

# ORM

TADP - 2021 - 1C - TP Metaprogramación

## Entrega Individual

El objetivo de esta entrega es extender el trabajo práctico grupal para agregar una nueva funcionalidad. Este agregado debe hacerse utilizando las herramientas vistas en clase, teniendo especial cuidado de no romper la funcionalidad provista en la entrega anterior.

## Inferencia de Atributos Persistentes

Para esta entrega se pide realizar los cambios necesarios al framework para permitir que los atributos con accesors serán automáticamente considerados persistentes.

De ahora en adelante, además de poder declarar atributos persistentes utilizando *has\_one* y *has\_many*, todos los atributos que sean declarados en una clase persistente utilizando *attr\_accessors* deben considerarse persistentes, de acuerdo a las siguientes reglas:

- Si el nombre del atributo termina en “s” debe asumirse que se trata de un *has\_many* (de lo contrario, se asume que es un *has\_one*).
- Si es un *has\_one* y su nombre en PascalCase existe como una constante y esta es un tipo persistente, el tipo del atributo es esa constante.
- Si es un *has\_many* y su nombre en singular (o sea, sin la “s” final) en PascalCase existe como una constante y esta es un tipo persistente, el tipo del atributo es esa constante.
- En cualquier otro caso se asume que el atributo no debe persistirse.

```
class Espadachin
  attr_accessors
    :nombre, # No se persiste: No hay ninguna clase Nombre
    :espada, # Persiste! Equivale a has_one Espada, named: :espada
    :enemigos # Persiste! Equivale a has_many Enemigo, named: :enemigos
  end
class Espada end

class Enemigo end
```

**Nota:** Puede utilizarse el método *to\_pascal\_case()*:

```
def to_pascal_case(str)
  return str.split('_').collect(&:capitalize).join
end
```

```
to_pascal_case("hello_world") #=> "HelloWorld"
```

Es **MUY IMPORTANTE** notar que el cambio solicitado debe complementar la funcionalidad actual, no reemplazarla! El uso anterior de *has\_one* y *has\_many* debe ser respetado.