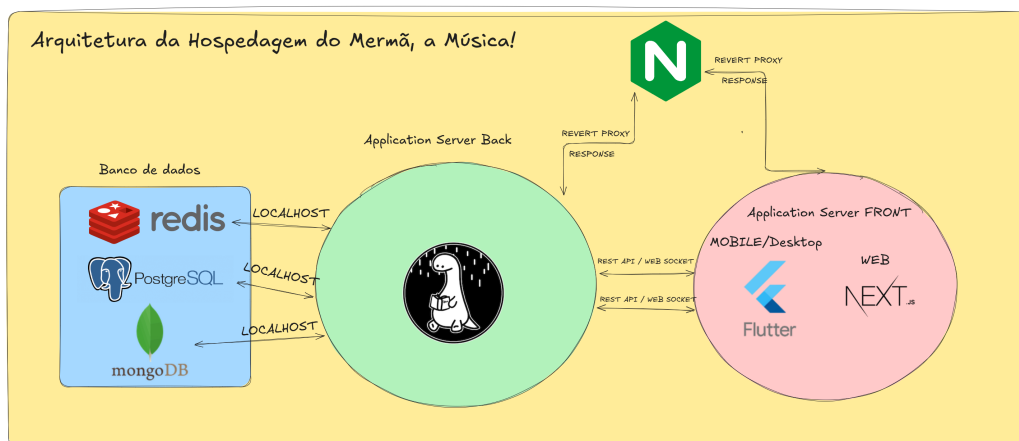


Figure 1: Arquitetura do Projeto Mermã, a Música!

2.3 CORE DOMAIN - Sistema de Partidas

O Sistema de Partidas é o Core Domain do jogo "Mermã, a Música!", responsável por gerenciar o núcleo da experiência de jogo, desde a criação da partida até a exibição dos resultados finais. Ele garante a sincronização em tempo real entre os jogadores, oferecendo uma experiência multiplayer fluida e dinâmica.

2.3.1 Bounded Context

- **Gerenciamento de Partidas:** Responsável por criar, configurar e gerenciar o ciclo de vida das partidas, desde o momento em que são criadas até a finalização e exibição dos resultados. Isso inclui o controle do estado da partida, gerenciamento de jogadores participantes, e aplicação das regras do modo de jogo selecionado.
- **Seleção e Reprodução de Músicas:** Este contexto lida com a seleção aleatória de músicas a partir das playlists dos jogadores, de acordo com as configurações da partida. Ele também é responsável por reproduzir os trechos das músicas para os jogadores adivinharem, garantindo a sincronização entre os jogadores.
- **Registro de Respostas e Pontuação:** Responsável por capturar, validar e processar as respostas dos jogadores durante as rodadas. Ele aplica as regras de acerto (nome da música, artista ou ambos) e calcula a pontuação de cada jogador com base no tempo de resposta e acertos.
- **Ranking e Progressão:** Gerencia o ranking dos jogadores dentro das partidas e no sistema como um todo. Isso inclui o cálculo do ELO em partidas ranqueadas, atualização do ranking global e progressão dos jogadores nos níveis.
- **Sincronização em Tempo Real:** Este contexto garante que todos os jogadores estejam sincronizados durante a partida, utilizando WebSockets para enviar e receber eventos em tempo real, como atualização do placar, mensagens de chat e status da partida.

2.3.2 Gerenciamento de Partidas

O Bounded Context "Gerenciamento de Partidas" é o coração do domínio "Sistema de Partidas", responsável por orquestrar todo o ciclo de vida de uma partida no jogo "Mermã, a Música!". Ele garante que as partidas sejam criadas, configuradas, iniciadas, jogadas e finalizadas de acordo com as regras estabelecidas.

2.3.3 Responsabilidades

- **Criação e Configuração:** Permite aos jogadores criar novas partidas, definindo o modo de jogo (casual, ranqueado, battle royale), as configurações (número de rodadas, tempo de resposta, tipo de acerto, playlists) e os jogadores participantes.
- **Gerenciamento do Ciclo de Vida:** Controla o estado da partida, desde a fase de "aguardando jogadores" até a finalização, passando pelos estados "em andamento", "pausada" e "cancelada".
- **Controle de Jogadores:** Gerencia a entrada e saída de jogadores na partida, garantindo que o número máximo de jogadores não seja excedido e que todos estejam prontos para iniciar.

- **Gerenciamento de Rodadas:** Divide a partida em rodadas, controla o tempo limite para cada rodada e garante a transição suave entre as rodadas.
- **Aplicação de Regras:** Aplica as regras específicas de cada modo de jogo, como o sistema de pontuação, o número de vidas no modo Battle Royale e os critérios de vitória.
- **Finalização e Ranking:** Finaliza a partida quando todas as rodadas forem concluídas ou quando algum critério de finalização for atingido, gera o ranking final da partida e, no caso de partidas ranqueadas, atualiza o ELO dos jogadores.

2.3.4 entidades

- **Partida:** Representa uma instância do jogo em andamento, armazenando informações sobre o modo de jogo, configurações, estado, jogadores, rodadas, ranking e datas de criação, início e fim.
- **JogadorPartida:** Representa um jogador dentro de uma partida específica, armazenando sua pontuação, respostas, vidas (no modo Battle Royale), tempo médio de resposta e estado (ativo, eliminado, desconectado).
- **Sala:** Representa o ambiente virtual onde os jogadores se reúnem antes de iniciar uma partida, armazenando informações sobre o nome da sala, dono, jogadores, configurações da partida, tipo de sala (pública ou privada), número máximo de jogadores, código de convite e estado.

2.3.5 Objetos de Valor

- **ConfiguracoesPartida:** Armazena as configurações da partida, como o tipo de acerto (música, artista ou ambos), número de rodadas, tempo limite para resposta, playlists selecionadas e visibilidade da sala.
- **RankingPartida:** Armazena o ranking final da partida, com a lista de jogadores classificados por pontuação, o vencedor, o modo de ranking (normal ou ranqueado) e as variações de ELO (apenas para partidas ranqueadas).

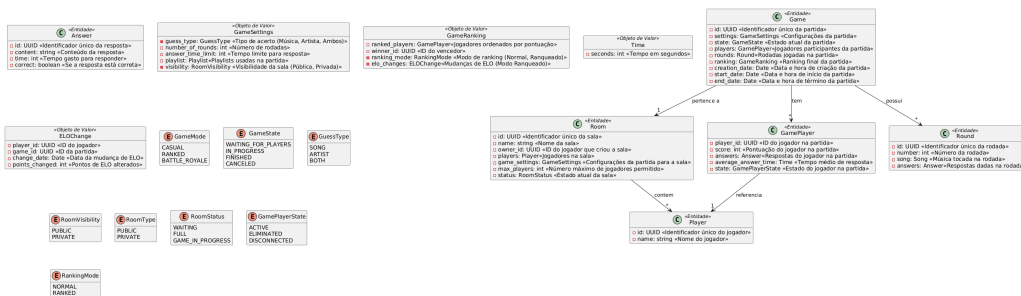


Figure 2: Entidades do Bounded Context "Gerenciamento de Partidas"

2.3.6 Relações com outros Bounded Context

- **Cliente do contexto "Seleção e Reprodução de Músicas":** Solicita músicas para serem utilizadas nas rodadas da partida.

- **Fornecedor para os contextos "Registro de Respostas e Pontuação", "Ranking e Progressão" e "Sincronização em Tempo Real":** Envia informações sobre as respostas dos jogadores, o fim da partida e outros eventos relevantes para esses contextos.

2.3.7 Caso de Uso 1: Criar uma Partida

Ator Principal: Jogador

Objetivo: Criar uma nova sala de espera para iniciar uma partida com outros jogadores, definindo as configurações da partida e a visibilidade da sala.

Pre-Requisito: O jogador deve estar autenticado no sistema.

Fluxo Principal:

1. O jogador solicita a criação de uma nova sala.
2. O sistema solicita ao jogador que defina as configurações da partida:

Modo de Jogo:

Casual: Sem ranking, apenas para diversão.

Ranqueado: Partida competitiva que afeta o ranking do jogador.

Battle Royale: Partida eliminatória onde o último jogador restante vence.

Número de Rodadas: Campo numérico para definir a quantidade de rodadas da partida (valor mínimo: 1, valor máximo: 20).

Tempo Limite de Resposta: Seletor de tempo com opções pré-definidas: (5 segundos, 10 segundos, 15 segundos, 20 segundos) ou campo numérico para definir o tempo limite para responder a cada pergunta (valor mínimo: 5 segundos, valor máximo: 30 segundos).

Tipo de Acerto:

Música: O jogador precisa acertar apenas o nome da música.

Artista: O jogador precisa acertar apenas o nome do artista.

Ambos: O jogador precisa acertar o nome da música e do artista.

Visibilidade da Sala:

Pública: Qualquer jogador pode entrar na sala.

Privada: Apenas jogadores convidados ou com a senha de acesso podem entrar na sala.

3. O jogador define as configurações desejadas.
4. Se o jogador escolher "Privada", o sistema solicita que ele defina uma senha de 4 dígitos para a sala.
5. O jogador define a senha da sala, caso não pula esse passo.
6. O sistema valida as configurações.

7. Se as configurações forem válidas, o sistema cria uma nova sala com as configurações definidas, o jogador como dono e a senha caso seja privada.
8. O sistema adiciona o jogador à sala.
9. O sistema informa ao jogador que a sala foi criada com sucesso, incluindo o nome da sala e a senha.

REGRAS DE NEGOCIO:

- **Sala Cheia:** Um jogador não pode entrar em uma sala que já atingiu o número máximo de jogadores definido na criação da sala.
- **Partida Iniciada:** Um jogador não pode entrar em uma sala se a partida já tiver sido iniciada.

- **Senha da Sala**

A senha deve ter exatamente 4 dígitos sendo todos numericos.

Para salas públicas, qualquer jogador pode entrar sem necessidade de senha.

Para salas privadas, o jogador precisa fornecer a senha correta para entrar.

Se o jogador fornecer uma senha incorreta, o sistema deve exibir uma mensagem de erro e impedir a entrada na sala.

- **Jogador já na Sala:** Um jogador não pode entrar em uma sala na qual ele já está presente.

- **Estado do Jogador:**

Um jogador não pode entrar em uma sala se estiver banido ou suspenso.

Um jogador que foi desconectado de uma partida em andamento pode reentrar na mesma sala, caso a partida ainda esteja acontecendo.

O criador da sala é classificado como dono da sala

O dono da sala pode definir configurações adicionais, como expulsar jogadores, iniciar a partida e alterar as configurações da sala.

O dono da sala sair da sala, a liderança é transferida para outro jogador presente na sala.

O dono da sala pode transferir a liderança para outro jogador presente na sala.

- **Disponibilidade da Sala:**

Uma sala pode ser removida da lista de salas disponíveis se a partida for iniciada ou se o dono da sala a cancelar.

Uma sala privada pode ser removida da lista de salas disponíveis se o dono da sala a fechar ou se tornar inacessível após um determinado tempo de inatividade.

- **Conexão com a Sala:**

O sistema deve garantir que o jogador tenha uma conexão estável com a sala.

Se o jogador perder a conexão com a sala, ele deve ser removido da lista de jogadores e notificado.

O sistema deve tentar reconectar o jogador automaticamente, caso a conexão seja perdida temporariamente.

2.3.8 Caso de Uso 2: Entrar em uma Sala

Ator Principal: Jogador

Objetivo: Entrar em uma sala de espera existente para participar de uma partida.

Pre-Requisito: O jogador deve estar autenticado no sistema.

Fluxo Principal:

1. O jogador solicita a entrada em uma sala, informando o nome ou código da sala.
2. O sistema verifica se a sala existe.
3. Se a sala existir, o sistema verifica se ela é pública ou privada.
4. Se a sala for privada, o sistema solicita a senha ao jogador.
5. O jogador informa a senha da sala (se solicitada).
6. O sistema verifica se a senha está correta (se aplicável).
7. O sistema verifica se a sala ainda não está cheia e se a partida ainda não foi iniciada.
8. Se a sala estiver disponível, o sistema adiciona o jogador à sala.
9. O sistema informa ao jogador que ele entrou na sala com sucesso.
10. O sistema notifica os demais jogadores na sala sobre a entrada do novo jogador.

REGRAS DE NEGOCIO:

- **Sala Cheia:** Um jogador não pode entrar em uma sala que já atingiu o número máximo de jogadores.
- **Partida Iniciada:** Um jogador não pode entrar em uma sala se a partida já tiver sido iniciada.
- **Senha da Sala:**
 - Salas públicas podem ser acessadas por qualquer jogador sem senha.
 - Salas privadas exigem a senha correta para acesso.
 - Se a senha estiver incorreta, o sistema informa o erro ao jogador e impede a entrada na sala.
- **Jogador já na Sala:** Um jogador não pode entrar em uma sala na qual ele já está presente.
- **Estado do Jogador:** Um jogador banido ou suspenso não pode entrar em uma sala.

2.3.9 Caso de Uso 3: Iniciar Partida

Ator Principal: Lider da Sala

Objetivo: Iniciar a partida na sala de espera, após a configuração da partida e a entrada dos jogadores.

Pre-Requisito: O lider da sala tem está em uma sala e todos os jogadores tem que está com estado de "pronto"

Fluxo Principal:

1. O dono da sala clica no botão "Iniciar Partida".
2. O sistema verifica se há pelo menos dois jogadores na sala.
3. O sistema verifica se todos os participantes então em estado de "pronto"
4. Se houver jogadores suficientes, o sistema altera o estado da sala para "Partida em Andamento".
5. O sistema cria uma nova instância de Partida com as configurações definidas na sala.
6. O sistema associa os jogadores da sala à instância da Partida.
7. O sistema obtém as playlists dos jogadores participantes.
8. O sistema inicia a primeira rodada da partida.
9. O sistema notifica todos os jogadores da sala que a partida foi iniciada.
10. O sistema redireciona todos os jogadores para a tela da partida.

REGRAS DE NEGOCIO:

- **Número Mínimo de Jogadores:** Uma partida só pode ser iniciada se houver pelo menos dois jogadores na sala.
- **Todos os jogadores devem está prontos:** A partida só inicia quando todos os jogadores estão em estado de "pronto".
- **Lider da Sala Inicia:** Apenas o Lider da Sala pode iniciar a partida.
- **Estado da Sala:** A sala deve estar no estado "Aguardando Jogadores" para que a partida possa ser iniciada.

2.3.10 Caso de Uso 4: Responder a uma Pergunta

Ator Principal: Jogador

Objetivo: Responder a uma pergunta sobre a música que está sendo tocada na rodada atual da partida, com a possibilidade de múltiplas tentativas até o tempo limite.

Pre-Requisito: A partida tem que ser válida e está em andamento

Fluxo Principal:

1. O sistema apresenta uma pergunta ao jogador sobre a música que está sendo tocada na rodada atual.

2. O jogador analisa a pergunta e as opções de resposta.
3. O jogador seleciona a opção de resposta que acredita ser a correta.
4. O sistema registra a resposta do jogador e o tempo em que ela foi dada.
5. O jogador pode continuar respondendo à mesma pergunta, alterando sua resposta quantas vezes desejar, até o tempo limite.
6. Quando o tempo limite da rodada expirar, o sistema considera a última resposta dada pelo jogador como a resposta final.
7. O sistema verifica se a resposta final do jogador está correta.
8. Se a resposta final estiver correta, o sistema calcula a pontuação do jogador com base no tempo da última resposta.
9. O sistema atualiza a pontuação do jogador na partida.
10. O sistema notifica o jogador sobre o resultado da sua resposta final (correta ou incorreta) e a pontuação obtida.
11. O sistema verifica se a rodada terminou (todos os jogadores responderam ou o tempo limite da rodada expirou).
12. Se a rodada terminou, o sistema inicia a próxima rodada ou finaliza a partida, caso seja a última rodada.

REGRAS DE NEGOCIO:

- **Tempo Limite:** O jogador tem um tempo limite para responder a cada pergunta, definido nas configurações da partida.
- **Múltiplas Tentativas:** O jogador pode responder à mesma pergunta várias vezes dentro do tempo limite.
- **Última Resposta:** O sistema considera apenas a última resposta dada pelo jogador como a resposta final.
- **Pontuação:** A pontuação do jogador é calculada com base no tempo da última resposta e no tipo de acerto (música, artista ou ambos), conforme definido nas configurações da partida.
- **Tipo de Pergunta:** O tipo de pergunta (música, artista ou ambos) varia de acordo com as configurações da partida.
- **Registro de tempo de resposta:** O sistema deve registrar todas as respostas do jogador e seus respectivos tempos, mas apenas a última resposta é considerada para o cálculo da pontuação.

2.3.11 Caso de Uso 5: Finalizar Rodada

Ator Principal: Sistema

Objetivo: Finalizar uma rodada da partida, calcular a pontuação dos jogadores e determinar o próximo passo do jogo (próxima rodada ou finalização da partida).

Pre-Requisito: A partida tem que ser válida e está em andamento

Fluxo Principal:

1. O sistema aguarda até que todos os jogadores tenham respondido à pergunta da rodada ou o tempo limite da rodada expire.
2. O sistema verifica as respostas de cada jogador e calcula a pontuação individual de acordo com as regras de pontuação definidas nas configurações da partida.
3. O jogador seleciona a opção de resposta que acredita ser a correta.
4. O sistema atualiza a pontuação total de cada jogador na partida.
5. O sistema exibe o resultado da rodada para todos os jogadores, incluindo a resposta correta, a pontuação de cada jogador e o ranking da rodada.
6. O sistema verifica se a rodada finalizada era a última rodada da partida.
7. Se houver mais rodadas, o sistema inicia a próxima rodada.
8. Se a rodada finalizada era a última, o sistema finaliza a partida.

REGRAS DE NEGOCIO:

- **Término da Rodada:** Uma rodada termina quando todos os jogadores responderam à pergunta ou o tempo limite da rodada expirou.
- **Cálculo da Pontuação:** A pontuação de cada jogador é calculada com base em sua resposta e no tempo gasto para responder, conforme definido nas configurações da partida.
- **Ranking da Rodada:** O sistema classifica os jogadores de acordo com sua pontuação na rodada.
- **Pontuação:** A pontuação do jogador é calculada com base no tempo da última resposta e no tipo de acerto (música, artista ou ambos), conforme definido nas configurações da partida.
- **Progressão da Partida:** O sistema controla a progressão da partida, verificando se há mais rodadas a serem jogadas ou se a partida deve ser finalizada.

2.3.12 Caso de Uso 6: Finalizar Partida

Ator Principal: Sistema

Objetivo: Finalizar a partida, calcular o ranking final, atualizar o ELO dos jogadores (se aplicável) e realizar outras ações necessárias ao término da partida.

Pre-Requisito: A partida tem que ser válida e está em andamento

Fluxo Principal:

1. O sistema verifica se a última rodada da partida foi finalizada.
2. Se a última rodada foi finalizada, o sistema calcula a pontuação final de cada jogador.
3. O sistema gera o ranking final da partida, classificando os jogadores por pontuação.
4. Se a partida for ranqueada, o sistema atualiza o ELO dos jogadores com base em seus resultados.
5. O sistema armazena os resultados da partida, incluindo o ranking final e as pontuações individuais.
6. O sistema exibe a tela de resultados da partida para todos os jogadores, mostrando o ranking final, a pontuação de cada jogador e outras estatísticas relevantes.
7. O sistema encerra a instância da **Partida**.
8. O sistema libera os jogadores da sala.
9. O sistema disponibiliza opções para os jogadores, como:
 - Iniciar uma nova partida fazendo os jogadores voltarem direto para a SALA.
 - Voltar para a tela inicial, fazendo os jogadores voltarem para a seleção de salas ou home.
 - Visualizar o perfil dos jogadores.
 - Compartilhar os resultados nas redes sociais.

REGRAS DE NEGOCIO:

- **Finalização da Partida:**A partida é finalizada quando a última rodada é concluída.
- **Cálculo do Ranking:**O ranking final é calculado com base na pontuação total de cada jogador na partida.
- **Atualização do ELO:**Em partidas ranqueadas, o ELO dos jogadores é atualizado de acordo com as regras de cálculo do ELO.
- **Armazenamento dos Resultados:** Os resultados da partida são armazenados para fins de histórico e estatísticas.

2.3.13 MAPA DE CONTEXTO

MAP CONTEXT DO DOMINIO: Sistema de Partidas

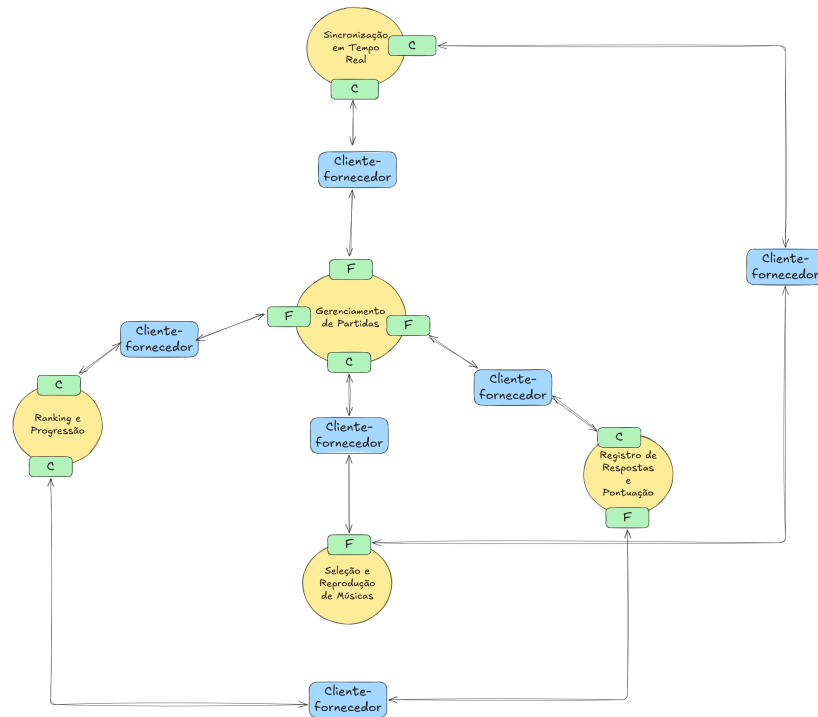


Figure 3: Mapa de Contexto do Sistema de Partidas;

Cliente - Servidor: O contexto "Clinte" requisita informações do contexto "Fornecedor" para realizar suas tarefas.

"C" -> Cliente || "F" -> Fornecedor