

1. Implementar una clase Ruby para representar y gestionar listas enlazadas (*Linked List*).

Por ejemplo, considere la lista (2,3), su representación como una lista enlazada sería:



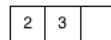
En Ruby, puesto que no existe un objeto `List`, se puede conseguir lo mismo con `Arrays`:

```

a = []
a.unshift 3
a.unshift 2

a = []
a << 2
a << 3
  
```

Internamente, Ruby creará una array como este:



Por lo tanto, los arrays de Ruby pueden funcionar como listas enlazadas, puesto que los métodos `push`, `unshift` e `insert` añaden un nuevo elemento a la lista y los métodos `pop` y `shift` eliminan un elemento de la lista. Sin embargo, se ha de abordar la implementación de una estructura de datos de lista enlazada.

Para el desarrollo de la clase utilizar la metodología de desarrollo dirigido por pruebas (*Test Driven Development - TDD*) y la herramienta ***RSpec***. **Empezar el desarrollo desde cero**, NO RECICLAR CÓDIGO.

Utilizar un `Struct` para su definición.

```

# create a Struct with :value and :next
Node = Struct.new(:value, :next)
  
```

Ejemplos de definición de expectativas son las siguientes:

```

Exam
Simple_Selection
  :Debe existir una pregunta
  :Se debe invocar a un método para obtener la pregunta
  :Deben existir opciones de respuesta
  :Se debe invocar a un método para obtener las opciones de respuesta
  :Se deben mostrar por la consola formateada la pregunta y las opciones de respuesta
  
```

```

Exam
Node
  #Debe existir un Nodo de la lista con sus datos y su siguiente
List
  #Se extrae el primer elemento de la lista
  #Se puede insertar un elemento
  #Se pueden insertar varios elementos
  #Debe existir una Lista con su cabeza
  
```

```

Finished in 0.00933 seconds
10 examples, 0 failures
  
```

2. Repartir las tareas entre los miembros del **Equipo de Trabajo**.

Utilizar el ‘repositorio de *Bitbucket*’ con el nombre del equipo creado en la práctica anterior. Todos los miembros del equipo, han de realizar al menos una confirmación e incorporarla al repositorio compartido.

Utilizar la estructura del ‘directorio de trabajo del equipo’ generado en la práctica anterior con *Bundler*.

Crear una prueba para la siguiente relación de preguntas de selección simple:

1.-) ¿Cuál es la salida del siguiente código *Ruby*?

```
class Xyz
  def pots
    @nice
  end
end

xyz = Xyz.new
p xyz.pots
```

a) #<Xyz:0xa000208>  
b) nil  
c) 0  
d) Ninguna de las anteriores

2.-) La siguiente definición de un *hash* en *Ruby* es válida:

```
hash_raro = {
  [1, 2, 3] => Object.new(),
  Hash.new => :toto
}
```

a) Cierto  
b) Falso

3.-) ¿Cuál es la salida del siguiente código *Ruby*?

```
class Array
  def say_hi
    "HEY!"
  end
end

p [1, "bob"].say_hi
```

a) 1  
b) bob  
c) HEY!  
d) Ninguna de las anteriores

4.-) ¿Cuál es el tipo del objeto en el siguiente código *Ruby*?

```
class Objeto
end
```

a) Una instancia de la clase **Class**  
b) Una constante  
c) Un objeto  
d) Ninguna de las anteriores

5.-) Es apropiado que una clase *Tablero* herede de una clase *Juego*.

a) Cierto  
b) Falso

3. Escribir la dirección HTTP del repositorio que se ha creado en *Bitbucket* en la tarea habilitada en el Campus Virtual.