

# Composer PHP

## EspecializaTi (Carlos Ferreira)

<https://www.youtube.com/watch?v=n57YVUVT2A&list=PLVSNL1PHDWvSrVzXyKxBc5cPM4sU7-c2T>

Resumo do curso feito por Roberto Pinheiro

### Aula 01 - O que é e para que serve?

O PHP é uma das linguagens de programação mais usadas no mundo, isso ninguém pode negar. Mas, em relação a outras, tais como Ruby, Python e Perl, PHP possui uma tremenda desvantagem: não possui um sistema totalmente confiável e unificado para gerenciamento de pacotes. Ruby tem Gems, Python, PIP; Perl, CPAN; e, estendendo um pouco, é possível citar o NPM do Node.js. Mas e o PHP?

Na verdade, o PHP até tem um sistema de pacotes, o PEAR, mas, convenhamos, trabalhar com o PEAR não é algo divertido ou produtivo

Composer é uma ferramenta para gerenciamento de dependências em PHP. Ele permite que você declare as bibliotecas dependentes que seu projeto precisa e as instala para você

Usando Composer, você simplesmente especifica quais pacotes (códigos reutilizáveis) seu projeto precisa – podendo estes pacotes também ter dependências – e ele vai, automaticamente, baixar isso e incluir nos locais apropriados de seu projeto!

Caso seja preciso acrescentar, remover ou atualizar algum pacote, sem problemas: o gerenciador também faz o trabalho todo! Acreditem: depois que se começa a trabalhar usando este tipo de ferramenta, realmente é difícil abrir mão de suas facilidades e comodidades; além disso, a produtividade vai às alturas.

## Download

<https://getcomposer.org>



Dependency Manager for PHP

Latest: v1.8.0



Authors: Nils Adermann, Jordi Boggiano and many community contributions

Sponsored by:



Clique no botão "Download".

# Download Composer

Run this in your terminal to get the latest Composer version:

```
curl -sS https://getcomposer.org/installer | php
```

Or if you don't have curl:

```
php -r "readfile('https://getcomposer.org/installer');" | php
```

This installer script will simply check some php.ini settings, warn you if they are set incorrectly, and then download the latest composer.phar in the current directory

No caso de estar usando Windows, basta fazer o download do arquivo executável Composer-Setup.exe:

## Download Composer Latest: v1.8.0

### Windows Installer

The installer will download composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

## Como funciona o Composer



O Composer funciona, basicamente, através de duas vertentes: um **repositório** para os pacotes (**Packagist**) e **instruções via linha de comando** para gerenciamento dos pacotes (para procurar, instalar, atualizar, remover, etc).

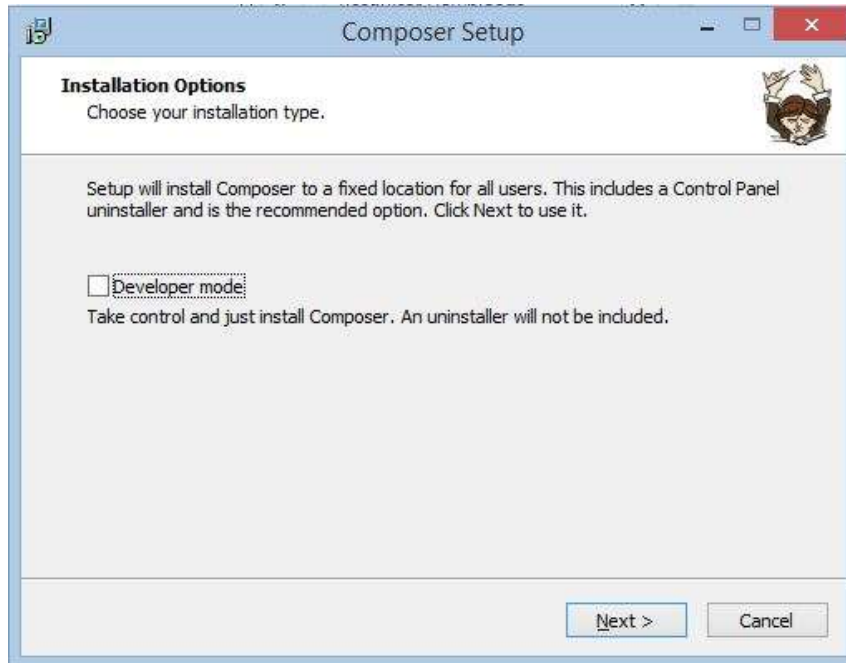
A instalação dos pacotes é feita *por projeto* e, *por default*, nada é instalado globalmente. Por isso o Composer é considerado mais um **Gerenciador de Dependências** do que um *Gerenciador de Pacotes* (mas usar o termo “pacotes”, no caso, também não é errado).

Depois de configurar corretamente o arquivo **composer.json** para instalar as dependências informadas – através do comando `install`, no Terminal – é feita uma verificação para ver se há algum erro de sintaxe, acontece a busca no repositório pelo pacote informado, o download é realizado e a “instalação” feita no diretório apropriado – por padrão, é o diretório `vendor`, na raiz do projeto, mas isso também é configurável.

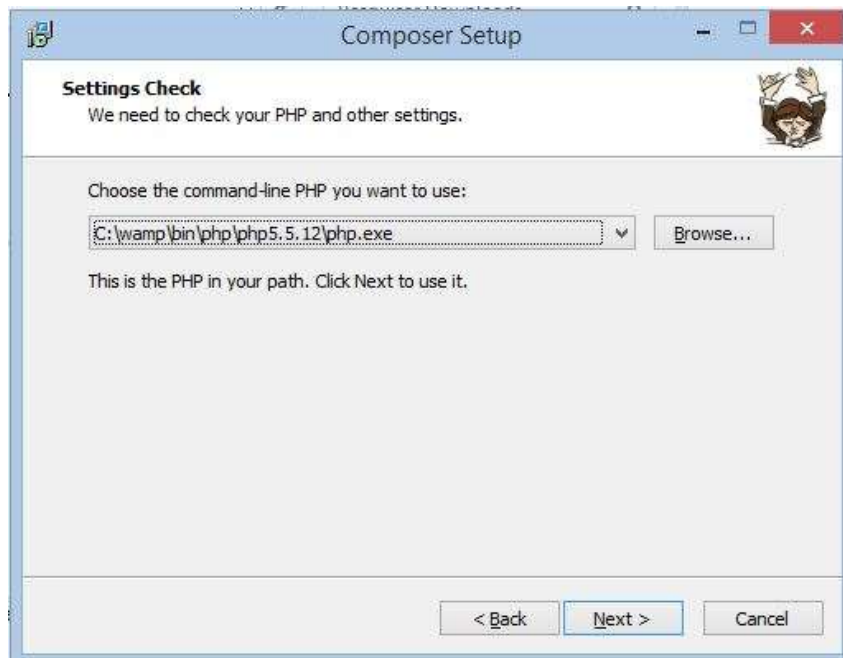
O primeiro passo para entrar no mundo de gerenciamento de pacotes PHP é instalar o Composer.

## Instalação

Para instalar o Composer, antes, instale o Wamp. Após o download do arquivo **Composer-Setup.exe**, clique nele para iniciar a instalação:

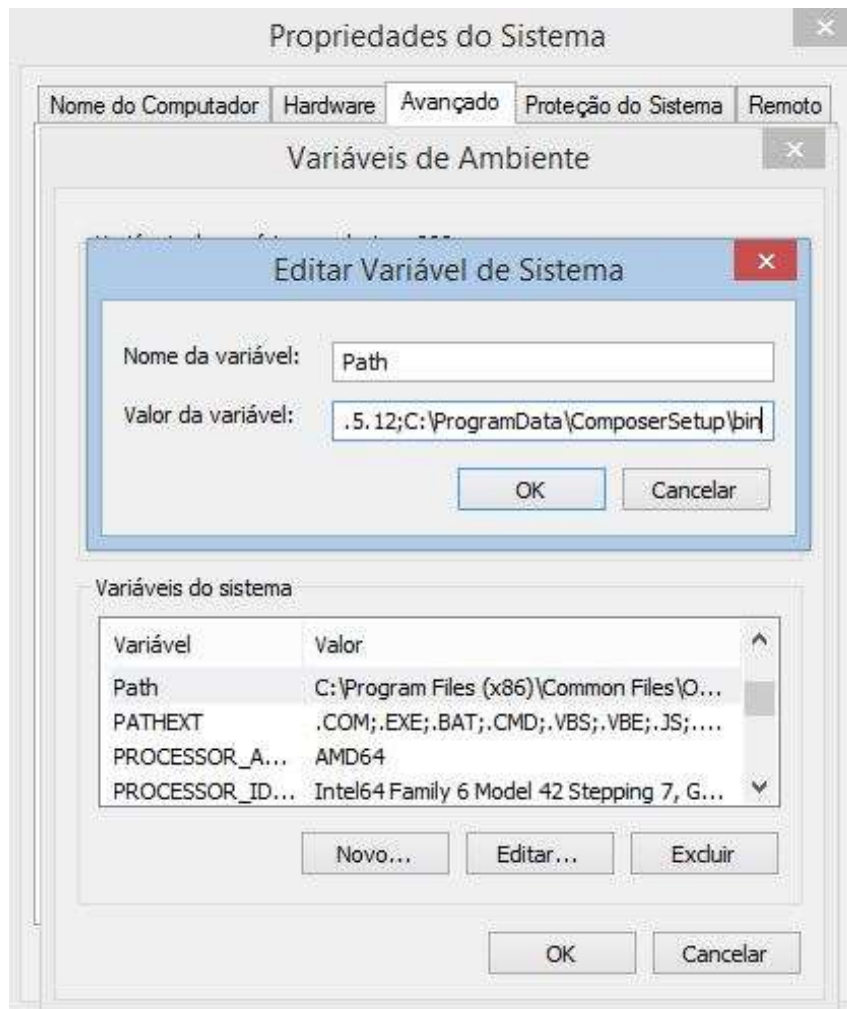


Clique no botão "Next".



Clique novamente no botão "Next". Na tela que se abrir, clique novamente no botão "Next" e finalmente clique no botão "Install" e confirme os demais passos até finalizar a instalação.

Após a instalação, reinicie o computador. Confirme, em "variáveis de ambiente" do Windows, se no **path** foi incluído o caminho para o Composer:



Na pasta `c:\wamp\www` crie a pasta "`curso_composer`".

No prompt do DOS (cmd), acesse a pasta criada e entre com o comando:

`composer`

Se estiver tudo ok, será apresentada a tela a seguir:

```
C:\wamp\www\curso_composer>composer

Composer version 1.8.0 2018-12-03 10:31:16

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi                   Force ANSI output
  --no-ansi                Disable ANSI output
  -n, --no-interaction     Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins             Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  -vvvv, --verbose         Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about                     Shows the short information about Composer.
  archive                  Creates an archive of this composer package.
  browse                   Opens the package's repository URL or homepage in your browser.
  check-platform-reqs     Check that platform requirements are satisfied.
  clear-cache              Clears composer's internal package cache.
  clearcache               Clears composer's internal package cache.
  config                   Sets config options.
  create-project           Creates new project from a package into given directory.
  depends                  Shows which packages cause the given package to be installed.
  diagnose                 Diagnoses the system to identify common errors.
  dump-autoload            Dumps the autoloader.
  dumpautoload             Dumps the autoloader.
  exec                     Executes a vendored binary/script.
  global                   Allows running commands in the global composer dir (<$COMPOSER_HOME>).
  help                     Displays help for a command
  home                     Opens the package's repository URL or homepage in your browser.
  i                         Installs the project dependencies from the composer.lock file if present, or falls back on the composer.json.
  info                     Shows information about packages.
  init                     Creates a basic composer.json file in current directory.
  install                  Installs the project dependencies from the composer.lock file if present, or falls back on the composer.json.
  licenses                 Shows information about licenses of dependencies.
  list                     Lists commands
  outdated                 Shows a list of installed packages that have updates available
```



## Aula 02 - Iniciando o Composer

Ainda na pasta criada, entre com o comando:

`composer init`

Como nome do pacote entre com: `eti/composer-php`

Em "**Description**" entre com: `Composer Curso de PHP`

Em "**author**" entre com `"Carlos Ferreira <carlos@especializati.com.br>"`

Em "**Minimum Stability**", "**Package Type**" e "**License**", tecla "Enter".

Nas próximas duas perguntas entre com "no".

```
C:\wamp\www\curso_composer>composer init

Welcome to the Composer config generator

This command will guide you through creating your composer.json config.
Package name (<<vendor>/<name>>) [betop_000/curso_composer]: eti/composer-php
Description []: Composer Curso de PHP
Author [, n to skip]: Carlos Ferreira <carlos@especializati.com.br>
Minimum Stability []:
Package Type (e.g. library, project, metapackage, composer-plugin) []:
License []:

Define your dependencies.

Would you like to define your dependencies (require) interactively [yes]? no
Would you like to define your dev dependencies (require-dev) interactively [yes]
? no

{
    "name": "eti/composer-php",
    "description": "Composer Curso de PHP",
    "authors": [
        {
            "name": "Carlos Ferreira",
            "email": "carlos@especializati.com.br"
        }
    ],
    "require": {}
}

Do you confirm generation [yes]? yes
C:\wamp\www\curso_composer>
```

Na última pergunta confirme, digitando "**yes**" a geração do arquivo. O arquivo `composer.json` será criado.

### Arquivo `composer.json`

Um dos principais arquivos para se trabalhar com Composer é o `composer.json`. É nele que as instruções sobre os pacotes que serão usados no projeto ficam contidas. Este é um arquivo `“.json”` comum que deve ficar na raiz de seu projeto. Existe uma gama de parâmetros possíveis para o gerenciamento de dependências, mas o mais usado, certamente, é o `require`.



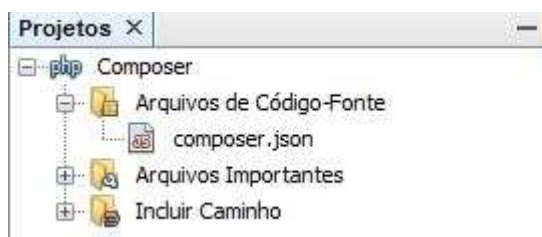
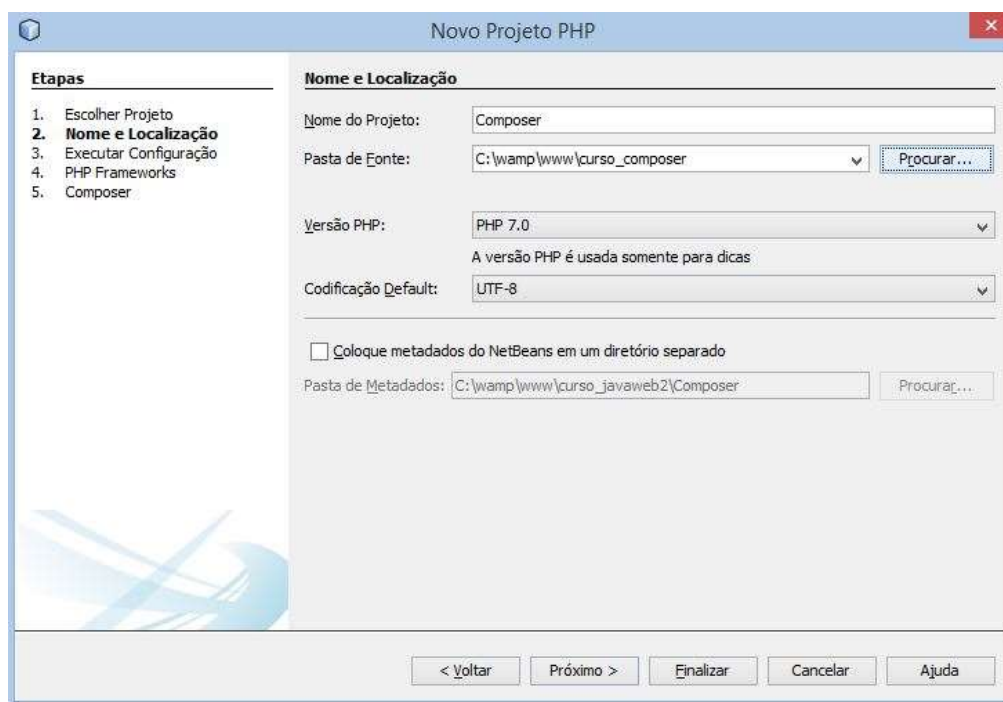
A diretiva “**require**” no composer.json informa ao Composer quais os pacotes necessários para o projeto. Como citado, o repositório central é o **Packagist** (<https://packagist.org/>), então, apesar de ser possível trabalhar com outras “fontes de pacotes”, por padrão é lá que os pacotes que você vai usar estão.

## Arquivo composer.lock

Depois de instalar as dependências informadas, o Composer vai criar automaticamente, também na raiz do projeto, o arquivo **composer.lock**, com informações de quais pacotes estão instalados e quais as versões destes pacotes.

## Trabalhando no NetBeans

Abra o NetBeans e crie um novo projeto (**Aplicação PHP**), na pasta criada (**curso\_composer**), com o nome de “**Composer**”.



Conteúdo do arquivo **composer.json**:

A screenshot of a code editor window titled 'composer.json'. The editor shows the following JSON code:

```
1 {  
2     "name": "eti/composer-php",  
3     "description": "Composer Curso de PHP",  
4     "authors": [  
5         {  
6             "name": "Carlos Ferreira",  
7             "email": "carlos@especializati.com.br"  
8         }  
9     ],  
10    "require": {}  
11 }
```

The code is displayed with syntax highlighting and line numbers on the left side of the editor.

O “name” é o nome de sua aplicação. Esta marcação é opcional mas recomendada.

O “description” é uma breve descrição do que sua aplicação se propõe a fazer. Também opcional.

Em “authors” aparecem os créditos de desenvolvedores que contribuíram com o projeto.

O “require” basicamente deixa claro quais são as dependências de sua aplicação. Neste caso se a versão do PHP for abaixo da 5.2.8 simplesmente uma mensagem de erro será lançada ao instalar as dependências lhe informando que não é possível prosseguir por nem todos os requisitos estarem satisfeitos.

## Aula 03 - Composer update e composer install

### Atualizando pacotes

A atualização de pacotes é feita com o comando **composer update**. Sem fazer nenhuma alteração, na pasta do projeto, entre com o seguinte comando:

**composer update**

```
C:\wamp\www\curso_composer>composer update
Loading composer repositories with package information
Updating dependencies (including require-dev)
Nothing to install or update
Generating autoload files
C:\wamp\www\curso_composer>
```

Foi informado que não foi feita nenhuma instalação ou atualização.



Foi criada uma pasta chamada "**vendor**" contendo uma subpasta chamada "**composer**" e um arquivo chamado **autoload.php**.

Todos os pacotes de terceiros instalados ficarão dentro da pasta "**vendor**".

### Arquivo autoload.php

Então, depois de corretamente instalados os pacotes e o arquivo **composer.lock** ser gerado, é preciso "ensinar" ao seu projeto como autocarregar as dependências para que sejam usadas quando necessário.

### Arquivo index.php

Agora já temos tudo. O composer gerenciando as dependências, as dependências definidas em nosso arquivo **composer.json**, e uma pasta contendo todas as dependências necessárias juntamente com o **autoloader** do composer que encarrega-se de registrar todos os namespaces dos arquivos baixados na pasta vendor. Com isso basta utilizarmos.

Dentro da pasta raiz, crie um arquivo chamado **index.php** e inclua o autoloader do composer conforme o exemplo abaixo.

```
<?php  
header('Content-Type: text/html; charset=utf-8');  
require_once './vendor/autoload.php';
```

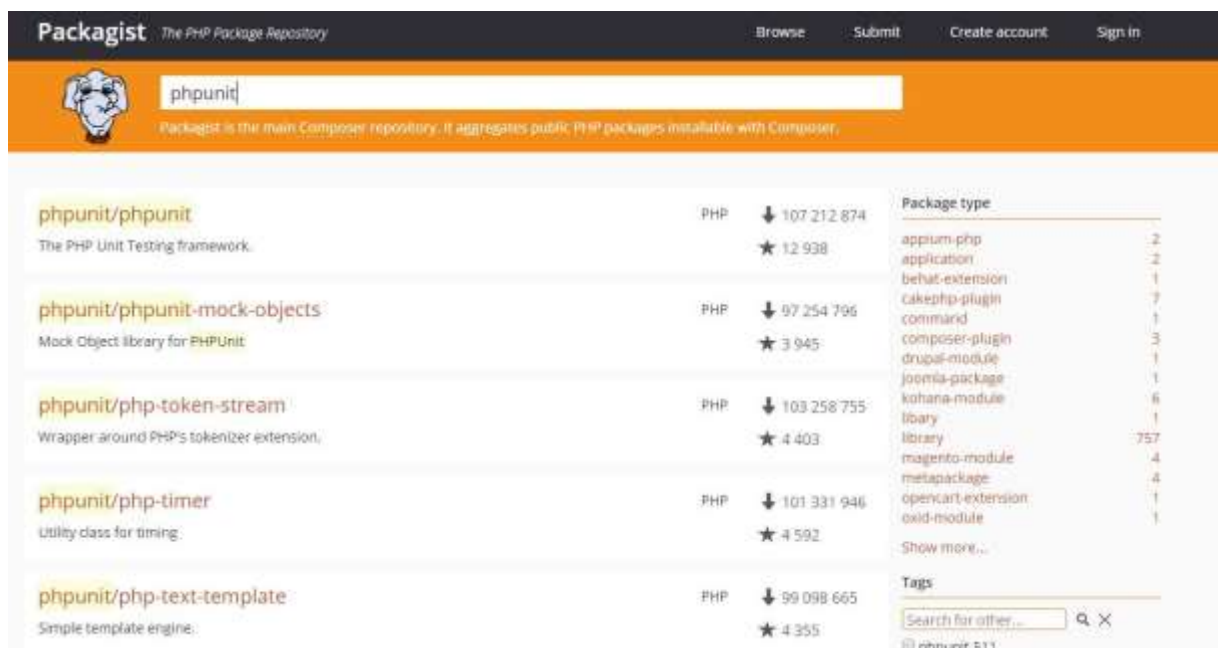
## Aula 06 - Escolhendo pacotes no Packagist

Agora que você já tem o esqueleto de seu composer configurado em sua aplicação falta incluir alguns pacotes.

O composer utiliza como seu repositório o Packagist (<https://packagist.org/>) onde qualquer desenvolvedor pode criar seus próprios pacotes e disponibilizá-los para a comunidade semelhante o github (<https://github.com/>). O Packagist lhe fornece o total de instalações dos pacotes por dia, mês e o total. O mais legal é que estas estatísticas são fiéis, ou seja, se alguém remover um pacote do seu `_composer.json_` o total de instalações é reduzido. Com esta informação restam contagens apenas aplicações que realmente estão utilizando determinado pacote.

### Instalando o pacote PHPUnit

O PHPUnit é utilizado para fazer testes de software.

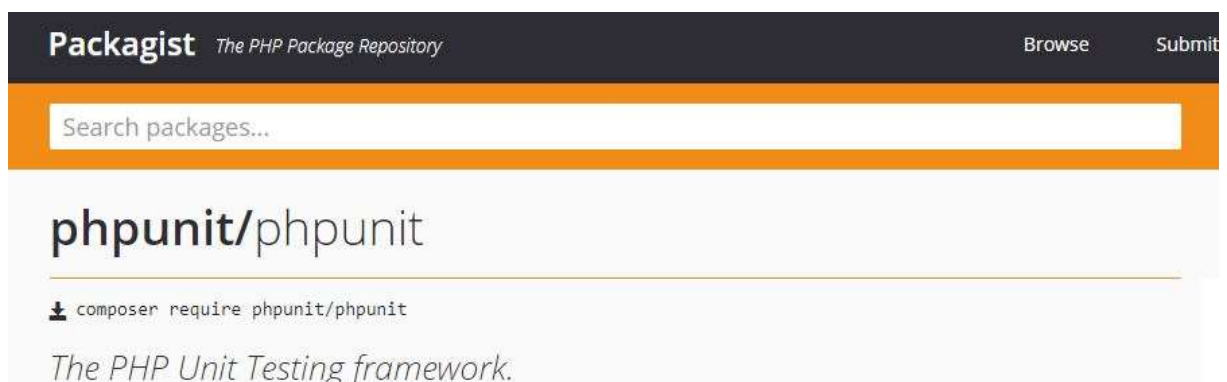


The screenshot shows the Packagist website interface. At the top, there's a navigation bar with 'Packagist The PHP Package Repository', 'Browse', 'Submit', 'Create account', and 'Sign in'. Below the navigation bar is a search bar with 'phpunit' entered. The main content area displays a list of search results for 'phpunit'. The first result is 'phpunit/phpunit' with a description 'The PHP Unit Testing framework.' and statistics: 107,212,874 downloads and 12,938 stars. To the right of the results is a 'Package type' sidebar with a list of categories and their counts. At the bottom right, there's a 'Tags' section with a search bar and a list of tags, including 'phpunit 5.11'.

Package	Type	Downloads	Stars
phpunit/phpunit	PHP	107,212,874	12,938
phpunit/phpunit-mock-objects	PHP	97,254,796	3,945
phpunit/php-token-stream	PHP	103,258,755	4,403
phpunit/php-timer	PHP	101,331,946	4,592
phpunit/php-text-template	PHP	99,098,665	4,355

Package type	Count
appium-php	2
application	2
behat-extension	1
cakephp-plugin	7
command	1
composer-plugin	3
drupal-module	1
joomla-package	1
kohana-module	6
library	757
magento-module	4
metapackage	4
opencart-extension	1
oxid-module	1

Tags: Search for other... phpunit 5.11



The screenshot shows the Packagist website interface for the 'phpunit/phpunit' package. At the top, there's a navigation bar with 'Packagist The PHP Package Repository', 'Browse', and 'Submit'. Below the navigation bar is a search bar with 'Search packages...'. The main content area displays the details for the 'phpunit/phpunit' package. The package name 'phpunit/phpunit' is prominently displayed. Below it, there's a command to install the package: 'composer require phpunit/phpunit'. The description 'The PHP Unit Testing framework.' is also visible.

Search packages...

# phpunit/phpunit

composer require phpunit/phpunit

The PHP Unit Testing framework.

Altere o arquivo **composer.json** para:

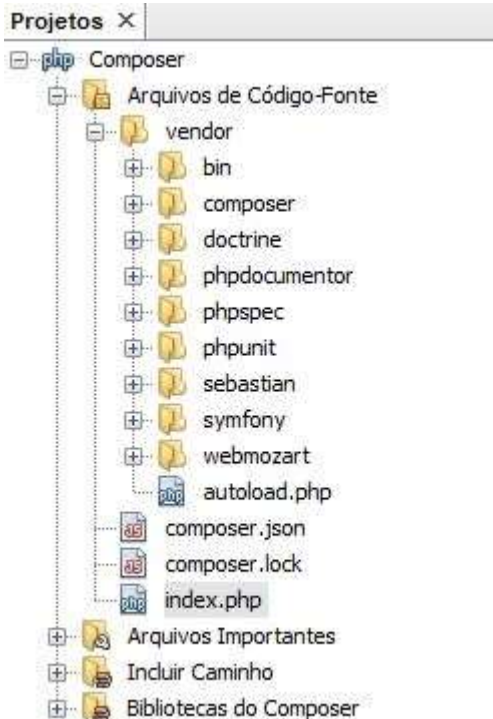
```
{
  "name": "eti/composer-php",
  "description": "Composer Curso de PHP",
  "authors": [
    {
      "name": "Carlos Ferreira",
      "email": "carlos@especializati.com.br"
    }
  ],
  "require": {
    "php": ">=5.3.3"
  },
  "require-dev": {
    "phpunit/phpunit": "^4.3"
  }
}
```

No prompt do DOS, na pasta do projeto, entre com o seguinte comando:

**composer update**

```
C:\wamp\www\curso_composer>composer update
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 22 installs, 0 updates, 0 removals
- Installing symfony/polyfill-ctype (v1.10.0): Downloading (100%)
- Installing symfony/yaml (v3.4.20): Downloading (100%)
- Installing sebastian/version (1.0.6): Downloading (100%)
- Installing sebastian/global-state (1.1.1): Downloading (100%)
- Installing sebastian/recursion-context (1.0.5): Downloading (100%)
- Installing sebastian/exporter (1.2.2): Downloading (100%)
- Installing sebastian/environment (1.3.8): Downloading (100%)
- Installing sebastian/diff (1.4.3): Downloading (100%)
- Installing sebastian/comparator (1.2.4): Downloading (100%)
- Installing doctrine/instantiator (1.0.5): Downloading (100%)
- Installing phpunit/php-text-template (1.2.1): Downloading (100%)
- Installing phpunit/phpunit-mock-objects (2.3.8): Downloading (connecting...)
Downloading (100%)
- Installing phpunit/php-timer (1.0.9): Downloading (100%)
- Installing phpunit/php-file-iterator (1.4.5): Downloading (100%)
- Installing phpunit/php-token-stream (1.4.12): Downloading (100%)
- Installing phpunit/php-code-coverage (2.2.4): Downloading (100%)
- Installing webmozart/assert (1.4.0): Downloading (100%)
- Installing phpdocumentor/reflection-common (1.0.1): Downloading (connecting.
Downloading (100%)
- Installing phpdocumentor/type-resolver (0.3.0): Downloading (100%)
- Installing phpdocumentor/reflection-docblock (3.2.2): Downloading (connectin
Downloading (100%)
- Installing phpspec/prophecy (1.8.0): Downloading (100%)
- Installing phpunit/phpunit (4.8.36): Downloading (100%)
symfony/yaml suggests installing symfony/console (For validating YAML files usin
g the lint command)
sebastian/global-state suggests installing ext-uopz (*)
phpunit/phpunit suggests installing phpunit/php-invoker (~1.1)
Writing lock file
Generating autoload files
C:\wamp\www\curso_composer>
```





Serão instalados as dependências do pacote PHPUnit dentro da pasta "vendor".

```
C:\wamp\www\curso_composer>composer update
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 22 installs, 0 updates, 0 removals
- Installing symfony/polyfill-ctype (v1.10.0): Downloading (100%)
- Installing symfony/yaml (v3.4.20): Downloading (100%)
- Installing sebastian/version (1.0.6): Downloading (100%)
- Installing sebastian/global-state (1.1.1): Downloading (100%)
- Installing sebastian/recursion-context (1.0.5): Downloading (100%)
- Installing sebastian/exporter (1.2.2): Downloading (100%)
- Installing sebastian/environment (1.3.8): Downloading (100%)
- Installing sebastian/diff (1.4.3): Downloading (100%)
- Installing sebastian/comparator (1.2.4): Downloading (100%)
- Installing doctrine/instantiator (1.0.5): Downloading (100%)
- Installing phpunit/php-text-template (1.2.1): Downloading (100%)
- Installing phpunit/phpunit-mock-objects (2.3.8): Downloading (connecting...)
Downloading (100%)
- Installing phpunit/php-timer (1.0.9): Downloading (100%)
- Installing phpunit/php-file-iterator (1.4.5): Downloading (100%)
- Installing phpunit/php-token-stream (1.4.12): Downloading (100%)
- Installing phpunit/php-code-coverage (2.2.4): Downloading (100%)
- Installing webmozart/assert (1.4.0): Downloading (100%)
- Installing phpdocumentor/reflection-common (1.0.1): Downloading (connecting.
Downloading (100%)
- Installing phpdocumentor/type-resolver (0.3.0): Downloading (100%)
- Installing phpdocumentor/reflection-docblock (3.2.2): Downloading (connectin
Downloading (100%)
- Installing phpspec/prophecy (1.8.0): Downloading (100%)
- Installing phpunit/phpunit (4.8.36): Downloading (100%)
symfony/yaml suggests installing symfony/console (For validating YAML files usin
g the lint command)
sebastian/global-state suggests installing ext-uopz (*)
phpunit/phpunit suggests installing phpunit/php-invoker (~1.1)
Writing lock file
Generating autoload files
C:\wamp\www\curso_composer>
```



## Instalando o pacote monolog

The screenshot shows the Packagist search results for the query 'monolog'. The search bar at the top contains the text 'monolog'. Below the search bar, a list of packages is displayed. The first package is 'monolog/monolog', which is highlighted. It has a description: 'Sends your logs to files, sockets, inboxes, databases and various web services'. To the right of the package name, it shows 'PHP' and a download count of '122 697 084'. Below the download count is a star icon and the number '13 497'. To the right of the package name, there is a 'Package type' section with a list of package types and their counts: 'aimeos-extension' (1), 'bundle' (6), 'cakephp-plugin' (2), 'formatter' (1), 'ladybug-plugin' (1), 'laravel-package' (1), 'library' (660), 'magento2-module' (3), 'monolog-handler' (1), 'netta-addon' (1), 'october-plugin' (1), 'opus-package' (1), 'package' (3), 'pff2-module' (1), and 'phalcon-library' (1). Below the 'Package type' section is a 'Tags' section with a search bar and a list of tags: 'monolog' (290). The second package in the list is 'symfony/monolog-bundle', which has a description: 'Symfony MonologBundle'. It has a download count of '36 620 436' and a star icon with the number '1 278'. The third package is 'symfony/monolog-bridge', which has a description: 'Symfony Monolog Bridge'. It has a download count of '5 291 721' and a star icon with the number '1 008'. The fourth package is 'easycorp/easy-log-handler', which has a description: 'A handler for Monolog that optimizes log messages to be processed by humans instead of software. Improve your productivity with logs that are easy to understand.' It has a download count of '1 284 367' and a star icon with the number '1 023'. The fifth package is 'maxbanton/cwh', which has a description: 'AWS CloudWatch Handler for Monolog library'. It has a download count of '512 854' and a star icon with the number '128'.

Package	Description	PHP	Downloads	Stars	Package type	Tags
monolog/monolog	Sends your logs to files, sockets, inboxes, databases and various web services	PHP	122 697 084	13 497	aimeos-extension bundle cakephp-plugin formatter ladybug-plugin laravel-package library magento2-module monolog-handler netta-addon october-plugin opus-package package pff2-module phalcon-library	monolog 290
symfony/monolog-bundle	Symfony MonologBundle	PHP	36 620 436	1 278		
symfony/monolog-bridge	Symfony Monolog Bridge	PHP	5 291 721	1 008		
easycorp/easy-log-handler	A handler for Monolog that optimizes log messages to be processed by humans instead of software. Improve your productivity with logs that are easy to understand.	PHP	1 284 367	1 023		
maxbanton/cwh	AWS CloudWatch Handler for Monolog library	PHP	512 854	128		

The screenshot shows the Packagist package page for 'monolog/monolog'. The search bar at the top contains the text 'monolog/monolog'. Below the search bar, the package name 'monolog/monolog' is displayed. To the left of the package name, there is a command to install the package: 'composer require monolog/monolog'. Below the command, there is a description: 'Sends your logs to files, sockets, inboxes, databases and various web services'. To the right of the package name, there is a 'Maintainers' section with a profile picture of the maintainer. Below the 'Maintainers' section is a 'Details' section with links to the package's GitHub repository, homepage, source, and issues. Below the 'Details' section is a table with statistics: 'Install: 122 823 530', 'Dependents: 3 925', 'Suggesters: 353', 'Stars: 13 049', 'Watchers: 337', 'Forks: 1 474', and 'Open Issues: 23'.

Package	Description	PHP	Downloads	Stars	Package type	Tags
monolog/monolog	Sends your logs to files, sockets, inboxes, databases and various web services	PHP	122 823 530	13 049	aimeos-extension bundle cakephp-plugin formatter ladybug-plugin laravel-package library magento2-module monolog-handler netta-addon october-plugin opus-package package pff2-module phalcon-library	monolog 290

Na pasta do projeto, entre com o seguinte comando:

```
composer require monolog/monolog
```

```

C:\wamp\www\curso_composer>composer require monolog/monolog
Using version ^1.24 for monolog/monolog
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 2 installs, 0 updates, 0 removals
  - Installing psr/log (1.1.0): Downloading (100%)
  - Installing monolog/monolog (1.24.0): Downloading (100%)
monolog/monolog suggests installing aws/aws-sdk-php (Allow sending log messages
to AWS services like DynamoDB)
monolog/monolog suggests installing doctrine/couchdb (Allow sending log messages
to a CouchDB server)
monolog/monolog suggests installing ext-amqp (Allow sending log messages to an A
MQP server (1.0+ required))
monolog/monolog suggests installing ext-mongo (Allow sending log messages to a M
ongoDB server)
monolog/monolog suggests installing graylog2/gelf-php (Allow sending log message
s to a GrayLog2 server)
monolog/monolog suggests installing mongodb/mongodb (Allow sending log messages
to a MongoDB server via PHP Driver)
monolog/monolog suggests installing php-amqp/lib/php-amqp/lib (Allow sending log m
essages to an AMQP server using php-amqp/lib)
monolog/monolog suggests installing php-console/php-console (Allow sending log m
essages to Google Chrome)
monolog/monolog suggests installing rollbar/rollbar (Allow sending log messages
to Rollbar)
monolog/monolog suggests installing ruflin/elastica (Allow sending log messages
to an Elastic Search server)
monolog/monolog suggests installing sentry/sentry (Allow sending log messages to
a Sentry server)
Writing lock file
Generating autoload files
C:\wamp\www\curso_composer>

```

O conteúdo do arquivo **composer.json** será alterado para:

```

{
  "name": "eti/composer-php",
  "description": "Composer Curso de PHP",
  "authors": [
    {
      "name": "Carlos Ferreira",
      "email": "carlos@especializati.com.br"
    }
  ],
  "require": {
    "php": ">=5.3.3",
    "monolog/monolog": "^1.24"
  },
  "require-dev": {
    "phpunit/phpunit": "^4.3"
  }
}

```

Na pasta **vendor** será incluída uma pasta para o pacote **monolog**:



Para usar o monolog basta consultar sua documentação.

## Aula 07 - Usando um pacote para validar dados (GUMP)

### Instalando o pacote gump

É utilizado para validação de dados.

The screenshot shows the Packagist website with the search bar containing 'gump'. The results list several packages, with 'wixel/gump' being the most popular. The 'Package type' sidebar on the right lists various categories like 'library', 'application', and 'plugin'.

Package	PHP	Downloads	Stars
wixel/gump	PHP	219 920	984
rwitchell/gump-form-builder	PHP	39	5
accexs/validator	PHP	25	0
ubcent/gump	PHP	16	0
romangorbatko/gump	PHP	239	0

The screenshot shows the detailed page for the 'wixel/gump' package. It includes the package description, a list of maintainers, and various statistics such as installs, dependents, and stars.

**wixel/gump**

composer require wixel/gump

A fast, extensible & stand-alone PHP input validation class that allows you to validate any data

**Maintainers:**

- github.com/Wixel/GUMP
- Homepage
- Source
- Issues

**Details:**

- Installs: 220 154
- Dependents: 25
- Suggesters: 0
- Stars: 977
- Watchers: 92
- Forks: 322
- Open Issues: 45

Na pasta do projeto, entre com o seguinte comando:

```
composer require wixel/gump
```

```
C:\wamp\www\curso_composer>composer require wixel/gump
Using version ^1.5 for wixel/gump
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Installing wixel/gump (v1.5.7): Downloading (100%)
Writing lock file
Generating autoload files
C:\wamp\www\curso_composer>
```

O conteúdo do arquivo **composer.json** será alterado para:

```
{
  "name": "eti/composer-php",
  "description": "Composer Curso de PHP",
  "authors": [
    {
      "name": "Carlos Ferreira",
      "email": "carlos@especializati.com.br"
    }
  ],
  "require": {
    "php": ">=5.3.3",
    "monolog/monolog": "^1.24",
    "wixel/gump": "^1.5"
  },
  "require-dev": {
    "phpunit/phpunit": "^4.3"
  }
}
```

## Usando o pacote

Altere o arquivo `index.php` para:

```
<?php

header('Content-Type: text/html; charset=utf-8');

use \GUMP;

require_once './vendor/autoload.php';

$dadosForm = ['nome' => 'Carlos Ferreira', 'email' => 'carlos@especializati.com.br'];
$regras = [
    'nome' => 'required|max_len,100|min_len,3',
    'email' => 'required|max_len,150|valid_email'
];

$is_valid = GUMP::is_valid($dadosForm, $regras);

if($is_valid === true) {
    echo "Cadastrar dados, correto!";
} else {
    print_r($is_valid);
}
```

