

Práctica Calificada 1

Programación Lógica y Funcional

Docente: Carlos R. P. Tovar

Ciclo 2025-2

Indicaciones

- Duración: 90 minutos.
- Responda justificando cada solución con código en Haskell.
- Puede usar `ghci` para validar sus funciones.
- Cada pregunta vale 7 puntos (total: 21).

Ejercicio 1 (7 puntos)

Implemente en Haskell una función:

```
1 clasificarNota :: Int -> String
```

que clasifique una nota de 0 a 20:

- 0 – 10 : "Desaprobado"
- 11 – 13 : Regular"
- 14 – 17 : "Bueno"
- 18 – 20 : "Excelente"
- Fuera de rango : "Nota inválida"

Prueba esperada:

```
1 ghci> clasificarNota 9
2 "Desaprobado"
3 ghci> clasificarNota 15
4 "Bueno"
5 ghci> clasificarNota 21
6 "Nota invalida"
```

Ejercicio 2 (7 puntos)

Defina en Haskell una función recursiva:

```
1 contarDigitos :: Int -> Int
```

que retorne la cantidad de dígitos de un número entero positivo.

Ejemplo de uso:

```
1 ghci> contarDigitos 7
2 1
3 ghci> contarDigitos 12345
4 5
```

Extensión: si el número es negativo, conviértalo primero a positivo.

Ejercicio 3 (7 puntos)

Implemente un programa principal recursivo:

```
1 menu :: IO ()
```

que muestre las siguientes opciones:

1. Calcular factorial de un número
2. Calcular el cuadrado de un número
3. Salir

El menú debe repetirse hasta que el usuario elija la opción 3 (Salir).

Ejemplo de ejecución:

Seleccione una opción:

```
1. Factorial
2. Cuadrado
3. Salir
> 1
```

Ingrese un número: 5

El factorial es 120

Seleccione una opción:

```
1. Factorial
2. Cuadrado
3. Salir
> 3
```

Adiós!

Criterios de Evaluación

- Uso correcto de funciones y guards (Ej. 1).
- Implementación recursiva sin usar funciones predefinidas (Ej. 2).
- Manejo de recursion y E/S en menú interactivo (Ej. 3).