# Binding parameters

El objetivo del binding parameters es agregar parámetros dinámicamente sin modificar la consulta y evitar problemas de seguridad como es la inyección sql.

Se puede realizar de 2 maneras con números {?1} o con cadenas{:value}.Cualquiera de las 2 es permitida pero no se pueden mezclar es decir o una u otra.

Ejemplo -1

<?php

// $qb instanceof QueryBuilder

$qb->select('u')

->from('User', 'u')

->where('u.id = ?1')

->orderBy('u.name', 'ASC')

->setParameter(1, 100); // Sets ?1 to 100, and thus we will fetch a user with u.id = 100

Ejemplo -2

<?php

// $qb instanceof QueryBuilder

$qb->select('u')

->from('User', 'u')

->where('u.id = :identifier')

->orderBy('u.name', 'ASC')

->setParameter('identifier', 100); // Sets :identifier to 100, and thus we will fetch a user with u.id = 100

## setParameter

En el caso de la fecha es bueno utilizar el 3 parámetro para ayudar en el rendimiento

$qb->setParameter('date', new \DateTimeImmutable(), Types::DATETIME\_IMMUTABLE);

Si son varios los parametros a pasar se puede utilizar en su lugar setParametrs y pasar un arreglo de parámetros. Ejemplo

$qb->setParameters(new ArrayCollection([

new Parameter('1', 'value for ?1'),

new Parameter('2', 'value for ?2')

]));