Apellido: Guaymas **Nombres:** Matías Julián **Legajo:** 23061/0

Conteste las preguntas CON LAPICERA EN ESTA HOJA, de otra forma no se considerará respondida la pregunta.

CyPLP 2025

EMT-3-A

Verdadero o Falso

Indique la respuesta.

☐ El componente principal del paradigma funcional es la transparencia referencial.

Lo falso.

Pregunta

1	2

fact2 :: Integer -> Integer
fact2 0 = 1
fact2 n = n * fact2(n-1)
	function fact(var n: integer): int;
	begin
	if (n = 0) then
	fact := 1
	else
	fact := n * fact(n - 1);
	end;

Sean los siguientes, códigos escritos en distintos paradigmas. indique a qué paradigma pertenece cada uno y qué característica de seguridad representa uno respecto del otro. Justifique la respuesta.

El código de la izquierda (1) pertenece al paradigma funcional y el código de la derecha al paradigma imperativo (o procedural).

La característica de seguridad la presenta el paradigma funcional (1) que pasa cumple con la parte conocida de transparencia referencial, que significa que dos expresiones sintácticamente idénticas darán siempre el mismo valor y esto reduce los errores y hace que no se produzcan efectos colaterales, a diferencia del imperativo, o sea código 2, donde se modifica el valor de n, y podría provocar efectos no deseados.

podría

Pregunta

Describa brevemente la diferencia entre la sentencia ***break*** y ***continue*** en una sentencia de iteración y muestre con pseudocódigo un ejemplo sencillo de su uso.

La diferencia está en que break finaliza el bucle por completo y la ejecución continúa en la siguiente instrucción después del bucle. continue lo que hace es saltar a la siguiente iteración actual del bucle y pasa a la siguiente iteración

```
//BÚSQUEDA DE UN ELEM
ARRAY DE 1 A 5 ELEM
DESDE i=1 HASTA 5
        SI EL ELEMENTO DEL ARRAY EN LA POS i ES EL QUE BUSCO
            BREAK
// CONTAR SOLO NUM PARES DE INDICE
DESDE i = 1 HASTA 4
        SI i MOD 2=0
            CONTINUE
        IMPAR := IMPAR + 1
```

Pregunta

Sea el siguiente código escrito en PHP indique de qué forma evitaría un error en el sistema si se pudieran manejar cualquiera de las excepciones que no se contemplan en este código. Justifique indicando cual es el resultado de la ejecución.

```
1  <?php
2   try{
3       $a=8;
4       $b=4;
5       $c=$b/$d;
6   }
7   catch(DivisionByZeroError $e){
8       print_r("dividió por 0");
9   }
10  finally{
11      print_r("SE PRODUJO UN ERROR EN LA DIVISIÓN");
12  }
13
14  ?>
```

Existen dos maneras para manejar las excepciones que no se contemplan en este código para evitar error:

- Usando el bloque de excepciones y aprovechando la clase Exception para genérica y en caso de cualquier excepción entrará a este catch. La manera de definirlo sería luego de la línea 9 y la sentencia es: `catch (Exception $e) { //código o print }`
- Redefiniendo la función set_exception_handler().

El resultado de la ejecución es: imprime "dividió por 0" en caso de producirse una división por cero (o el print del código del manejador en caso de surgir otra excepción (el que para evitar error)) y por la sentencia finally, que se ejecuta siempre haya ocurrido una excepción o no, se imprime "SE PRODUJO UN ERROR EN LA DIVISIÓN", finalizando la ejecución.