

Migração para PHP 8.2 - Principais Mudanças e Ajustes Necessários

A versão do PHP utilizada pelo sistema Ilii está passando por uma atualização significativa.

A versão será migrada de **PHP 7.3.33** para **PHP 8.2.28**.

Com essa mudança, é esperado que enfrentemos algumas incompatibilidades e a necessidade de ajustes no código.

Além disso, mensagens de erro podem surgir nos logs, exigindo análise e correções pontuais.

Para auxiliar nesse processo, destacamos algumas das mudanças mais relevantes:

1. Acesso a índices inexistentes em arrays agora gera Warning

Antes (PHP 7.3):

Ao acessar um índice inexistente em um array associativo, era gerado um "Notice".

```
// Exemplo 1
$array = array();
echo $array['chave'];

// Exemplo 2
$valor['filhos'] = false;
$valor['filhos'][] = "Um Valor Qualquer";

// Exmplo 3
$array[$row["id_acesso"]][$row["id"]] = true;
```

Agora, esse mesmo acesso gera um *Critical*.

```
$array = [];  
echo $array['chave'];
```

Automatic conversion of false to array is deprecated

Depois (PHP 8.2):

Solução Recomendada:

Para evitar esse erro, utilize o operador de coalescência nula (`??`) ou verifique a existência da chave com `isset()`, conforme abaixo:

```
// Exemplo 1
$array = [];  
  
// Exemplo 2
$valor['filhos'] = [];  
$valor['filhos'][] = "Um Valor Qualquer";  
  
// Exemplo 3
$array = [];  
if (empty($array[$row["id_acesso"]])) {  
    $array[$row["id_acesso"]] = [];  
}  
$array[$row["id_acesso"]][$row["id"]] = true;
```

Referências:

[RFC: Deprecate implicit conversion of incompatible float keys](#)

[RFC: Deprecate implicit conversion of array keys](#)

2. Mudanças no retorno de funções de ordenação (`sort()`, `asort()`, etc.)

Antes (PHP 7.3):

As funções `sort()`, `asort()`, `ksort()`, entre outras, modificavam o array e retornavam `true` ou `false`, mas também podiam ser usadas diretamente no `var_dump()`.

```
$array = [3, 1, 2];  
var_dump(sort($array));
```

Depois (PHP 8.2):

Essas funções continuam modificando o array, mas agora retornam apenas `true` ou `false`.

```
$array = [3, 1, 2];  
if (sort($array))  
{  
    print_r($array);  
}
```

Solução Recomendada:

Evite usar `var_dump(sort($array))` para visualizar o resultado. Em vez disso, use `if (sort($array))` e imprima o array modificado separadamente.

Referências:

3. Novas funções para manipulação de arrays

A partir do PHP 8.1, foram introduzidas novas funções que facilitam a manipulação de arrays.

Antes (PHP 7.3):

Não existiam funções nativas para verificar se um array era indexado sequencialmente ou para obter a primeira e a última chave de um array.

Depois (PHP 8.1+):

Agora, contamos com:

- `array_is_list()`: Verifica se um array é uma lista sequencial.
- `array_key_first()`: Retorna a primeira chave de um array.
- `array_key_last()`: Retorna a última chave de um array.

Exemplo de uso:

```
$array = [10, 20, 30];  
if (array_is_list($array))  
{  
    echo "O array é uma lista válida";  
}
```

```
$array = ['a' => 1, 'b' => 2];  
echo array_key_first($array); // Retorna "a"  
echo array_key_last($array); // Retorna "b"
```

Referências:

PHP 8.1: `array_is_list()`

PHP 8.0: `str_contains()`, `str_starts_with()`, `str_ends_with()`

PHP 8.3: `array_take()`, `array_drop()`

4. Erro: "Creation of dynamic property ... is deprecated"

Antes (PHP 7.3):

A partir do **PHP 8.2**, a criação dinâmica de propriedades em classes que não declaram explicitamente essas propriedades se tornou **obsoleta** (*deprecated*).

Isso significa que, se um objeto tentar atribuir um valor a uma propriedade que não foi previamente definida na classe, o PHP gerará um aviso de depreciação.

```
class Vendarapida {  
    // Nenhuma propriedade $md5Hash declarada aqui  
}  
  
$obj = new Vendarapida();  
$obj->md5Hash = 'abc123'; // ⚠ ERRO: Criando propriedade dinamicamente!
```

Saída do log:

Destaque para as linhas 4 e 7.

```
[critical] -> require [/index.php:2]
[critical] --> require_once [/backend/index.php:191]
[critical] ---> vendarapida::__construct [/backend/system/core/CodeIgniter.php:309]
[critical] ----> errorBygeneral [/backend/application/controllers/pdv/vendarapida.php:132]
[critical] -----> dump [/backend/application/libraries/dump.php:76]
[critical] -----> __get_backtrace [/backend/application/libraries/dump.php:477]
[critical] [string] Creation of dynamic property vendarapida::$md5Hash is deprecated
```

Depois (PHP 8.1+):

Declare a propriedade explicitamente dentro da classe antes de usá-la:

```
class Vendarapida {
    public string $md5Hash; // ☐ Correto: propriedade definida na classe
}

$obj = new Vendarapida();
$obj->md5Hash = 'abc123'; // Agora funciona sem erro
```

Referências:

[RFC: Deprecate Dynamic Properties](#)

[PHP 8.2 - Dynamic Properties Deprecated](#)

5. Erro: "Return type of ... with IteratorAggregate::getIterator(): Traversable, or the

#[\ReturnTypeWillChange] attribute should be used to temporarily suppress the notice"

Antes (PHP 7.3):

A partir do PHP 8.1, métodos que implementam interfaces internas do PHP (como `IteratorAggregate::getIterator()`) devem declarar explicitamente seus tipos de retorno.

Caso contrário, o PHP gerará um aviso de depreciação

Destaque para as linhas 8 e 20.

```
[critical] -> require [/index.php:2]
[critical] --> require_once [/backend/index.php:191]
[critical] ---> MY_Controller::__construct [/backend/system/core/CodeIgniter.php:309]
[critical] ----> MY_Controller::apiAuth [/backend/application/core/MY_Controller.php:2]
[critical] -----> usuario::autenticar [/backend/application/core/MY_Controller.php:2]
[critical] -----> usuario_grupo_usuario::autenticar [/backend/application/controllers/usuario/usuario.php:1099]
[critical] -----> dispositivo::alterar [/backend/application/controllers/usuario/usuario_grupo_usuario.php:179]
[critical] -----> MY_Controller::salvar [/backend/application/controllers/usuario/dispositivo.php:110]
[critical] -----> Doctrine\ORM\EntityManager::flush [/backend/application/core/MY_Controller.php:2]
[critical] -----> Doctrine\ORM\UnitOfWork::commit
[/backend/vendor/doctrine/orm/lib/Doctrine/ORM/EntityManager.php:378]
[critical] -----> Doctrine\ORM\UnitOfWork::executeInserts
[/backend/vendor/doctrine/orm/lib/Doctrine/ORM/UnitOfWork.php:423]
[critical] -----> Doctrine\ORM\Persisters\Entity\BasicEntityPersister::executeInserts
[/backend/vendor/doctrine/orm/lib/Doctrine/ORM/UnitOfWork.php:1122]
[critical] -----> Doctrine\DBAL\Connection::prepare
[/backend/vendor/doctrine/orm/lib/Doctrine/ORM/Persisters/Entity/BasicEntityPersister.php:273]
[critical] -----> Composer\Autoload\ClassLoader::loadClass
[/backend/vendor/doctrine/dbal/lib/Doctrine/DBAL/Connection.php:1244]
[critical] -----> Composer\Autoload\includeFile [/backend/vendor/composer/ClassLoader.php:428]
[critical] -----> include [/backend/vendor/composer/ClassLoader.php:571]
[critical] -----> errorBygeneral [/backend/vendor/doctrine/dbal/lib/Doctrine/DBAL/Statement.php:27]
[critical] -----> dump [/backend/application/libraries/dump.php:76]
```

```
[critical] -----> __get_backtrace [/backend/application/libraries/dump.php:477]
[critical] [string] Return type of Doctrine\DBAL\Statement::getIterator() should either be compatible with
IteratorAggregate::getIterator(): Traversable, or the #[\ReturnTypeWillChange] attribute should be used to
temporarily suppress the notice
```

```
class MinhaClasse implements IteratorAggregate {
    public function getIterator() { // ⚠ Erro: falta o tipo de retorno
        return new ArrayIterator(["maça", "banana", "uva"]);
    }
}
```

Antes (PHP 8.1+):

Adicione o tipo de retorno `Traversable`, garantindo compatibilidade com a interface `IteratorAggregate`:

```
class MinhaClasse implements IteratorAggregate {
    public function getIterator(): Traversable { // ✅ Correto: tipo de retorno declarado
        return new ArrayIterator(["maça", "banana", "uva"]);
    }
}
```

Referências:

[RFC: Deprecate Implicit Incompatible Method Signatures](#)

[PHP 8.1 - Changes to IteratorAggregate::getIterator\(\)](#)

Documentação Oficial do PHP 8.x

PHP 8.0 Release Notes:

<https://www.php.net/releases/8.0/>

PHP 8.1 Release Notes:

<https://www.php.net/releases/8.1/>

PHP 8.2 Release Notes:

<https://www.php.net/releases/8.2/>

Agradecemos a colaboração de todos nesse processo de migração.

Caso encontrem outras inconsistências ou precisem de suporte, por favor, entrem em contato.

Revisão #: contagem de revisões

Criado: duração de tempo por usuário

Atualizado: duração de tempo por usuário