

Considere la jerarquía de clases Ruby para representar una lista de citas y referencias bibliográficas en formato de la Asociación de Psicología Americana (APA), esto es, una *bibliografía*.

1. Diseñar e implementar un Lenguaje de Dominio Específico *Domain Specific Language - DSL*, siguiendo la filosofía de Ruby que permita la definición de una bibliografía.

Los DSL son herramientas útiles que permiten expresar fácilmente la lógica específica de un problema particular (dominio) que de otro modo sería difícil o farragoso de escribir en otro idioma. Por lo general, se trata de definir una gramática que se asemeja más al léxico utilizado por el dominio de destino. Por ejemplo, un matemático que trabaja con matrices no piensa en bucles, iteradores o arrays, sino que piensa en términos de vectores, productos y transformaciones. El uso de un lenguaje de propósito general, como Ruby, con sólo arrays e iteradores requeriría que el matemático practicara gimnasia mental para traducir mentalmente entre el dominio de su problema y el del lenguaje con el que escribe el código (Ruby). El uso de un DSL diseñado para las operaciones que le interesan eliminaría esta traducción mental y proporcionaría un código más conciso.

Los DSL tienen dos formas: externos e internos. Los DSL externos existen independientemente de cualquier otro lenguaje. Los DSL internos están alojados dentro de otro lenguaje de programación - por ejemplo, Rails es un DSL interno que se aloja en el lenguaje de programación Ruby.

El DSL que diseñe ha de permitir la definición de referencias o citas bibliográficas de una forma natural. Por ejemplo:

```
article = Article.new("Ejemplo") do
  author    :surname => "Apellido",
            :name    => "Nombre"
  journal   :name    => "Titulo de la revista",
            :volume  => 20,
            :issue   => 3
  title     "Titulo"
  abstract  "Resumen"
end
```

2. Utilizar la metodología de desarrollo dirigido por pruebas (*Test Driven Development - TDD*) y la herramienta ***RSpec***.
3. Documentar la gema utilizando una herramienta de generación automática de documentación (RDOC o YARD).
4. Crear una cuenta en rubygems.org. El *'handle'* que se solicita es el nombre de usuario.
5. Puesto que ya se ha terminado de escribir código, se está listo para construir y publicar la gema.
 - Para *construir* la gema, desde el directorio raíz creado con Bundler ejecutar: **rake build**
 - Para crear la *version 0.1.0* de la gema ejecutar: **rake release**
 - Para *instalar* gema ejecutar: **rake install**
 - Para comprobar que ha ido bien la instalación, ejecutar: **gem list**