***WHMCS***

📘 ***Introdução ao WHMCS:***

O **WHMCS (Web Host Manager Complete Solution)** é um sistema pago de gerenciamento e monitoramento de vendas de serviços e produtos por assinatura.

O serviço é contratado por empresas que trabalham com **Hospedagem de Sites**, **Revenda de Serviços Digitais**, **Registro de Domínios**, e muitos outros.

O sistema automatiza toda a parte administrativa e operacional da venda do produto ou serviço, cadastrando clientes, emitindo faturas, recebimento de pagamentos, ativação de serviços, controle de domínios, e tem implementado em si um sistema de suporte técnico, onde os clientes da empresa contratante do WHMCS poderam dar suporte ao seus clientes mediante suas requisições.

Ele pode gerenciar qualquer tipo de venda de produtos ou serviços online, mas é amplamente reconhecido na área de **Web Hosting** que é a venda de Hospedagem em servidores.

## 🧩 ***Estrutura do Sistema WHMCS:***

* **Clientes:** Cada cliente terá um painel exclussivo para visualizar os serviços contratados pela empresa, verificar suas faturas, seus chamados de suporte e muito mais.
* **Produtos e Serviços:** A empresa poderá cadastrar todos os seus serviços e produtos.
* **Faturamento:** O WHMCS avisará seu cliente gerando e enviando mensagens automáticas a respeito de suas faturas em aberto, vencimentos e seus pagamentos.
* **Pagamentos (Gateways):** É possível integrar plataformas de pagamento como PayPal, Mercado Pago, PagSeguro, Stripe, e outras, processando e adminitrando toda a parte de pagamento das faturas dos clientes.
* **Domínios:** Para empresas vendedoras de domínios ela tem integração com registradores de domínios.
* **Automação de Servidores:** Tem integração com cPanel/WHM, Plesk, DirectAdmin e outros, gerenciando todo o registro do cliente da empresa nos painéis contratados.
* **Suporte Técnico:** Os clientes da empresa poderam solicitar suporte por meio de tickets, que são reclamações ou pedidos de ajuda.
* **Relatórios e Logs:** O WHMCS apresenta todas as informações administrativas, financeiras e operacionais por meio de relatórios textuais, gráficos e Logs.

## 🔌 ***Módulos no WHMCS:***

**Os **módulos** são os componentes do sistema que permitem a conexão com os servidores de registro de domínio, que fazem a conexão com as APIs de pagamento, conexão com as plataformas de configuração de servidores como o cPanel e entre outros serviços e aplicações externas.**

## 🧠 ***Funcionamento Prático de um Módulo:***

1. O cliente entra no site e escolhe um plano de hospedagem.
2. O WHMCS gera uma fatura automaticamente com o módulo de pagamento.
3. O cliente efetua o pagamento.
4. O WHMCS valida o pagamento através do gateway e utiliza o módulo de servidor para **criar a conta de hospedagem automaticamente**.
5. Os dados de acesso são enviados ao cliente.
6. Se houver renovação automática, o sistema repete esse processo mensalmente ou conforme o ciclo definido.

## 🧰 *****O que são Hooks?*****

**Nos módulos é possível progamar **hooks** que são trechos de código que de declaração de toda a regra de negócio do sistema.**

**Podemos gerenciar depois de quantas faturas o sistema será bloqueado, envios de notificações automáticas aos clientes, e até publicações de marketing. É só deixar a imaginação fluir.**

## **📌 ***Módulos:*****

**🔹***Módulo de Provsionamento ou Módulo de Servidor:*****

**O módulo de provisionamento ou módulo de servidor tem a finidalidade de intermediar uma aplicação externa e o **WHMCS**, automatizando e permitindo a manipulação desses sistemas externos dentro do próprio **WHMCS**. Apartir de um formulário para passar dados e botões podemos interagir com sistemas externos utilizando suas APIs.**

**📁 ***Em qual diretório ficam os módulos no WHMCS?*****

**📄** /modules/servers/nome\_do\_modulo/nome\_do\_modulo.php

***Atenção:*** ⚠️ O nome do arquivo **tem que ser igual ao nome da pasta!**

**⚙️***Métodos padrões do WHMCS:*****

**💠 CreateAccount() → Cria um novo serviço no sistema externo.**

**🛑 SuspendAccount() **→** Suspende o serviço quando a fatura atrasa.**

**🔓 UnsuspendAccount() **→** Reativa o serviço quando o cliente paga.**

**❌ TerminateAccount() **→** Remove o serviço quando o cliente cancela.**

**🔄 ChangePackage() **→** Atualiza um serviço (mudar plano, espaço, etc).**

**🔑 ChangePassword() **→** Atualiza a senha do serviço.**

## ****🛠️ *Estrutura básica do módulo:*****

**O módulo tem dois métodos obrigatórios, sendo:**

****1️⃣ ConfigOptions – Configuração do Módulo:****

**Aqui definimos quais propriedades podem ser configuradas no produto ou serviço representado no módulo.**

📝 **Código Exemplo:**

function nome\_do\_modulo\_ConfigOptions() {

return [

'Plano' => [

'Type' => 'text',

'Size' => '25',

'Default' => 'Básico',

'Description' => 'Nome do Plano que será usado na criação do serviço.',

],

'Email' => [

'Type' => 'text',

'Size' => '30',

'Default' => '',

'Description' => 'E-mail do cliente para integração com a API externa.',

],

];

}

**2️⃣ CreateAccount – Criando um Serviço:**

**Aqui faremos a interconexão com o serviço externo, neste caso de exemplo que iremos mostrar, estamos enviando os dados dos campos da configuração do módulo para a API do servidor externo, e receberemos uma resposta.**

**📝 Código Exemplo:**

function nome\_do\_modulo\_CreateAccount($params){

try{

$nome = $params['clientsdetails']['firstname'];

$email = $params['clientsdetails']['email'];

$plano = $params['configoption1'];

$ch = curl\_init();

curl\_setopt($ch, CURLOPT\_URL, "url");

curl\_setopt($ch, CURLOPT\_POST, 1);

curl\_setopt($ch, CURLOPT\_POSTFIELDS, json\_encode([

'nome' => $nome,

'email' => $email,

'plano' => $plano

]));

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);

$response = curl\_exec($ch);

curl\_close($ch);

if ($response){

return 'success';

} else{

return 'Erro ao criar conta!';

}

} catch (Exception $e){

return 'Erro: ' . $e->getMessage();

}

}