

---

# **Elysium.NET**

***Versão master***

**10 out., 2021**



<b>1</b>	<b>Ajuda</b>	<b>3</b>
<b>2</b>	<b>Introdução</b>	<b>5</b>
2.1	Visão Geral . . . . .	5
2.2	Instalando . . . . .	6
2.3	Compilando . . . . .	8
<b>3</b>	<b>Servidor</b>	<b>13</b>
3.1	Comandos do Servidor . . . . .	13
3.2	Rodando no Linux . . . . .	13
<b>4</b>	<b>Cliente</b>	<b>15</b>
4.1	Sistemas dos Clientes . . . . .	15
4.2	Comandos de Cliente . . . . .	15
<b>5</b>	<b>Extensões</b>	<b>19</b>
5.1	Gerenciador de Plugins . . . . .	19
<b>6</b>	<b>Problemas Específicos</b>	<b>27</b>
6.1	Perguntas Mais Frequentes . . . . .	27
<b>7</b>	<b>Outros</b>	<b>29</b>
7.1	Notas de Lançamento . . . . .	29
7.2	Contribuindo para a Elysium.NET . . . . .	29
7.3	Versionamento . . . . .	30





Seja bem-vindo à documentação oficial da Elysium.NET, série 1, um motor de jogos online 2D totalmente livre e gratuito. Se você é novo nesta documentação, sugerimos que leia a seção “Visão Geral” para ter uma ideia melhor sobre toda a aplicação.

A tabela de conteúdos, contida nesta página, e o menu lateral devem te dar fácil acesso a todos os aspectos desta documentação, podendo você ir diretamente ao seu tópico de interesse. Também é possível usar a barra de busca.



# CAPÍTULO 1

---

## Ajuda

---

Mesmo olhando a tabela abaixo, ainda está com problemas? Gostaríamos de ajudar!

- Tente olhar o [FAQ](#) – ele tem respostas para as perguntas mais comuns.
- Procurando por alguma informação específica? Tente usar o [genindex](#) ou o [modindex](#).
- Faça uma pergunta no nosso canal no [Discord](#),
- Utilize nossa [comunidade](#) para procurar/fazer dúvidas mais técnicas ou reportar bugs.





### 2.1 Visão Geral

A Elysium é uma *engine* (em português, motor de jogos) que é específica para fazer jogos online 2D de uma maneira fácil e intuitiva, totalmente em português. Para tanto, todos os sistemas mais comuns já encontram-se implementados bem como todas as edições realizadas no jogo são vistas e atualizadas em tempo real.

#### 2.1.1 Histórico

A Elysium.NET tem suas raízes na engine Mirage Source, basicamente uma aplicação escrita em Visual Basic 6. A Mirage Source inspirou muitos desenvolvedores que, com base nela, desenvolveram a Konfuze Engine, a Elysium original (em inglês) e, com base nesta última, a Elysium Engine Brasil. A MMODEV surgiu originalmente, em 2007, como forma de dar suporte à EEB.

Acontece que, com a evolução tecnológica, tanto a tecnologia de DirectX 7 quanto a Visual Basic 6 ficaram obsoletas e, em algum ponto, pararam de ser suportadas pelo próprio Windows. Foi inspirado nessa ideia que a Eclipse Engine surgiu como sucessora espiritual da Mirage Source, e consequentemente muitas outros motores apareceram baseados nesta. A Elysium.net é baseada na Orion+ Engine, originalmente desenvolvida pelo Damian666 e por um dos fundadores da MMODEV, Lucas Tardivo.

Ao tempo que se passa o desenvolvimento da Elysium.NET, cada vez menos semelhanças com o passado vão ficando, mas a gratidão a todos que contribuíram é sempre eterna.

#### 2.1.2 Como usar a documentação?

Esta documentação é dividida em seis seções principais:

- A seção “Introdução” tem como objetivo ensinar o básico da Elysium.NET, principalmente às pessoas que nunca mexeram com softwares similares ou nunca programaram.
- A seção “Servidor” visa discutir a estrutura do servidor em detalhes, bem como suas utilidades.
- Em “Cliente”, você encontra uma discussão detalhada sobre cada sistema.

- Em “Extensões”, há informações sobre os blocos oficialmente suportados pela MMODEV (no momento, apenas o Gerenciador de Plugins).
- Os “Problemas Específicos” contém as perguntas mais frequentes e, ocasionalmente, *workarounds* para certos problemas.
- Finalmente, em “Outros”, há mais detalhes concernentes ao desenvolvimento da Elysium.NET, aos interessados em participar ou ajudar de alguma forma.

A documentação está em constante construção e evolução, ainda mais nessa fase inicial; esperamos que você divirta-se fazendo seu próprio mundo!

### 2.1.3 Lista de Desenvolvedores

Em ordem alfabética, segue a lista das pessoas que fazem contribuições periódicas para a Elysium.NET ou a comunidade MMODEV como um todo:

- Andrei Duarte (andreidl)
- Daniel Araújo (rajenki)
- Lucas Tardivo (boasfesta)

## 2.2 Instalando

Fazer seu primeiro jogo com a Elysium.NET é muito fácil, e deve levar alguns poucos minutos. Entretanto, este tutorial também irá discutir alguns detalhes mais técnicos para aprofundar a compreensão do desenvolvedor.

### 2.2.1 Requisitos

É muito provável que você já tenha instalado o .NET Framework no seu Windows, em virtude de ser amplamente utilizado por outras aplicações. Caso não o possua, faça o download no [website da Microsoft](#). Certifique-se que a versão do mesmo é 4.0 ou superior.

### 2.2.2 Baixando a versão mais recente

O primeiro passo envolve baixar a versão mais recente, costumeiramente chamada de latest, e pode ser encontrada [aqui](#). Para extrair os arquivos, você precisará de uma aplicação como o WinRAR ou o 7-zip.

### 2.2.3 Estrutura dos Diretórios

A versão baixada contém apenas os arquivos executáveis, sem o código fonte. Isso limita um pouco o que pode ser feito com a Elysium.NET, apesar de que todos os parâmetros de jogos (como mapas, classes, habilidades) podem ser alterados. Ao descompactar, existem duas pastas no diretório-raiz:

- **Server:** Possui os arquivos do servidor, responsável por gerenciar todo o banco de dados do jogo, como contas, mapas, NPCs, etc.
- **Client:** É o jogo em si, contendo os arquivos gráficos, sons e a lógica de comandos a serem enviadas para o servidor processar.

## 2.2.4 Inicializando o Servidor

Na versão latest da MMODEV, o servidor sempre vem com um mapa exemplo, com NPCs e a conta padrão de login **mmodev** e senha **mmodev**. Inicialize o servidor e espere o carregamento completo de todas as telas, onde ele mostrará a logo da Elysium.NET.



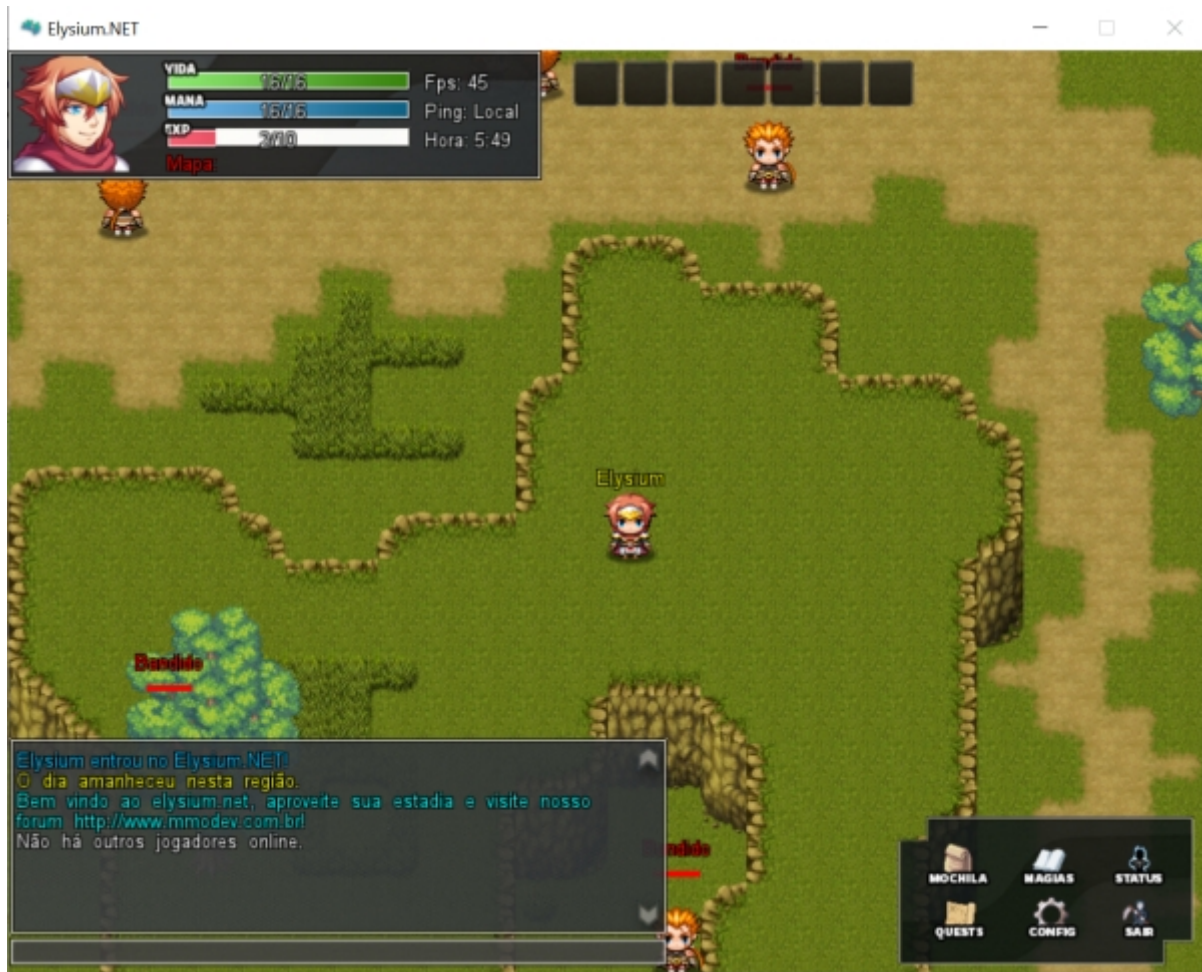
A partir deste momento, o cliente conseguirá detectar o servidor e conectar a ele. Você pode usar o comando `/ajuda` na própria tela do servidor para obter uma lista de comandos que podem ser usados.

**Aviso:** Por questão de segurança, delete o arquivo **AsyncKeys.xml** antes de inicializar o servidor. Ele contém uma chave criptografica que serve de comunicação entre o servidor e o cliente e, por ser padrão na versão compilada, outras pessoas podem conseguir hackear seu jogo se não trocada. Ao deletar, o servidor cria, em sua inicialização uma chave nova, a qual pode servir para a distribuição do seu jogo.

## 2.2.5 Trabalhando com o Cliente

Com o servidor preparado, você pode abrir o arquivo **Elysium Cliente.exe** e esperar que o estado do servidor fique Online. A partir daí, crie uma nova conta ou faça o login com a conta padrão de login **mmodev** e senha **mmodev**.

Caso você opte por criar uma conta própria, utilize o comando `/setadmin NOME-DO-PERSONAGEM 4` na tela do servidor para dar o máximo de poder de administração ao personagem.



Os comandos mais básicos do cliente são:

- W-A-S-D: Movimento para cima, baixo, esquerda e direita.
- CTRL: Interaç o com o mapa e/ou ataca os inimigos.
- INSERT: Abre o painel de administra  o.

Com o painel de administra  o aberto, a cria  o do mundo fica mais intuitiva e   atualizada em tempo real. Para mais informa  es, verifique a p gina correspondente de cada sistema.

## 2.2.6 Distribuindo seu jogo

No momento, a  nica maneira de distribuir seu jogo   compilando-o e alterando o IP do servidor dentro do c digo. Isso est  para ser alterado nas pr ximas vers es.

Repare que, mesmo voc  criando uma vers o do zero,   poss vel copiar o diret rio do servidor antigo para o novo, preservando o conte do de mapas, contas e habilidades.

## 2.3 Compilando

O processo de compila  o da Elysium.NET   importante para que se possa fazer a distribui  o do seu jogo. Em suma,   transformar todo o c digo-fonte, escrito em Visual Basic.NET, em um execut vel do Windows. No momento,

algumas alterações, como o IP do Servidor, só podem ser realizadas através do processo de compilação, e isso é o que este guia vai ensinar.

### 2.3.1 Requisitos

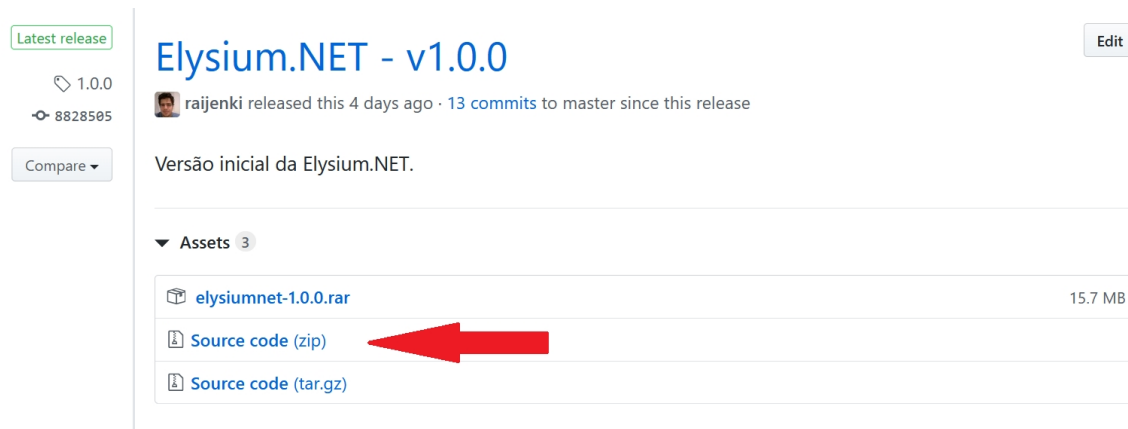
Diferentemente de quando você necessita rodar a aplicação, a versão do .NET Framework necessária neste caso é o *Software Development Kit* (SDK), que inclui todas as bibliotecas necessárias para desenvolvimento na plataforma .NET. Além disso, também é necessário a utilização do Visual Studio, haja visto que é a interface de desenvolvimento utilizada pela MMODEV.

- O Visual Studio Community, cuja versão é gratuita, pode ter seu download realizado [aqui](#).
- O .NET Framework pode ser baixado [aqui](#). Certifique-se de estar instalando o “Dev Pack” e a versão .Net Framework 4.0 ou superior.

Você pode optar por baixar o .NET Framework através do próprio instalador do Visual Studio, que está em português. Certifique-se de estar instalando as ferramentas de desenvolvimento em .NET (que incluirá C#, VB.NET e F#).

### 2.3.2 Pegando o Código-Fonte da Elysium.NET

O código-fonte da Elysium.NET encontra-se no [GitHub](#) de desenvolvimento. Escolha uma das versões lançadas e baixe o *Source Code*, indiferente do formato (.zip ou .tar.gz).



**Aviso:** Usuários mais avançados podem se sentir mais a vontade para fazer cópia direta do repositório da Elysium.NET e ter acesso às novidades mais recentes; entretanto, ressalta-se que tais versões não são amplamente testadas e são bastante suscetíveis a bugs.

### 2.3.3 Estrutura do Diretório

Ao descompactar os arquivos, esta é a estrutura de diretório-raiz do código-fonte da Elysium.NET:

- **docs:** A documentação do Sphinx, exatamente este texto que você está lendo.
- **packages:** Os pacotes de dependências; no momento, somente a SFML é dependência.
- **Source:** O código fonte da Elysium.NET como um todo.

Dentro da pasta Source, há a seguinte estrutura:

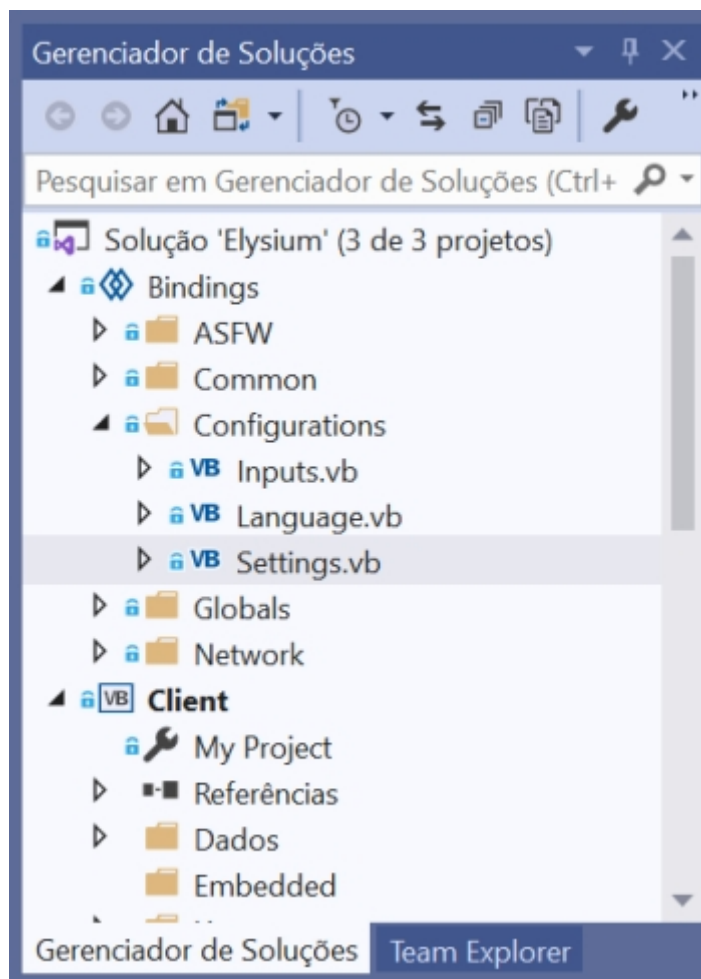
- **Client:** Código exclusivo ao cliente, relativo principalmente a renderização gráfica e sonora.

- **Server:** Código exclusivo do servidor, como gerenciamento de contas e processamento de ações (danos, por exemplo).
- **Bindings:** Trata-se de código comum ao cliente e ao servidor, como criptografia, comunicação e acesso a diretórios.

Em especial, o cliente terá outros diretórios de formulários (Forms), que dizem respeito aos editores e a parte de renderização gráfica, e a parte lógica (Modules) do jogo em si.

### 2.3.4 Alterando Configurações

O processo mais básico a se fazer é a alteração das configurações básicas e a geração de um executável. Com o Visual Studio, abra o arquivo **Elysium.sln**, que estará no diretório-raiz. Ao abrir, procure no gerenciador de soluções o caminho Bindings/Configurations/Settings.vb.



Esse arquivo é o responsável por todas as configurações iniciais tanto do cliente quanto do servidor. Comece procurando por:

```
<XmlIgnore()> Public Ip As String = "127.0.0.1"
<XmlIgnore()> Public Port As Integer = 7001

<XmlIgnore()> Public GameName As String = "Elysium.NET"
<XmlIgnore()> Public Website As String = "http://www.mmodev.com.br"
```

A primeira linha será usada pelo cliente para procurar uma conexão pelo servidor, enquanto a segunda se refere à porta de conexão; o padrão é 7001, mas é possível alterá-la caso esteja bloqueada por algum motivo. Se você não sabe seu IP, utilize sites como o [WhatIsMyIp.com](http://WhatIsMyIp.com). `GameName` e `Website` são, respectivamente, o nome do jogo e o website.

Logo abaixo, também há configurações do servidor:

```
Public GameName As String = "Elysium.NET"
Public Website As String = "http://www.mmodev.com.br"

Public Welcome As String = "Bem vindo ao Elysium.net, aproveite sua estadia e visite_
↔nosso portal http://www.mmodev.com.br!"
Public Port As Integer = 7001

Public StartMap As Integer = 1
Public StartX As Integer = 13
Public StartY As Integer = 7
```

Atente-se que os parâmetros `GameName`, `Website` e `Port` devem ser iguais para o servidor e o cliente. O `Welcome` é a mensagem que o cliente irá mostrar ao conectar com o servidor, enquanto `StartMap`, `StartX` e `StartY` se referem a localização que um novo personagem irá aparecer quando criando (respectivamente, mapa, posição X e posição Y).

### 2.3.5 Compilando

Com as modificações realizadas, atente-se em mudar os parâmetros do cliente e do servidor para o modo `Release`. Isso porque o modo `Debug` emite muitas informações desnecessárias ao servidor, servindo mais para procurar bugs do sistema e afins.



Feito isso, acesse o menu `Compilação` e escolha a opção `Compilar Solução`. Automaticamente, o diretório `Build` será criado no diretório-raiz, com as pastas `Cliente` e `Servidor`.

A partir daí, você já pode zipar a pasta do cliente e distribuir para os jogadores. Caso tenha um servidor anterior, faça a migração de toda a pasta `Database` para o novo servidor.

**Visão Geral** Entenda o que é a Elysium.NET, seus requisitos e o que você pode fazer com ela.

**Instalando** Faça seu primeiro jogo com a Elysium.NET.

**Compilando** Faça modificações mais profundas e compile a Elysium.NET facilmente.





### 3.1 Comandos do Servidor

Os comandos do servidor são utilizados somente pelas pessoas que tem acesso direto à interface/terminal do mesmo. Tais comandos visam ajudar no gerenciamento do mundo, e não são executadas quaisquer controle de permissão para sua execução.

- **/ajuda:** Mostra a mensagem de ajuda, com todos os comandos.
- **/sair:** Encerra o servidor de forma segura.
- **/chutar <NOME-DO-JOGADOR>:** Retira temporariamente o usuário de nome <NOME-DO-JOGADOR>. Ele pode refazer o login e entrar no mundo novamente.
- **/banir:** Retira permanentemente o jogador de nome <NOME-DO-JOGADOR>, e ele não pode entrar no jogo até que seja desbanido.
- **/setadmin <NOME-DO-JOGADOR> <NÍVEL-DE-ACESSO>:** Configura o acesso de administração ou moderação de determinado jogador, com o valor <NÍVEL-DE-ACESSO> sendo um inteiro variando de 0 a 4. Para mais informações, verifique a página de Comandos do Cliente.

### 3.2 Rodando no Linux

**Aviso:** O uso desta versão requer conhecimentos básicos de utilização de sistemas Linux, como Debian, Ubuntu ou CentOS.

Apesar desta seção ser majoritariamente focada em servidores Linux, é possível rodar também esta versão específica no Windows. Isso porque a biblioteca .NET Core 3.0 é utilizada para tal fim.

### 3.2.1 Requisitos

Inicialmente, precisaremos instalar as dependências que serão necessárias para executar o servidor:

```
apt-get install -y git
apt-get install -y dotnet-sdk
```

Feito isso, basta clonar o repositório da Elysium e dar *checkout* na versão server-core:

```
git clone https://github.com/raijenki/elysium.git
cd elysium/
git checkout server-core
```

### 3.2.2 Compilando e Executando

Finalmente, basta compilar e executar:

```
dotnet build
cd Servidor/
dotnet run
```

**Aviso:** A compilação não move automaticamente os arquivos de dados do servidor para a pasta do binário! Lembre-se de fazer isso manualmente ou terá erro na execução.

### 3.2.3 Cliente em .NET Core

Por motivos de tecnologia, é atualmente impossível rodar o cliente em outros sistemas operacionais diferentes do Windows. Isso porque a biblioteca WinForms se utiliza de APIs exclusivas apenas para o Windows. Enquanto é possível futuramente migrar o cliente para a plataforma .NET Core, o que envolve uma ampla reestruturação do código, esta limitação ainda se aplicará.

*Comandos do Servidor* Aprenda os comandos que te auxiliam a gerenciar o servidor.

*Rodando no Linux* Aprenda a rodar o servidor em sistemas baseados em Linux.

### 4.1 Sistemas dos Clientes

- Quests
- Moradia
- Atributos
- Pets
- Projéteis

### 4.2 Comandos de Cliente

Esta página visa ilustrar todos os comandos que o lado do cliente possui para uso dos desenvolvedores e dos jogadores.

#### 4.2.1 Estrutura Administrativa

A Elysium possui uma estrutura administrativa embutida a fim de dar diferentes poderes para quem contribui na construção do jogo, de alguma forma. Idealmente, tal estrutura deve seguir um formato de pirâmide, onde as pessoas com maior nível de acesso ficam no topo e em menor quantidade, enquanto os usuarios e as pessoas de menor acesso, na base maior.

A estrutura é separada em:

- Usuário (Nível de acesso: 0)
- Monitor (Nível de acesso: 1)
- Mapeador (Nível de acesso: 2)
- Desenvolvedor (Nível de acesso: 3)
- Administrador (Nível de acesso: 4)

## 4.2.2 Lista de Comandos

### Usuário (0)

Os comandos são mostrados em lista:

- **/emote <INTEIRO>**: Manda o emote número <INTEIRO> para o mapa.
- **/ajuda**: Mostra uma mensagem com todos os comandos disponíveis.
- **/convidarmoradia <NOME-DO-JOGADOR>**: Convida <NOME-DO-JOGADOR> para a moradia do jogador que está executando o comando.
- **/vendermoradia**: Vende a moradia do jogador que está executando o comando.
- **/info <NOME-DO-JOGADOR>**: Recebe informações do <NOME-DO-JOGADOR>.
- **/quem**: Recebe lista de quem está online.
- **/fps**: Mostra o FPS atual do jogo.
- **/lps**: Mostra o LPS atual do jogo.
- **/atributos**: Mostra os atributos do jogador que está executando o comando.
- **/equipe <NOME-DO-JOGADOR>**: Cria um convite de equipe com o jogador executor e o jogador <NOME-DO-JOGADOR>.
- **/entrar**: Aceita o convite para criar a equipe.
- **/sair**: Sai da equipe atual.
- **/soltarpet**: Solta o pet em posse do jogador executor do comando.

### Monitor (1)

Além de todos os comandos do nível de usuário, os Monitores tem acesso aos seguintes comandos:

- **/admin**: Mostra a ajuda da administração.
- **/resetartarefa <INTEIRO>**: Reseta os atributos da tarefa de número <INTEIRO>.
- **/chutar**: Retira temporariamente o jogador. Este pode refazer o login e entrar novamente.

### Mapeador (2)

Além de todos os comandos do nível de Monitor, os Mapeadores tem acesso aos seguintes comandos:

- **/loc**: Retorna a localização em coordenadas X,Y.
- **/melevarpara <NOME-DO-JOGADOR>**: Leva o jogador executor para a coordenada de <NOME-DO-JOGADOR>.
- **/metraga <NOME-DO-JOGADOR>**: Traz <NOME-DO-JOGADOR> para o jogador executor.
- **/irpara <INTEIRO>**: Transporta para o mapa de número <INTEIRO>.
- **/sprite <INTEIRO>**: Altera o número da sprite para o de <INTEIRO>.
- **/mapreport**: Retorna o MapReport.
- **/boasvindas <STRING>**: Seta a mensagem de boas-vindas para <STRING>.
- **/respawn**: Refaz a geração do mapa em que o Monitor está.

- **/banlist**: Retorna a lista de jogadores banidos.
- **/banir <NOME-DO-JOGADOR>**: Remove permanentemente o jogador.

No painel de administração, os mapeadores podem acessar os editores de mapas, moradia e o gerador procedural de mapas.

### Desenvolvedor (3)

Além de todas as funções englobadas pelo nível de mapeador, os desenvolvedores podem acessar os editores de animações, classes, habilidades, itens, lojas, NPCs, pets, projéteis, receitas, recursos e de tarefas.

### Criador (4)

Além de todas as funções englobadas pelo nível de desenvolvedor, os criadores podem acessar a função de gerar item e usar o seguinte comando:

- **/acesso <NOME-DO-JOGADOR> <INTEIRO>**: Seta o acesso para o valor <INTEIRO>, variando de 0 a 4, ao <NOME-DO-JOGADOR>.

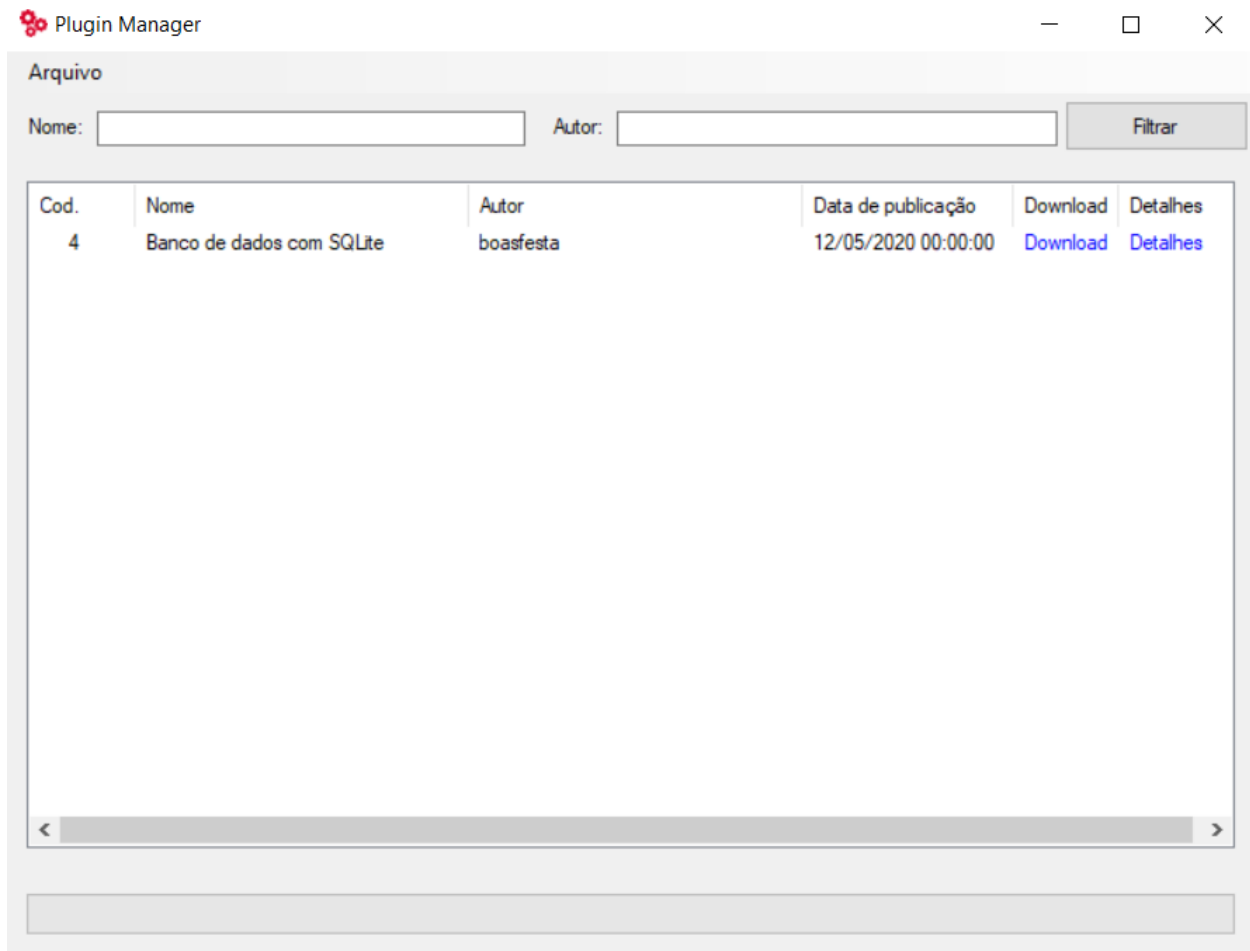
*Comandos de Cliente* Descubra quais são os comandos que podem ser utilizados pelo cliente, bem como a estrutura dos níveis de acesso.

*Sistemas dos Clientes* Saiba quais são os sistemas disponíveis no cliente.



#### 5.1 Gerenciador de Plugins

O **Gerenciador de Plugins** ou **Plugin Manager** é um gerenciador capaz de instalar novas tecnologias na sua Elysium.NET de maneira automática. Sem a necessidade de tutoriais que acabam por requerir muito trabalho tanto do autor como o do usuário. A proposta do gerenciador é a capacidade de implementar modificações da Elysium.NET através de um simples clique.



### 5.1.1 Utilidades

- Criar plugins de forma dinâmica definindo a ação efetuada com cada bloco de código.
- Exibir e instalar plugins verificados e disponibilizados pela comunidade.
- Publicar plugins em prol de colaborar com a comunidade.
- Permitir uma instalação manual do plugin em forma de tutorial.

### 5.1.2 Como utilizar

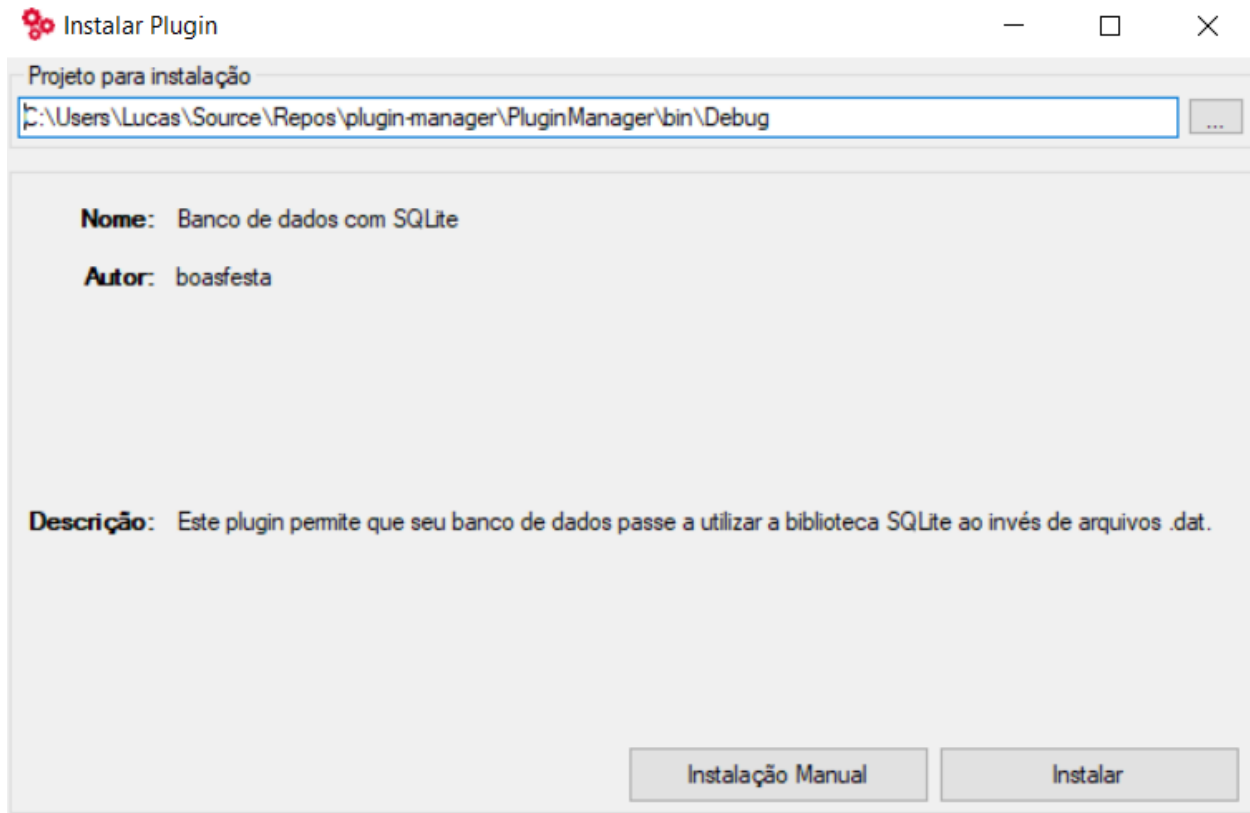
#### Instalando o Plugin Manager

O Plugin Manager vem acompanhado de um instalador. Após a instalação, é possível abri-lo através do Menu Iniciar.

#### Instalando um novo Plugin

Os plugins verificados da comunidade são automaticamente listados na tela principal do Plugin Manager, permitindo o download e instalação dos mesmos a qualquer momento utilizando de um simples clique no botão «Download». Da mesma forma, qualquer plugin externo pode ser instalado através do menu **Arquivo -> Instalar** e selecionando o arquivo .plugin.



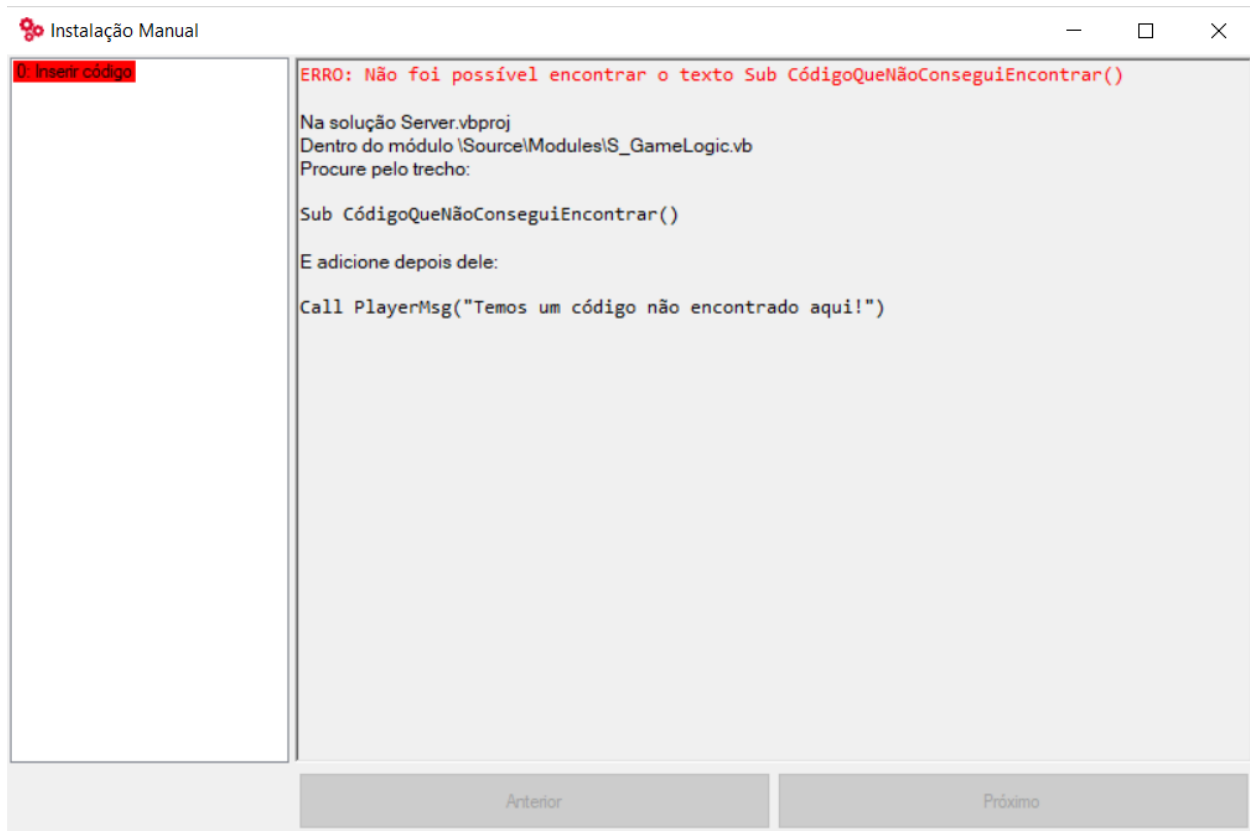


Para instalar o plugin, basta configurar a pasta destino do projeto na caixa de texto superior e clicar no botão «Instalar». Observando de que esta pasta é a pasta raiz de um projeto Elysium.NET, antes da pasta «Source».

**Alerta:** É altamente recomendável de que um backup do código seja feito antes de cada instalação e o Visual Studio se encontre fechado.

### Lidando com possíveis erros

Alguns plugins podem ter problemas ao serem instalados em sua Elysium, principalmente em casos onde o desenvolvedor a modificou bastante e impediu o Plugin Manager de encontrar algum trecho de código específico. Neste caso, a instalação ressaltará o número de erros e permitirá que o usuário veja a lista de ações do plugin, destacando as ações falhas e seu respectivo problema. Atualmente, o usuário terá de seguir a orientação da ação manualmente para efetuar a correção.



### 5.1.3 Criando um novo Plugin

Os plugins trabalham de maneira similar a um autor escrevendo um tutorial. Instruções como «Procure por tal trecho de código e substitua-o por isso» ou «Encontre a função X e abaixo dela insira este trecho» são descritas de maneira automatizada pelo Plugin Manager como uma lista de ações.

**Criar novo plugin**

Nome: Banco de dados com SQLite

Autor: boasfesta

Descrição: Este plugin permite que seu banco de dados passe a utilizar a biblioteca SQLite ao invés de arquivos .dat.

Ações: Ação: Modificar código

0: Modificar código

Adicionar Remover

Localização Código

Solução: Server.vbproj

Módulo: \\Source\\Modules\\S\_GameLogic.vb

Sub/Struct/Etc?

☐ Início

☐ Fim

Procurar por:

☒ Bloco de código

Sub CódigoQueNãoConseguirEncontrar()

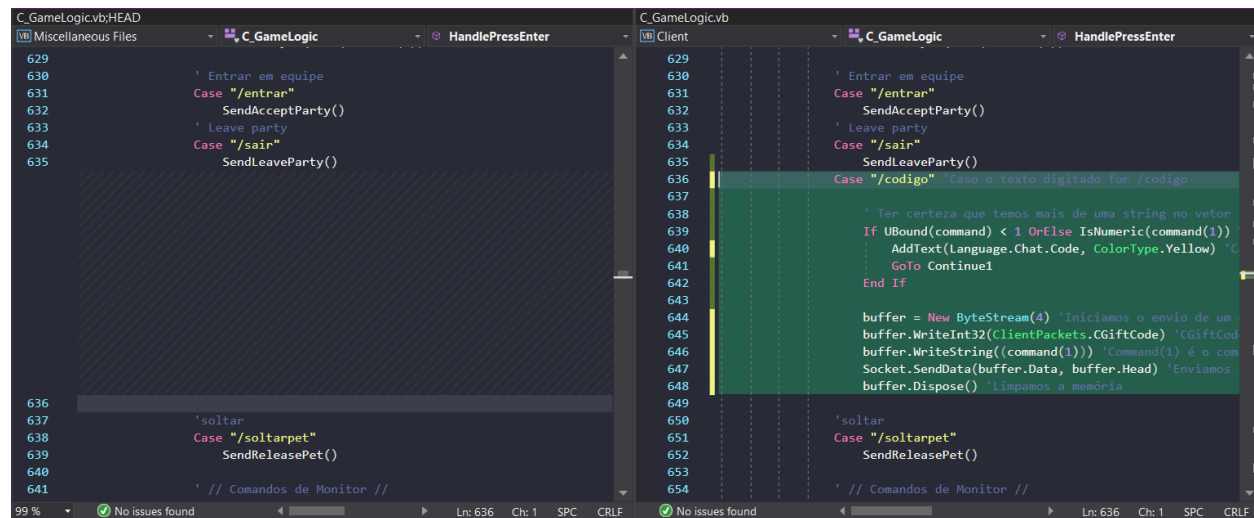
☒ Inserir após ☐ Inserir antes ☐ Substituir

Salvar Plugin

A região esquerda contém a lista de ações do plugin, enquanto a região central direita descreve os dados da ação correspondente. A aba «Localização» permite que o usuário oriente onde a ação deverá atuar, informando a solução, classe, opcionalmente uma possível estrutura (como uma Structure, Sub, Function, etc. . . ), ou até mesmo um bloco de código específico, e apenas neste ultimo caso, é possível selecionar se o seu código customizado será adicionado após o bloco em questão, antes ou o substituirá. A aba «Código» irá conter seu código customizado.

## Recomendações

Antes de iniciar a criação do Plugin, é recomendável criar um repositório Git para facilitar a visualização de modificações no código. Como este trecho de código realçado em Verde que foi **adicionado**:



O repositório ajuda a exibir as modificações em blocos de códigos e de certa forma «traduz» as ações que devem ser feitas no Plugin Manager, como adicionar, remover ou substituir códigos.

### Configurações de uma ação

- **Aba Localização**
- **Solução:** A solução em questão a ser modificada (Cliente, Servidor ou ambos).
- **Módulo:** Módulo da solução em questão (Caso a ação for **Criar novo módulo**, este é o módulo que será adicionado e a configuração acaba aqui).
- **Sub/Struct/Etc? (Opcional):** Caso a alteração seja especificamente em uma Sub, Struct, Function ou qualquer estrutura parecida, a mesma deve ser informada nesta linha, isso facilitará o Plugin Manager de encontrar a estrutura e tornará o plugin mais compatível (reduzindo problemas em Elysiums já modificadas ou com mais plugins).
- **Procurar por:**
- **Início:** O código será aplicado no início do módulo (Ou estrutura caso algo seja informado em **Sub/Struct/Etc?**)
- **Fim:** O código será aplicado no fim do módulo (Ou estrutura caso algo seja informado em **Sub/Struct/Etc?**)
- **Bloco de código:** O código será aplicado **Após, antes ou substituirá** o bloco de código informado para pesquisa.
- **Aba Código**
- Código em questão a ser inserido/substituído na Elysium. É neste espaço que será inserido o **código do plugin**. No caso da ação **Criar novo módulo**, este campo deverá conter **todo o código do módulo**.

**Alerta:** É recomendável testar o plugin em uma Elysium.NET sem modificações para validar sua compatibilidade. O critério de validação dos plugins para a lista da comunidade utilizam de uma Elysium.NET limpa mais recente como teste, caso contrário, o plugin não será validado ao público.

#### 5.1.4 Publicando um plugin

Após concluir o plugin, o usuário pode optar por compartilhá-lo com a comunidade da MMODEV. Para isso, o link <https://www.plugins.mmodev.com.br/> é disponibilizado ao público para submeter seus respectivos plugins.

## Recomendações para publicação

- Testar o Plugin instalando-o em uma Elysium original e limpa.
- Tomar cuidado com a identificação do código, afinal o Plugin Manager não efetua identificação automática e o plugin pode acabar por «sujar» o código dos usuários.
- Ser descritivo na funcionalidade do Plugin e tomar cuidado para não esquecer detalhes, os Administradores devem entender a utilidade e função do Plugin facilmente.
- Criar um tópico na categoria «Extensões» do fórum explicando sobre o Plugin, é extremamente aconselhável e evitará a rejeição do Plugin desnecessariamente.

*Gerenciador de Plugins* Entenda o gerenciador de plugins.



---

### Problemas Específicos

---

## 6.1 Perguntas Mais Frequentes

### 6.1.1 Qual a segurança da Elysium.NET na comunicação entre servidor e cliente?

Por motivos de compatibilidade entre o .NET Framework e o .NET Core, a Elysium.NET utiliza criptografia RSA com chave de 128bits combinado ao algoritmo AES-256.

### 6.1.2 O cliente roda no Linux?

Não, tanto porque o módulo de Windows Forms não é suportado oficialmente pela .NET Core quanto pelo fato que, mesmo quando for, estes dependem de APIs exclusivas do Windows - o que requer uma reescrita completa do código dos formulários.

*Perguntas Mais Frequentes* Tenha as respostas para as perguntas mais frequentes.





## 7.1 Notas de Lançamento

### 7.1.1 Elysium.NET 1.0.1 (Junho-2020)

Destaques:

- Correções de bugs reportados pela comunidade
- Tradução das pastas do Cliente e Servidor
- Criação do editor de detalhes do Automapper

### 7.1.2 Elysium.NET 1.0.0 (Maio-2020)

Destaques:

- Lançamento Inicial

## 7.2 Contribuindo para a Elysium.NET

Há muitas maneiras de contribuir com a Elysium.NET. Aqui estão algumas delas:

- Escreva ao mundo sobre os jogos que você fez com a Elysium.NET.
- Reporte bugs e peça por funcionalidades no nosso [GitHub](#).
- Escreva patches para novas funcionalidades ou correções de bugs.
- Responda perguntas na nossa [comunidade](#).

## 7.3 Versionamento

Existem três números em uma versão da Elysium.NET: *A.B.C*

- *A* é a versão principal. Isso raramente muda e vai significar mudanças muito profundas.
- *B* é o número de lançamento. Significa a inclusão de muitas mudanças, incluindo funções e coisas que podem quebrar a retrocompatibilidade de versões.
- *C* é alterado quando ocorre meros consertos de bugs.

Retro-incompatibilidade são mencionadas explicitamente nas *release notes*, e podem necessitar de atenção especial antes do upgrade.

*Notas de Lançamento* Veja o que mudou nas versões recentes da Elysium.

*Contribuindo para a Elysium.NET* Aprenda como contribuir para o projeto da Elysium.NET.

*Versionamento* Entenda o versionamento da Elysium.NET.