

# Apuntes irene

AD

Gestor de dependencias, COMPOSER.

NETBEANS. Classroom material aparte de esto.

Al abrir NETBEANS → File → New Project → PHP → PHP with existing resources → Activating PHP

## CONECTORES CON LA BASE DE DATOS Y LA APLICACIÓN

Mysqli es la clase con la que conectamos la app con la base de datos.

Mysqli depende del lenguaje de programación y depende de la tecnología de la base de datos.

DAO= Objeto de acceso a datos. Nos da una serie de métodos con un nivel de abstracción mayor que el de php? Porque nosotros no tenemos que hacer la conexión.

ORM= Objeto de Mapeo Relacional. Vamos a mapear la base de datos, pasa a ser una CLASE(Class), cada tabla de la bbdd pasa a ser una CLASE. Tenemos un método para acceder a la base de datos.

ORM conector tiene nombre y en nuestro caso de Yii2 se llama (ACTIVE RECORD) también vamos a usar HIBERNATE.

Por ejemplo: una tabla ciclista pasa a ser un Class Ciclista.

Con el ORM da igual la tecnología usada o el lenguaje de programación utilizados.

EXAMEN: UTILIZAR indistintamente ORM o DAO.

MVC= Modelo Vista Controlador.

CONTROLADOR acumula las acciones que se pueden realizar en una aplicación. Manipula el MODELO. En el controlador está la lógica. Hay métodos que realizan acciones."organizar evento, por ejemplo.

EL MODELO es la representación de los datos de la aplicación. Modifica la VISTA. Están las clases para acceder a la información. Clase ciclista, clase puerto, ejemplos. Propiedades de los modelos, serán los campos de las tablas.

VISTA es la interfaz de la aplicación, es la parte del front. El usuario OBSERVA la VISTA. Php(HTML).

Framework=se puede crear un entorno de programación, se tendrá una estructura u otra. Base sobre la que programamos(aunque va a contener errores), IMPORTANTEà nos da unos protocolos o normas a la hora de nomenclatura en Yii2à((actionNombremetodo)).

Saber dónde colocar el código?→ LOCALIZACIÓN está estricta.

Create Yii2 project en Google

composer create-project --prefer-dist yiisoft/yii2-app-basic basic

VENDOR carpeta donde está el Núcleo de Yii2 y NUNCA NUNCA se TOCA.

PHP Version 7.2 es la que vamos a usar

INSTALACIÓN PROYECTO FINAL

1. Entrar en GitLabà
2. Abre xamp y start Apache y mysql.
3. Ir a Htdocs: C:/àxampà
4. Crear en htdocs un carpeta llamada yii(en minúsculas).
5. Crear un proyecto en GitLab llamado: "My awesome first project".
6. Ir a la carpeta yii dentro de htdocsàclick derechoàgit bash here
7. Clonar el https de git lab del proyecto nuevo.
8. En consolaà git clone + https la url
9. En Google poner: create yii2 project basic
10. Borrar la carpeta .git en la carpeta yii de htdocs.
11. La primera cuenta darla y copiar esto:
- 12.composer create-project --prefer-dist yiisoft/yii2-app-basic
1. darle a espacio y escribir .
2. Esperamos a que cargue esto.
3. Escribimos "y" para aceptar la descarga y damos intro.
4. Git init en consola y intro
5. Copiar https de git lab del proyectoawesome

6. En consola git remote add origin + https + intro
7. Git add .
8. Git commit -m "Proyecto instalado" + intro
9. Git push origin master
10. Refrescar Git lab y ver todos las cositas instaladas.
11. Clickar en NETBEANS darle a la carpeta+ , new proyect
12. Phpàphp with existing sources
13. NEXT
14. Sources folderàC:\xampp\htdocs\yii
15. "My awesome first" Project en Project name
16. Versión de php será 7.2
17. NEXT
18. INDEX FILEà
19. FINISH
- 20.
21. Seleccionar el GOOGLE CHROME
22. VENDOR NO SE TOCA. NUNCA SE METE EN VENDO Y\_ NUNCA SE CODIFICA.
23. FLECHA VERDE y se ve esto
- 24.
- 25.
26. Configàphpà para config la base de datos
27. 'dsn' ⇒ 'mysql:host=localhost;dbname=ciclistas', Cambiar lo que pone por ciclistas
28. Abrir Web.php
- 29.
- 30.
31. Cambiamos el false a true en línea 51àen ShowScriptName

32. En consola `git add .` + intro
  33. En consola `git commit -m "Aplicación configurada"` + intro
  34. `git push origin master` + intro
  35. Ir a Git lab + actualizar la pagina
  36. Code, commits, commit de aplicación configurada **PARA VER LO QUE HEMOS HECHO, TIPO APUNTES AUTOMÁTICOS**
  37. <http://localhost/yii/web/index.php/gii> en google chrome
  38. Darle a star Model generator.
  39. Table name , pulsar
  - 40.
  - 41.
  42. Preview
  43. GENERATE
  44. Hacemos el CRUD para la tabla de ciclista
  - 45.
  46. Model Class `app\models\Ciclista` en mayúsculas porque es una clase
  47. En Controller Class poner `app\controllers\CiclistaController`
  48. Preview, y Generate
  - 49.
  - 50.
  51. Guardar en NETBEANS
  52. En CONSOLA `git add .` + intro
- Consola `git commit -m "Modelo y CRUD de ciclista creado + Menú añadido"`
1. Consola `git push origin master`
  - 2.
  3. Hfghf

URLS base de una app de yii2 acaba en  
https://.....index.php/controlador/acción

https://....index.pgp/site/indexà lo que

CONTROLADOR/ACCION

CONTROLADOR/ACCION

CONTROLADOR/ACCION

Una acción de un controlador puede hacer una consulta, manipula info que devuelve la consulta, trabajar con variables, hacer cuentas matemáticas, etc.

DEVOLVER EL REDERIZADO DE UNA VISTA: la acción termina lanzando la presentación de los datos que se han modificado.

:: operador de resolución de ámbito, permitir al programador acceder a métodos estáticos.

```
Public function actionConsulta(){  
    $dataProvider = new ActiveDataProvider([  
        'query' => Ciclista::find() -> select("edad" -> distinct(),  
    ]);  
    Return $this->render("resultado",[  
        "resultados" => $dataProvider,  
        "campos" => ['edad'],  
        "titulo" => "Consulta 1 con Active Record",  
        "enunciado" => "Listar las edades de los ciclistas sin repeticiones",  
        "sql" => "SELECT DISTINCT edad FROM ciclista",  
    ]);  
    Public function actionConsulta1(){  
        $dataProvider=new SqlDataProvider({  
            'sql' => 'SELECT DISTINCT edad FROM ciclista',  
        ]);
```

La vista esta en VIEWS en Resultado dentro de site:

```
<?php
```

```
Use yii\grid\GridView;
```

¿>

<div class="jumotron">

<h2><?=\$titulo?></h2>

<p class="lead"><?=\$enunciadp?></p>

<div class="well">

<?=\$sql?>

</div>

<?= GridView::widget([

'dataProvider' => \$resultados,

'columns'=>\$campos

]); ?>

DAO:

BOOSTRAP

READ THE DOCS página para cosas de buscarà

<https://about.readthedocs.com/?ref=readthedocs.com>

USAR VALIDATORS REGULAR EXPRESSION VALIDATOR | MATCH

## **ANTEPROYECTO**

PRESENTACIÓN DEL PROBLEMA PLANTEADO (INTRODUCCIÓN)

Cual es la necesidad que vais a cubrir., o el problema que vamos a solucionar.

OBJETIVOS DE VUESTRO PROYECTO (SOLUCIÓN PROPUESTA Y  
ESPECIFICACIÓN FUNCIONAL)

Requisitos funcionales que va a tener nuestra aplicación.

Cuando tenemos:

1. Dar de alta
2. Dar de baja
3. Modificar
4. Consultar

1,2,3,4 son CRUD y son GESTIONAR.

DIAGRAMA DAFO DEL PROYECTO

## DESCRIPCIÓN TÉCNICA(TECNOLOGÍAS Y LENGUAJES DE PROGRAMACIÓN QUE VAIS A USAR)

Las tecnologías que planeamos usar en nuestro proyecto.

## PLANIFICACIÓN DEL TRABAJO Y ESTIMACIÓN TEMPORAL(DEDICACIÓN SEMANAL PREVISTA EN HORAS/SEMANA Y A QUÉ TAREAS O FASES SE VA A DEDICAR EL TIEMPO)/DIAGRAMA DE GANTT

Distribución de qué vamos a hacer cada semana.

Usar el diagrama de Gantt.

## DESCRIPCIÓN DE LA DOCUMENTACIÓN A ENTREGAR EN EL PROYECTO.

### **Mínimos de documentación a entregar cuando el proyecto finalice**

María Hoyos Gutiérrez

- 

4 dic

- Productos de la planificación: el anteproyecto.
- Productos del análisis. la documentación más típica sería el diagrama ER y el de casos de uso.
- Productos del diseño: podríais tener un prototipo de la interfaz, esquemas de la interfaz, un diagrama de clases, el paso a tablas de la base de datos, documentación relativa a color y tipografía, etc.
- Productos de la implementación: aquí no hay mucho que añadir porque probablemente documentéis el código en comentarios.
- Productos de las pruebas. las pruebas realizadas y los resultados obtenidos
- Memoria del proyecto: Diario de trabajo con las tareas que vais realizando, las dificultades con las que os vais encontrando, los cambios o problemas con los que os encontráis, las soluciones que aportáis, etc.

## BIBLIOGRAFÍA

Libros que hayamos consultado.

## OBSERVACIONES

Todo lo que nos quede por poner y que no entre en los anteriores apartados.

## EXAMEN::

¿Como se resuelve este error?

PHP notice - tiibase/errorException

- -Accediendo como si fuera un objeto (→)

Por qué este código no funciona?

- -La clave primaria no puede ser null

Por qué este código no funciona?

- -No puede haber 2 auto-increment

Dada una url de una aplicación de Yii2: `..index.php/recetas/mostrarrecetas`

¿Cuál es el nombre del método que se ejecuta y cuál es la ruta donde podemos encontrarlo?

- -nombre: `actionMostrarrecetas`, y lo encontramos en:  
`app/controllers/recetasController()`

Corrige el diagrama:

- -(poner un sistema y añadir atributo principal)

¿Qué devuelve esta consulta?

`SELECT provincia, poblacion`

`FROM provincias`

`ORDER BY 2;`

- - una tabla con 2 columnas (provincia y poblacion), ordenadas de menor a mayor por población

¿Qué acción se ejecuta en `https://localhost/feb22/web/index.php/`

- -`actionPuertosEquipo()`

¿Qué devuelve esta consulta?

`Select provincia, MAX(poblacion)`

`FROM provincias`

`GROUP BY autonomia;`

- -Dos tablas una con el nombre de la provincia, y la otra con la población ordenada por mayor a menor

¿Qué devuelve esta consulta?

`SELECT provincia`



```
FROM provincias
WHERE autonomia = (
Select * from provincias
Where(poblacion>100);)
ORDER BY 1, ....ETC
```

- -Da error por que hay un ; despues de poblaci3n>100);)

¿Como se resuelve este error?

Error - Cass 'html' not found

- -use yii\helpers\Html;

¿Por que no funciona el c3digo?

- -Hay una coma al final del c3dgo, despu3s del PRIMARY KEY(id)

::MAS TEORÍA::

::PROGRAMACIÓN::

Hay vida m3s all3 de Java.

CONTENIDOS

- Estado actual del desarrollo de software
- Metodolog3a de desarrollo de software
- Lenguajes m3s populares
- Paradigmas de programaci3n
- Caracter3sticas de los lenguajes de programaci3n..
- Entornos de programaci3n
- Arquitecturas de software.
- Buenas pr3cticas.

Estado actual del desarrollo de software:

Chairs report 2020: Beyond infinity:

OBJETIVOS-PRESUPUESTO-PLANIFICACI3N TEMPORAL son los 3 factores que se analizan a la hora de calificar un proyecto de succesful, challenged o failed.

Problemas en el desarrollo de software.

- Falta de comunicación efectiva con los clientes.
  - Hay que consensuar con el cliente aunque haya cambios de necesidades que provengan del usuario final.
- Falta de comunicación efectiva con el usuario final.
  - El cliente no tiene porqué ser el usuario final.
- Requisitos no definidos.
  - Van a fallar si fallan los dos puntos anteriores. Porque no hace lo que debería hacer.
- Requerimientos demasiado rígidos.
  - El proyecto falla si no permitimos que haya modificaciones. Hay que tener un grado de flexibilidad dentro de la redacción de los requisitos.
- Falta de planificación.
  - Sin planificación, seguro que se va de tiempo y de costes y el proyecto va a fallar.
- Falta de pruebas.
  - Tiene que dar tiempo a probarlo. A hacer pruebas formales.
- Falta de /pobre gestión del proyecto.
  - La gente por encima de nosotros tiene que hacer un buen trabajo para que el proyecto no falle.
- Resistencia al cambio.
  - Hay que adaptarse a los cambios.
- Alta rotación del personal.
  - Es muy alto el nivel de rotación de personal.
  - Tiene que ver con una filosofía capitalista. Sale barato ir cambiando que ir subiendo sueldos.
- Equipos con falta de diversidad.
  - Tiene que haber diferentes perfiles de seres humanos en los equipos de trabajo. Así el proyecto es más completo. Se cubren más requerimientos porque hay más mentes diversas pensando. Que el equipo esté más equilibrado y tenga más logros.

- Falta de formación.
  - Es el punto del que más se quejan los empresarios. Es de lo que menos influye en realidad, porque se tiene que formar a los empleados.

## METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Metodologías en cascada, es ideal si el proyecto pequeño y hay pocos desarrolladores trabajando en el proyecto.

Planificación→ Análisis→Diseño→Implementación→Pruebas→Mantenimiento.

Cuando aumenta el número de desarrolladores y el ámbito o áreas que vana estar dentro del proyecto es recomendable pasar a una metodología AGIL.

Metodologías ágiles

::SUSTITUIR:: Comentarios→Mantenimiento.

Discusiones y planificación → Análisis de requerimientos.

## RANKING DE POPULARIDAD (GOOGLE TRENDS)

- Python(Librerías)
- Java(OO)
  - Paradigma de orientación a objetos.
- JavaScript (Back/Front)
  - Versatilidad para trabajar en front y back.
- C y C++(gaming-sistemas.acceso a memoria subido)
  - No hay orientación a objetos, es estructurado, imperativo. Da acceso directo a memoria donde se cargan las instrucciones. Punteros->apuntan a la memoria.
  - Afecta al hardware del ordenador , en lugar de dar excepciones.
  - C++ → También llamado C con clase o C con clases.
- C# (gaming-versatilidad-bajado)
  - Como tendencia ha subido C# por encima de C++.
- R(ciencia de datos-subido)
  - Cálculos de estadística y representación de datos.
- PHP(servidor cms-bajado)

- El 75% de las webs están hechos con un CMS, y por ende con PHP.  
Hay muchos proyectos que están basados en estructuras de PHP+SQL.  
(Frameworks→Laravel,CodeIgniter.)
- TypeScript(FT)-subido 2
  - Fuertemente Tipado.
  - Es un superconjunto de JavaScript.
- Swift (2014)-Apple
- Objective C-bajado 2-Apple

::MÁS::

## PARADIGMAS DE PROGRAMACIÓN.

Sienta las bases de un conjunto de lenguajes de programación, sienta los preceptos en los que se basan el conjunto de lenguajes de programación.

IMPERATIVO: (CÓMO). Se le da una "receta" con unos pasos que tiene que seguir para obtener un resultado.

- Programación estructurada.(Ejemplo C)
- Programación orientada a objetos (Ejemplo C++, C#, Java). Características:
  - Clases y métodos.
  - El conjunto de los métodos decimos que es el comportamiento (behaviours). Es lo que pueden hacer los métodos.
  - Las propiedades son privadas o protegidos, se oculta la información por medio del encapsulamiento

DECLARATIVO: (QUÉ). Ejemplo: SQL. Le decimos que queremos y nos abstraemos de cómo SQL hace internamente las instrucciones para obtener lo que le hemos pedido.

- Programación lógica. (Ejemplo: Prolog)
- Programación funcional. (Ejemplo: Haskell)

Características de los lenguajes de programación GENERALES:

- Abstracción (Nivel de). Abstracción baja se acerca a lenguaje de máquina, abstracción alta se acerca al lenguaje humano. Nivel abstracción bajo: c después C++, después C# que es de más alto nivel de abstracción.
- Lenguajes fuertemente tipados. Saber qué tipo de dato tienen las variables que vamos a utilizar.
- Lenguajes débilmente tipados. Hasta el tiempo de ejecución no se sabe el tipo de las variables o no se le asigna.
- Lenguajes compilados/Interpretados. Lenguaje compilado: el código pasa por un compilador y lo que sale del compilador son 0,1,0,1, etc. Los lenguajes interpretados son más rápidos, perdemos el control de errores. Los lenguajes interpretados: Un **lenguaje interpretado** es un **lenguaje** de programación para el que la mayoría de sus implementaciones ejecuta las instrucciones directamente, sin una previa compilación del programa a instrucciones en **lenguaje** máquina.

::MAS::MAS TEORÍA::

<https://www.stackscale.com/es/blog/lenguajes-programacion-mas-populares/>

<https://www.youtube.com/watch?v=QFNNuio-Q38&t=4s>

<https://pypl.github.io/PYPL.html>

Es mejor un lenguaje fuertemente tipado para comenzar a aprender a programar desde cero. Si falla, sabes por qué etc.

## ENTORNOS DE PROGRAMACIÓN

- Usar un framework.
- Programar "a mano".

## KARTIK—KARJEE

- json
- Composer update

ESTO ES LA FIRMA de un método:

Public void nombre del método (parámetros)

## ARQUITECTURAS DE SOFTWARE

Cliente-Servidor

- Modelo de dos capas.

#### Modelo-Vista-Controlador

- Modelo de tres capas.

El usuario usa el controlador que son acciones que son funcionalidades. Es donde está la lógica.

El controlador necesita datos del modelo para manipularlos.

Con los datos del modelo vamos a rellenar la vista.

La vista es la presentación de las operaciones con los datos.

Porqué se evolucionó hacia la arquitectura de modelo-vista-controlador.

#### PATRONES DE DISEÑO (GRASP)

##### Buenas Prácticas

- Bajo acoplamiento y alta cohesión. Es lo mejor.
  - Acoplamiento es la dependencia con otros elementos de la programación, de las clases con otras clases o de los métodos con otros métodos.
- Propiedades
  - Bajo coste de mantenimiento
  - Código
    - Legible
    - Reusable
    - Fiable
  - Menor coste de producción

```
<?php
```

```
use yii\helpers\ArrayHelper;
```

```
$this->title = "Los ciclistas";
```

```
?>
```

```
<div class="col-sm-6 col-md-12">
```

```
<div class="thumbnail noticias" style="max-height: 900px;">
```

```
<div class="caption">
```

```
<h2><?= $model→nombre ?></h2>
<hr class="my-4">
<<h4>Peculiaridades del ciclista:</h4>
<div class="btn-group btn-group-justified" style="margin: 11px;">
  <?= "Dorsal: " . $model→dorsal ?>
</div>
<div class="btn-group btn-group-justified" style="margin: 11px;">
  <?= "Edad del ciclista: " . $model→edad ?>
</div>
<div class="btn-group btn-group-justified" style="margin: 11px;">
  <?= "Nombre del equipo al que pertenece: " . $model→nomequipo ?>
</div>
<hr class="my-4">
</div>
</div>
</div>
```