

## **Mudança da Arquitetura da Plataforma Badiu.Net na Versão 2.0**

### ***Atualização do Symfony de 2.5 para 6.0***

Badiu - Soluções Em Tecnologia da Informação Para Web  
Lino Vaz Moniz

15 de outubro de 2024

<b>I - Estrutura de Arquivos da Versão 1 do Badiu.Net</b>	<b>3</b>
Estrutura do Symfony versão 2.5	4
1.2 - Arquivos de Configuração	4
1.3 Arquivos de log e cache	5
Pastas de configuração do cache e log	5
1.4 - Módulos do Sistema	6
<b>II - Configuração do Sistema da Versão I do Badiu.Net</b>	<b>7</b>
2.2- Arquivos de Aplicação	8
1.3 - Servidor de Envio de E-mail	9
1.4 - Integração de Login com Gov.br	9
1.5 - Permissão dos Arquivos	11
<b>III - Estrutura de Arquivos da Versão II do Badiu.Net</b>	<b>11</b>
3.1 - Estrutura do Sistema	12
3.2 - Arquivos de Configuração	14
3.3 Arquivos de log e cache	14
3.4 - Módulos do Sistema	15
<b>IV - Configuração do Sistema da Versão II do Badiu.Net</b>	<b>16</b>
4.2 - Acesso ao Banco de Dados	17
4.3 - Servidor de Envio de E-mail	18
4.5 - Permissão dos Arquivos	20
<b>5 - Impacto da Mudança de Versão</b>	<b>21</b>
5.2 Permissão	23
5.2 - Mudança de Configuração do Badiu.Net	23
<b>VI - Considerações Finais</b>	<b>25</b>

## I - Introdução

A primeira versão da plataforma Badiu.Net foi lançada em 2015, introduzindo o módulo Badiu SGA – um sistema de gestão acadêmica. Em 2017, foi incorporado o módulo MReport, um sistema de relatórios para a plataforma Moodle. Desde então, diversos outros módulos foram desenvolvidos. Em 2021, o módulo GC2 – um sistema de gestão corporativa de cursos – foi lançado. Atualmente, em 2024, está em desenvolvimento o módulo A2I - um assistente de inteligência artificial. Todo o core de desenvolvimento foi baseado na arquitetura Symfony 2.5, e a atualização recente migrou o core para o Symfony 6.

A versão 2.0 da plataforma Badiu.Net é o resultado da mudança no core do Symfony. Este tutorial tem como objetivo abordar as alterações na estrutura de arquivos da plataforma Badiu.Net, da versão 1 para a versão 2. A principal mudança consiste na atualização do Symfony: a primeira versão utiliza o Symfony 2.5, enquanto a versão 2 adota o Symfony 6. Compreender essas mudanças é essencial para entender a nova estrutura de configuração e os impactos da atualização para a versão mais recente. A nova arquitetura afetará todos os módulos e sistemas baseados na plataforma Badiu.Net.

A explicação sobre as mudanças será estruturada em cinco partes. As partes 1 e 2 abordarão, respectivamente, a estrutura e a configuração da primeira versão. As partes 3 e 4 tratarão da estrutura e da configuração da versão 2. Por fim, a parte 5 descreverá as alterações ocorridas e os cuidados necessários no processo de atualização.

## I - Estrutura de Arquivos da Versão 1 do Badiu.Net

Nesta parte, exploraremos a estrutura de arquivos que compõe a primeira versão da plataforma Badiu.Net.

## 1.1 - Estrutura do Sistema

A tabela a seguir apresenta as pastas que compõem a estrutura do Symfony versão 2.5, que serve como base para o Badiu.Net I.

Estrutura do Symfony versão 2.5	
Pasta	Descrição
app	Pasta de configuração da aplicação Symfony. Essa pasta tem os arquivos que registram os bundles e arquivos e a configuração do banco de dados.
src	Pasta onde fica os bundles (módulos) criados no sistema Symfony
vendor	Nesta pasta fica as bibliotecas core do symfony e também as bibliotecas externas necessárias para o funcionamento dos bundles
web	Nesta pasta ficam os arquivos de acesso web que gerencia a rota para acesso aos bundles. Também ficam publicados arquivos css, javascript e imagens.

A figura a seguir apresenta as pastas que compõem a estrutura do Symfony versão 2.5

 app src vendor web

## 1.2 - Arquivos de Configuração

Na pasta app ficam arquivos importantes de configuração. A tabela a seguir

especifica esses arquivos.

Arquivos de configuração da pasta app	
Pasta/Arquivo	Descrição
app/config/parameters.yml	Arquivo de configuração. Fica a configuração do banco de dados, SMTP, endereço de gravação de arquivos no sistema
app/config/routing.yml	Arquivo de configuração da rota de cada bundle
app/AppKernel.php	Arquivo que registra cada bundle criados no Symfony

## 1.3 Arquivos de log e cache

A tabela a seguir apresenta as pastas de configuração do cache e log.

Pastas de configuração do cache e log	
Pasta	Descrição
app/cache	Pasta que armazena cache do sistema
app/logs	Pasta que armazena logs do sistema

Os comandos a seguir, fazem a limpeza de log e cache.

```
rm -rf app/logs/*
```

```
rm -rf app/cache/*
```

Ao limpar cache, o sistema será resetado. Esse procedimento é recomendado após atualização ou alteração de arquivos.

## 1.4 - Módulos do Sistema

Os módulos do Badiu.Net ficam na pasta src/Badiu. Os módulos ficam organizados em pastas. Na pasta src/Badiu ficam as categorias dos módulos. A tabela a seguir lista as categorias padrões que vem no sistema Badiu.Net.

Categoria de Módulos		
Categoria dos módulos		Descrição
src/Badiu/System		Nessa pasta ficam os módulos que são core do sistema Badiu.Net
src/Badiu/Theme		Nessa pasta ficam os módulos do thema, ou seja, projeto visual. O módulo padrão do thema é usado pelo core do sistema.
src/Badiu/Util		Nessa pasta ficam os módulos acessórios. Alguns são usados pelos módulos do sistema
src/Badiu/Admin		Nessa pasta ficam os módulos administrativos. Alguns são usados pelos módulos do sistema ou qualquer outro módulo como o caso do módulo de geração de formulário.

src/Badiu/Auth	Nessa pasta ficam os módulos de autenticação.
src/Badiu/Local	Esta pasta é reservada para módulos adicionais que não seja da distribuição badiu. Caso queira criar um novo módulo para estender ou customizar as funcionalidades da Badiu.Net é recomendado criar dentro dessa pasta.

## II - Configuração do Sistema da Versão I do Badiu.Net

Nesta parte, exploraremos a configuração geral dos arquivos que compõem a primeira versão da plataforma Badiu.Net.

Todas as configurações estão concentradas no arquivo `app/config/parameters.yml`. Neste arquivo, há um conjunto de variáveis que define a configuração global do sistema.

### 2.1 - Acesso ao Banco de Dados

A tabela a seguir apresenta as variáveis de configuração de acesso ao banco de dados do sistema.

Configuração de Acesso ao Banco de Dados	
Variável	Descrição
database_host	Endereço ou IP do banco de dados
database_port	Porta de conexão do banco de dados
database_name	Nome do banco de dados
database_password	Senha de acesso ao banco de dados

Configuração de Acesso ao Banco de Dados	
database_user	Usuário do banco de dados
badiu.system.file.defaultpath	<p>Endereço da pasta para upload de arquivos. Essa pasta deve ficar fora da raiz da web. Se não for especificado, por padrão usa a pasta <code>BADIUNET_DIR_INSTALL/app/data</code></p> <p>Caso queira alterar a configuração padrão, crie a pasta com nome <code>badiunetdata</code> (ou com qualquer outro nome) e coloque fora da raiz da web. Dê permissão de escrita nesta pasta para o usuário do servidor web.</p> <p>No exemplo exemplo do caminho que estamos ilustrando, a pasta de <code>badiunetdata</code> ficaria no seguinte endereço <code>/var/www/badiunetdata</code></p>

## 2.2- Arquivos de Aplicação

A variável `badiu.system.file.defaultpath` define o endereço da pasta para upload de arquivos feitos no sistema. Essa pasta deve ficar fora da raiz da web. Se não for especificado, por padrão será utilizada a pasta `BADIUNET_DIR_INSTALL/app/data`.

Caso deseje alterar a configuração padrão, crie uma pasta com o nome *badiunetdata* (ou qualquer outro nome) e coloque-a fora da raiz da web. Conceda permissão de escrita para o usuário do servidor web nessa pasta.



## 1.3 - Servidor de Envio de E-mail

A tabela a seguir apresenta as variáveis de configuração de envio de e-mail.

Configuração de Envio de E-mail	
Variável	Descrição
mailer_host	O endereço do servidor de e-mail, como "smtp.exemplo.com".
mailer_port	A porta usada pelo servidor de e-mail, tipicamente 465 para SSL ou 587 para TLS.
mailer_user	O nome de usuário usado para autenticar no servidor de e-mail.
mailer_password	A senha usada para autenticar no servidor de e-mail.
mailer_from	O endereço de e-mail que aparece como remetente das mensagens enviadas.
mailer_encryption	O tipo de criptografia usada para conectar ao servidor de e-mail, geralmente "ssl" ou "tls".

## 1.4 - Integração de Login com Gov.br

A tabela a seguir apresenta as variáveis de configuração sobre integração com sistema de login do gov.br.

Configuração de Integração de Login com Gov.br	
Variável	Descrição
badiu.auth.ssogovbr.config.providerurl	URL do provedor de autenticação do sistema gov.br. Este é o endereço do servidor onde as solicitações de autenticação são enviadas. Geralmente, aponta para o serviço de autenticação unificado do gov.br.

badiu.auth.ssogovbr.config.clientid	Identificador único do cliente, fornecido pelo sistema gov.br durante o processo de registro da aplicação que deseja integrar com o serviço de login. Este valor permite que o sistema gov.br reconheça o sistema que está solicitando autenticação.
badiu.auth.ssogovbr.config.secret	Segredo do cliente (client secret) é uma chave privada usada para autenticar o sistema ao gov.br e garantir que apenas sistemas autorizados possam se conectar. Este valor deve ser mantido em segurança.
badiu.auth.ssogovbr.config.redirecturl	URL de redirecionamento, que é para onde o sistema gov.br envia o usuário após ele ter feito login com sucesso. Esse endereço deve estar registrado no sistema gov.br para garantir que o redirecionamento seja válido e seguro.

## 1.5 - Permissão dos Arquivos

É necessário atribuir permissão de escrita na pasta da aplicação. A pasta da aplicação deve ter permissão de escrita. Caso a variável `badiu.system.file.defaultpath`, que especifica a pasta de dados `BADIUNET_DATA`, não siga o padrão, também será necessário definir permissão de escrita em `BADIUNET_DATA`, além de vinculá-la ao grupo do servidor web.

A pasta da aplicação deve estar vinculada ao grupo do servidor web. No sistema Debian/Ubuntu, o grupo utilizado é o `www-data`.

Para configurar permissões usando usando use os seguintes comandos:

```
chgrp www-data BADIUNET_DIR_INSTALL -R
cd BADIUNET_DIR_INSTALL
chmod g+wx app/data -R
```

Ao mudar a expressão para os endereços especificados no exemplo desse tutorial, ficaria assim:

```
chgrp www-data /var/www/html/badiunet -R
cd /var/www/html/badiunet
chmod g+wx app/data -R
```

Para limpar o cache caso precise, use esses comandos:

```
rm -rf app/data/logs/*
rm -rf app/data/cache/*
```

## III - Estrutura de Arquivos da Versão II do Badiu.Net

Nesta parte, exploraremos a estrutura de arquivos que compõe a segunda versão da plataforma Badiu.Net A atualização do portal requer a implementação dessa nova estrutura.

## 3.1 - Estrutura do Sistema

A tabela a seguir apresenta as pastas que compõem a estrutura do Symfony versão 6.0, que serve como base para o Badiu.Net 2.

Estrutura do Symfony versão 6.0	
Pasta	Descrição
Badiu	Contém os arquivos específicos do sistema Badiu.Net.
assets	Armazena arquivos estáticos como CSS, JavaScript e imagens, usados na interface do usuário.
bin	Contém executáveis, incluindo o script Symfony Console para executar tarefas de linha de comando.
config	Contém todos os arquivos de configuração da aplicação, como serviços, pacotes e rotas.
data	Usada para armazenar cache e arquivos enviados pelos usuários, como uploads de arquivos.
migrations	Armazena as migrações do banco de dados que ajudam a gerenciar as alterações na estrutura do banco de dados.
public	Pasta raiz da web, contendo o arquivo index.php que é o ponto de entrada da aplicação.
src	Contém os códigos-fonte PHP da aplicação, como controladores, serviços e modelos.
templates	Armazena templates Twig, que são usados para renderizar as páginas HTML.
tests	Contém os testes automáticos, como testes unitários e funcionais.
translations	Armazena arquivos de tradução usados para internacionalização da aplicação.
var/cache/dev	Contém arquivos de cache específicos do ambiente de desenvolvimento.
vendor	Contém bibliotecas de terceiros que a aplicação depende, gerenciadas pelo Composer.

A figura a seguir apresenta as pastas que compõem a estrutura do Symfony versão 6.0



## 3.2 - Arquivos de Configuração

A tabela a seguir apresenta os principais arquivos de configuração da nova versão do sistema.

Arquivos de configuração	
Arquivo	Descrição
.env	O arquivo .env é usado para armazenar variáveis de ambiente. Essas variáveis são geralmente configurações que diferem entre os ambientes de desenvolvimento, teste e produção e podem incluir sensíveis como credenciais de banco de dados, chaves de API e outras configurações de infraestrutura.
config/services.yaml	Arquivo de configuração. Fica a configuração do SMTP, endereço de gravação de arquivos no sistema e outros itens
config/routes.yaml	Arquivo de configuração da rota de cada bundle
config/bundles.php	Arquivo que registra cada bundle criados no Symfony

## 3.3 Arquivos de log e cache

A tabela a seguir apresenta as pastas de configuração do cache e log.

Pastas de configuração do cache e log	
Pasta	Descrição
var/cache	Pasta que armazena cache do sistema

var/log	Pasta que armazena logs do sistema
---------	------------------------------------

Os comandos a seguir, fazem a limpeza de log e cache.

```
rm -rf var/log/*
```

```
rm -rf var/cache/*
```

Ao limpar cache, o sistema será resetado. Esse procedimento é recomendado após atualização ou alteração de arquivos.

## 3.4 - Módulos do Sistema

Os módulos do Badiu.Net ficam na pasta Badiu na raiz do sistema. Os módulos ficam organizados em pastas. Na pasta Badiu ficam as categorias dos módulos. A tabela a seguir lista as categorias padrões que vem no sistema Badiu.

Categoria de Módulos	
Categoria dos módulos	Descrição
Badiu/System	Nessa pasta ficam os módulos que são core do sistema Badiu.Net
Badiu/Theme	Nessa pasta ficam os módulos do thema, ou seja, projeto visual. O módulo padrão do thema é usado pelo core do sistema.
Badiu/Util	Nessa pasta ficam os módulos acessórios. Alguns são usados pelos módulos do sistema

Badiu/Admin	Nessa pasta ficam os módulos administrativos. Alguns são usados pelos módulos do sistema ou qualquer outro módulo como o caso do módulo de geração de formulário.
Badiu/Auth	Nessa pasta ficam os módulos de autenticação.
Badiu/Local	Esta pasta é reservada para módulos adicionais que não seja da distribuição badiu. Caso queira criar um novo módulo para estender ou customizar as funcionalidades da Badiu.Net é recomendado criar dentro dessa pasta.

## IV - Configuração do Sistema da Versão II do Badiu.Net

Nesta parte, exploraremos a configuração geral dos arquivos que compõem a segunda versão da plataforma Badiu.Net.

Todas as configurações estão concentradas nos arquivos `.env` e `config/services.yaml`. Esses arquivos contêm um conjunto de variáveis que definem a configuração global do sistema.

### 4.1 - Configuração Básica

A tabela a seguir apresenta as variáveis de configuração do arquivo `.env`.

Configuração do arquivo <code>.env</code>	
Variável	Descrição
DATABASE_URL	Define a URL completa de conexão com o banco de dados, incluindo o nome do usuário, senha, host, porta e nome do banco de dados. Esta URL é usada para configurar a conexão do banco de dados na aplicação. Os parâmetros seguem esse layout: "dbtype://user:passwod@host:port/dbname"
KERNEL_DEBUG	Controla o nível de debug do kernel do Symfony. Quando ativado (true), o sistema fornece informações detalhadas sobre erros, o que



	é útil durante o desenvolvimento. Deve ser desativado (false) em produção para melhor performance.
APP_DEBUG	Ativa ou desativa o modo de depuração da aplicação. Quando habilitado (true), a aplicação exibirá mensagens de erro detalhadas, que são úteis durante o desenvolvimento. Deve ser desativado (false) em ambientes de produção para evitar a exposição de informações sensíveis.

## 4.2 - Acesso ao Banco de Dados

A tabela a seguir apresenta as variáveis de configuração de acesso ao banco de dados do sistema. A configuração deve ser feita no arquivo `.env`, localizado na raiz da instalação, na variável `DATABASE_URL`. Essa variável segue o seguinte formato: `DATABASE_URL="dbtype://user:password@host:port/dbname"`

Configuração de Acesso ao Banco de Dados	
Variável	Descrição
dbtype	Protocolo ou tipo de banco de dados usado na conexão.
user	Nome do usuário utilizado para autenticar no banco de dados.
password	Senha do usuário para acesso ao banco de dados.
host	Endereço do servidor onde o banco de dados está hospedado.
port	Porta através da qual a conexão com o banco de dados é estabelecida.
dbname	Nome do banco de dados específico a ser acessado.

## 4.3 - Servidor de Envio de E-mail

A tabela a seguir apresenta as variáveis de configuração de envio de e-mail, que estão localizadas no arquivo `config/services.yaml`. As variáveis de configuração são as mesmas da versão I e permanecem inalteradas, com a única mudança sendo o local de configuração. A tabela abordada no item 1.3 descreve todas as variáveis de configuração de envio de e-mail, portanto, a mesma tabela se aplica à versão II da plataforma Badiu.Net.

## 4.4 - Integração de Login com gov.br

As variáveis de configuração de integração de login com o Gov.br estão localizadas no arquivo `config/services.yaml`. Assim como na configuração do SMTP, as variáveis permaneceram inalteradas, havendo apenas a mudança no local de configuração. A tabela abordada no item 1.4 descreve cada variável de configuração da integração de login, e as mesmas configurações aplicadas na versão 1 também se aplicam na versão 2.

## 4.5 - Variáveis Gerais de Configuração

A tabela a seguir apresenta um conjunto de variáveis de configuração, algumas das quais foram inseridas na nova versão do sistema.

Variáveis Gerais de Configuração		
Variável	Valor Padrão	Descrição
<code>badiu.system.access.defaultentity</code>	1	Define a entidade padrão do sistema.

badiu.system.user.recoverpwd.defaultentity	1	Define a entidade padrão utilizada no sistema para recuperação de senha.
badiu.system.access.usermanageentity	1	Define id do usuário com permissão geral, ou seja, usuário root.
badiu.system.file.defaultpath		Caminho padrão onde os arquivos do sistema são armazenados.
badiu.system.file.appcache		Define o caminho onde o cache da aplicação é armazenado.
badiu.system.core.wwwroot		Define a URL raiz do sistema (geralmente a URL principal do site ou aplicação).
badiu.system.core.forcehttps		Determina se o sistema deve forçar o uso do protocolo HTTPS (geralmente configurado como 0 ou 1).
badiu.system.core.route.withoutsession		Define rotas que podem ser executados sem que a sessão do usuário esteja ativa.
badiu.system.core.update.link		URL ou link utilizado para atualizações do sistema.
badiu.system.core.passowrdtoexeccommand		Senha utilizada para executar comandos administrativos ou críticos no sistema.
badiu.system.core.type.instalation		Define o tipo de instalação do sistema, como múltiplas entidades ou entidade simples.

<code>badiu.system.core.instance.shortname</code>		Nome curto da instância atual do sistema, geralmente uma forma abreviada da identificação da instância. Deve ser configurado em contexto de múltipla entidade
<code>badiu.system.core.lang.config</code>	system	Configuração do idioma do sistema, geralmente define se o idioma segue a configuração do sistema, browser ou usuário
<code>badiu.system.core.lang.default</code>	pt	Idioma padrão utilizado pelo sistema.
<code>badiu.system.core.criptkey.default</code>		Chave criptográfica padrão utilizada pelo sistema para garantir segurança nas operações, como criptografia de dados com JWT.

## 4.5 - Permissão dos Arquivos

Na nova versão do sistema, as permissões de arquivos e pastas seguem a mesma lógica da primeira versão, ou seja, os arquivos de log, cache e upload da aplicação devem ter permissão de escrita. A diferença em relação à versão anterior é que esses arquivos mudaram de local, no entanto, a lógica permanece a mesma.

A pasta da aplicação deve ter permissão de escrita. Caso a variável `badiu.system.file.defaultpath` (no arquivo `config/services.yaml`), que especifica a pasta de dados `BADIUNET_DATA`, não siga o padrão, é necessário também definir permissão de escrita em `BADIUNET_DATA`, além de vinculá-la ao grupo do servidor web.

A pasta da aplicação deve estar vinculada ao grupo do servidor web. No sistema Debian/Ubuntu, o grupo utilizado é o `www-data`.

Para configurar permissões, utilize os seguintes comandos:

```
chgrp www-data BADIUNET_DIR_INSTALL -R
cd BADIUNET_DIR_INSTALL
chmod g+wx data -R
chmod g+wx var -R
```

Ao ajustar os comandos para os diretórios especificados neste tutorial, ficaria assim:

```
chgrp www-data /var/www/html/badiunet -R
cd /var/www/html/badiunet
chmod g+wx data -R
chmod g+wx var -R
```

Para limpar o cache, caso necessário, use os seguintes comandos:

```
rm -rf var/log/*
rm -rf var/cache/*
```

## 5 - Impacto da Mudança de Versão

Nesta etapa, vamos abordar os principais impactos da mudança da versão 1 para a versão 2 da plataforma Badiu.Net, no que se refere à estrutura de arquivos e configurações. Essas mudanças são fundamentais e devem ser levadas em conta por programadores ao realizar ajustes e atualizações no sistema.

Nas partes 1 e 2, foram apresentadas as estruturas e configurações da primeira versão. Já nas partes 3 e 4, foi abordado o mesmo conteúdo referente à segunda versão. Nesta parte, vamos focar nas alterações significativas entre a primeira e a segunda versão. Essa abordagem oferece uma visão ampla do processo de atualização e auxilia o desenvolvedor a entender a arquitetura da nova versão do sistema.

### 5.1 - Mudança da Arquitetura do Symfony

A versão 6 do Symfony alterou a estrutura de muitos arquivos e pastas. A

seguir, vamos destacar os principais arquivos e pastas que tiveram sua estrutura modificada em comparação com a versão 2.5.

## **src**

A pasta `src` na versão 6 do Symfony continua com a mesma funcionalidade que havia na versão 2.5, ou seja, é a pasta que organiza o código-fonte dos módulos, conhecidos como Bundles. No entanto, na versão 6, o suporte padrão é para um único bundle, que deve ser programado dentro da pasta `src`. Já na versão 2.5, havia suporte para múltiplos bundles, que eram programados dentro da mesma pasta. Sendo assim, todos os bundles que existiam na versão 2.5 foram migrados para a nova versão, ficando na pasta `Badiu`, na raiz da aplicação, ao invés de na pasta `src`.

## **public**

A pasta `public` na versão 6 do Symfony substitui, em parte, as funcionalidades que existiam na pasta `web` da versão 2.5. Nessa pasta ficam os arquivos que permitem o acesso à web e que fazem roteamento de todo o sistema. O arquivo `index.php`, que está localizado nessa pasta, desempenha essa função, substituindo o arquivo `app.php`, que ficava na pasta `web` na versão 2.5.

## **var**

Os arquivos de log e cache são armazenados na pasta `var` nesta nova versão do Symfony. Na versão anterior, 2.5, esses arquivos ficavam na pasta `app`.

## **config**

A pasta `config` centraliza os principais arquivos de configuração. Dentro dessa pasta, temos o arquivo `bundles.php`, que contém o registro de todos os bundles, ou seja, módulos habilitados no sistema. Na versão anterior do Symfony, essa configuração estava no arquivo `AppKernel.php`, localizado na pasta `app`. O

arquivo `routes.yaml` centraliza a configuração de todas as rotas do sistema; na versão anterior, esse arquivo ficava na pasta `app/config`. Já o arquivo `services.yaml` centraliza a configuração das variáveis do sistema e de alguns serviços globais. Esse arquivo substitui as configurações que eram feitas no arquivo `parameters.yml`, que estava na pasta `app/config`.

## **.env**

O arquivo `.env` foi adicionado na nova versão do Symfony, atendendo melhor aos padrões de segurança do sistema. Esse arquivo contém as configurações de credenciais de acesso ao banco de dados e outras configurações. Na versão anterior, essas configurações eram feitas no arquivo `parameters.yml`, localizado na pasta `app`.

## **5.2 Permissão**

A permissão de escrita deve ser atribuída na pasta `var` e também na pasta `data` na nova versão 6 do Symfony. Já na versão anterior, essa permissão deveria ser atribuída na pasta `app/data`.

## **5.2 - Mudança de Configuração do Badiu.Net**

Na nova versão do Symfony, as configurações da plataforma Badiu.Net sofreram algumas alterações, tanto no layout quanto no local das configurações. Além disso, novas variáveis foram adicionadas na versão 2 do Badiu.Net. A seguir, apresentaremos as alterações mais importantes.

### **Configuração do Banco de Dados**

A configuração de acesso ao banco de dados passou a ser feita no arquivo `.env`, utilizando a variável

`DATABASE_URL="dbtype://user:password@host:porta/dbname"`. Esta variável segue um formato de string específico que deve ser preenchido com os dados de acesso ao banco. Em comparação com a versão anterior, houve uma grande mudança nesse item, pois a configuração era feita no arquivo `app/config/parameters.yml`, com um conjunto de variáveis explícitas.

## Configuração do SMTP

O layout de configuração do SMTP continua o mesmo em relação às variáveis, no entanto, o local de configuração foi alterado. Atualmente, na nova versão, a configuração deve ser feita no arquivo `config/services.yml`. Já na versão anterior, essa configuração era realizada no arquivo `app/config/parameters.yml`. Tanto na versão anterior quanto na nova, as variáveis de configuração foram mantidas.

## Novas variáveis de configuração

Na versão 2 da plataforma Badiu.Net, algumas novas variáveis inseridas merecem atenção, especialmente se estiver migrando da versão 1 para a versão 2 da plataforma. A seguir, apresentaremos algumas dessas variáveis e suas descrições.

### **`badiu.system.core.criptkey.default`**

Define uma chave base utilizada para gerar tokens e chaves de autenticação para várias operações dentro da plataforma que envolvem a tecnologia JWT. Essa chave deve ser gerada durante o processo de instalação ou atualização. Uma vez gerada, ela não pode mais ser alterada, pois sua alteração invalida a autenticação e todas as outras chaves derivadas dela.

### **`badiu.system.core.forcehttps`**



A variável `badiu.system.core.forcehttps` permite que o sistema seja acessado exclusivamente via HTTPS. Nesse caso, não é necessário definir o endereço base na variável `badiu.system.core.wwwroot`. Ao habilitar essa configuração, o domínio configurado será automaticamente forçado para HTTPS, o que resolve o erro de processamento de formulário que ocorria na versão anterior quando o HTTPS era usado sem a configuração da configuração da variável `badiu.system.core.wwwroot`.

## **`badiu.system.file.appcache`**

A variável `badiu.system.file.appcache` define o local onde o cache do sistema deve ser gerado, referente às variáveis do sistema. Na versão 2, há uma separação dos registros de sessão: os registros globais são encaminhados para o cache, enquanto os registros de sessão personalizados de cada usuário são tratados individualmente, sem serem replicados para todos. Isso resulta em uma redução significativa do consumo de memória e armazenamento, que era um problema na versão anterior, onde as variáveis globais eram replicadas para todos os usuários, sobrecarregando os recursos.

O local definido por essa variável para armazenar os arquivos deve ter permissão de escrita. Caso essa variável não seja configurada, o sistema usará automaticamente a pasta de cache para gravar os arquivos relacionados às variáveis do sistema.

## **VI - Considerações Finais**

A atualização da plataforma Badiu.Net tem como objetivo acompanhar a modernização tecnológica e garantir maior segurança, alinhada às novas versões de software. A nova versão exige o uso do PHP 8, a versão mais atualizada do momento. Além disso, permite a adoção de novas tecnologias, como a integração com Docker, proporcionando a implementação de serviços mais eficientes e escaláveis.

As próximas versões da plataforma Badiu.Net irão focar na incorporação de funcionalidades de inteligência artificial, integrando os módulos do sistema com o uso de linguagem natural e as potencialidades oferecidas pela IA para a execução de diversas tarefas.

Manter o portal atualizado com a versão mais recente da plataforma Badiu.Net permitirá acompanhar a evolução tecnológica, garantindo maior segurança e aproveitando as atualizações contínuas do sistema Badiu.Net, que serão disponibilizadas ao longo do tempo.