****

UNAN-LEÓN

**Ciencias y Tecnologías.**

**Ing. En Sistemas de Información.**

**Componente: Patrones de Diseño**

**Elaborado por:**

* **Derian Ricardo Córdoba Pérez**
* **Brandon Steven Santeliz Vargas**
* **Jonathan Enrique Mendoza Herrera.**
* **Bryan Jose Luna Escoto.**

**Fecha: 20/09/2021**

**A la libertad por la universidad**

**Introducción**

Los patrones de diseño (design patterns) son soluciones habituales a problemas comunes en el diseño de software. Cada patrón es como un plano que se puede personalizar para resolver un problema de diseño particular de tu código. Son unas técnicas para resolver problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

La programación es una actividad muy valorada y con gran demanda en el mercado, ya que la transformación digital es una realidad, especialmente para páginas web, blogs y otros canales web. Sin embargo, aquellas personas que no tienen tanto conocimiento sobre lenguajes de programación generalmente no saben qué significa PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML, es un lenguaje de programación de uso general, favoreciendo la conexión entre los servidores y la interfaz de usuario. Entre los factores que hicieron que PHP se volviera tan popular, se destaca el hecho de que es de código abierto esto significa que cualquiera puede hacer cambios en su estructura. En esta práctica utilizamos PHP8 con Laravel el cual tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades. Su filosofía es desarrollar código PHP de forma elegante y simple.

**Metodología**

**¿Qué son los patrones de diseño?**

Un patrón de diseño es básicamente una forma reutilizable de resolver un problema común en el desarrollo de software. Si la forma de solucionar este problema se puede extraer, explicar y reutilizar en múltiples ámbitos y en muchas ocasiones, en ese caso nos hemos encontrado con un patrón de diseño.

Saber identificar qué patrón de diseño encaja con el problema al que nos estamos enfrentando en un momento es el punto más complicado. Esto solo lo vas a conseguir practicando mucho y a base de realizar pruebas de ensayo-error, hasta que llega un momento en el que aprendes que hay patrones que encajan con ciertos tipos de problemas.

los patrones pueden clasificarse por su propósito. Este libro cubre tres grupos generales de patrones:

* Los patrones creacionales proporcionan mecanismos de creación de objetos que incrementan la flexibilidad y la reutilización de código existente.
* Los patrones estructurales explican cómo ensamblar objetos y clases en estructuras más grandes a la vez que se mantiene la flexibilidad y eficiencia de la estructura.
* Los patrones de comportamiento se encargan de una comunicación efectiva y la asignación de responsabilidades entre objetos.

Existen muchísimos patrones de diseño, y además van a seguir apareciendo, por lo que es imposible conocerlos todos. Lo que sí es interesante es tener un catálogo con los patrones más conocidos y los más frecuentes, los que se suelen llamar patrones clásicos, en esta práctica utilizaremos solamente algunos patrones muy comunes, el cual lo mencionaremos a continuación.

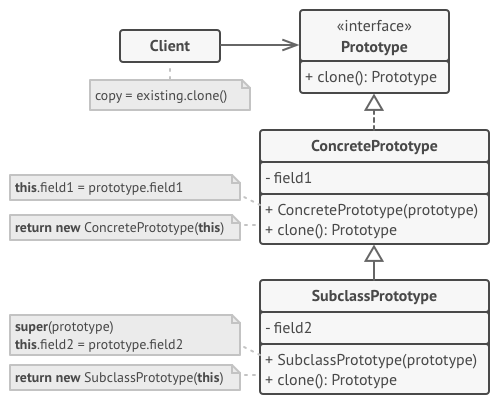
1. **Prototype**

Es un patrón de diseño creacional que nos permite copiar objetos existentes sin que el código dependa de sus clases, el patrón Prototype delega el proceso de clonación a los propios objetos que están siendo clonados. El patrón declara una interfaz común para todos los objetos que soportan la clonación. Esta interfaz nos permite clonar un objeto sin acoplar el código a la clase de ese objeto. Normalmente, dicha interfaz contiene un único método clonar.

La implementación del método clonar es muy parecida en todas las clases. El método crea un objeto a partir de la clase actual y lleva todos los valores de campo del viejo objeto, al nuevo. Se puede incluso copiar campos privados, porque la mayoría de los lenguajes de programación permite a los objetos acceder a campos privados de otros objetos que pertenecen a la misma clase.

Un objeto que soporta la clonación se denomina prototipo. Cuando tus objetos tienen decenas de campos y miles de configuraciones posibles, la clonación puede servir como alternativa a la creación de subclases.

**Estructura**



1. **MVC Modelo-Vista-Controlador**

**Modelo de la aplicación**

El modelo de la aplicación es un conjunto de clases que se relacionan con el modelo del dominio, que tienen conocimiento de las vistas y que implementan los mecanismos necesarios para notificar a éstas últimas sobre los cambios que se pudieren dar en el modelo del dominio. El modelo de la aplicación es llamado también coordinador de la aplicación.

**Las vistas de la aplicación**

Las vistas son el conjunto de clases que se encargan de mostrar al usuario la información contenida en el modelo. Una vista está asociada a un modelo, pudiendo existir varias vistas asociadas al mismo modelo; así, por ejemplo, se puede tener una vista mostrando la hora del sistema como un reloj analógico y otra vista mostrando la misma información como un reloj digital. Una vista obtiene del modelo solamente la información que necesita para desplegar y se actualiza cada vez que el modelo del dominio cambia por medio de notificaciones generadas por el modelo de la aplicación.

**El controlador de la aplicación**

El controlador es un objeto que se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él. A partir de estos mensajes, el controlador se encarga de modificar el modelo o de abrir y cerrar vistas. El controlador tiene acceso al modelo y a las vistas, pero las vistas y el modelo no conocen de la existencia del controlador.

**Estructura**

Diagram

Description automatically generated

1. **Delegación**

En Ingeniería de software, el patrón de diseño de delegación es una técnica en la que un objeto de cara al exterior expresa cierto comportamiento, pero en realidad delega la responsabilidad de implementar dicho comportamiento a un objeto asociado en una relación inversa de responsabilidad. Consiste en dar una responsabilidad a otro objeto para que complemente un trabajo, como vemos este patrón es muy simple y lo más normal es que muchos de nosotros lo hayamos utilizado sin saber, básicamente podemos extender nuestro sistema utilizando funcionalidades de otras clases, sin necesidad de utilizar la herencia

**Estructura**

Diagram

Description automatically generated

1. **Abstract Factory**

Es un patrón de diseño creacional que nos permite producir familias de objetos relacionados sin especificar sus clases concretas. Este patrón se puede aplicar cuando:

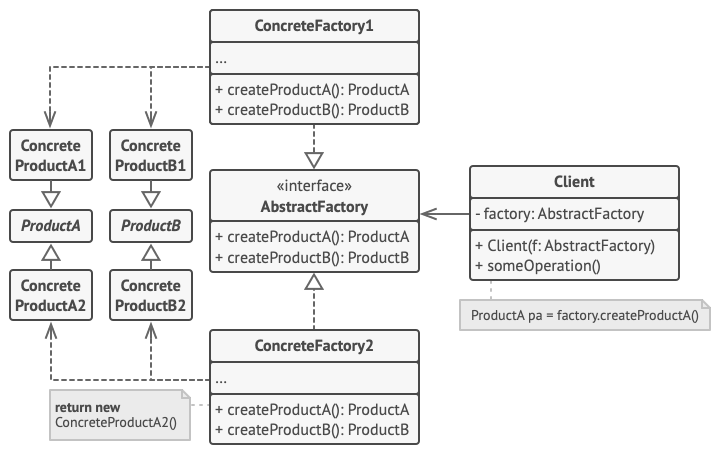
* Un sistema debe ser independiente de cómo sus objetos son creados.
* Un sistema debe ser ‘configurado’ con una cierta familia de productos.
* Se necesita reforzar la noción de dependencia mutua entre ciertos objetos.

Este patrón busca agrupar un conjunto de clases que tiene un funcionamiento en común llamadas familias, las cuales son creadas mediante un Factory, este patrón es especialmente útil cuando requerimos tener ciertas familias de clases para resolver un problema, sin embargo, puede que se requieran crear implementaciones paralelas de estas clases para resolver el mismo problema, pero con una implementación distinta.

La estructura de Abstract Factory puede resultar muy enredosa ya que tiene muchos componentes y éstos aparentemente se mezclan entre sí.

**Estructura**





**PHP8:**

Es un lenguaje de scripting de uso general popular que es especialmente adecuado para el desarrollo web. Rápido, flexible y pragmático, PHP impulsa todo, desde su blog hasta los sitios web más populares del mundo.

PHP es el lenguaje de programación para el desarrollo del backend más popular. También uno de los lenguajes más usados, cómo se refleja en todos los índices globales.

Con PHP puedes desarrollar sitios y aplicaciones de todo tipo. PHP es el motor además de los CMS más populares, como WordPress, Joomla, Drupal o Magento, entre muchos otros. En PHP también encontramos frameworks potentes y muy usados como Laravel o Symfony.

Una de las ventajas de PHP es su facilidad de uso. Cualquier persona con muy pocos conocimientos puede comenzar a usar PHP para el desarrollo de páginas web. Sin embargo, esa misma ventaja también puede ser un inconveniente, pues si no se dispone de la debida formación es fácil caer en malas prácticas. Otra gran ventaja es su disponibilidad, ya que lo encontramos en la mayoría de los alojamientos web de todo tipo.

PHP es el lenguaje más popular para el desarrollo backend, es decir, es la tecnología más usada en lo que respecta a lenguajes del lado del servidor. El motivo de este éxito no es solo la facilidad con la que PHP permite desarrollar páginas, sino también formar parte del stack de tecnologías más habitual en Internet: LAMP (Linux, Apache, MySQL y PHP). La mayoría de los planes de hosting soportan PHP y la mayoría de los proyectos de gestores de contenido populares están construídos con PHP.

PHP 8.0 es la última versión principal del lenguaje PHP. Introduce varios cambios sustanciales, mejoras de rendimiento y muchas características nuevas, como argumentos con nombre, compilador JIT, tipos de unión, expresiones de coincidencia y más.

**Instalación de PHP8 Linux**

Habilita el repositorio de PHP

Ondřej Surý, un desarrollador de Debian mantiene un repositorio que incluye múltiples versiones de PHP. Para agregar el repositorio, ejecute el siguiente comando:

* $ sudo apt install software-properties-common
* $ sudo add-apt-repository ppa:ondrej/php

Una vez que el PPA está habilitado, puede instalar PHP 8.

Instalar PHP 8 con Apache. Si usa Apache como su servidor web, puede ejecutar PHP como un módulo de Apache o PHP-FPM.Instalar PHP como un módulo de Apache es simple:

* $ sudo apt update
* $ sudo apt install php8.0-fpm libapache2-mod-fcgid

Una vez que los paquetes estén instalados, reinicie Apache para cargar el módulo PHP:

* $ sudo systemctl restart apache2

Configurar Apache con PHP-FPM

Php-FPM es un administrador de procesos FastCGI para PHP. Ejecute el siguiente comando para instalar los paquetes necesarios:

* $ sudo apt update
* $ sudo apt install php8.0-fpm libapache2-mod-fcgid

De forma predeterminada, PHP-FPM no está habilitado en Apache. Para habilitarlo, ejecute:

* $ sudo a2enmod proxy\_fcgi setenvif
* $ sudo a2enconf php8.0-fpm

Para activar los cambios, reinicie Apache:

* $ sudo systemctl restart apache2

¿Qué es Laravel?

Laravel es un popular framework de PHP. Permite el desarrollo de aplicaciones web totalmente personalizadas de elevada calidad, laravel es un framework PHP. Es uno de los frameworks más utilizados y de mayor comunidad en el mundo de Internet. Como framework resulta bastante moderno y ofrece muchas utilidades potentes a los desarrolladores, que permiten agilizar el desarrollo de las aplicaciones web.

Laravel pone énfasis en la calidad del código, la facilidad de mantenimiento y escalabilidad, lo que permite realizar proyectos desde pequeños a grandes o muy grandes. Además, permite y facilita el trabajo en equipo y promueve las mejores prácticas. El framework Laravel trabaja con una arquitectura de carpetas avanzada, de modo que promueve la separación de los archivos con un orden correcto y definido, que guiará a todos los integrantes del equipo de trabajo y será un estándar a lo largo de los distintos proyectos. Por supuesto, dispone también de una arquitectura de clases también muy adecuada, que promueve la separación del código por responsabilidades. Su estilo arquitectónico es MVC, contiene además un amplio conjunto de características, que sirven para realizar la mayoría de las aplicaciones web. Entre ellas podemos encontrar:

* Un sistema de rutas, mediante las cuales es fácil crear y mantener todo tipo de URLs amistosas a usuarios y buscadores, rutas de API, etc.
* Un sistema de abstracción de base de datos, con un ORM potente pero sencillo de manejar, mediante el que podemos tratar los datos de la base de datos como si fueran simples objetos.
* Un sistema para creación de colas de trabajo, de modo que es posible enviar tareas para ejecución en background y aumentar el rendimiento de las aplicaciones.
* Varias configuraciones para envío de email, con proveedores diversos
* Un sistema de notificaciones a usuarios, mediante email, base de datos y otros canales
* Una abstracción del sistema de archivos, mediante el cual podemos escribir datos en proveedores cloud, y por supuesto en el disco del servidor, con el mismo código.
* Gestión de sesiones
* Sistema de autenticación, con todo lo necesario como recordatorios de clave, confirmación de cuentas, recordar un usuario logueado, etc.
* La posibilidad de acceder a datos en realtime y recibir notificaciones cuando éstos se alteran en la base de datos

**Instalación de Laravel en Linux**

Actualice los paquetes del sistema a las últimas versiones:

* $ sudo apt update && sudo apt upgrade

Instalar PHP

PHP 7.4, que es la versión predeterminada de PHP en Ubuntu 20.04 LTS, es totalmente compatible con Laravel.

Ejecute el siguiente comando para instalar PHP y todos los módulos PHP necesarios:

* $ sudo apt install php7.4-common php7.4-cli php7.4-gd php7.4-mysql php7.4-curl php7.4-intl php7.4-mbstring php7.4-bcmath php7.4-imap php7.4-xml php7.4-zip

Instalar Composer

Composer es un administrador de dependencias para PHP y lo usaremos para descargar el núcleo de Laravel e instalar todos los componentes de Laravel necesarios.

Para instalar composer globalmente, descargue el instalador de Composer con curl y mueva el archivo al /usr/local/bin:

* $ curl -sS https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin --filename=composer

Verifique la instalación imprimiendo la versión del compositor:

* $ composer --version

**Conclusion**

De acuerdo con las encuestas de NetCraft, PHP es ahora el módulo más popular para el servidor Apache, creciendo un 4% mensual sobre la totalidad de sitios de Internet. Las cuatro grandes características: **Velocidad:** No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, está listo para ser utilizado. **Estabilidad:** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para lo bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable. **Seguridad:** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini. **Simplicidad:** Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

Como muchas otras cosas en la caja de herramientas de un programador, PHP ha sido creado casi accidentalmente. Empezando como experimento de una persona, modificado para adaptarse a las necesidades de muchos otros programadores de un lenguaje rápido, simple, robusto que pueda trabajar en conjunto a otros componentes.

La potencia de Laravel radica en su integración, escalabilidad y facilidad de mantenimiento respecto a otros desarrollos en lenguajes 100% nativos y por lo tanto es una opción más que a tener en cuenta a la hora de decidir usar este framework en nuestros desarrollos. Laravel es un framework de PHP para ayudarnos en un tipo de desarrollo sobre aplicaciones escritas en este lenguaje de programación. Esté framework o más bien podría llamarlo compañero de ahora en adelante, nos ayuda en muchas cosas al desarrollar una aplicación, por medio de sus sistema de paquetes y de ser un framework del tipo MVC (Modelo-Vista-Controlador) da como resultado que podamos “despreocuparnos” (por así decirlo) en ciertas aspecto del desarrollo, cómo instanciar clases y métodos para usarlos en muchas partes de nuestra aplicación sin la necesidad de escribirlo y repetirlos muchas veces con lo que eso conlleva a la hora de modificar algo en el código. Funciona como muchos otros ya, desde la línea de comandos con el famoso Artisan que es el nombre que le dan a esta interfaz por comandos para ejecutar muchas funcionalidades como ver todas las rutas de la aplicación disponible, o poner a correr la aplicación o pararla.

**Bibliografía**

Normas APA.

<https://normasapa.co/como-aplicar-normas-apa-a-tu-monografia/>

Open Webinars

<https://openwebinars.net/blog/que-son-los-patrones-de-diseno/>

GURU

<https://refactoring.guru/es/design-patterns/prototype>

Scielo

<http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1683-07892004000100005>

Reactive Programming

<https://reactiveprogramming.io/blog/es/patrones-de-diseno/abstract-factory>

Noviello

<https://noviello.it/es/como-instalar-laravel-en-ubuntu-20-04-lts/>

<https://noviello.it/es/como-instalar-php-8-en-ubuntu-20-04-lts/>