

Evidencia Programación Orientada a Objetos con PHP

Aprendiz:

Nicolas David Peña Gomez

Instructor:

Ing. Néstor Montaña

Tecnólogo Análisis y Desarrollo de Software

Ficha: 3064241

SENA Centro de Diseño y Metrología

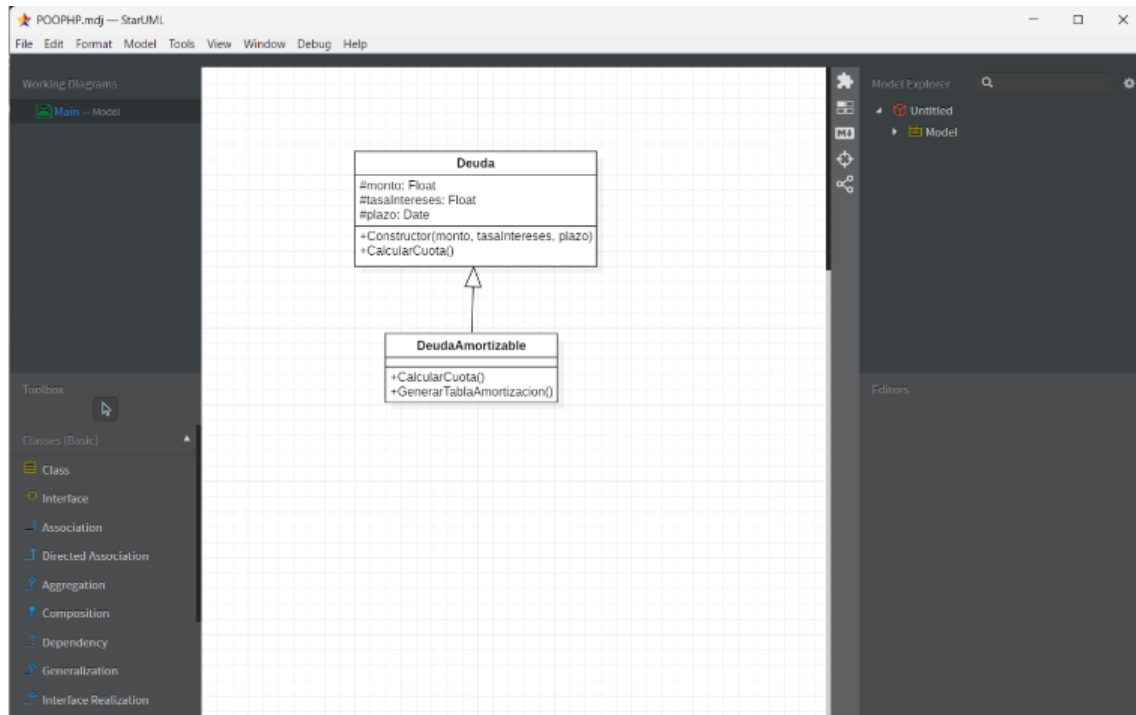
Descripción del Proyecto

Este proyecto implementa un simulador de tabla de amortización desarrollado en PHP

aplicando los principios de la Programación Orientada a Objetos (POO).

Se diseñó un diagrama UML con dos clases principales:

- **Deuda:** clase base que contiene los atributos comunes como monto, tasa de interés y plazo.
- **DeudaAmortizable:** clase que hereda de Deuda y genera la tabla de amortización mensual con la cuota, interés, amortización y saldo.



El programa permite al usuario ingresar los datos del préstamo a través de un formulario

web y genera automáticamente la tabla con todos los cálculos financieros.

Además, incluye

validaciones en el formulario y un diseño visual limpio y organizado.

Repositorio del proyecto

El código completo de mi simulador en PHP está disponible en mi repositorio oficial:

<https://github.com/NicolasP03/Clase-Nestor>

Dentro del repositorio se encuentra la carpeta **poophp**, donde está el archivo **index.php**, que contiene toda la lógica del sistema: clases, formulario y generación de la tabla de amortización.

Cómo ejecutarlo

1. **Instalar XAMPP** si aún no lo tienes.
2. Abrir el panel de XAMPP y **encender Apache**.
3. Copiar la carpeta del proyecto dentro de:
4. C:\xampp\htdocs\

Ejemplo:

C:\xampp\htdocs\poophp\

5. Asegúrate de que dentro esté el archivo:
6. index.php
7. Abrir el navegador y entrar a:
8. <http://localhost/poophp/>
9. Llenar los datos del formulario (monto, interés y plazo) y presionar **Calcular**.

El sistema mostrará la cuota mensual y toda la tabla de amortización generada con POO.

Conclusión

Este proyecto demuestra el uso de Programación Orientada a Objetos en PHP aplicando herencia y reutilización de código.

La estructura del proyecto está basada en un diagrama UML, lo que permitió organizar las clases y métodos de forma clara.

El resultado final es un simulador funcional y fácil de usar, capaz de generar la tabla de amortización de cualquier deuda ingresada por el usuario.