

# Acceso a bases de datos en Yii

Ricardo Pérez López

IES Doñana, curso 2020/2021

## Índice general

<b>1. DAO</b>	<b>2</b>
1.1. <code>yii\db\Connection</code> . . . . .	2
1.2. <code>yii\db\Connection::createCommand()</code> . . . . .	2
1.3. Consultas SQL . . . . .	2
1.3.1. <code>queryAll()</code> . . . . .	2
1.3.2. <code>queryOne()</code> . . . . .	2
1.3.3. <code>queryColumn()</code> . . . . .	2
1.3.4. <code>queryScalar()</code> . . . . .	2
1.4. Sentencias no <code>SELECT</code> . . . . .	2
1.4.1. <code>execute()</code> . . . . .	2
1.4.2. <code>insert()</code> . . . . .	3
1.4.3. <code>update()</code> . . . . .	3
1.4.4. <code>delete()</code> . . . . .	3
<b>2. Query Builder</b>	<b>3</b>
2.1. <code>yii\db\Query</code> . . . . .	3
2.2. Creación de consultas . . . . .	3
2.2.1. <code>select()</code> . . . . .	3
2.2.2. <code>from()</code> . . . . .	3
2.2.3. Condiciones y filtrado de filas . . . . .	3
2.2.4. <code>orderBy()</code> . . . . .	3
2.2.5. <code>groupBy()</code> . . . . .	3
2.2.6. Condiciones y filtrado de grupos . . . . .	3
2.2.7. <code>limit()</code> . . . . .	3
2.2.8. <code>offset()</code> . . . . .	4
2.2.9. Combinaciones . . . . .	4
2.2.10. <code>union()</code> . . . . .	4
2.3. Recogida de resultados . . . . .	4
2.3.1. <code>all()</code> . . . . .	4
2.3.2. <code>one()</code> . . . . .	4
2.3.3. <code>column()</code> . . . . .	4
2.3.4. <code>scalar()</code> . . . . .	4
2.3.5. <code>exists()</code> . . . . .	4

2.3.6. <code>count()</code> . . . . .	4
2.3.7. Funciones de grupo . . . . .	4
2.3.8. <code>indexBy()</code> . . . . .	4
2.4. Consultas por lotes . . . . .	4
2.4.1. <code>batch()</code> . . . . .	4
2.4.2. <code>each()</code> . . . . .	4
<b>3. Active Record</b> . . . . .	<b>4</b>
3.1. <code>findOne()</code> . . . . .	4
3.2. <code>findAll()</code> . . . . .	4
3.3. <code>save()</code> . . . . .	4
3.4. ActiveQuery . . . . .	5
3.4.1. <code>find()</code> . . . . .	5
3.5. Atributos sucios . . . . .	5
3.6. Relaciones . . . . .	5
3.6.1. Encadenamiento de relaciones . . . . .	5
3.7. <code>joinWith()</code> . . . . .	5
3.8. Atributos virtuales . . . . .	5
3.8.1. Siete técnicas . . . . .	5
<b>4. Metadatos</b> . . . . .	<b>5</b>
4.1. Objetivos de la unidad . . . . .	5
4.2. Resultados de aprendizaje y criterios de evaluación asociados . . . . .	5
4.2.1. RA2 . . . . .	5
4.2.2. RA3 . . . . .	5
4.2.3. RA4 . . . . .	5
4.2.4. RA5 . . . . .	5
4.2.5. RA6 . . . . .	6
4.2.6. RA9 . . . . .	6

## 1. DAO

### 1.1. `yii\db\Connection`

### 1.2. `yii\db\Connection::createCommand()`

### 1.3. Consultas SQL

#### 1.3.1. `queryAll()`

#### 1.3.2. `queryOne()`

#### 1.3.3. `queryColumn()`

#### 1.3.4. `queryScalar()`

### 1.4. Sentencias no SELECT

#### 1.4.1. `execute()`

**1.4.2. insert()**

**1.4.3. update()**

**1.4.4. delete()**

## **2. Query Builder**

**2.1. yii\db\Query**

**2.2. Creación de consultas**

**2.2.1. select()**

**2.2.2. from()**

**2.2.3. Condiciones y filtrado de filas**

**2.2.3.1. where()**

**2.2.3.2. Formatos de condiciones**

De cadena

De array

De operadores

**2.2.3.3. andWhere()**

**2.2.3.4. orWhere()**

**2.2.3.5. filterWhere()**

**2.2.3.6. andFilterWhere()**

**2.2.3.7. orFilterWhere()**

**2.2.4. orderBy()**

**2.2.5. groupBy()**

**2.2.6. Condiciones y filtrado de grupos**

**2.2.6.1. having()**

**2.2.6.2. filterHaving()**

**2.2.6.3. andFilterHaving()**

**2.2.6.4. orFilterHaving()**

**2.2.7. limit()**

**2.2.8. offset()****2.2.9. Combinaciones****2.2.9.1. join()****2.2.9.2. innerJoin()****2.2.9.3. leftJoin()****2.2.9.4. rightJoin()****2.2.10. union()****2.3. Recogida de resultados****2.3.1. all()****2.3.2. one()****2.3.3. column()****2.3.4. scalar()****2.3.5. exists()****2.3.6. count()****2.3.7. Funciones de grupo****2.3.7.1. sum()****2.3.7.2. average()****2.3.7.3. max()****2.3.7.4. min()****2.3.8. indexBy()****2.4. Consultas por lotes****2.4.1. batch()****2.4.2. each()****3. Active Record****3.1. findOne()****3.2. findAll()****3.3. save()**

### 3.4. ActiveQuery

#### 3.4.1. find()

### 3.5. Atributos sucios

### 3.6. Relaciones

#### 3.6.1. Encadenamiento de relaciones

### 3.7. joinWith()

### 3.8. Atributos virtuales

#### 3.8.1. Siete técnicas

##### 3.8.1.1. Calcular a mano cuando/donde haga falta

##### 3.8.1.2. Usar vistas SQL

##### 3.8.1.3. Sobreescibir el método find() del modelo para que se use siempre en lugar del heredado de ActiveRecord

##### 3.8.1.4. Sobreescibir el método afterFind() para rellenar el atributo a mano cada vez que se hace un find()

##### 3.8.1.5. Capturar el evento EVENT\_AFTER\_FIND del modelo

##### 3.8.1.6. Usar una propiedad con getter y setter

##### 3.8.1.7. Crear un método findEspecial() que se usará en lugar de find() cuando haga falta

##### 3.8.1.8. La mejor opción, en la mayoría de los casos: combinar las dos anteriores

Ejemplo

## 4. Metadatos

### 4.1. Objetivos de la unidad

### 4.2. Resultados de aprendizaje y criterios de evaluación asociados

#### 4.2.1. RA2

#### 4.2.2. RA3

#### 4.2.3. RA4

##### 4.2.3.1. CE4.g

#### 4.2.4. RA5

4.2.4.1. CE5.f

4.2.4.2. CE5.g

4.2.4.3. CE5.h

4.2.5. RA6

4.2.5.1. CE6.a

4.2.5.2. CE6.b

4.2.5.3. CE6.c

4.2.5.4. CE6.d

4.2.5.5. CE6.e

4.2.5.6. CE6.f

4.2.5.7. CE6.g

4.2.5.8. CE6.h

4.2.6. RA9

4.2.6.1. CE9.e

4.2.6.2. CE9.f

4.2.6.3. CE9.g