



A continuació trobaràs enllaços a recursos i tutorials, que t'ajudaran a resoldre l'activitat d'aquest apartat:

Tasca -> [PHP Avançat](#)



Fins ara hem après bàsicament a traduir a PHP el que vas fer a fonaments (a més de maquetació bàsica i ús d'algunes eines elementals del gremi).

En aquest tema desenvoluparem algunes habilitats que són pròpies de llenguatge PHP i que no necessàriament hauriem de veure en altre llenguatge similar.

Moltes de les utilitats que veurem aquí estan especialment pensades pel context en el qual PHP és fort, el **desenvolupament d'aplicacions web**.

Endavant!

## Scope de variables

Segurament ja estareu familiaritzats amb l'àmbit de les variables de la vostra formació a fonaments. Tot i això, convé tenir al cap el particular tractament d'aquest tema en el context de PHP.

### Àmbit de variables a PHP

-> [Ámbito de las variables en PHP](#)

### PHP Variables Scope

-> [PHP Variables Scope](#)

## Variables SuperGlobals

A més dels àmbits esmentats en l'anterior secció, a PHP a més tenim l'àmbit superglobal. Aquest àmbit permet tractar amb variables que tenen com a context tota l'aplicació PHP i en tenim força tipus ja definits al mateix llenguatge.

-> [Interfaces](#)

### Superglobals

-> [Superglobals](#)

### Explicació de les variables superglobals a PHP

-> [Explicación de las variables superglobales en PHP: con una hoja de referencia](#)

## Sessions

Una de les variables superglobals més utilitzades és la variable `$_SESSION`, en aquesta secció entendrem per què.

### Sessions a PHP

-> [Sesiones en PHP](#)

## PHP Sessions

->[PHP Sessions](#)

## Constants predefinides (o màgiques)

PHP porta de sèrie tot un grup de constants de diferent significat que convé conèixer.

### Constants Predefinides

->[Constantes predefinidas](#)

## Mètodes màgics



Sabem que podem reprogramar el que passa quan instanciem una classe gràcies al mètode `__construct()`.

Però encara tenim més comportaments per defecte de les classes que podem redefinir mitjançant la sobreescritura dels mètodes màgics.

->[Métodos mágicos](#)

### Mètodes màgics explicats

->[PHP Magic Methods Explained](#)

## Enums

Quan hem de definir un tipus de variable que té un nombre de valors acotat, però no cap comportament associat, convé pensar en enums per representar aquests tipus de dades de forma més acurada.

### PHP 8.1 Enums

->[PHP 8.1 Enums](#)

### Enumerats a PHP

->[PHP 8.1 Enumerados](#)

## Traits

Un altre mecanisme menys conegut, però també existent a PHP de cara a millorar la reutilització de codi font són els Traits.

### Traits a PHP

->[Rasgos \(Traits\)](#)

### Reutilitzant codi font amb Traits a PHP

->[Using namespaces: Basics](#)

## Diferències entre interfície, classe abstracta, i trait a PHP

->[The difference between Traits, Interfaces, and Abstract Classes in PHP](#)

## Namespaces

Quan els programes creixen molt, es pot donar el cas que anomenem classes de manera idèntica però referint diferents contexts. Els namespaces apareixerien aquí per evitar conflictes de noms.

### Namespaces a PHP

->[Namespaces en PHP](#)

### Utilitzants namespaces a PHP

->[Using namespaces: Basics](#)

## Utilitzant llibreries externes a PHP

Si volem utilitzar aplicacions PHP de tercers per donar suport a la nostra pròpia aplicació, hem de donar la benvinguda a les llibreries. Podem instal·lar-les "a mà" però el normal és usar un gestor de dependències. A PHP, el gestor per excel·lència és Composer i tot i que no el farem servir en aquest Sprint és d'obligat coneixement per a qualsevol programador/a PHP l'existència d'aquest gestor.

### Composer

->[A Dependency Manager for PHP](#)

## Com organitzar fitxers a PHP.

Quan els programes creixen molt (cosa que veuràs molt clarament a l'Sprint 3) necessitem crear una estructura de carpetes per decidir on van, per exemple, les classes responsables d'un determinat tipus de tasques, els recursos d'imatges, les pàgines HTML, el codi CSS... Etcètera. En aquest sentit, pots trobar la teva pròpia manera de fer-ho, però aquí et deixem algunes idees. Per altra banda, pensa que si fas servir un framework (cosa que passarà a partir de l'Sprint 4) l'estructura et vindrà imposada pel propi framework.

### Estructura projectes PHP

->[Estructura](#)

### Com organitzar l'estructura de carpetes?

¿Cómo organizar la estructura de carpetas? en un proyecto MVC PHP - MYSQL

