PEP 1 Paradigmas de Programación (2020-2)

Noviembre

NOMBRE:_____ RUT: _____ Profesor: _____

Instrucciones

- Tiempo estimado de respuesta: 90 minutos
- Tiempo disponible para entregar respuestas desde que tuvo acceso al enunciado: 3 horas hasta las 23:59 del 4 de noviembre de 2020.
- Lugar de entrega: Campus Virtual.
- Escriba sus respuestas en hojas separadas
- Puede escribir sus respuestas directamente en un documento electrónico o bien puede hacerlo en papel. Escoja el que más le acomode.
- Recomendación: Para asegurar acceso continuo a este enunciado en caso de cortes de suministro eléctrico o en el acceso a Internet, procure descargar este enunciado y si puede, imprimalo.
- Identifique cada hoja de respuestas con su nombre y rut, lo que sirve de complemento a la declaración que ha aceptado en Campus Virtual.
- Identifique cada hoja de respuesta con el profesor de su sección.
- Al terminar la evaluación,
    - Si desarrolló sus respuestas con lápiz y papel, saqué una fotografía o escanee cada una de sus hojas de respuestas. Asegúrese que las imágenes de sus capturas son legibles (abra el archivo y verifique). Luego, agrupe todas las imágenes en un PDF (en Windows basta con seleccionar las imágenes desde el explorador de archivos -> clic derecho -> Imprimir -> Escoger Impresora PDF de Microsoft u otra que disponga) o bien en un archivo comprimido *.zip* o *.rar*.
    - Si desarrolló sus respuestas directamente en formato digital (ej: documento Word, Google Docs, Latex, etc.) guardar el documento como PDF. Se recomienda este formato para evitar problemas en la visualización de estilos.
- Finalmente, subir el archivo al espacio creado para tales efectos en Campus Virtual. Una vez subido el archivo, descárguelo y asegúrese que el archivo se puede abrir (para evitar situaciones de archivos dañados).
- Excepción: En la eventualidad que no pueda subir su trabajo a Campus Virtual en el plazo disponible, enviarlo inmediatamente vía correo a su profesor.

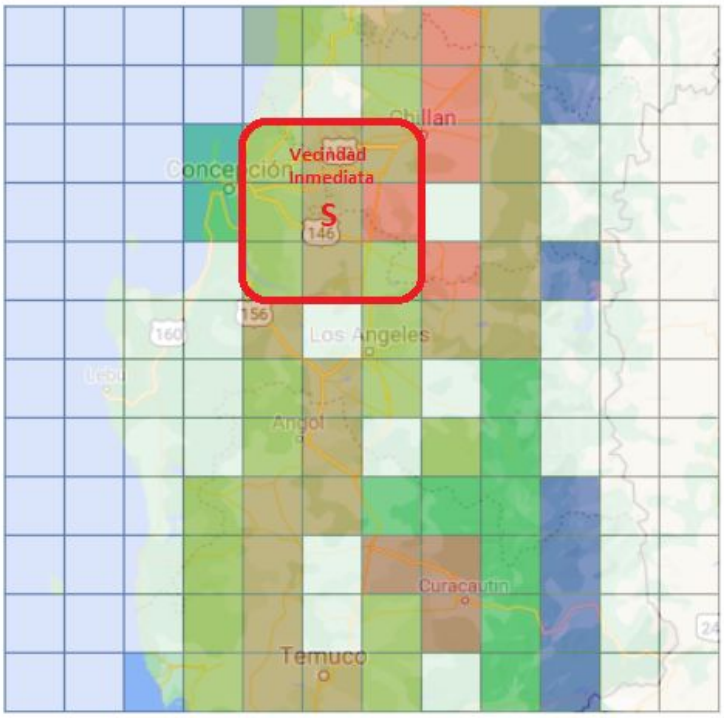
Sobre la evaluación. Todo ítem se evalúa en una escala de 3 puntos: 0 (sin respuesta o respuesta no aborda el problema), 0.5, 1 (respuesta completa sin errores), según grado de cumplimiento.

NOTA 1: Procure hacer un adecuado uso del paradigma funcional (principalmente declarativo en la medida de lo posible) de manera de no redundar en esfuerzos de implementación. Si lo estima conveniente, puede implementar funciones propias del TDA y otras funciones complementarias que pueden ser usadas en ambas preguntas. La implementación innecesaria y/o redundante de funciones que desaprovechen el paradigma funcional tendrán una penalización de hasta **2 puntos** por cada ítem afectado.

NOTA 2: Recuerde documentar todas sus funciones indicando dominio, recorrido, descripción de su funcionalidad, tipo de recursión (si aplica) y ejemplo de uso (una sola línea, es un ejemplo de cómo se evalúa ej: (función 3 4 5)). El no hacerlo puede significar un descuento de hasta **1 punto** por cada ítem afectado.

NOTA 3: Para cada pregunta, puede usar las siguientes funciones de Scheme: car, cdr, if, cond, define, let, null?, =, eqv?, eq? equal?, +, *, /, -, list?, null, let, lazy, lambda, force, not, and, or, eq?, cons, list, round (para redondear), length. Cualquier otra función no indicada en esta lista deberá ser implementada.

1. **(30 pts)** El servicio meteorológico dispone de datos de las temperaturas (en °C) del país las que se encuentran sectorizada en un mapa como se ilustra en la siguiente imagen:



Cada color representa un rango de temperaturas que se ha medido en alguna estación de monitoreo que esté ubicada en ese sector, mientras que los sectores de color blanco significa que no se tiene estación de monitoreo.

Parte del software de análisis del servicio meteorológico requiere de un módulo de búsqueda e interpolación de información, el cual debe estar implementado en scheme siguiendo los lineamientos del paradigma funcional.

a. (**5 pts.**) Especifique un TDA con lo necesario para el módulo de búsqueda e interpolación del servicio meteorológico (ver las siguientes preguntas). Luego implemente el TDA procurando que su representación permita almacenar sólo los sectores que tienen información de las estaciones de monitoreo.

b. (**5 pts.**) Implemente una función en scheme que permita obtener los sectores del mapa cuyas temperaturas cumplan con un criterio C especificado por el usuario (ej: temperaturas en un rango, temperaturas menor que, temperaturas mayor que).

c. (**10 pts.**) Implemente una función en scheme que permita obtener los sectores que están en la vecindad inmediata de cierto sector S (ver imagen ejemplo).

d. (**10 pts.**) Implementar una función que permita estimar la temperatura de cierto sector S basándose en la temperatura de la vecindad. La estimación se debe hacer a través de una función *f* de agregación estadística (ej: min, max, avg, moda) que ingresa como parámetro.

Rúbrica Pregunta 1:

a) TDAs (5 ptos)
   i) Especifica TDAs válidos y coherentes al problema, solo considera sectores con datos (0.5)
   ii) Define selectores adecuados para el problema (0.25)
   iii) Define tanto el TDA sector como el TDA mapa, en el caso del TDA mapa no es solo una lista, también debe tener las dimensiones del mapa (0.25)

b) Desarrollo de una función filter o equivalente (5 ptos)
   i) Entradas y salidas correctas, debe recibir una función de filtrado como argumento (0.25)
   ii) Caso base correcto (0.25)
   iii) Desarrollo recursivo correcto, agrega a lista solo si se cumple con la función de filtrado (0.5)

c) Desarrollo de función para obtener sectores en la vecindad (10 ptos)

i) Entradas y salidas correctas (0.25)
ii) Caso base correcto (si es que no reutiliza función de pregunta b) (0.25)
iii) Desarrollo recursivo correcto, condición de filtrado correcta para obtener la vecindad (si es que no reutiliza función de pregunta b) (0.5)

d) Desarrollo de función para estimar temperatura de sector a partir de vecindad (10 ptos)
i) Entradas y salidas correctas, debe recibir una función estadística como entrada (0.25)
ii) Obtiene vecindad (0.25)
iii) Aplica función estadística sobre la temperatura de la vecindad (0.5)

Descuentos:
1. Uso incorrecto del paradigma funcional (hasta -8 ptos)
2. Documentación poco completa o incoherente con el código (hasta -4 ptos)
3. Funciones auxiliares usadas, pero no especificadas y declaradas (hasta -4 puntos)

Pauta de Ejemplo Pregunta 1:

a) Representación TDAs

```
(define sector (lambda (x y t) (list x y t)))
(define mapa (lambda (ancho alto sectores) (list ancho alto sectores)))
(define sectorGetX (lambda (sector) (car sector)))
(define sectorGetY (lambda (sector) (cadr sector)))
(define sectorGetTemperatura (lambda (sector) (caddr sector)))
(define mapaGetAncho (lambda (mapa) (car mapa)))
(define mapaGetAlto (lambda (mapa) (cadr mapa)))
(define mapaGetSectores (lambda (mapa) (caddr mapa)))
(define (sector? sector)
      (and
            (list? sector)
            (= 3 (length sector))
            (integer? (sectorGetX sector)) (>= (sectorGetX sector) 0)
            (integer? (sectorGetY sector)) (>= (sectorGetY sector) 0)
            (number? (sectorGetTemperatura sector))
      )
)
;Ejemplo de sector: '(1 1 30)
;Ejemplo de mapa de sectores: '(20 20 (1 1 30) (1 2 25) (1 3 20) (3 3 20)
...)
```

b) Desarrollo de una función filter o equivalente

```
;Descripción: filtra algún elemento de una lista por una función criterio
;Dominio: function X list sectores
;Recorrido: list sectores
;Tipo recursión: Natural
;Ejemplo de uso: (filter (lambda (s) (> (sectorGetTemperatura s) 0) )
sectores)
(define filter (lambda (crit sectores)
            (if (null? sectores)
                  mapa
                  (if (crit (car sectores))
                        (cons (car sectores) (filter crit (cdr sectores)))
                        (filter crit(cdr sectores))
                  )
            )
      )
)
;Descripción: filtra solo los sectores de un mapa que cumplan un criterio
;Dominio: function X Mapa
;Recorrido: lista de sectores del mapa que cumplen el criterio
;Ejemplo de uso: (filterSectores
```

```
 (lambda (s) (> (sectorGetTemperatura s) 0) ) mapa)
(define filterSectores (lambda (crit mapa)
           (filter crit (mapaGetSectores mapa) )
     )
)
```

c) Desarrollo de una función para obtener sectores de la vecindad

```
;Descripción: Verifica si un sector está en la vecindad de x,y
;Dominio: sector X integer X integer
;Recorrido: boolean
;Ejemplo: (esVecindad '(4 4 30) 4 5)
(define (esVecindad? s x y)
     (and
          (>= (- x 1) (sectorGetX s))
          (<= (+ x 1) (sectorGetX s))
          (>= (- y 1) (sectorGetY s))
          (<= (+ y 1) (sectorGetY s))
     )
)

;Descripción: Busca los sectores de la vecindad de una posición x,y del map
;Dominio: mapa X integer X integer
;Recorrido: Lista de sectores
;Ejemplo: (buscaVecindad mapa 3 4)
(define (buscaVecindad mapa x y)
     (filterSectores (lambda (s) (esVecindad? s x y)) mapa)
)
```

d) Desarrollo de una función para estimar temperatura a partir de vecindad

```
;Descripción: Estima la temperatura de una posición x,y del mapa a partir
de la temperatura de la vecindad
;Dominio: function X mapa X integer X integer
;Recorrido: number
;Ejemplo: (estimarTemperatura avg mapa 3 4)
(define (estimarTemperatura f mapa x y)
     (f (map sectorGetTemperatura (buscaVecindad mapa x y) ) )
)
```

2. RacketFlix™ es el nuevo sistema de moda para ver películas por *streaming*. El sistema de RacketFlix™ dispone de “películas”, las cuales son compradas por los “usuarios” y son agregadas al “perfil del usuario”. Por lo que cada usuario consta con su propio perfil que despliega las películas adquiridas en este sistema.

   Cada película se identifica con un id (*int*), nombre (*string*), duración en minutos (*int*), año (*int*) y una lista de tags (*string*). Los tags corresponden a etiquetas asociadas a cada película. Por ejemplo, “comedia”, “clasico”, “acción”. Asimismo, cada usuario se identifica con su rut (*string*), username (*string*), una lista de preferencias de estilo.

   RacketFlix utiliza su propio sistema de criptomonedas, el cual es llamado Racket Dollars™ que corresponde a un número entero. Cada usuario tiene una cartera de Racket Dollars, que indica la cantidad disponible a gastar. Del mismo modo, cada película tiene un precio asociado en este tipo de moneda.

   Dentro de la compañía quieren evaluar la posibilidad de migrar RacketFlix a una implementación basada en el paradigma funcional.

   a. **(2 pts)** Especificar TDAs que permitan cubrir con lo necesario para cubrir con las demandas de RacketFlix de acuerdo a los requerimientos de las siguientes preguntas. Luego para la implementación del TDA señalar las representaciones escogidas para cada uno.
   b. **(5 pts)** Implementar una función que permita registrar un nuevo usuario con nombre de usuario y contraseña tomando como base la representación escogida.
   c. **(5 pts)** Implementar una función que permita comprar contenido procurando descontar los montos de su cartera de RacketDollas.
   d. **(5 pts)** Implementar una función que permita seleccionar películas de acuerdo a un criterio C determinado.
   e. Implementar **SOLO UNA** de las siguientes funciones que pueden ser usadas como criterios de la función del apartado d.
      i. **(5 pts)** Películas cuya duración se encuentre en un rango de minutos [min,max]
      ii. **(5 pts)** Películas que contengan TODAS las etiquetas presentadas en una lista L de entrada (etiqueta1, etiqueta2, etiqueta3). Las películas pueden tener otras etiquetas, pero debe contener todas las de la lista para ser seleccionada.
   f. **(8 pts)** Implementar una función que permita aplicar una rebaja dada por una función *f* a un conjunto de películas que cumplan con un criterio C. A modo de ejemplo la función *f* podría ser una rebaja del 30% y el criterio de selección aplicaría para todas las películas del catálogo etiquetadas como comedia.

Rúbrica Pregunta 2:

A. TDAs (2 ptos)
    a. Especifica TDAs válidos y coherentes al problema (0.5)
    b. Define selectores adecuados para el problema (0.25)
    c. Define listas de usuarios y peliculas para facilitar su procesado (0.25)
B. Registrar nuevo usuario (5 ptos)
    a. Entrada correcta (0.25)
    b. Utiliza el constructor de TDA Usuario (0.25)
    c. Añade al usuario a la lista de usuarios (0.5)
C. Comprar película (5 ptos)
    a. Entradas y salidas correctas (0.25)
    b. Caso base correcto (0.25)
    c. Desarrollo recursivo correcto (0.5)
D. Seleccionar película según criterio C (5 ptos)
    a. Entradas y salidas correctas (0.25)
    b. Caso base correcto (0.25)
    c. Desarrollo recursivo correcto (0.5)
E. Funciones de filtrado (5 ptos)
    a. Función filtro puede ser usada en el punto anterior (0.5)
    b. El filtro retorna lo solicitado (0.5)
F. Aplicar Rebaja (8 ptos)
    a. Entradas y salidas correctas (0.25)
    b. Caso base correcto (0.25)
    c. Desarrollo recursivo correcto (0.5)

Descuentos:
1. Uso incorrecto del paradigma funcional (hasta -6 ptos)
2. Documentación poco completa (hasta -6 ptos)

Pauta de Ejemplo Pregunta 2:

```
a) Representación TDAs
(define (pelicula id nombre duracion anio tags precio)
    (list id nombre duracion anio tags precio)
)

(define (usuario rut username password preferencias cartera peliculas)
    (list rut username password preferencias cartera peliculas)
)

(define (listaPeliculas . peliculas) peliculas)

(define (listaUsuarios . usuarios) usuarios)

(define (getId pelicula) (car pelicula))
(define (getNombre pelicula) (cadr pelicula))
(define (getDuracion pelicula) (caddr pelicula))
(define (getAnio pelicula) (cadddr pelicula))
(define (getEtiquetas pelicula) (caddddr pelicula))
(define (getPrecio pelicula) (cadddddr pelicula))

(define (getRut usuario) (car usuario))
(define (getUsername usuario) (cadr usuario))
```

```
(define (getPassword usuario) (caddr usuario))
(define (getPreferencias usuario) (cadddr usuario))
(define (getCartera usuario) (caddddr usuario))
(define (getPeliculas usuario) (cadddddr usuario))
```

b) Registrar nuevo usuario

```
(define (registro rut username password listaUsuarios)
    (if (null? listaUsuarios)
        (cons (usuario rut username password ‘’ 0 null) null)
        (cons (car listaUsuarios) (registro rut username password (cdr listaUsuarios)))
    )
)
```

c) Comprar película

```
(define (agregar lista elem)
    (if (null? lista)
        (cons elem null)
        (cons (car lista) (agregar (cdr lista) elem))
    )
)

(define (comprar usuario pelicula)
    (if (<= (getPrecio pelicula) (getCartera usuario))
        (usuario (getRut usuario)
            (getUsername usuario)
            (getPassword usuario)
            (getPreferencias usuario)
            (- (getCartera usuario) (getPrecio pelicula)
            (agregar (getPeliculas usuario) pelicula)
        )
        usuario
    )
)
```

d) Seleccionar películas segun criterio C

```
(define (filtroPeliculas listaPeliculas c)
    (if (null? listaPeliculas)
        null
        (if (c (car listaPeliculas))
            (cons (car listaPeliculas) (filtroPeliculas (cdr listaPeliculas) c))
            (filtroPeliculas (cdr listaPeliculas) c)
        )
    )
)
```

e) Criterios para filtrar

```
(define (duracionPelicula pelicula min max)
    (if (and (<= (getDuracion pelicula) max) (>= (getDuracion pelicula) min))
        #t
        #f
)

(define (etiquetaPelicula pelicula listaEtiquetas)
    (aux (getEtiquetas pelicula) listaEtiquetas)
)

(define (aux etiquetasPelicula listaEtiquetas)
    (if (null? listaEtiquetas)
        #f
        (if (null? etiquetasPelicula)
            #t
```

```
        (if (eqv? (car etiquetasPelicula) (car (listaEtiquetas)))
            (aux (cdr etiquetasPelicula) (cdr listaEtiquetas))
            (aux etiquetaPelicula (cdr listaEtiquetas))
    )
)
```

f) Aplicar rebaja según función F y criterio C

```
(define (aplicarRebaja listaPeliculas f c)
    (if (null? listaPeliculas)
        null
        (if (c (car listaPeliculas))
            (cons (pelicula (getId (car listaPeliculas))
                (getNombre (car listaPeliculas))
                (getDuracion (car listaPeliculas))
                (getAnio (car listaPeliculas))
                (getEtiquetas (car listaPeliculas))
                (f (getPrecio (car listaPeliculas)))
                (aplicarRebaja (cdr listaPeliculas))
            )
            (aplicarRebaja (cdr listaPeliculas))
         )
    )
)
```