

# Segundo parcial - 2025-06-05



Borrador guardado

## Preguntas

Considerar el siguiente código. ¿Cuál es el propósito de dosync?

1 punto

```
(def cuenta1 (ref 100))
```

```
(def cuenta2 (ref 200))
```

```
(dosync
```

```
  (alter cuenta1 - 50)
```

```
  (alter cuenta2 + 50))
```

- ☐ Provoca que cada uno de los refs se muta en un hilo propio
- ☒ Garantiza que los dos refs mutan en una transacción atómica
- ☐ Garantiza que las variables no pueden ser negativas
- ☐ Ejecuta todas las operaciones en forma secuencial

Borrar selección



```
List<Integer> nums = List.of(1, 2, 3, 4, 5, 6);  
List<Integer> result = nums.stream()  
    .filter(n -> n % 2 == 0)  
    .map(n -> n * n)  
    .collect(Collectors.toList());
```

1 punto

¿Cuál es el contenido de result?

- ☐ 1, 9, 25
- ☒ 4, 16, 36
- ☐ 1, 4, 9, 16, 25, 36
- ☐ 2, 4, 6

Borrar selección

¿Cuál de las siguientes expresiones es equivalente a  $\lambda x. \lambda y. x$  según la regla  $\alpha$ ? 1 punto

- ☐  $\lambda x. \lambda x. x$
- ☒  $\lambda z. \lambda w. z$
- ☐  $\lambda x. \lambda y. y$
- ☐  $\lambda a. \lambda b. b$

Borrar selección



(defn numbers [] (range))

1 punto

¿Qué ocurre al evaluar (first (numbers))?

- ☒ Devuelve 0
- ☐ Ciclo infinito
- ☐ Devuelve "n"
- ☐ Error

Borrar selección

¿Cuál de los siguientes describe mejor el comportamiento de Prolog ante la falta de información? 1 punto

- ☒ Si algo no se puede probar mediante las reglas y hechos, se asume falso.
- ☐ Si algo no se puede probar mediante las reglas y hechos, se asume verdadero.
- ☐ Si algo no se puede probar mediante las reglas y hechos, se lanza un error.
- ☐ Prolog utiliza inferencia probabilística para adivinar los valores faltantes.

Borrar selección

¿Qué estrategia de búsqueda sigue Prolog para probar las metas y submetas?

1 punto

- ☒ Depth-first
- ☐ Breadth-first
- ☐ Random
- ☐ In-order

Borrar selección



¿Cuál es el principal riesgo si hay múltiples hilos que acceden a una variable mutable compartida sin un mecanismo de sincronización? 1 punto

- ☐ Deadlock
- ☐ Incremento en el tiempo de cómputo
- ☐ Error de compilación
- ☒ Condición de carrera

Borrar selección

$(\text{map } \#(* \% 3)$   
 $(\text{filter } \#(> \% 4) [1\ 3\ 5\ 7]))$

1 punto

¿Cuál es el resultado?

- ☐ (3 5 7)
- ☐ (5 7)
- ☐ (3 9 15 21)
- ☒ (15 21)

Borrar selección

¿Cuál de las siguientes expresiones no tiene forma normal (por entrar en un ciclo infinito al reducir)? 1 punto

- ☐  $(\lambda x.x) y$
- ☐  $\lambda x.(\lambda y.y) x$
- ☒  $(\lambda x.x\ x) (\lambda x.x\ x)$
- ☐  $(\lambda x.\lambda y.x) a\ b$

Borrar selección



Dada la siguiente base de datos Prolog:

1 punto

progenitor(alicia, bruno).

progenitor(bruno, clara).

ancestro(X, Y) :- progenitor(X, Y).

ancestro(X, Y) :- progenitor(X, Z), ancestro(Z, Y).

Indicar cuál será el resultado de la consulta: ancestro(X, clara).

- ☐ Primero X = alicia y luego X = bruno
- ☐ Solo X = bruno
- ☒ Primero X = bruno y luego X = alicia
- ☐ Solo X = alicia

Borrar selección

¿Dado el predicado "misterio", cuál de las siguientes consultas da éxito?

1 punto

misterio([], []).

misterio([H|T], [H2|T2]) :- H2 is H + H, misterio(T, T2).

- ☐ misterio([2, 6, 9], [2, 2, 6, 6, 9, 9]).
- ☐ misterio([2, 6, 9]) = [2, 6, 9].
- ☒ misterio([2, 6, 9], [4, 12, 18]).
- ☐ misterio([2, 6, 9], [2, 6, 9, 2, 6, 9]).

Borrar selección



¿Cuál de los siguientes describe mejor el concepto de transparencia referencial?

1 punto

- ☐ Una llamada a una función puede ser evaluada en forma perezosa.
- ☐ Todas las referencias deben ser inmutables
- ☐ Los argumentos de una función deben ser pasados por referencia
- ☒ Una llamada a una función puede ser reemplazada por su valor de retorno.

Borrar selección

¿Cuál de las siguientes NO es una cláusula de Horn?

1 punto

- ☒  $P \vee Q$
- ☐  $P$
- ☐  $\neg P \vee \neg Q \vee R$
- ☐  $\neg P \vee Q$

Borrar selección

Una expresión en Cálculo Lambda puede ser una:

1 punto

- ☐ Instrucción, expresión o constante
- ☐ Variable, función o invocación
- ☐ Variable, expresión o declaración
- ☒ Variable, abstracción o aplicación

Borrar selección



¿Cuál de las siguientes NO es una función pura?

1 punto

```
int add(int a, int b) {  
    return a + b;  
}
```

```
int nextId() {  
    return (int)(Math.random() * 1000);  
}
```

```
int length(String s) {  
    return s.length();  
}
```

```
boolean isEmpty(List<?> list) {  
    return list.isEmpty();  
}
```

- ☐ length
- ☐ isEmpty
- ☐ add
- ☒ nextId

Borrar selección

¿Cuál de las siguientes afirmaciones es correcta con respecto a las estrategias de evaluación en Cálculo Lambda?

1 punto

- ☐ Call-by-name y call-by-value siempre llegan al mismo resultado.
- ☐ El orden normal garantiza que siempre se llega a la forma normal.
- ☐ El orden aplicativo garantiza que siempre se llega a la forma normal.
- ☒ El orden normal garantiza que siempre se llega a la forma normal, si existe.

Borrar selección



El siguiente programa Python está escrito principalmente en estilo \_\_\_\_:

1 punto

```
numeros = [1, 2, 3, 4, 5, 6]  
resultado = sum(map(lambda x: x**2, filter(lambda x: x % 2 == 0, numeros)))  
print(resultado)
```

- ☐ Concurrente
- ☒ Funcional
- ☐ Objetos
- ☐ Lógico

Borrar selección

En la expresión  $\lambda x.(x\ y)$ , ¿cuál de los siguientes es verdadero?

1 punto

- ☐ Ambas x e y son ligadas
- ☐ Ambas x e y son libres
- ☒ x es ligada, y es libre
- ☐ x es libre, y es ligada

Borrar selección





Considerar el siguiente código. ¿Cuál de los siguientes es FALSO?

1 punto

(doall (pmap println (range 10)))

- ☐ doall es necesario porque pmap devuelve una secuencia perezosa
- ☒ Puede que imprima menos de 10 números
- ☐ Se invoca a println en múltiples hilos
- ☐ Los números se imprimen en orden no determinístico

Borrar selección

Dadas las definiciones:

1 punto

TRUE =  $\lambda x. \lambda y. x$

FALSE =  $\lambda x. \lambda y. y$

IF =  $\lambda p. \lambda q. \lambda r. p \text{ } q \text{ } r$

¿Cuál es el resultado de reducir la expresión IF FALSE a b?

- ☐ a
- ☐  $\lambda y. a$
- ☒ b
- ☐  $\lambda x. b$

Borrar selección



¿Dado el siguiente código, existe la posibilidad de deadlock?

1 punto

```
// en un thread:  
synchronized (objetoA) {  
    synchronized (objetoB) {  
        // ...  
    }  
}
```

```
// en otro thread:  
synchronized (objetoB) {  
    synchronized (objetoA) {  
        // ...  
    }  
}
```

- ☐ No, porque los candados implícitos evitan el deadlock
- ☐ Sí, porque hay un candado por cada thread
- ☒ Sí, porque los candados se toman en orden inverso
- ☐ No, porque se utiliza synchronized

Borrar selección

Para poder utilizar recur en Clojure:

1 punto

- ☐ No debe haber ninguna variable mutable en el ámbito
- ☐ La llamada recursiva no puede recibir parámetros
- ☒ La llamada recursiva debe estar en posición de cola
- ☐ La llamada recursiva debe estar dentro de una función anónima

Borrar selección



Un hilo intenta leer datos de una conexión de red que nunca llegan, y queda esperando sin límite. Mientras tanto, mantiene adquirido un candado. ¿Cuál es la consecuencia más probable? 1 punto

- ☐ El sistema operativo mata el hilo automáticamente después de un tiempo
- ☐ Otros hilos que intentan tomar ese candado pueden tomarlo sin problemas
- ☐ El candado se libera automáticamente después de un tiempo
- ☒ Otros hilos que intentan tomar ese candado quedan bloqueados indefinidamente

Borrar selección

¿Cuál de los siguientes es FALSO en relación a los lenguajes de programación declarativos? 1 punto

- ☐ Permiten describir el problema a resolver, en lugar del algoritmo que hay que aplicar
- ☐ Hacen énfasis en la evaluación de expresiones
- ☐ Apuntan a minimizar o eliminar los efectos colaterales
- ☒ Son siempre más eficientes que los lenguajes imperativos

Borrar selección

```
(defn g [f x]  
  (f (f x)))
```

1 punto

¿Cuál es el resultado de (g inc 5)?

- ☐ 10
- ☒ 7
- ☐ Error
- ☐ 6

Borrar selección



Considerar el siguiente programa. ¿Qué ocurrirá al ejecutarlo?

1 punto

```
class Actividad {  
    public synchronized void esperar() {  
        try {  
            this.wait();  
        } catch (InterruptedException _) {}  
    }  
  
    public synchronized void despertar() throws InterruptedException {  
        this.notify();  
    }  
}  
  
public class Main {  
    public static void main(String[] args) throws InterruptedException {  
        Actividad act = new Actividad();  
  
        Thread t = new Thread(() -> { act.esperar(); });  
        t.start();  
        t.join();  
  
        act.despertar();  
    }  
}
```

- ☐ El método notify() lanza una excepción porque no hay ningún hilo esperando
- ☐ El programa finaliza exitosamente
- ☒ El programa nunca termina porque el hilo se queda bloqueado en el método wait()
- ☐ El método wait() lanza una excepción porque falta adquirir el candado

Borrar selección



¿Cuál de los siguientes describe mejor el concepto de condición de carrera? 1 punto

- ☐ Cuando dos o más hilos están bloqueados esperando para utilizar un recurso
- ☐ Cuando el tiempo de ejecución depende de la cantidad de hilos
- ☐ Cuando ocurre un bug en un sistema multicore
- ☒ Cuando el comportamiento del sistema depende del orden de las operaciones, que no es determinístico

Borrar selección

¿Cuál de los siguientes describe mejor la característica de "evaluación perezosa"? 1 punto

- ☐ Permite crear listas inmutables
- ☒ Permite representar secuencias potencialmente infinitas
- ☐ Permite crear funciones recursivas
- ☐ Permite pasar funciones por parámetro

Borrar selección

¿Cuál de los siguientes mecanismos de java provee exclusión mutua mediante candados intrínsecos? 1 punto

- ☐ Thread.sleep()
- ☒ synchronized
- ☐ ReentrantLock
- ☐ Condition

Borrar selección



Una función de orden superior:

1 punto

- ☒ Recibe o devuelve funciones
- ☐ Invoca métodos de otras clases
- ☐ Tiene una complejidad computacional elevada
- ☐ Opera únicamente con tipos de datos de orden superior, como árboles o grafos

Borrar selección

Atrás

Enviar

Página 2 de 2

Borrar formulario

Nunca envíes contraseñas a través de Formularios de Google.

Este formulario se creó en Facultad de Ingeniería - Universidad de Buenos Aires.  
¿Parece sospechoso este formulario? [Informe](#)

Google Formularios

