

# Relatório de Atualização e Análise do Plugin Vincere System

Data: 08 de Maio de 2025

## 1. Introdução

Este relatório detalha as atividades realizadas na análise, correção e melhoria do plugin "game-system-plugin" (Vincere System) para o site Vincere Club. O foco principal foi a revisão de segurança, implementação de funcionalidades críticas de controle de acesso baseadas em planos de assinatura e a preparação do plugin para futuras expansões.

## 2. Análise Inicial e Auditoria

- **Análise do Código-Fonte:** Todos os arquivos principais do plugin foram revisados para entender a arquitetura, funcionalidades existentes e a estrutura do banco de dados.
- **Identificação de Funcionalidades:** Foram identificadas as funcionalidades já implementadas (sistemas de PUGs, Lobbys, Elo, Ranking, etc.) e aquelas que estavam pendentes ou parcialmente implementadas.
- **Auditoria de Segurança e Boas Práticas:**
  - **Nonces:** Verificada a utilização de nonces em formulários e ações AJAX.
  - **Validação e Sanitização:** Revisadas as entradas de dados ( `$_POST` , `$_GET` ). Foram aplicadas sanitizações mais robustas (e.g., `absint()` para IDs numéricos, validação contra listas de valores permitidos para tipos de ação).
  - **Prevenção contra SQL Injection:** Confirmado o uso de `$wpdb->prepare()` .
  - **Prevenção contra XSS:** Verificada a necessidade de escape em dados exibidos.
  - **Controle de Acesso/Permissões (Capabilities):** Identificada a ausência de verificação de planos de assinatura e capabilities granulares como um ponto CRÍTICO.
  - **Boas Práticas WordPress:** Avaliado o uso de hooks, enfileiramento de scripts/estilos, internacionalização (identificada a necessidade de melhoria), prefixos (identificada a necessidade de padronização) e estrutura geral.

## 3. Melhorias e Correções Implementadas (Itens Prioritários)

As seguintes melhorias e correções foram implementadas nos arquivos do plugin:

### 3.1. Controle de Acesso e Planos de Assinatura (CRÍTICO)

- **Função `vincere_user_has_access()` :**
  - Criada no arquivo `includes/helpers.php` .
  - Esta função serve como um placeholder para a lógica de verificação do plano de assinatura do usuário (Básico/Premium) que deverá ser integrada com WooCommerce Subscriptions ou JetEngine.
  - Atualmente, simula o acesso para `user_id = 1` (Premium) e `user_id = 2` (Basic) para fins de desenvolvimento e teste.
- **Aplicação da Verificação de Acesso:**
  - A função `vincere_user_has_access()` foi integrada nos shortcodes e funções AJAX dos sistemas de Filas (PUGs) e Lobbys para restringir o acesso com base no plano simulado.
  - Arquivos modificados: `includes/sistema-de-filas/queue-shortcode.php` e `includes/sistema-de-lobby/lobby-shortcodes.php` .

### 3.2. Capabilities do WordPress (CRÍTICO)

- **Definição e Registro de Capabilities:**
  - Novas capabilities foram definidas e são registradas na ativação do plugin ( `game-system-plugin.php` ):
    - `vincere_access_pugs`
    - `vincere_create_lobby`
    - `vincere_join_lobby`
    - `vincere_access_premium_content` (uso principal via `vincere_user_has_access` )
  - Utilizado o prefixo `vincere_` para consistência.
- **Uso de `current_user_can()` :**
  - As verificações de `is_user_logged_in()` foram complementadas ou substituídas por `current_user_can()` utilizando as novas capabilities nos arquivos:
    - `includes/sistema-de-filas/queue-shortcode.php`
    - `includes/sistema-de-lobby/lobby-shortcodes.php`

### 3.3. Validação e Sanitização de Entradas

- **IDs Numéricos:**
  - Em `queue-shortcode.php` e `lobby-shortcodes.php`, os IDs recebidos via AJAX (e.g., `queue_id`, `team_id`) agora são sanitizados utilizando `absint()`.
- **Tipos de Ação:**
  - Em `queue-shortcode.php`, o parâmetro `action_type` é validado contra uma lista de valores permitidos (`join`, `leave`).

### 3.4. Boas Práticas WordPress

- **Internacionalização (i18n):**
  - Adicionadas funções de tradução do WordPress (e.g., `__()`, `_e()`, `esc_html__()`, `_n()`) com o text domain `game-system-plugin` em diversas strings visíveis ao usuário nos arquivos:
    - `includes/sistema-de-filas/queue-shortcode.php`
    - `includes/sistema-de-lobby/lobby-shortcodes.php`
  - O arquivo principal `game-system-plugin.php` foi atualizado para carregar o text domain.
- **Prefixos:**
  - Padronizado o uso do prefixo `VINCERE_PREFIX` (definido como `vincere_`) para constantes, hooks de ativação, categorias de widget Elementor e handles de scripts/estilos no `game-system-plugin.php`.
  - A versão do plugin foi atualizada para `1.0.1`.
- **Comentários e Estrutura:**
  - Adicionados comentários explicativos em diversas partes do código modificado.
  - Realizadas pequenas refatorações para melhorar a clareza e a robustez (e.g., verificações `is_array` antes de iterar sobre `$queues` e `$teams`).
- **Hooks AJAX:**
  - Removidas as actions `wp_ajax_nopriv_` para `game_system_process_queue` e `game_system_get_queue_state`, pois estas ações requerem que o usuário esteja logado e tenha as permissões/assinaturas adequadas.

## 4. Funcionalidades Pendentes e Próximos Passos

Apesar das melhorias implementadas, diversas funcionalidades centrais e outras melhorias ainda estão pendentes e são cruciais para a completude do plugin:

## 4.1. Integração Real com Planos de Assinatura

- A função `vincere_user_has_access()` precisa ser conectada ao sistema real de planos de assinatura (WooCommerce Subscriptions / JetEngine) para verificar o status e o tipo de plano do usuário de forma dinâmica.

## 4.2. Sistema de Filas (PUGs)

- **Criação Automática de Partida:** Implementar um mecanismo robusto (e.g., WP Cron ou AJAX aprimorado) para disparar `QueueSystem::createMatch` quando a fila atingir 10 jogadores.
- **Página da Partida ( `[game_match]` ):**
  - Desenvolver a interface para exibir informações da partida (times, jogadores, mapa sorteado, lados GR/BL).
  - Implementar o sistema de votação/reporte do time vencedor.
- **Finalização da Partida:** Lógica no backend para processar o resultado da votação, chamar `QueueSystem::finishMatch`, distribuir pontos de ranking e ELO.
- **Sorteio de Lados (GR/BL):** Implementar e exibir na página da partida.
- **Correção de Bug:** Investigar e corrigir o potencial bug de dupla contagem de score em `QueueSystem::finishMatch`.

## 4.3. Sistema de Lobby

- **Gerenciamento de Time pelo Líder:** Implementar a funcionalidade para o líder remover jogadores.
- **Exclusão de Time:** Implementar a lógica de exclusão de time pelo líder (após remover todos os membros).
- **Sistema de Desafios:** Desenvolver interface e lógica para times completos desafiarem outros, e para aceitar/recusar desafios.
- **Criação de Partida de Lobby:** Após desafio aceito, criar a partida, definir/sortear mapa e lados.
- **Página da Partida de Lobby ( `[lobby_match]` ):** Desenvolver interface, sistema de votação e finalização da partida, similar ao PUG.
- **Refatoração `class-lobby-system.php`:** Esta classe parece ser um placeholder e precisa ser desenvolvida ou integrada corretamente com `LobbyManager.php`.

## 4.4. Sistema de ELO

- Revisar e garantir que a lógica de ganho/perda de ELO seja justa e equilibrada para PUGs e Lobbys após a implementação completa das partidas.

## 4.5. Segurança e Boas Práticas (Continuação)

- **Validação em `admin-configurations.php`** : Implementar validação para `delete_map` e `delete_feedback_category` .
- **Internacionalização (i18n)**: Continuar a revisão de todos os arquivos para garantir cobertura completa de tradução.
- **Prefixos**: Garantir consistência de prefixos em todas as funções globais, classes e outros elementos do plugin.
- **Comentários e Documentação (PHPDoc)**: Adicionar documentação mais completa em todo o código.
- **Tratamento de Erros e Logging**: Padronizar e aprimorar o sistema de logs para facilitar o debugging.

## 4.6. Interface do Usuário (UX) e Frontend

- Revisar e melhorar a usabilidade e o design das interfaces geradas pelos shortcodes e widgets Elementor.
- Garantir responsividade para dispositivos móveis.

## 4.7. Testes Funcionais

- É crucial que você realize testes funcionais completos em um ambiente WordPress com WooCommerce/JetEngine configurado para simular os planos de assinatura.  
Testar:
  - Controle de acesso por plano (Básico/Premium) para PUGs e Lobbys.
  - Funcionamento das capabilities.
  - Fluxos de entrar/sair de filas e times.
  - Criação de times.
  - Validação de entradas e mensagens de erro.

## 5. Conclusão

As modificações realizadas estabeleceram uma base mais segura e robusta para o plugin Vincere System, especialmente no que tange ao controle de acesso. No entanto, uma quantidade significativa de trabalho ainda é necessária para completar as funcionalidades principais e refinar o plugin. Recomenda-se priorizar a implementação das funcionalidades pendentes dos sistemas de PUGs e Lobbys, seguida pela integração real com os sistemas de assinatura.

## 6. Arquivos Modificados (Principais)

- `game-system-plugin.php` (arquivo principal do plugin)
- `includes/helpers.php`
- `includes/sistema-de-filas/queue-shortcode.php`
- `includes/sistema-de-lobby/lobby-shortcodes.php`

O código-fonte completo e atualizado do plugin é fornecido em anexo ( `plugin_game_system_atualizado.zip` ).