

# Paradigmas de la Programación – Recuperatorio del Segundo Parcial

21 de Junio de 2016

Apellido y Nombre: \_\_\_\_\_

1. [20 pt.] Estos dos programas están escritos en Perl. ¿Cuál de los dos es funcional? Identifique en el programa no funcional las partes del código que no son funcionales y explique cómo se expresa la misma semántica en la versión funcional para que sea funcional.

```
my @numbers = qw(1 3 5 6 7 8);

my $sum = 0;
for my $num (@numbers) { $sum += $num;}

say $sum;
```

```
my @numbers = qw(1 3 5 6 7 8);

sub reduce (&@) {
    my ($code, $first, @rest) = @_;
    return $first unless @rest;

    return $code->(
        $first,
        reduce($code, @rest),
    );
}

sub sum {
    return reduce { $_[0] + $_[1] } @_;
}

say sum @numbers;$
```

2. [10 pt.] En el siguiente programa en Elixir, identifique porciones del programa que tienen semántica puramente concurrente. Las porciones del programa que ha señalado, ¿son construcciones lingüísticas para forzar atomicidad en porciones del código?

```
task = Task.async fn -> perform_complex_action() end
other_time_consuming_action()
Task.await task
```

3. [20 pt.] A partir del siguiente programa en pseudocódigo, diagrame los diferentes momentos por los que pasa la pila de ejecución, y señale en esos momentos qué variables pueden ser recolectadas por el recolector de basura (*garbage collector*). En los *activation records* sólo necesita diagramar las variables locales. Asuma que todas las variables se representan mediante punteros en los *activation records*, y que el valor de la variable se encuentra en el heap, por lo tanto necesitan ser

recolectadas por el recolector de basura. Explique si la memoria asociada a alguna variable podría haber sido liberada antes de que el recolector de basura la libere, por ejemplo, manualmente.

```
int bla = ...
{ int bli = ...
  int blu = ...
  int blo = ...
  {
    int ble = bli * bla
  }
  bli = ...
  {
    int ble = bli + blo
  }
}
```

4. [10 pt.] El siguiente script en AppleScript busca en Computerworld.com el contenido del texto que hay en el portapapeles. Identifique los elementos del programa que lo caracterizan dentro del paradigma de scripting y explíquelos.

```
set url1 to "http://www.computerworld.com/action/googleSearch.do?cx=014..."
set url2 to "&x=0&y=0&cof=FORID%3A9#1214"

tell application "System Events"
  set myquery to the clipboard
end tell

set thequery to SaR(myquery, " ", "+")

tell application "Chrome"
  activate
  tell application "System Events"
    tell process "Chrome"
      click menu item "New Tab" of menu "File" of menu bar 1
    end tell
  end tell
  set theURL to url1 & thequery & url2
  set URL of document 1 to theURL
end tell
```

5. [10 pt.] Dada la siguiente base de conocimiento en Prolog, explique cómo sería la sucesión de objetivos que hay que satisfacer para contestar la consulta `ama(X,mia)`.

```
hombre(vincent).
mujer(mia).
hombre(jules).
hombre(john).
hija(mia, john).
hijo(jules, john).
ama(X,Y):- padre(X,Y).
padre(Y,Z):- hombre(Y), hijo(Z,Y).
padre(Y,Z):- hombre(Y), hija(Z,Y).
```

6. [10 pt.] Explique cómo se usa la firma de código en los mecanismos de seguridad de Java.
7. [20 pt.] En Occam los programas se construyen con las siguientes primitivas lingüísticas:

**SEQ** : una secuencia de comandos que suceden en orden estricto.

**PAR** : una lista de comandos que se ejecutan sin especificar el orden.

**CHAN** canales de comunicación definidos para que los procesos se comuniquen (lectura o escritura).

En el siguiente programa en Occam:

```
CHAN a
VAR x, y
PAR
SEQ
x := 1
a!x
SEQ
a?y
y := y + 2
```

Usando este programa como marco de comparación, explique cómo los actores implementan concurrencia sin condiciones de carrera, explicando cómo se implementaría este programa en programación con actores.