Pregunta 1:

What purpose does the **case** statement serve?

- ○ Upcase all strings in the Ruby file
- ○ Store various documents and folders within a Ruby file.
- ○ Specify the capitalization of every string in the `if` statement.
- ◉ Consolidate multiple `if / elsif` statements into a simpler design.

Pregunta 2:

How can the following code be expressed otherwise?

```ruby
age = 30

if age != 25
    puts "Hello there!"
end
```

- ○
```ruby
if !age == 25
    puts "Hello there!"
end
```
- ◉
```ruby
unless age == 25
    puts "Hello there!"
end
```
- ○
```ruby
if age !! 25
    puts "Hello there!"
end
```

Pregunta 3:

How can the following code be refactored?

```ruby
if 3 * 2 == 6
    puts "The two are equal"
end
```

- ○ `puts (if 3 * 2 == 6) "The two are equal"`
- ○ `{if 3 * 2 == 6} puts "The two are equal"`
- ◉
```ruby
puts "The two are equal" if 3 * 2 == 6
```

Pregunta 4:

How can you express the code below in a different way?

```ruby
i = 0
while i <= 10
    puts i
    i += 1
end
```

- ○
```ruby
i = 0
i += 1
until i > 10
    puts i
end
```
- ○
```ruby
while i <= 10
    i = 0
    puts i
    i += 1
end
```
- ○
```ruby
i = 0
until i > 10
    puts i
    i++
end
```
- ◉
```ruby
i = 0
until i > 10
    puts i
    i += 1
end
```

Pregunta 5:

What will `x` be equal to?

```
name = "Boris"
x = name[5]
x ||= "Z"
```

- ○ `nil`
- ○ s
- ◉ Z
- ○ sZ

¡Buen trabajo! Ya estás listo para pasar a la siguiente clase.

Has acertado 5 de 5.

✔ **Lo que sabes**

What purpose does the case statement serve?

How can the following code be expressed otherwise? age = 30 if age != 25 puts "Hello there!" end

How can the following code be refactored? if 3 * 2 == 6 puts "The two are equal" end

How can you express the code below in a different way? i = 0 while i <= 10 puts i i += 1 end

What will x be equal to? name = "Boris" x = name[5] x ||= "Z"