

**Universidad de La Laguna. Escuela Técnica Superior de Ingeniería Informática**  
**Tercero del Grado de Informática**  
**LENGUAJES Y PARADIGMAS DE PROGRAMACION. PRIMERA PARTE**  
5 páginas

Nombre: \_\_\_\_\_ Alu: \_\_\_\_\_

1. Asocie cada una de las siguientes sentencias en inglés con uno o mas de los cuatro principales paradigmas de programación (La primera contiene un ejemplo del tipo de respuesta que se espera):

- a) Digital hardware technology and the ideas of Von Neumann: IMPERATIVO
- b) Mathematics and the theory of functions
- c) Information hiding is used to protect internal properties of an object
- d) Functions are first class values
- e) Fits well with computations driven by needs
- f) Automatic proofs within artificial intelligence
- g) The values produced are non-mutable
- h) Execution of computational steps in an order governed by control structures
- i) Data as well as operations are encapsulated in objects
- j) Answer a question via search for a solution
- k) Fortran, Algol, Pascal, Basic, C

2. ¿Que números son 010, 0x1F, 0b1111?

3. `s = "hello"`. ¿Cual es el valor de las siguientes expresiones?

- `s[0,2]`
- `s[-1,1]`

4. ¿Cuantos elementos tiene este array? `[-10..0, 0..10,]`

5. ¿Que es esto? `%W[\t \n]`

6. ¿Que queda en `c`?

```
>> c = Array.new(3) { |i| 2*i }
```

7. ¿Cual es el resultado de cada una de estas operaciones?

```
>> a = ('a'...'e').to_a
=> ["a", "b", "c", "d", "e"]
>> a[1,1]
=>
>> a[-2,2]
=>
>> a[0..2]
=>
>> a[0...1]
=>
>> a[-2..-1]
=>
```

8. ¿Cual es el resultado de cada una de estas operaciones?

```
>> a
=> ["a", "b", "c", "d", "e"]
>> a[0,2] = %w{A B}
=> ["A", "B"]
>> a
=>
>> a[2..5] = %w{C D E}
=> ["C", "D", "E"]
>> a
=>
>> a[0,0] = [1,2,3]
=> [1, 2, 3]
>> a
=>
>> a[0,2] = []
=> []
>> a
=>
>> a[-1,1] = [ 'Z' ]
=> ["Z"]
>> a
=>
>> a[-2,2] = nil
=> nil
>> a
=>
```

9. ¿Cual es el resultado de cada una de estas operaciones?

```
>> a = (1...4).to_a
=>
>> a = a + [4, 5]
=>
>> a += [[6, 7, 8]]
=>
>> a = a + 9
```

10. ¿Cual es el resultado de cada una de estas operaciones?

```
>> x = %w{a b c b a}
=>
>> x = x - %w{b c d}
=>
```

11. ¿Cual es el resultado de cada una de estas operaciones?

```
>> a = [1, 1, 2, 2, 3, 3, 4]
=> [1, 1, 2, 2, 3, 3, 4]
>> b = [5, 5, 4, 4, 3, 3, 2]
=> [5, 5, 4, 4, 3, 3, 2]
```

```
>> c = a | b
=>
>> d = b | a
=>
>> e = a & b
=>
>> f = b & a
=>
```

12. ¿Que resultados dan las siguientes operaciones?

```
>> a = 1..10
=> 1..10
>> a.class
=> Range
>> a.to_a
=>
>> b = 1...10
=> 1....10
>> b.to_a
=>
>> b.include? 10
=>
>> b.step(2) {|x| print "#{x} " }

>> 1..3.to_a
```

13. ¿Que resultados dan las siguientes operaciones?

```
>> r = 0...100
=> 0....100
>> r.member? 50
=>
>> r.include? 99.9
=>
>> r.member? 99.9
=>
```

14. ¿Que resultados dan las siguientes operaciones?

```
>> true.class
=>
>> false.class
=>
>> puts "hello" if 0

>> puts "hello" if nil

>> puts "hello" if ""

>>
```

15. ¿Que resultados dan las siguientes operaciones?

```

>> x = { :a => 1, :b => 2 }
=> {:b=>2, :a=>1}
>> x.keys
=>
>> x.values
=>
>> x[:c] = 3
=> 3
>> x
=>
>> x.delete('a')
=> nil
>> x
=>
>> x.delete(:a)
=> 1
>> x
=>
>> x = { :a => 1, :b => 2, :c => 4 }
=> {:b=>2, :c=>4, :a=>1}
>> x.delete_if { |k,v| v % 2 == 0 }
=>
>> x
=>

```

16. ¿Cuales son los resultados?

```

>> x, y = 4, 5
=>
>> z = x > y ? x : y
=>
>> x,y,z = [1,2,3]
=>

```

17. ¿Que hace `yield 4, 5`?

18. ¿Que retorna `block_given?` ?

19. ¿Que hace la siguiente sentencia?

```
counts = Hash.new(0)
```

¿Que diferencia hay con la asignación `counts = {}`?

20. ¿Que retorna esta expresión?

```
'hello world, hello PP2'.scan /\w+|,/
```

21. ¿Cómo se describe una tarea en **Rake**?

22. ¿Cual es el comando para generar una pareja clave-privada clave-pública? ¿Con que argumentos se llama?

23. ¿Que permisos debe tener la clave privada y donde debe estar?

24. ¿Con que opción se le especifica al cliente `ssh` que haga uso de una clave específica?

25. ¿Como se llama y en que lugar está el fichero en el que especificamos las opciones para el cliente `ssh`?
26. ¿Que formato tiene la entrada en el fichero de configuración del cliente `ssh` para una conexión `ssh`?
27. ¿Que opción de `ssh` debo usar para descubrir un problema con el comportamiento del cliente `ssh`?
28. ¿Cuales son las principales diferencias entre un VCS distribuido y uno centralizado?
29. ¿Que es el `stage` o `index`?

30. Complete los comandos para configurar en `git` su nombre e email de forma que esten disponibles en todos los proyectos en la máquina actual:

```
$ git config ----- "Firstname Lastname"
$ git config ----- "your_email@youremail.com"
```

31. ¿Como puedo ver los valores de mi configuración `git`?
32. ¿Cómo puedo hacer que `git` ignore ciertos ficheros?
33. ¿Como creo un repositorio?
34. ¿Como se clona un repositorio cuya url es `git@github.com:crguezl/dsl-latex-simple.git`?
35. ¿Como añado cambios para que estén bajo el control de `git`? ¿En que consiste añadir los cambios?
36. ¿Cómo envío los cambios añadidos al repositorio?
37. ¿Cómo configuro que editor se ejecuta para los mensajes de log?

```
git config --global -----
```

38. ¿ Para que sirven y en que difieren los ficheros `/etc/gitconfig`, `~/.gitconfig` y `~/.git/config`?
39. ¿Cómo veo el estado de mis ficheros con respecto al repositorio?
40. ¿En que estados puede estar un fichero con respecto al repositorio `git`?
41. ¿Que es comparado por `git diff` cuando no se le pasan opciones?
42. ¿Que opción debo pasarle a `git commit` para especificar el mensaje de log y evitar que abra el editor?
43. ¿Que opción debo pasarle a `git commit` para que añada todos los cambios efectuados en ficheros rastreados (*tracked*)?
44. ¿Cómo nos referimos a un commit específico? ¿Cómo se caracterizan las versiones en `git`?
45. ¿Que hace la opción `--graph` de `git log`?
46. ¿Que muestra el comando `git remote -v`?
47. ¿Cómo añado un nuevo remoto?
48. ¿Como me descargo la rama `master` de un remoto `remoto`?
49. ¿Que remoto queda establecido cuando hacemos `git clone`? ¿Que rama será rastreada en el remoto por tu rama `master`?
50. ¿Cómo envío los cambios en el repositorio local en la rama `master` al repositorio remoto apuntado por `origin`?