

Índice

- [Atributos de Doctrine](#)
 - [¿Qué es Doctrine?](#)
 - [#\[ORM\Id\]](#)
 - [#\[ORM\GeneratedValue\]](#)
 - [#\[ORM\Column\]](#)
 - [#\[ORM\OneToMany\]](#)
 - [#\[ORM\ManyToOne\]](#)
 - [#\[ORM\JoinColumn\]](#)
 - [#\[ORM\OneToOne\]](#)
 - [#\[ORM\ManyToMany\]](#)
 - [#\[ORM\Index\]](#)
 - [#\[ORM\UniqueConstraint\]](#)
 - [#\[ORM\Table\]](#)
 - [#\[ORM\Embeddable\]](#)
 - [#\[ORM\Embedded\]](#)

Atributos de Doctrine

¿Qué es Doctrine?

Doctrine ORM es una herramienta que permite mapear entidades de PHP (clases y objetos) a tablas de bases de datos relacionales, gestionando automáticamente las relaciones, las consultas y las operaciones CRUD.

Desde Doctrine ORM 2.9, se introdujeron atributos PHP como una alternativa moderna a las tradicionales anotaciones de Doctrine (comentarios con @ en PHPDoc). Estos atributos usan la sintaxis moderna de PHP (con #[...]) y ofrecen las mismas capacidades que las anotaciones anteriores.

#[ORM\Id]

- Declara que el campo es la clave primaria de la tabla asociada a la entidad.
- **Ejemplo:**

```
#[ORM\Id]
#[ORM\Column(type: "integer")]
#[ORM\GeneratedValue]
private int $id;
```

#[ORM\GeneratedValue]

- Indica que el valor del campo será generado automáticamente por la base de datos (generalmente usado con claves primarias incrementales).
- **Ejemplo:**

```
#[ORM\Id]
#[ORM\GeneratedValue]
#[ORM\Column(type: "integer")]
private int $id;
```

#[ORM\Column]

- Define una columna en la base de datos.
- Atributos:
 - **type**: Define el tipo de datos.
 - **length**: Longitud máxima (solo aplica para tipos como string).
 - **nullable**: Permite valores nulos.
 - **unique**: Define que el valor debe ser único.
- Tipos comunes:
 - **integer**: Números enteros.
 - **string**: Cadena de texto.
 - **boolean**: Verdadero o falso.
 - **decimal**: Números decimales con precisión y escala.
 - **datetime**: Fechas y horas.
- Ejemplos:

```
#[ORM\Column(type: "string", length: 100)]
private string $nombre;

#[ORM\Column(type: "decimal", precision: 10, scale: 2)]
private float $precio;

#[ORM\Column(type: "boolean", nullable: true)]
private ?bool $activo = null;
```

#[ORM\OneToMany]

- Declara una relación de uno a muchos con otra entidad.
- Atributos:
 - **mappedBy**: Campo en la otra entidad que mapea esta relación.
- Ejemplo:

```
#[ORM\OneToMany(mappedBy: "categoria", targetEntity: Producto::class)]
private Collection $productos;
```

#[ORM\ManyToOne]

- Declara una relación de muchos a uno con otra entidad.
- Atributos:

- **targetEntity**: Clase de la entidad relacionada.
- **inversedBy**: Campo en la otra entidad que representa esta relación.
- **Ejemplo:**

```
#[ORM\ManyToOne(targetEntity: Categoria::class, inversedBy: "productos")]
#[ORM\JoinColumn(name: "categoria_id", referencedColumnName: "id", nullable:
false)]
private Categoria $categoria;
```

#[ORM\JoinColumn]

- Define la columna de unión en una relación.
- Atributos:
 - **name**: Nombre de la columna.
 - **referencedColumnName**: Campo en la entidad relacionada.
 - **nullable**: Si la columna acepta valores nulos.
- **Ejemplo:**

```
#[ORM\JoinColumn(name: "categoria_id", referencedColumnName: "id", nullable:
false)]
private Categoria $categoria;
```

#[ORM\OneToOne]

- Declara una relación de uno a uno con otra entidad.
- Atributos:
 - **targetEntity**: Clase de la entidad relacionada.
 - **mappedBy**: Campo en la otra entidad que mapea esta relación.
- **Ejemplo:**

```
#[ORM\OneToOne(mappedBy: "perfil", targetEntity: Usuario::class)]
private Usuario $usuario;
```

#[ORM\ManyToMany]

- Declara una relación de muchos a muchos con otra entidad.
- Atributos:
 - **targetEntity**: Clase de la entidad relacionada.
 - **inversedBy**: Campo en la otra entidad que mapea esta relación.
 - **JoinTable**: Define la tabla intermedia.
- **Ejemplo:**

```
#[ORM\ManyToMany(targetEntity: Rol::class, inversedBy: "usuarios")]
#[ORM\JoinTable(name: "usuarios_roles")]
private Collection $roles;
```

#[ORM\Index]

- Define un índice en una o más columnas.
- Atributos:
 - **columns**: Columnas incluidas en el índice.
 - **name**: Nombre del índice.
- **Ejemplo**:

```
#[ORM\Index(columns: ["email"], name: "idx_usuario_email")]
```

#[ORM\UniqueConstraint]

- Define una restricción de unicidad en una o varias columnas.
- Atributos:
 - **columns**: Lista de columnas que forman la restricción.
- **Ejemplo**:

```
#[ORM\UniqueConstraint(columns: ["email"])]
```

#[ORM\Table]

- Declara una tabla de base de datos personalizada para una entidad.
- Atributos:
 - **name**: Nombre de la tabla.
 - **indexes**: Define índices para la tabla.
 - **uniqueConstraints**: Define restricciones de unicidad.
- **Ejemplo**:

```
#[ORM\Table(name: "usuarios", indexes: [#[ORM\Index(columns: ["email"],
name: "idx_email")])])]
```

#[ORM\Embeddable]

- Define una clase que puede ser reutilizada como un objeto incrustado dentro de otra entidad.
- **Ejemplo**:

```
#[ORM\Embeddable]
class Direccion {
    #[ORM\Column(type: "string", length: 255)]
    private string $calle;

    #[ORM\Column(type: "string", length: 100)]
    private string $ciudad;
}
```

#[ORM\Embedded]

- Permite incluir una clase embeddable dentro de una entidad.
- **Ejemplo:**

```
#[ORM\Embedded(class: Direccion::class)]
private Direccion $direccion;
```