

1. A partir de l'exemple MVC-01 fer un nou model: Nota, amb les propietat privades nom, titol.

```

notaController.php M  autoload.php U  mostrarNotes.php U  nota.php M X
UF2 > 2PHP > MVC-01 > models > nota.php
1  <?php
2  require "modelBase.php";
3  // ---- 1 ---- modul
4  class Nota extends ModelBase{ // ---- 4 herencia ----
5      private $nom;
6      private $titol;
7

```

2. Fes el mètodes corresponents setters i getters.

```

8      // ---- 2 ---- get i set
9      /**
10     * Get the value of nom
11     */
12     public function getNom()
13     {
14         return $this->nom;
15     }
16     /* Set the value of nom
17     * @return self
18     */
19     public function setNom($nom)
20     {
21         $this->nom = $nom;
22         return $this;
23     }

```

3. Fes una classe modelBase a la carpeta models, nom modelBase.php que té el mètode aconseguirTots().

```

notaController.php M  autoload.php U  mostrarNotes.php U  nota.php M  modelBase.php >
UF2 > 2PHP > MVC-01 > models > modelBase.php
1  <?php
2  // ---- 3 ----
3  class ModelBase{
4      public function aconseguirTots(){
5          return "nota";
6      }
7  }
8  ?>

```

4. has de modificar les classes nota i usuari per tal que heredin de modelBase el mètode aconseguirTots. Recorda incloure la classe amb un require.

```

notaController.php M  autoload.php U  mostrarNotes.php U  nota.php M X
UF2 > 2PHP > MVC-01 > models > nota.php
1  <?php
2  require "modelBase.php";
3  // ---- 1 ---- modul
4  class Nota extends ModelBase{ // ---- 4 herencia ----
5      private $nom;
6      private $titol;
7

```

5. Fes un controlador per a nota (notaController), amb el mètode mostrarTots().

```

// ---- 5 ---- crear controller nota
class NotaController {
    // El controlador encapsula accions
    // una acció de crear usuari, mostrar usuari, ...

    //fixem-nos el nom del mètode
    // carreguem el model
    // fem un usuari nou
    // a una variable li assignem el resultat del
    // mètode aconseguir tots
    // carreguem a la vista

    public function mostrarTots(){
        require_once 'E:\Programas info\xampp\htdocs\PHP\UF2\2PHP\MVC-01\models\nota.php';

        $nota = new Nota();
        $totesLesNotes=$nota->aconseguirTots();

        require_once 'E:\Programas info\xampp\htdocs\PHP\UF2\2PHP\MVC-01\views\notes\mostrarTots.php';
    }
}

```

6. Fes la vista corresponent.

```

notaController.php M  autoload.php U  mostrarNotes.php U  nota.php M  mostrarTots.php X
UF2 > 2PHP > MVC-01 > views > notes > mostrarTots.php
1  <!-- La vista és específica per cada acció -->
2  <!-- ---- 6 vista de nota ---- -->
3  <h1><?=$totesLesNotes?></h1>
4

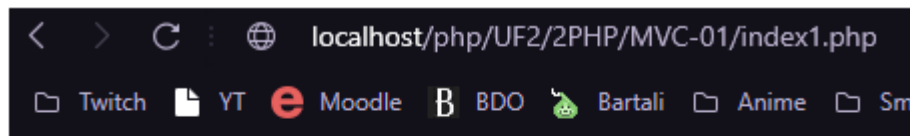
```

7. Comprova que funciona passant paràmetres per get. Tal i com hem vist en l'exemple.

```

UF2 > 2PHP > MVC-01 > index1.php
1  <?php
2  echo "<h1>PHP MVC </h1>";
3
4  // require_once 'controllers/usuariController.php';
5  require_once 'controllers/notaController.php';
6
7  $controlador = new NotaController();
8  $controlador->mostrarTots();

```



PHP MVC

nota

8. Fes un nou mètode a notaController.php anomenat crear. En aquest mètode hauràs de crear un objecte nota, assignar-li valors a través dels setters i getters ("Nota1" i "Una nota qualsevol"), fer la vista corresponent (mostra el nom de la nota en h2, i el títol i la descripció en h3, cridar la vista des del controlador i comprova que funciona.

```
UF2 > 2PHP > MVC-01 > controllers > notaController.php
1  <?php
2  // ---- 5 ---- crear controller nota
3  class NotaController {
4      // El controlador encapsula accions
5      // una acció de crear usuari, mostrar usuari, ...
6
7      //fixem-nos el nom del mètode
8      // carreguem el model
9      // fem un usuari nou
10     // a una variable li assignem el resultat del
11     // mètode aconseguir tots
12     // carreguem a la vista
13
14     public function mostrarTots(){
15         require_once 'E:\Programas info\xampp\htdocs\PHP\UF2\2PHP\MVC-01\models\nota.php';
16
17         $nota = new Nota();
18         $totesLesNotes=$nota->aconseguirTots();
19
20         require_once 'E:\Programas info\xampp\htdocs\PHP\UF2\2PHP\MVC-01\views\notes\mostrarTots.php';
21     }
22     // ---- 8 ----
23     public function crear(){
24         require_once 'E:\Programas info\xampp\htdocs\PHP\UF2\2PHP\MVC-01\models\nota.php';
25
26         $nota = new Nota();
27         $nota->setNom("Nota1");
28         $nota->setTitol("8");
29         $nota->setDescripcio("molt be");
30         require_once 'E:\Programas info\xampp\htdocs\PHP\UF2\2PHP\MVC-01\views\notes\mostrarNotes.php';
31     }
32 }
33 }
```

```
UF2 > 2PHP > MVC-01 > views > notes > mostrarNotes.php
1  <!-- ----- 8 ----- -->
2  <h2> <?=$nota->getNom()?> </h2>
3  <h3> <?=$nota->getTitol()?> </h3>
4  <h3> <?=$nota->getDescripcio()?> </h3>
```

```

UF2 > 2PHP > MVC-01 > index1.php
1  <?php
2  echo "<h1>PHP MVC </h1>";
3
4  // require_once 'controllers/usuariController.php';
5  require_once 'controllers/notaController.php';
6
7  $controlador = new NotaController();
8  $controlador->mostrarTots();
9  // ---- 8 ----
10 $controlador->crear();
11

```

localhost/php/UF2/2PHP/MVC-01/index1.php

Twitch YT Moodle BDO Bartali Anime Smix Software

PHP MVC

nota

Nota1

8

molt be

9. Fer que a index.php es carreguin els controladors amb un autoload.php.

```

MVC > MVC-ex-01 > autoload.php > ...
1  <?php
2
3
4  function app_autoloader($classname){
5      //var_dump($classname);
6      //die();
7      require_once 'controllers/'.strtolower($classname).'.php';
8  }
9
10 // fem servir la funció de php spl_autoload_register
11 // li passem com a paràmetre el nom de la funció que carrega les classes
12 // busca totes les clazsses que hi ha en la carpeta que li indicuem
13 // i les carrega
14
15 spl_autoload_register('app_autoloader');
16
17 ?>
18

```

```

notaController.php M  autoload.php U X  mostrarNotes.php U  not
UF2 > 2PHP > MVC-01 > autoload.php
1  <!-- ---- 9 ---- -->
2  <?php
3
4  function app_autoloader($classname){
5      var_dump($classname);
6      require_once 'controllers/'.strtolower($classname).'.php';
7  }
8  sql_autoload_register('app_autoloader');
9  ?>

```

10. Creeu la base de dades que hi ha al moodle.

The screenshot shows the phpMyAdmin interface. On the left, the database structure tree is visible, showing 'notes_m7' with sub-databases 'notes' and 'usuaris'. The main panel shows the 'Estructura' (Structure) tab for the 'notes_m7' database. It lists two tables: 'notes' and 'usuaris'. The 'notes' table has columns: 'id' (InnoDB, utf8mb4_general_ci, 32.0 KB), 'usuari_id' (InnoDB, utf8mb4_general_ci, 32.0 KB), 'titol' (InnoDB, utf8mb4_general_ci, 64.0 KB), and 'descripcio' (InnoDB, utf8mb4_general_ci, 64.0 KB). The 'usuaris' table has columns: 'id' (InnoDB, utf8mb4_general_ci, 32.0 KB), 'nom' (InnoDB, utf8mb4_general_ci, 32.0 KB), 'cognoms' (InnoDB, utf8mb4_general_ci, 32.0 KB), 'email' (InnoDB, utf8mb4_general_ci, 32.0 KB), and 'password' (InnoDB, utf8mb4_general_ci, 32.0 KB).

11. Afegiu la base de dades uns quants usuaris fake i notes fake.

The first screenshot shows the 'usuaris' table in phpMyAdmin. The table has columns: 'id', 'nom', 'cognoms', 'email', 'password', and 'data'. It contains two rows of fake data:

id	nom	cognoms	email	password	data
1	jori	oliveda	jordi@gmail.com	Admin@123	2022-01-30
3	alberto	garcia	alb@gmail.com	Admin@123	2022-01-30

The second screenshot shows the 'notes' table in phpMyAdmin. The table has columns: 'id', 'usuari_id', 'titol', 'descripcio', and 'data'. It contains two rows of fake data:

id	usuari_id	titol	descripcio	data
1	1	6	ta bien	2022-01-30
2	3	10	molt be	2022-01-30

12. Afegeix la classe, adapta-la al teu entorn

```
MVC-02-I-EX > config > database.php > ...
1  <?php
2
3
4  // fem una classe base de dades per connectar a la base de dades
5  // afegim una afunció connectar a la base de dades
6
7  // a modelBase.php
8
9
10 class database {
11     public static function connectar() {
12
13         $host = "localhost";
14         $usuari="root";
15         $password="1234";
16         $database="notes_m7";
17
18
19         $connexio = new mysqli($host,$usuari, $password, $database);
20         $connexio->query("SET NAMES 'utf8'");
21
22         return $connexio;
23     }
24 }
25 ?>
```

```
UF2 > 2PHP > MVC-01 > config > basedatos.php
1  <?php
2  // ---- 12 ----
3  class database{
4      public static function connectar(){
5          $servername = "localhost";
6          $username = "admin";
7          $password = "OEh8lNo#nRZB";
8          $db= "notes_m7";
9
10         $conec = new mysqli ($servername, $username, $password, $db);
11         $conec->query("SET NAMES'utf8'");
12         if ($conec->connect_error) {
13             die("Connection failed: " . $conec->connect_error);
14             return FALSE;
15         }
16         return $conec;
17     }
18 }
19 ?>
```

13. Modifica modelBase.php per a que es connecti a la base de dades.

```

MVC-02-I-EX > models > modelBase.php > ...
1  <?php
2
3  // Com tenir mètodes en comú
4
5  require_once 'config/database.php';
6
7  class ModelBase {
8
9      // tenim una propietat que és la base de dades
10     // al constructor es connecta a la base de dades
11     // http://localhost/mp7-php/UF3/MVC/MVC-02/?controller=usuari&action=mostrarTots
12
13     // a les classes que hereden d'aquest cal heredar el constructor també
14     // per tenir accés a la base de dades
15
16
17
18     public $db;
19
20     public function __construct() {
21
22         $this->db = database::connectar();
23
24     }
25
26     public function aconseguirTots($taula) {
27
28         $query = $this->db->query ("SELECT * FROM $taula ORDER BY id DESC");
29
30         //var_dump($this->db);
31         return $query;
32         //"Llistat de tots el elements";
33     }
34
35 }
36

```

```

UF2 > 2PHP > MVC-01 > models > modelBase.php
1  <?php
2  // ---- 3 ----
3  // ---- 13 ----
4  require_once 'config/basedatos.php';
5  class ModelBase{
6
7      public $db;
8      public function __construct(){
9          $this->db= database::connectar();
10     }
11     public function aconseguirTots($taula){
12         $query = $this->db->query("SELECT * FROM $taula ORDER BY id DESC");
13         return $query;
14     }
15 }
16 ?>

```

14. Adapta el model nota.php als camps de la base de dades, les propietats i els setters i getters.

15. Refés el que calgui per a que vista aconseguirTots mostri les dades en forma de llista tant per a usuaris, com per les notes

16. Fer un mètode desar nota a nota.php. Ajuda

```
$sql = "INSERT INTO notes(usuari_id, titol, descripcio, data) VALUES
({$this->usuari_id},      '{$this->titol}',      '{$this->descripcio}',
CURDATE());";
```

deseu la consulta en una variable p.ex \$sql

aquesta seria l'execució de la consulta

```
$desat = $this->db->query($sql);
```

i retorneu \$desat;

```
UF2 > 2PHP > MVC-01 > models > nota.php
58     return $this;
59     }
60     // ---- 16 ----
61     public function desa ()
62     {
63         $sql = "INSERT INTO notes(usuari_id, titol, descripcio, data)VALUES ({$this->usuari_id},
64         {$this->titol}, '{$this->descripcio}', CURDATE());";
65         $desat = $this->db->query($sql);
66         return $desat;
67     }
68     }
69     ?>
```

17. Al controlador de nota, en el mètode crear, fes una nota de prova i comprova que s'ha desat correctament amb el llistat que ja has fet.

```
UF2 > 2PHP > MVC-01 > controllers > notaController.php
22     // ---- 8 ----
23     public function crear(){
24         require_once 'E:\Programas info\xampp\htdocs\PHP\UF2\2PHP\MVC-01\models\nota.php';
25
26         $nota = new Nota();
27         $nota->setNom("Nota1");
28         $nota->setTitol("8");
29         $nota->setDescripcio("molt be");
30         // ---- 17 ----
31         $nota->desa();
32         require_once 'E:\Programas info\xampp\htdocs\PHP\UF2\2PHP\MVC-01\views\notes\mostrarNotes.php';
33
34     }
35 }
36
37 ?>
```