

**NOMBRE DE ASIGNATURA**

**SEMANA X**

Nombre del estudiante

Fecha de entrega

Carrera

# **DESARROLLO**

*Texto

Descripción generada automáticamente*

**

*b.) Herencia: Cuando creamos los objetos estos deben poder dejar propiedades descendentemente, aplicando esto en los datos de los alumnos, ya que deben tener una*

*conexión de forma directa en sus notas, creando igualdad de propiedades refiriéndome a ambos campos.*

*La herencia: Se aplica sobre las clases. O sea, de alguna forma las clases pueden tener descendencia, y ésta heredará algunas características de las clases “padres”. Si disponemos las clases con un formato de árbol genealógico, tenderemos lo que se denomina una estructura jerarquizada de clases*

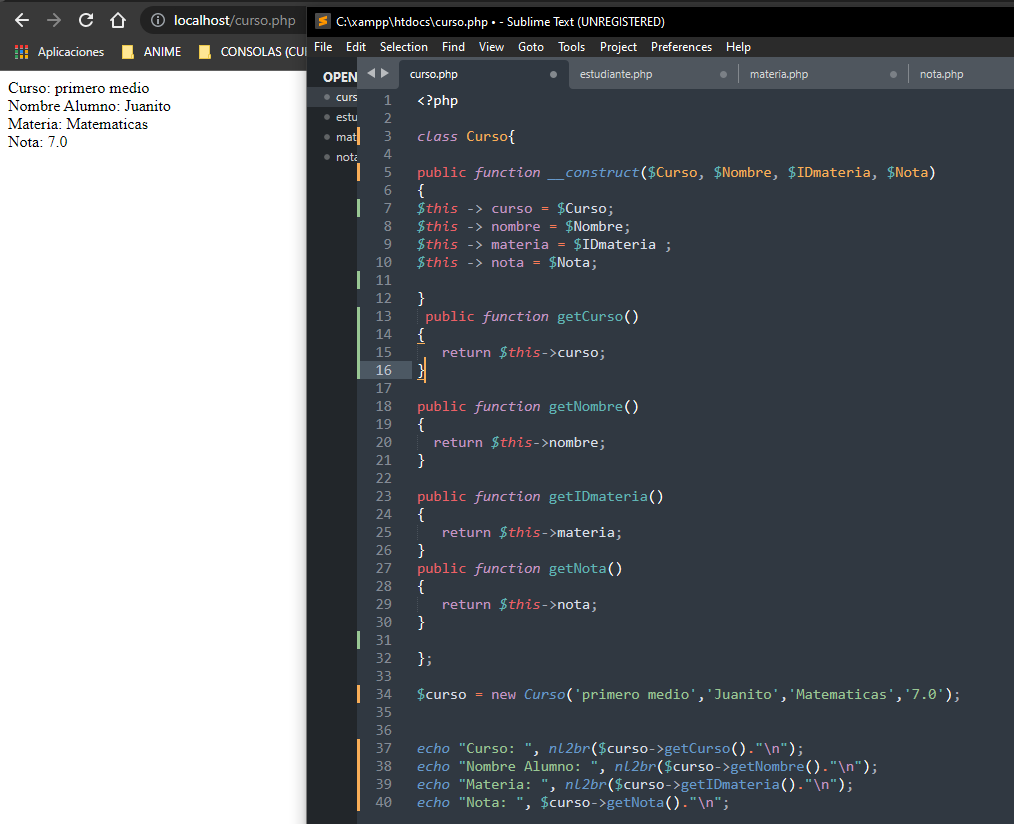
**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *clase* | *Curso* | *Estudiante* | *Materia* | *Nota* |
| *objetos* | *Nombre*  *Curso*  *IDmateria*  *Nota* | *Nombre*  *Curso* | *Nombre IDmateria* | *Nombre*  *IDmateria*  *Nota* |



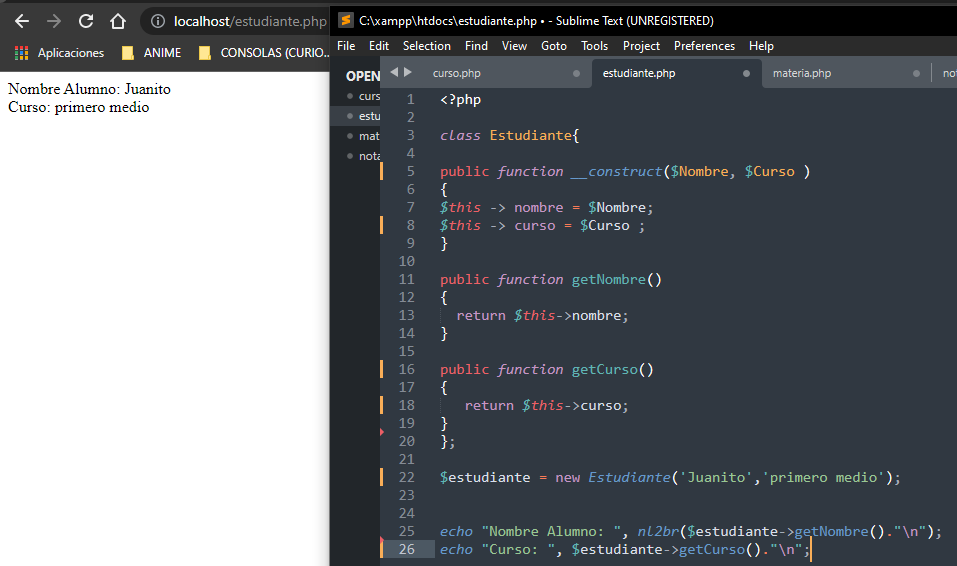
Clase curso

|  |
| --- |
| <?php  class Curso{  public function \_\_construct($Curso, $Nombre, $IDmateria, $Nota)  {  $this -> curso = $Curso;  $this -> nombre = $Nombre;  $this -> materia = $IDmateria;  $this -> nota = $Nota;  }  public function getCurso()  {  return $this->curso;  }  public function getNombre()  {  return $this->nombre;  }  public function getIDmateria()  {  return $this->materia;  }  public function getNota()  {  return $this->nota;  }  };  $curso = new Curso('primero medio','Juanito','Matematicas','7.0');  echo "Curso: ", nl2br($curso->getCurso()."\n");  echo "Nombre Alumno: ", nl2br($curso->getNombre()."\n");  echo "Materia: ", nl2br($curso->getIDmateria()."\n");  echo "Nota: ", $curso->getNota()."\n"; |



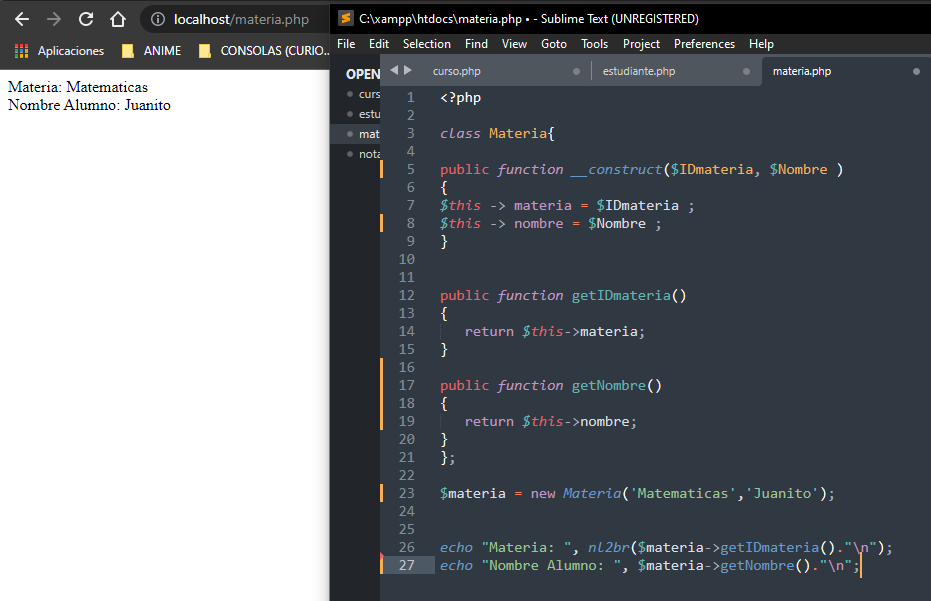
Clase Estudiante

|  |
| --- |
| <?php  class Estudiante{  public function \_\_construct($Nombre, $Curso )  {  $this -> nombre = $Nombre;  $this -> curso = $Curso ;  }  public function getNombre()  {  return $this->nombre;  }  public function getCurso()  {  return $this->curso;  }  };  $estudiante = new Estudiante('Juanito','primero medio');  echo "Nombre Alumno: ", nl2br($estudiante->getNombre()."\n");  echo "Curso: ", $estudiante->getCurso()."\n"; |



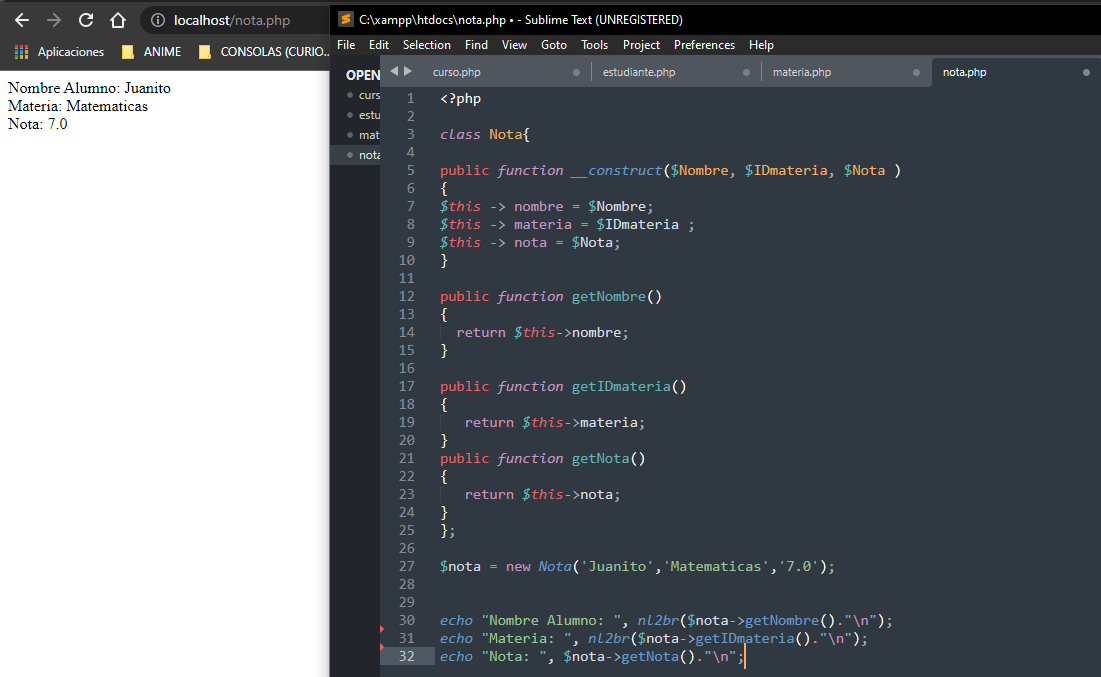
Clase Materia

|  |
| --- |
| <?php  class Materia{  public function \_\_construct($IDmateria, $Nombre )  {  $this -> materia = $IDmateria ;  $this -> nombre = $Nombre ;  }  public function getIDmateria()  {  return $this->materia;  }  public function getNombre()  {  return $this->nombre;  }  };  $materia = new Materia('Matematicas','Juanito');  echo "Materia: ", nl2br($materia->getIDmateria()."\n");  echo "Nombre Alumno: ", $materia->getNombre()."\n"; |



Clase Nota

|  |
| --- |
| <?php  class Nota{  public function \_\_construct($Nombre, $IDmateria, $Nota )  {  $this -> nombre = $Nombre;  $this -> materia = $IDmateria ;  $this -> nota = $Nota;  }  public function getNombre()  {  return $this->nombre;  }  public function getIDmateria()  {  return $this->materia;  }  public function getNota()  {  return $this->nota;  }  };  $nota = new Nota('Juanito','Matematicas','7.0');  echo "Nombre Alumno: ", nl2br($nota->getNombre()."\n");  echo "Materia: ", nl2br($nota->getIDmateria()."\n"); |



# **REFERENCIAS BIBLIOGRÁFICAS**

*Soni, S. (2018, 4 diciembre). PHP orientado a objetos con clases y objetos. Code Envato Tuts+. https://code.tutsplus.com/es/tutorials/basics-of-object-oriented-programming-in-php--cms-31910*