**Universidad de Santiago de chile**

**Facultad de ingeniería**

**Departamento de ingeniería informática**

**Paradigma de programación**

**Informe Laboratorio 1***Simón Sáez*

Profesor: Daniel Gacitua

**Índice**

*Introducción 3*

*Descripción del problema 3*

*Descripción del paradigma y sus conceptos 4*

*Análisis del problema 5*

*Diseño de la solución 5*

*Aspectos de implementación. 6*

*Instrucciones de uso 7*

*Resultado y autoevaluación 8*

*Conclusión 8*

*Referencias 9*

**Introducción**

En el siguiente informe se tratará de varios aspectos de la implementación del paradigma funcional en un proyecto de un parecido red social.

Para ello se explicará mediante varias secciones del informe:

**1. Descripción del problema:** Se describe el problema

**2. Descripción del paradigma y sus conceptos:** Se describe el paradigma

**3. Análisis del problema:** Se hace un análisis al problema planteado

**4. Diseño de solución:** Se describe como se resolverá el problema a detalle

**5. Aspectos de implementación:** Como se implemento la solución y en que

**6. Instrucciones de uso:** Cuales son las instrucciones para usar el programa

**7. Resultado y Autoevaluación:** Resultados obtenidos, alcance obtenido y problemas tenidos

**8. Conclusión**

**Descripción del problema**

Se creará un programa que simule un parecido a una red social como Facebook, Instagram, Twitter, entre otras redes sociales.

En donde esa red social podría hacer lo siguiente:

* Crear una cuenta
* Iniciar sesión de la cuenta
* Hacer una publicación
* Seguir a una persona
* Compartir una publicación
* Comentar una publicación
* Dar likes a publicaciones y/o comentarios

En donde cumplen funciones muy importantes dentro de una red social

**Descripción del paradigma y sus conceptos**

El paradigma funcional son funciones matemáticas por el cual tiene una entrada y una salida sin que tenga estados internos y efectos laterales

En los conceptos usados en este proyecto son nada más ni nada menos que:

* Currificación = Reciben como máximo un parámetro.
* Funciones = Concepto fundamental del paradigma y con el que está ligado toda la estructura.
* Inmutabilidad = No se pueden modificar los datos, solo emula el cambio.
* Recursividad = Es una función que se llama a si misma dentro de ella.
* Orden Superior = Pueden recibir el nombre de una función como parámetro.

**Análisis del problema**

Para el problema se planteará hacer las siguientes cosas:

Los TDA’s es crear:

* Socialnetwork
* Publicacion
* Cuenta
* Comentario

Entonces dentro de esos TDA’s se tendrá que crear los:

* Representación del TDA
* Constructores
* Selectores
* Modificadores
* Pertenencia
* Otras funciones según necesidad

Y luego crear las funciones esenciales del programa:

* register
* login
* post
* follow
* share
* socialnetowork->string
* comment
* like

Pero dentro de código se tendrá que usar obligadamente lo que es llamado recursión para poder recorrer listas, buscar cosas, hacer ciclos, entre otras razones. Y además de el uso de la currificación

**Diseño de la solución**

­Como se dijo anteriormente, se crearán funciones donde:

* register 🡪 Permite registrar un usuario
* login 🡪 Permite iniciar sesión
* post 🡪 Permite hacer una publicación en tu cuenta o en la de otros usuarios
* follow 🡪 Permite seguir a una persona
* share 🡪 Permite compartir una publicación a tu cuenta o a la de otros usuarios
* socialnetwork->string 🡪 Permite mostrar el contenido de toda la socialnetwork o de un usuario
* comment 🡪 Permite el comentar una publicación
* like 🡪 Permite dar like a una publicación o a un comentario

Aunque todas esas funciones son provenientes de login, exceptuando register. Por lo que primero habrá que hacer login y luego la función que se desea hacer.

Ahí es donde se implementará la currificación, permitiendo colocar más parámetros dependiendo de la función.

Aunque en el 90% de esas funciones se edita la socialnetwork, por lo que se necesitará retornar una nueva socialnetwork con los nuevos datos y modificaciones necesarias a ella.

Y como también se trabaja con listas en la social, por ejemplo, la listas de publicaciones, se necesitara si o si la recursión para recorrer aquella lista

**Aspectos de implementación.**

Se usó la versión de racket 7.8 con fondo negro para la vista.

La estructura para el proyecto fue con una división a las funciones en donde se separó por categoría de TDA cada función, así en el archivo fuente solo quedaron las funciones importantes.

Debido a ello se tuvo que importar cada función al archivo fuente (u otro archivo en que lo necesitaran), además de la importación de la librería principal de racket (#lang racket).

Cabe destacar que gracias a eso el orden en el proyecto fue más cómodo por la simple razón de que así puedo saber dónde está cada función y no tener que buscarlo a cada rato mientras escribo.

**Intrucciones de uso**

Las siguientes instrucciones son más que suficiente para probar en la red social

(define encryptFn (lambda (s) (list->string (reverse (string->list s)))))

(define emptyFB (socialnetwork "fb" (day 25 10 2021) encryptFn encryptFn))

(define FB (register (register (register (register (register emptyFB (day 25 10 2021) "user1" "pass1") (day 25 10 2021) "user2" "pass2") (day 25 10 2021) "user3" "pass3") (day 25 10 2021) "user4" "pass4") (day 25 10 2021) "user5" "pass5")

(define FB1 (((login FB "user1" "pass1" post) (day 28 10 2021)) "text" "0st post"))

(define FB2 (((login FB1 "user2" "pass2" post) (day 28 10 2021)) "text" "1th post" "user1"))

(define FB3 (((login FB2 "user3" "pass3" post) (day 28 10 2021)) "text" "2th post" "user2" "user3"))

(define FB4 (((login FB3 "user4" "pass4" post) (day 29 10 2021)) "photo" "meme.png"))

(define FB5 (((login FB4 "user5" "pass5" post) (day 29 10 2021)) "video" "vidaFantastica.mp4" "user2" "user5"))

(define FB6 (((login FB5 "user1" "pass1" follow) (day 29 10 2021)) "user2"))

(define FB7 (((login FB6 "user2" "pass2" follow) (day 29 10 2021)) "user3"))

(define FB8 (((login FB7 "user3" "pass3" follow) (day 30 10 2021)) "user3"))

(define FB9 (((login FB8 "user4" "pass4" follow) (day 30 10 2021)) "user1"))

(define FB10 (((login FB9 "user5" "pass5" follow) (day 1 11 2021)) "user5"))

(define FB11 (((login FB10 "user1" "pass1" share) (day 5 11 2021)) 2))

(define FB12 (((login FB11 "user1" "pass1" share) (day 5 11 2021)) 2 "user2" "user3"))

(define FB13 (((login FB12 "user3" "pass3" share) (day 6 11 2021)) 3 "user2" "user1" "user4"))

(define FB14((((login FB13 "user1" "pass1" comment)(day 9 11 2021)) 2) "Este comentario 1"))

(define FB15((((login FB14 "user3" "pass3" comment)(day 12 11 2021)) 1) "Este comentario 2"))

(define FB16((((login FB15 "user3" "pass3" comment)(day 15 11 2021)) 1) "Este comentario 3"))

(define FB17(((login FB16 "user2" "pass2" like) (day 23 11 2021)) 3))

(define FB18(((login FB17 "user3" "pass3" like) (day 28 11 2021)) 3))

(define FB19(((login FB18 "user5" "pass5" like) (day 28 11 2021)) 2))

(define FB20(((login FB19 "user1" "pass1" like) (day 3 12 2021)) -1))

(define S1 (login FB20 "user1" "pass1" socialnetwork->string))

(define S2 (login FB20 "user2" "pass2" socialnetwork->string))

(define S3 (login FB20 "user3" "pass3" socialnetwork->string))

(define S4 (socialnetwork->string FB20))

**Resultados y autoevaluación**

Los resultados obtenidos alcanzaron un buen porcentaje de logrado. Aunque las funciones no logran enviar un post o hacer un share respecto a si sigue a alguien o no, sino que envía directamente el mensaje.

Se completaron todas las funciones y se hicieron mas funciones que no son obligatorias.

**Conclusión**

El paradigma fue difícil de abordar por el hecho de que no se puede almacenar los datos y modificar, además el sistema muy cerrado que tiene racket hace un poco más difícil trabajarlo, pero está claro que estas funciones son efectuar cierta función que no es necesaria el almacenar los datos, sino que solo basta con el resultado simulado que da.

Referencia

U.F. (s. f.). *Paradigma funcional*. uqbar-wiki. Recuperado 17 de noviembre de 2020, de https://wiki.uqbar.org/wiki/articles/paradigma-funcional.html#:%7E:text=Este%20paradigma%20se%20basa%20en,ej.