

Bot de Discord para Gestão de Criaturas no Jogo Ark: Survival Ascended

Mayko D. M. Amaral, Rafael D. F. Feitosa

Instituto Federal Goiano – Campus Ceres (IFGoiano) – Ceres, GO – Brasil

mayko.amaral@estudante.ifgoiano.edu.br, rafael.feitosa@ifgoiano.edu.br

Abstract. *The development of a Discord bot focused on organizing creature information in ARK: Survival Ascended provides a practical solution for players struggling to manage data manually. With features to add, list, edit, and delete information, the bot promotes organized and accessible management in a familiar environment. Developed in Python, with support from Docker and GitHub, it ensures efficiency and security while complying with Discord's privacy policies. The project envisions future improvements, such as automating data capture, further enhancing its functionality.*

Resumo. *O desenvolvimento de um bot para Discord voltado à organização das informações das criaturas em ARK: Survival Ascended oferece uma solução prática para jogadores que têm dificuldade em gerenciar dados manualmente. Com funcionalidades para adicionar, listar, editar e excluir informações, o bot promove uma gestão organizada e acessível em um ambiente familiar. Desenvolvido em Python, com suporte de Docker e GitHub, garantindo eficiência e segurança, em conformidade com as políticas de privacidade do Discord. O projeto prevê melhorias futuras, como a automação da captura de dados, ampliando sua funcionalidade.*

1. Introdução

A utilização de plataformas se destaca por facilitar o acesso e a personalização da informação necessária para o dia a dia, além de aprimorar processos de decisão e centralizar informações de diversas fontes, cumprindo uma função integradora (Bax & Barbosa, 2009).

Segundo Batista (2023), a utilização de tecnologias de informação e comunicação é fundamental, pois fornece agilidade, praticidade e segurança na gestão de informações essenciais. Esses recursos registram e processam dados, tornando o gerenciamento mais eficiente quando comparado a métodos tradicionais, como o uso de papel.

Ferramentas que permitem o armazenamento seguro e a consulta ágil de dados contribuem para uma administração eficaz e menos sujeita a erros. No contexto de jogos *online*, na qual a complexidade dos dados e a necessidade de atualizações constantes são evidentes, ferramentas para salvar e consultar informações tornam-se ainda mais relevantes.

ARK: Survival Ascended (2023), uma versão remasterizada de *ARK: Survival Evolved* (2015) produzida pela *Studio Wildcard*, oferece uma experiência única. Nele, o jogador, como sobrevivente, enfrenta adversidades climáticas e biológicas em uma ilha misteriosa enquanto evolui por meio de um sistema de progressão. Nesse estilo de

sobrevivência *online*, cada jogador coleta recursos, caça, doma criaturas e interage com outros jogadores em um ambiente compartilhado (ARK Wiki, s.d.).

Mais de cem criaturas selvagens podem ser domesticadas no *ARK: Survival Ascended*, cada uma com um processo de domesticação próprio e estatísticas individuais, como saúde, estamina, oxigênio, fome, peso e dano de combate. As criaturas domesticadas podem ser cruzadas para criar descendentes com características selecionadas, utilizando herança genética recombinante. Esse processo envolve a incubação de ovos e ciclos de gestação no caso de mamíferos (ARK Wiki, s.d.).

As criaturas são utilizadas em batalhas contra outras criaturas e jogadores, o que gera a necessidade de possuir espécimes com estatísticas elevadas. Isso incentiva um elevado número de acasalamentos entre elas, configurando um ciclo contínuo voltado à melhoria progressiva dessas estatísticas.

Gerenciar as estatísticas das criaturas no *ARK: Survival Ascended* é uma tarefa complexa e suscetível a erros quando feita manualmente. Para ajudar os jogadores, este trabalho propõe o desenvolvimento de um *bot* para *Discord* que registre manualmente as estatísticas, salvando-as e disponibilizando-as de forma acessível sempre que necessário. Esse *bot* busca facilitar o armazenamento e a consulta dessas informações, contribuindo para uma gestão mais organizada e eficiente.

A relevância deste trabalho tornou-se evidente após uma investigação detalhada na plataforma *Tog.gg*, uma base amplamente utilizada para busca e publicação de *bots* para *Discord*. Embora existam *bots* dedicados ao jogo *ARK: Survival Ascended*, suas propostas estão voltadas para funcionalidades diferentes. Nenhum deles aborda a gestão específica das estatísticas de criaturas, evidenciando uma lacuna no atendimento a essa demanda. Essa constatação reforça a importância do presente trabalho em oferecer uma solução inovadora e direcionada às necessidades dos jogadores.

A ferramenta proposta busca substituir o gerenciamento manual, geralmente realizado em blocos de notas ou planilhas. Com base nos requisitos levantados, o desenvolvimento busca alinhar-se aos termos de serviço e às políticas de privacidade do *Discord*, garantindo uma solução conforme as diretrizes e boas práticas da plataforma.

2. Fundamentação Teórica

Os *bots* são softwares que variam desde *scripts* automatizados até agentes que realizam tarefas de forma independente, acionando-se com base em condições preestabelecidas (Santhanam et al., 2022).

O *Discord* é uma plataforma que facilita a comunicação virtual entre usuários de diferentes sistemas operacionais. A aplicação oferece recursos variados, incluindo mensagens de texto, chamadas de áudio e vídeo, além de permitir o uso de jogos na interface. Esses recursos são utilizados em conversas pessoais ou em grupos, conhecidos como servidores, que são coleções de canais organizados para a interação entre grupos de pessoas (Abhinand & Balasubramanian, 2022).

No *Discord*, os servidores conectam usuários e podem incluir *bots* que desempenham atividades automatizadas, simulando ações humanas. Esses *bots* respondem automaticamente a eventos e comandos, sendo amplamente utilizados para atender a diversas necessidades na plataforma e facilitar a interação entre membros em várias aplicações (Sielliagia et al., 2024).

3. Materiais e Métodos

O desenvolvimento do *bot* foi estruturado em duas etapas principais: levantamento de requisitos e desenvolvimento do código, adotando um ciclo de desenvolvimento contínuo que permitiu a integração e testes regulares de novas funcionalidades e melhorias. Na etapa de levantamento de requisitos, foram identificadas e documentadas as necessidades e funcionalidades do *bot*, com foco na interação com o *Discord* e nos dados a serem gerenciados. Na fase de desenvolvimento do código, foram realizadas a concepção e implementação do *bot* conforme os requisitos definidos.

Para sua implementação, o *bot* empregou várias tecnologias. O *Discord* (versão 1.0.9172) foi utilizado como plataforma de comunicação; *Python* (versão 3.12.3) foi a linguagem de programação principal, enquanto *Discord.py* (versão 2.3.2) possibilitou a integração do *bot* com a *API* do *Discord*. O *SQLAlchemy* (versão 2.0.29) foi usado para gerenciar operações com o banco de dados *MySQL* (versão 8.0.39), administrado via *Adminer* (versão 4.8.1). O *Docker* (versão 26.1.1) garantiu a consistência do ambiente de desenvolvimento em diferentes sistemas, enquanto o *Visual Studio Code* (versão 1.95.3) facilitou a codificação. O *GitHub* (versão 3.15.0) foi utilizado para controle de versão, e o *Visual Paradigm Online* (versão 17.2.0) auxiliou na criação de diagramas *UML*, facilitando a visualização e documentação dos processos.

4. Resultados e Discussão

O desenvolvimento do *bot* para *Discord* voltado à gestão de estatísticas em *ARK: Survival Ascended* resultou em uma ferramenta multifuncional, que permite aos jogadores gerenciar não apenas as informações das criaturas, mas também realizar a gestão de tribos, consultar ajuda sobre comandos e utilizar os recursos do *bot* em múltiplos idiomas. Na figura 1 é apresentado a interface visual do comando `/ajuda` e `/tribo`.

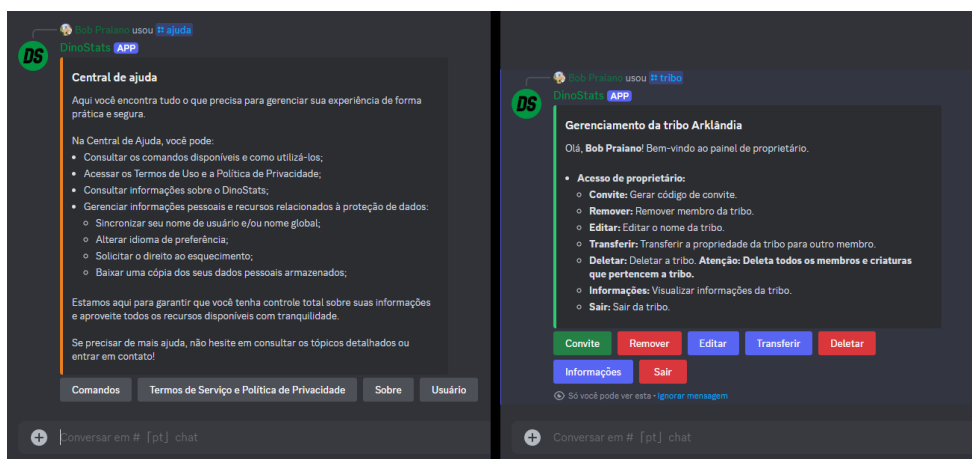


Figura 1. Comandos `/ajuda` e `/tribo`

No entanto, algumas limitações do *Discord* influenciaram o design da interface e a implementação das funcionalidades. Restrições no número de campos e caracteres nos *embeds* limitaram a quantidade de informações que podem ser exibidas em uma única mensagem. A estrutura de *views* também possui um limite de cinco linhas com cinco itens cada, na qual o menu *select* ocupa todo o limite.

Para contornar essas restrições, foi utilizado sistema de atualização da mensagem enviada no canal de texto pelo *bot* que adiciona e remove *views* conforme a

necessidade do usuário. Buscando evitar o envio de múltiplas mensagens ao listar várias criaturas ou tribos, desenvolveu-se um sistema de paginação que permite ao jogador navegar entre diferentes páginas de informações na mesma mensagem, atualizando os dados ao clicar em um botão. Essa solução proporciona uma experiência de navegação mais fluida e organizada.

5. Considerações Finais

O desenvolvimento de um *bot* para *Discord* voltado à gestão das estatísticas das criaturas no jogo *ARK: Survival Ascended* tem se mostrado uma solução prática e eficiente para jogadores que lidam com o complexo processo de gerenciamento manual de dados.

Embora o projeto ainda esteja em andamento, ele já possibilita a exploração de tecnologias e práticas de desenvolvimento atuais, como o uso de *APIs* e o gerenciamento de banco de dados. Testes de carga e monitoramento de recursos estão sendo realizados para avaliar o desempenho, tempo de resposta e consumo eficiente do bot. A análise e a conformidade com as políticas de privacidade do *Discord* garantem que o *bot* respeitará os parâmetros de segurança e privacidade exigidos pela plataforma.

O trabalho aponta para futuras melhorias, como a possibilidade de um *plugin* para automação da captura de dados, que facilitará ainda mais o processo para jogadores de *Desktop*. Assim, o projeto abre caminho para evoluções tecnológicas que podem torná-lo ainda mais completo e adaptado às necessidades dos jogadores, consolidando-o como uma ferramenta de apoio no gerenciamento das informações do jogo.

Referências

- Bax, M. P., & Barbosa, D. M. (2009). Gestão de Informação Empresarial com Portais Plone. *Encontros Bibli: Revista eletrônica De Biblioteconomia E Ciência Da informação*, 14(27), 69–84. <https://doi.org/10.5007/1518-2924.2009v14n27p69>
- Batista, I. G. S. (2023). Aplicação web para gestão de prontuários eletrônicos de atendimento aos pacientes da equoterapia do if goiano - campus ceres. RIIIF Goiano. <https://repositorio.ifgoiano.edu.br/handle/prefix/3796>
- ARK Wiki. (n.d.). ARK: Survival Ascended. ARK Wiki. https://ark.wiki.gg/wiki/ARK:_Survival_Ascended
- ARK Wiki. (n.d.). ARK: Survival Evolved. ARK Wiki. https://ark.wik.gg/wiki/ARK:_Survival_Evolved#Full_Release
- Santhanam, S., Hecking, T., Schreiber, A., & Wagner, S. (2022). Bots in software engineering: a systematic mapping study. *PeerJ Computer Science*, 8, e866. doi:10.7717/peerj-cs.866
- Abhinand G, & Roshni Balasubramanian. (2022). Study on the Development and Implementation of Ubiquitous Bots for the Discord Interface. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 212–221. <https://doi.org/10.32628/cseit228137>
- Sielliagia, A. A., Suwandi, M. H., Limiarto, L., & Hasani, M. F. (2024). ChirpMap: Python based Discord Bot for Tweet Data Visualization using Geographical Information System. *Procedia Comput. Sci.*, 227(C), 772–778. doi:10.1016/j.procs.2023.10.582