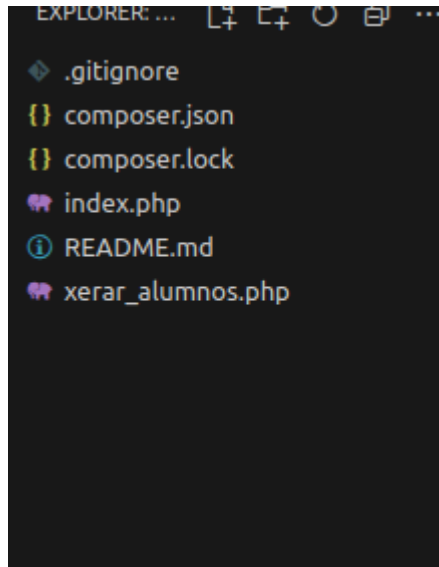


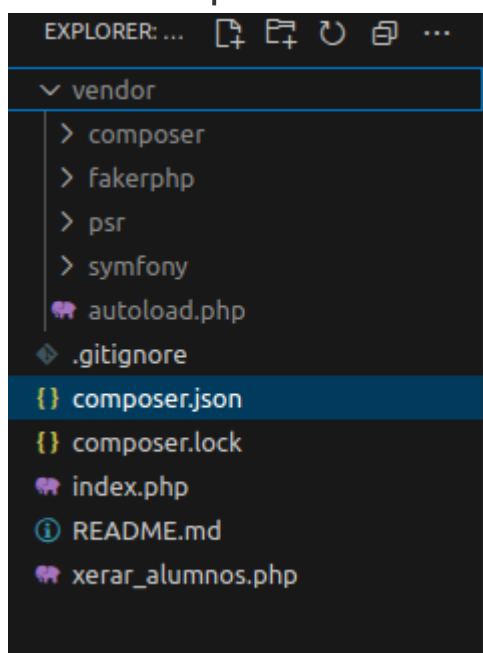
1. Inicia a máquina virtual `dapw_iniciais_server_01` e conéctate por SSH dende o Visual Studio Code.
2. Sitúate no directorio `home` de `dadmin` e executa o seguinte comando para borrar todo o contido: `rm -R * -f.`

Comezaremos **clonando o repositorio** que contén o proxecto de PHP e **instalaremos as librerías** necesarias para o correcto funcionamento **do proxecto**.

1. Clona o repositorio `iniciaist0201_1`. A continuación abre o directorio do repositorio en Visual Studio Code.
2. Comproba que non temos o directorio `vendor`. **Realiza unha captura** desta comprobación.



3. Utilizando `Composer` instala tódolos paquetes que necesita o proxecto para funcionar.
4. **Realiza unha captura** do contido do directorio `vendor`.



Se nos fixamos no ficheiro `composer.json` podemos observar que sempre se instalará unha versión maior se existe a que se indica no ficheiro. Realizaremos unhas modificacións pertinentes para que sempre se instale a mesma versión e non ter problemas nun futuro con incompatibilidades.

1. Elimina o directorio `vendor`. Deste xeito borraremos tódalas librerías instaladas.
2. Modifica o ficheiro `composer.json` para que instale unha versión concreta dos paquetes instalados.

Realiza unha captura do contido deste ficheiro.

```
{ } composer.json M X
{ } composer.json > { } require > fkerphp/faker
1  {
2      "name": "dadmin/d25vm-project",
3      "authors": [
4          {
5              "name": "david"
6          }
7      ],
8      "require": {
9          "fakerphp/faker": "1.24"
10     }
11 }
12
```

3. Instala tódolos paquetes que se indica no ficheiro `composer.json`. Deberíache saír unha mensaxe similar a seguinte:

```
Verifying lock file contents can be installed on current platform.
Warning: The lock file is not up to date with the latest changes in composer.json. You
may be getting outdated dependencies. It is recommended that you run `composer update`
or `composer update <package name>`.
```

4. Esta mensaxe indícanos que `composer.json` e `composer.lock` non están sincronizados. Se queremos sincronizalos teremos que executar o comando `composer update`.
5. Executa o *script* co seguinte comando `php xerar_alumnos.php` para ver que todo funciona correctamente. **Realiza unha captura** do ficheiro creado por este *script*.

```
EXPLORER: ...  { } composer.json M  alumnos.csv X
out
alumnos.csv
> vendor
.gitignore
composer.json M
composer.lock M
index.php
README.md
xerar_alumnos.php

out > alumnos.csv
1  Nome,Apellido,"Data de Nacemento"
2  Diana,Domínguez,1999-11-19
3  Emilia,Roybal,1982-07-18
4  "Rosa María",Vásquez,2006-04-30
5  "Francisco Javier",Oliva,1975-11-13
6  Claudia,Elizondo,1986-09-24
7  Álvaro,Gonzáles,2004-07-03
8  Sergio,Alemán,1977-10-06
9  Alejandro,Gastélum,2001-04-21
10 Verónica,Delatorre,1984-04-10
11 Erik,Estévez,1971-08-25
12
```

Para evitarnos futuros problemas, **no momento de instalar liberías podemos indicar a versión concreta que desexamos**. Neste caso vamos instalar unha versión antiga do paquete `friendsofphp/php-cs-fixer`. Este paquete é unha ferramenta para formatar e corrixir o estilo do código PHP automaticamente. Ademais é un **paquete de desenvolvemento**, polo que so nese contorno debe ser instalado (nunca se debe instalar nun contorno de produción).

1. Vai a páxina web de `Composer`: <https://getcomposer.org/>. Preme en `Browse Packages`. E busca por `php-cs-fixer`. Entra na páxina do paquete que desexamos instalar.
2. Na parte dereita da web, poderedes observar as versións dispoñibles. Instala a penúltima versión. **Realiza unha captura** do comando utilizado para realizar esta instalación. Recorda utilizar a opción `--dev` para que só se instale nun contorno de desenvolvemento.
`composer require --dev friendsofphp/php-cs-fixer`
3. **Realiza unha captura** do contido do contido do ficheiro `composer.json`.

```
{ } composer.json > ...
1  {
2      "name": "dadmin/d25vm-project",
3      "authors": [
4          {
5              "name": "david"
6          }
7      ],
8      "require": {
9          "fakerphp/faker": "1.24"
10     },
11     "require-dev": {
12         "friendsofphp/php-cs-fixer": "3.88"
13     }
14 }
15
```

4. Realiza un `commit` e un `push` para subir o repositorio a GitLab.com.
5. **Realiza unha captura** do repositorio en GitLab.com que se vexa o contido deste.



composercosas

david vidal authored 43 seconds ago

1f241d37



History

Name	Last commit	Last update
.gitignore	xerar alumnos	1 day ago
README.md	arranxando a cagada	3 days ago
composer.json	composercosas	43 seconds ago
composer.lock	composercosas	43 seconds ago
index.php	index.php	1 day ago
xerar_alumnos.php	xerar alumnos	1 day ago

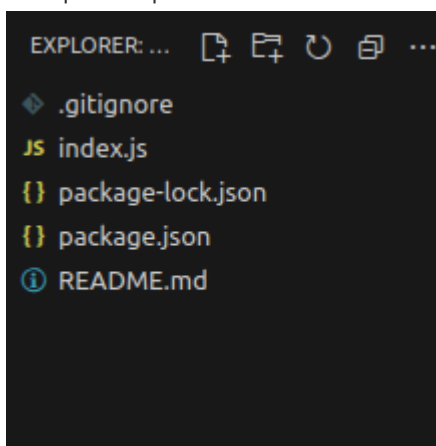
README.md

Readme editado

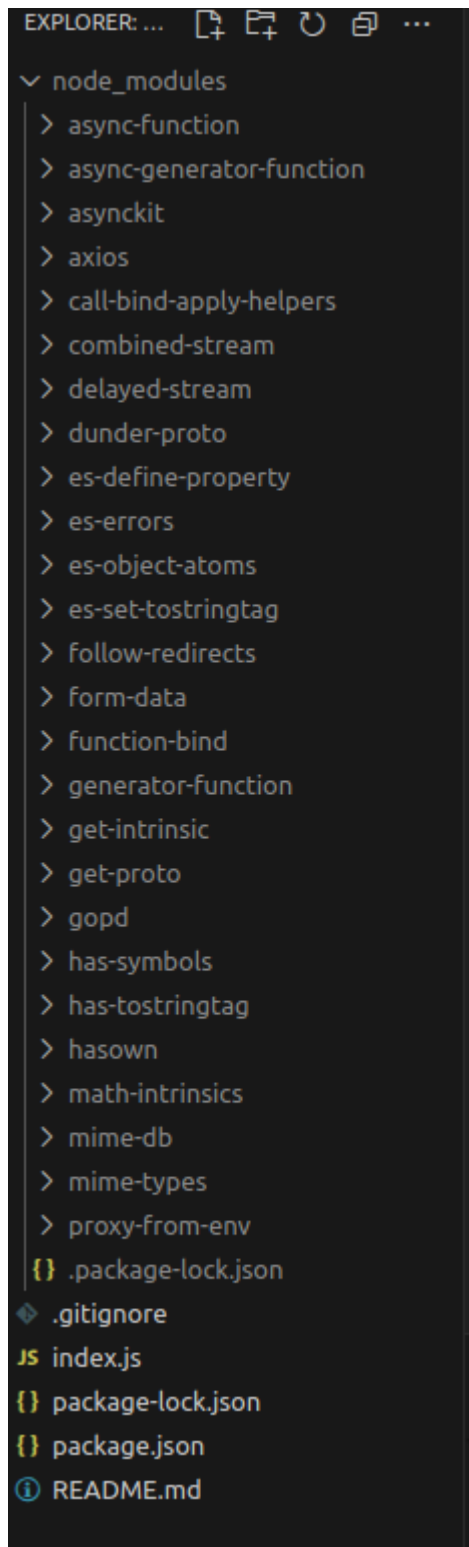
Exercicio 2: Reprodución de contorno de desenvolvemento con NPM

Neste exercicio faremos algo similar ao realizado no exercicio anterior, pero esta vez utilizando `npm` en lugar de `Composer`.

1. Clona o repositorio `iniciaist0201_2`. A continuación abre o directorio do repositorio en Visual Studio Code.
2. Comproba que non temos o directorio `node_modules`. **Realiza unha captura** desta comprobación.



3. Utilizando `npm` instala tódolos paquetes que necesita o proxecto para funcionar.
4. **Realiza unha captura** do contido do directorio `node_modules`.



Se comprobas o ficheiro `package.json` verás que non se indica unha versión concreta dos paquetes. Vamos modificar isto.

1. Borra o directorio `node_modules`. Así eliminamos todos os módulos instalados.
2. Modifica o ficheiro `package.json` para que se instale unha versión concreta. **Realiza unha captura** do

contido deste ficheiro.

```
{ } package.json M X
{ } package.json > { } dependencies > axios
1  {
2    "name": "dvmt0201_2",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC",
11   "dependencies": {
12     "axios": "1.12.2"
13   }
14 }
15
```

3. Instala os paquetes que se indican no ficheiro `package.json`.
4. Executa o comando `node index.js` para ver que todo funciona correctamente. **Realiza capturas** da saída da execución do *script*.

```
dadmin@dvmserver01:~/dvmt0201_2$ node index.js
🌍 Cidade: Santiago de Compostela
🌡 Temperatura: 22.9 °C
💨 Vento: 5.3 km/h
🌪 Dirección do vento: 62°
🕒 Hora da medición: 2025-09-30T14:15
```

Para finalizar veremos como se **instala un paquete de desenvolvemento** nunha versión concreta. Neste caso vamos instalar o paquete `eslint`. Esta é unha ferramenta para analizar e corrixir automaticamente o estilo e os erros potenciais no teu código JavaScript

1. Vai a páxina de `npm` (<https://www.npmjs.com/>). Busca o paquete `eslint` e vaite a súa páxina en `npm`. Preme na lapela de `versions`.
2. Instala a penúltima versión do paquete en modo desenvolvemento. **Realiza captura** do comando utilizado.

```
$ npm install eslint@9.35.0 --save-dev
```
3. **Realiza unha captura** do contido do ficheiro `packages.json`.

```
{ } package.json M X
{ } package.json > ...
1  {
2    "name": "dvmt0201_2",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "author": "",
10   "license": "ISC",
11   "dependencies": {
12     "axios": "1.12.2"
13   },
14   "devDependencies": {
15     "eslint": "^9.35.0"
16   }
17 }
18
```

4. Realiza un `commit` e un `push` para subir o repositorio a GitLab.com.
5. **Realiza unha captura** do repositorio en GitLab.com que se vexa o contido deste.


September 30, 2025

main dvmt0201_2

+






Find file


Code

 **eslint**
david vidal authored 24 seconds ago

4c5e379b

History

Name	Last commit	Last update
 .gitignore	node modules	48 minutes ago
 README.md	Initial commit	1 hour ago
 index.js	ventiño	44 minutes ago
 package-lock.json	eslint	25 seconds ago
 package.json	eslint	25 seconds ago

 README.md

dvmt0201_2