**¿Qué namespace?**

Son una forma de encapsular elementos como clases, interfaces, funciones y constantes. Son particularmente útiles en proyectos grandes y complejos, ya que permiten organizar mejor el código y evitar conflictos de nombres entre clases y funciones.

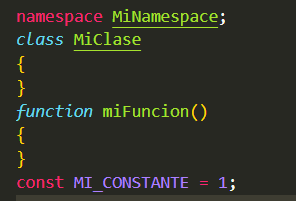


Ilustración 1. Declaración de Namespace

Para declarar un namespace se usa la palabra clave “**namespace”** seguido del nombre. Todos los elementos definidos después de esta declaración pertenecerán al namespace especificado.

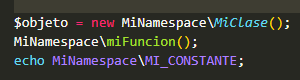


Ilustración 2. Uso de Namespace

Puedes acceder a las clases, funciones y constantes dentro de un **namespace** utilizando el nombre completo del namespace.

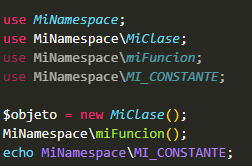


Ilustración 3.Importación de Namespace.

Para importar namespaces o elementos específicos dentro de un namespace usando la palabra clave “**use**”.

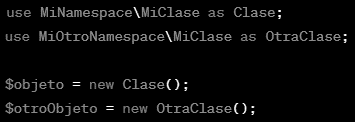
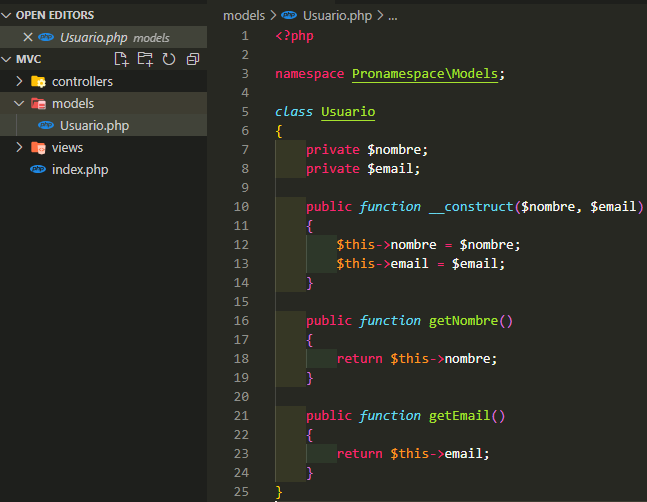
Crear alias para namespaces o elementos específicos para acortar el nombre o evitar conflictos de nombres.

Ilustración 4. Alias de Namespace

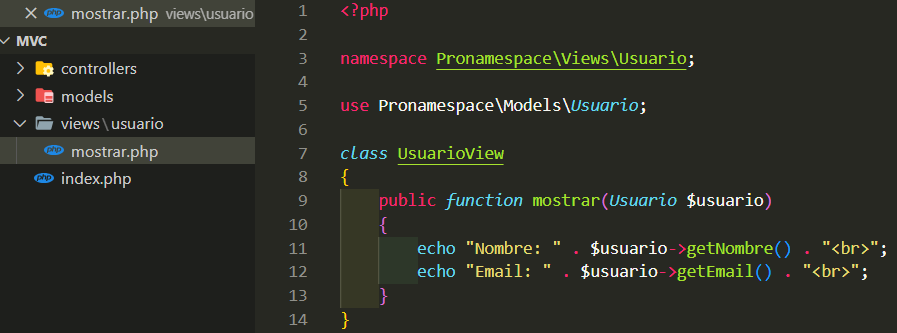
Para la mayor comprensión del Namespace.

1. Realice el siguiente proyecto ,y anexe el link del repositorio.

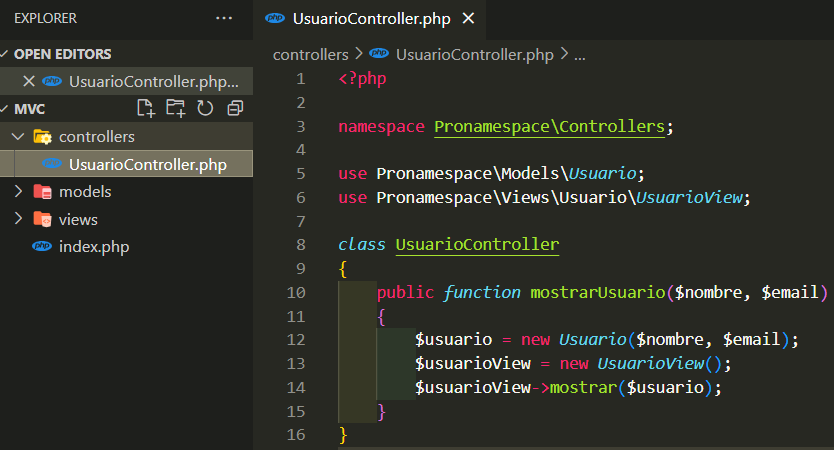
|  |
| --- |
|  |



Modelo: Usuario.php



Vista: mostrar.php

0

Controlador: UsuarioController.php



El modelo **Usuario** contiene la lógica relacionada con los datos del usuario.

La vista **UsuarioView** es responsable de mostrar los datos del usuario.

El controlador **UsuarioController** actúa como intermediario entre el modelo y la vista.

En **index.php**, se crea una instancia del controlador, y se llama al método **mostrarUsuario** para mostrar la información de un usuario.  
  
Los namespaces en PHP están estandarizados bajo una serie de recomendaciones conocidas como **PSR (PHP Standard Recommendations)**, específicamente en la PSR-4, que trata sobre la autocarga de clases a partir de rutas de archivos.

**PSR-4: Autocarga de Carga Automática de Clases**

La PSR-4 proporciona una especificación para la autocarga de clases a partir de rutas de archivos. Al seguir esta especificación, puedes estructurar tus directorios y archivos de manera que un cargador automático (como el de Composer) pueda cargar automáticamente tus clases sin necesidad de require o include. Puntos clave de la PSR-4

**Namespace completo:** La cadena completa después de la palabra clave namespace debe tener un prefijo de espacio de nombres superior ("namespace base") seguido de cero o más subespacios de nombres.

**Rutas de archivo:** La estructura de directorios y los nombres de archivo deben corresponder al espacio de nombres completo y al nombre de la clase.

**Cargador automático:** Debe ser capaz de cargar clases utilizando un único camino de inclusión.

**Transformación:** Los nombres de espacio de nombres y clase deben transformarse a rutas de archivos reemplazando el separador de espacio de nombres (\) por el separador de directorio (/), y los caracteres subrayados (\_) por el separador de directorio.

1. Teniendo en cuenta lo visto en formación desarrolle las siguientes preguntas.
   1. ¿Qué es un namespace en PHP?

|  |
| --- |
| Un namespace no es más que un directorio para clases, traits, interfaces, funciones y constantes. |

* 1. ¿Para qué se utilizan principalmente los namespaces?

|  |
| --- |
| Los namespaces en PHP se utilizan principalmente para resolver problemas de colisión de nombres y para organizar el código en proyectos grandes. |

* 1. ¿Es posible asignar un alias a un namespace o clase importados?

|  |
| --- |
| Sí, en PHP es posible asignar alias a los namespaces o clases que importas mediante la palabra clave use. Esto se puede hacer para simplificar la referencia a esos elementos en tu código, especialmente cuando trabajas con namespaces largos o cuando hay posibilidad de conflictos de nombres. |

* 1. ¿Cómo se declara un namespace?

|  |
| --- |
| La declaración de namespace generalmente se coloca al principio del archivo PHP, antes de cualquier otra declaración, se puede declarar un namespace utilizando la palabra clave “namespace” seguido del nombre de la carpeta. |

* 1. ¿Qué sucede si intentas usar una clase de un namespace sin importarlo?

|  |
| --- |
| Si intentas utilizar una clase de un namespace sin importarlo previamente, es posible que te encuentres con un error de tipo "Clase no encontrada" (Class not found). Esto se debe a que PHP no podrá resolver el nombre de la clase correctamente sin una referencia explícita al namespace. |

* 1. ¿Qué es la autocarga de clases en PHP y cómo se relaciona con los namespaces?

|  |
| --- |
| La autocarga de clases en PHP es un mecanismo que permite cargar automáticamente las clases cuando son necesarias, evitando así tener que incluir manualmente los archivos de clase en cada punto del código donde se utilicen. Este proceso de carga automática es especialmente útil en aplicaciones grandes y complejas, ya que facilita la organización del código y mejora el rendimiento. |

* 1. ¿Qué significa PSR-4 en relación con los namespaces?

|  |
| --- |
| PSR-4 (PHP Standards Recommendation 4) es una especificación de estándares de codificación propuesta por la organización PHP-FIG (PHP Framework Interop Group). PSR-4 describe un conjunto de reglas para la autoloading (carga automática) de clases en aplicaciones PHP, y está estrechamente relacionado con el uso de namespaces.  En el contexto de PSR-4, se establecen convenciones para organizar la estructura de directorios y archivos de tus clases de acuerdo con los namespaces. La idea es que puedas seguir una estructura consistente y predecible en tu proyecto, lo que facilita la implementación de la autoloading y la carga automática de clases. |

* 1. ¿Es posible acceder a una constante dentro de un namespace desde fuera de él?

|  |
| --- |
| Sí, es posible acceder a una constante dentro de un namespace desde fuera de él. Al igual que las clases, funciones y otras estructuras en PHP, las constantes definidas en un namespace son accesibles desde fuera mediante el uso del nombre completo del namespace y el operador de resolución de ámbito ::. |

1. Suba en Git y anexe los proyectos de la guía en un archivo .zip documentos requeridos.
2. Subir la evidencia al LMS