Robert Salazar Arias 11rsahome@gmail.com

OOP

Overview

The aim is to understand the fundamental principles of the OOP

It's a tool that detects errors and won't let you move forward until you correct it

Table of contents

Phase I - Project Initiation

Project requirements

- We will use PHP to do the practice of the OOP project
    - We have to install whether it's a development dependency or not.
    - Checking the new directory called Vendor

Project specifications

- The project must be developed in PHP or j.
- Create a Git repository
- Don't raise dependence
- The directory structure of the project must be well defined and organized.
- The code must be documented correctly using the English language.
- Your code must use a *camelCase style.*
- If you use HTML do not use inline styles
- The project must not contain unused files.
- The project must be developed using *git*, using explicit and concise deconfirmation messages.
- Delete files that are not necessary to evaluate the project
- The project must contain a *README* file written in *Markdown* that shows a brief description and the steps for runél.

Phase II - Project Planning

Reasoning

This pill will split 3 part implementationa case study about PHP performing the tests, explaining the presentation how it develops.

- Class, object and instance
- Encapsulation
- Abstraction
- Inheritance
- Static Class
- Polymorphism
- Overload
- Override

We will also conduct research earlier on the concepts to be clarified.

- What is **object-oriented programmingin**general?
- What is a **class?**
- What is an **object?**
- What is an **instance?**
- What is a **property?**
- What is a **method?**
- What is the difference between a **function** and a **method?**
- What is meant by **builder**?
- What is the difference between a **class** an **object** and an **instance?**
- What do we mean by the concept of **encapsulation**?
- What do we mean by the concept of **abstraction?**
- What do we mean by the concept of **inheritance?**
- What do we mean by the concept of **polymorphism?**
- What do we mean by the concept of **Overload**?
- What do we mean by the concept of **Override**?
- What are the differences between the concept of **Overload** and **Override?**
- What is a **static class?**
- Look for 3 advantages over object-oriented programming over other programming paradigms
- Look for disadvantages of this paradigm.

Arguments and examples.

Concepts for understanding the theoric part of object-oriented programming

- Explains when you consider the use of **object-oriented programming** as necessary against other programming paradigms
- Explains the differences between a **class** an **object** and an **instance.**
- Explain in which cases you consider it appropriate to make use of **encapsulation.**
- Explain in which cases you consider it appropriate to make use of **abstraction.**
- Explain in which cases you consider it appropriate to make use of **inheritance.**
- Explain in which cases you consider it appropriate to use a **static class.**
- Explain when you consider it appropriate to make use of **polymorphism.**

Implementationof the theorist

Of the concepts we need to clarify

General terms object-oriented programming

It is a programming paradigm, dividing everything into classes, objects, entities, allows us to realize

an structured and organized program, comes to innovate the way we get results. Objects manipulate input data for obtaining specific output data, where each object offers special functionality.

Classes

Classes are used to represent entities or concepts, such as nouns in the language. Each class is a model that defines a set of variables - the state, and appropriate methods for operating on that data - the behavior. Each object created from the class is called an instance of the class.

Objects

Un **object** is an object-oriented computer program thing consisting of a state and a behavior, which in turn consist of stored data and tasks that can be performed during runtime.

Instance

It is an object created from a class

Method

It is a set of statements associated with a class, which perform a certain task and can be invoked by a name

Property

They are like 'variables' that can take a single or multiple value and can be of a different type or

Method

It is a set of statements associated with a class, which perform a certain task and can be invoked by a name

Difference between methods and functions

Functions are defined outside the classes. Methods are defined within and are part of the classes.

What is meant by **builder**?

a **constructor** is a subroutine whose mission is to initialize an object of a class. The constructor assigns the initial values of the new object.

What is the difference between a **class** an **object** and an **instance?**

The set of messages that an **object** can respond to is called the **object** protocol. **Instance**: Any object that derives from some other **object** is called. In this way, all **objects** are **instances** of some other, except the Object class that is the mother of all.

What do we mean by the concept of **encapsulation**?

It is the process of storing data as methods, in which you can manipulate or change this data. It consists of separating the external aspects of an object (can be accessed from other objects)

What do we mean by the concept of **Abstraction?**

It is a class that contains at least one abstract method. Abstract method is the one that is declared but not implementedIt is possible to create objects of abstract classes, its goal is to define formatting of methods for the subclasses that should over-establish them. It is oriented to the

Heritage Class Estatica

What do we mean by the concept of inheritance?

Inheritance makes it easy to create objects from existing ones and implies that a child subclass gets all the behavior of the methods of another parent class. You can inherit from another parent class

What do we mean by the concept of **polymorphism?**

It allows the same method name to represent different code, as a result it can express many different behaviors

What do we mean by the concept of **Overload**?

Php overload provides the means to dynamically create properties and methods. These dynamic entities are processed by magic methods that can be set in a class for various actions.

What do we mean by the concept of **Override**?

It is a feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its parent classes or superclasses.

What are the differences between the concept of **Overload** and **Override?**

Overload is the overload of methods, is that in the same class you can have two methods that are called in the same way but that are diferiencen through the amount of parametersOverride is method overwrite is used when inheriting the methods of a class and an m all redefines in the daughter class this matters more when you combine it with polymorphism

What is a **static class?**

It allows the same method name to render different code, as a result it can express many different behaviors.

Advantages of using OOP

The first advantage of the concept of objects is that all the code that has something to do with spacecraft is in one place.

Another advantage is that objects can have inherent attributes of the class to which they belong, for example, spacecraft and asteroids could both have an XY position because all objects belonging to the class of moving objects have a XY position.

Another advantage is that POO makes large programs more manageable. If all windows belong to a hierarchy of window classes and all code that refers to a particular window is within that window, all window manipulations can be written as a simple message transfer.

Disadvantages

If you want to read some data, make it simple and write again, you don't need to define classes and objects

Developing a presentation

We'll explain how the pill has been organized

- How pill tasks have been organized
- Knowledge learned
- What difficulties have arisen during the pill
- What conclusions do you draw from OOP?
- In what cases you recommend the use of the OOP paradigm
- Define the following concepts:
    - Class, object and instance
    - Encapsulation
    - Abstraction
    - Inheritance
    - Static class
    - Polymorphism
    - Overload
    - Override

Organize the code

In this small project we have focused on testing **PHP,**so we have not taken into account the organization of our code. It is very important that you organize properly. Create the following directories:

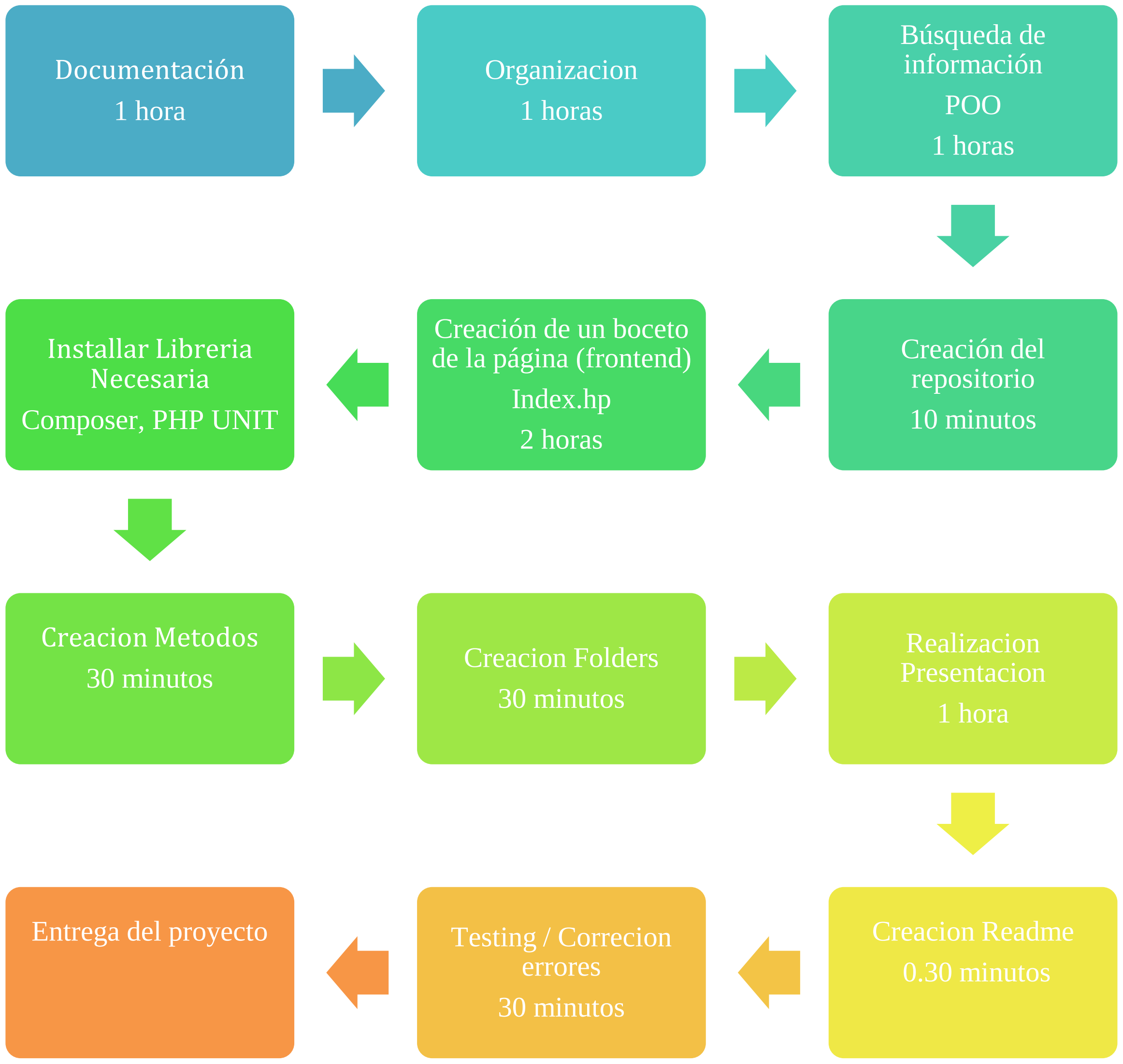
- app (will be responsible for containing the source code of your app)
- test (will be responsible for containing the tests)
    - test/app (this directory is created to maintain the same structure as the original app to facilitate the location of the tests of each of the files)

Task planning

LISTA DE TAREAS A REALIZAR

Task	Priority	Hours	Difficulty	ID
Documentation	High	1,00	High	1
Organization	High	1,00	High	2
Pre-search for information	Normal	1,00	Normal	3
Repository creation	Low	0,15	Low	4
Index php structure	Low	0.30	Normal	5
Folders creation	Normal	0.30	Normal	6
Creation of Poo Methods	High	2.00	High	7
Presentation making	High	1.00	High	8
CREATION README	Low	0,30	Low	9
Testing / Correction Errors	High	0,30	Normal	10
Project delivery	High	0.20	High	11

Project Calendar Tracking

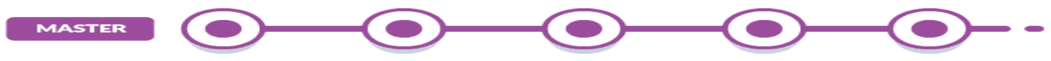

Documentación
1 hora
Organizacion
1 horas
Búsqueda de información
POO
1 horas
Creación del repositorio
10 minutos
Creación de un boceto de la página (frontend)
Index.hp
2 horas
Installar Libreria Necesaria
Composer, PHP UNIT
Creacion Metodos
30 minutos
Creacion Folders
30 minutos
Realizacion Presentacion
1 hora
Creacion Readme
0.30 minutos
Testing / Correcion errores
30 minutos
Entrega del proyecto

GIT WORKFLOW documentation

- Creating Git Hub **https://github.com/robertfox11/PooPills.git**
- We make commits of the structure of the main page.
- Chance of it occurring 80%
- Project impact 60%
- Possible alternative (mitigation) Ask colleagues for help
- Chance of it occurring 30%
- Project impact 60%
- Possible alternative (mitigation) Ask colleagues for help
- Not easily finding information related to the project
- Chance of it occurring 30%
- Project impact 60%
- Alternative alternative (mitigation)
- Ask colleagues for help

From the realization of the structure, work continued only on the "master" branch, through the Workflow "Gitflow".

But information --> https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow



Tools

Different tools were used in the development of the project. They are as follows:

- ***git: A powerful version control system*** that helps track changes in the work tree.
- ***Visual Studio Code: A code editor*** optimized for creating and debugging modern web applications.
- ***WampServer,*** *comes integrated Apache Web server, openSSL for SSL support, MySQL database and PHP language*
- ***Google Chrome Developer Tools:*** Used to debug JavaScript code and to test design settings.
- ***Google Docs:*** Used to write project documentation.
- ***W3C Validator***– Used to validate HTML and CSS code.
- ***ESLint***– Used to validate JavaScript code.
- ***nano: A basic text editor that uses the*** command-line interface.
- ***curl: A*** command-line tool used to transfer data using various network protocols.
- ***Google Docs:*** Used to write project documentation.

Phase III - Project execution

Concepts

Which is a class

Incidents

None, luckily!

Lessons

All tasks were completed without having to face any major obstacles.

Phase IV - Project closure

General comments

The pill was successfully completed in the time interval that was predicted in task *planning.*