

Stackoverflow

Laboratorio 1

Sebastian Villalobos

Profesor: Roberto Gonzalez

Ayudante: Nicolas Alarcón

Santiago – Chile

2 semestre – 2020

Contenido

Introducción.....	3
Contexto.....	4
Diseño de base de datos de conocimiento.	5
Representación de los TDA.....	6
Conclusiones.....	11

Introducción

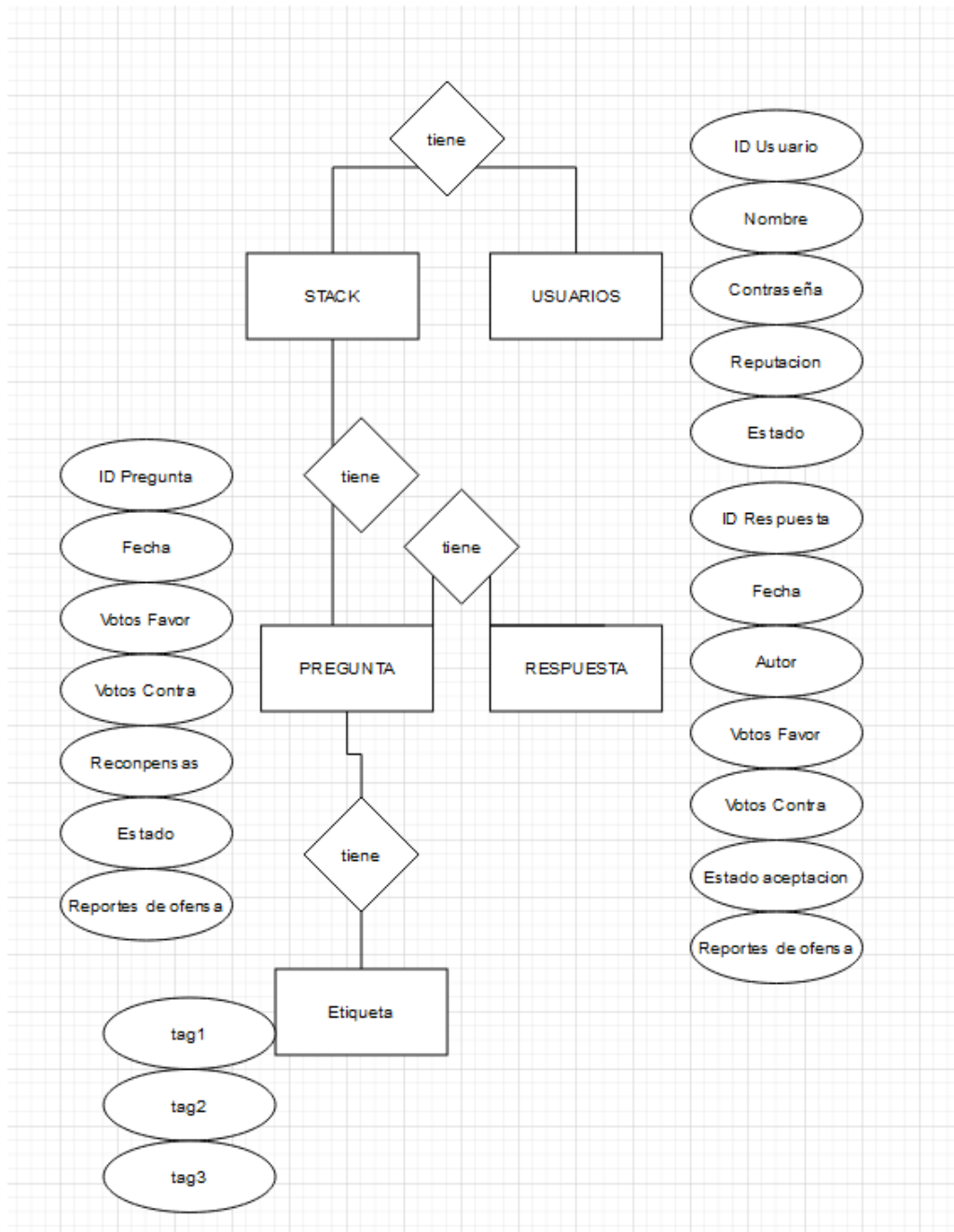
El presente informe tiene por objeto registrar el trabajo realizado para el taller de la clase Paradigmas de programación.

El desarrollo de este proyecto ha sido usando el lenguaje de programación Scheme como respuesta al requerimiento inicial del profesor, además de considerar el paradigma funcional, revisado amplia y detalladamente durante las cátedras. Asimismo, la abstracción de la problemática fue formada con lo que denominamos TDAs, que son representaciones que nos sirven para establecer “cualquier cosa” como una unidad manejable por el lenguaje de programación en el marco del paradigma funcional.

Contexto

El fin de todo esto, es lograr una app que constituye una base de conocimiento, permitiendo plantear preguntas que luego pueden ser respondidas por otros usuarios que simula las funcionalidades y comportamientos de un sistema de preguntas y respuestas como stackoverflow.

Diseño de base de datos de conocimiento.



Representación de los TDA

El programa considera una serie de abstracciones para su ejecución basado en la terminología conocida como TDA, a fin de encapsular las partes esenciales de la aplicación y poder ser codificadas mediante la siguiente estructura técnica definida:

; TDA = Etiquetas

;Representacion

; Las etiquetas son representadas con una lista de elementos, que representan los temas tratados en la pregunta,

; para facilitar la busqueda, en esta representación las etiquetas serán 3 por pregunta.

; no se pueden agregar más etiquetas, solo modificar una menos representativa por una mejor.

;Dominio: 3 strings, tag1, tag2, tag3

;Recorrido: Una Etiqueta, representada por 3 strings

; Tag1: un string etiqueta

; Tag2: un string etiqueta

; Tag3: un string etiqueta

; TDA = Pregunta

```
;Representacion
```

```
; ID Pregunta: Int, identificador unico para cada pregunta.
```

```
; Fecha de creación: Representado por un Int , con el formato AñoMesDiaHoraMinuto, ej 202011051221 -
```

```
> 05 Noviembre del año 2020 hora: 12:21.
```

```
; Autor: String que representa quien formula la pregunta.
```

```
; Votos Favor: Int que representa la cantidad de votos a favor.
```

```
; Votos Contra: Int que representa la cantidad de votos en contra.
```

```
; Recompensa: Int, representa cuantos puntos se dan por responder la pregunta.
```

```
; Estado: int que representa el estado de la pregunta, 1 para abierto (recibe respuestas) o 0 cerrado (no recibe respuestas).
```

```
; Reportes: Int que representa con un numero la cantidad de reportes negativos en la pregunta.
```

```
; Visualisaciones: Int, que representa con un numero la cantidad de visualisaciones de la pregunta.
```

```
; Etiquetas: Una lista representada en el TDA Etiquetas
```

```
; Respuestas: Una lista representada en el TDA Respuestas
```

```
; Ejemplo de una pregunta representada por una lista.
```

```
; (1 202011101930 "Sebastian Villalobos" 14 3 20 1 0 500 "dios existe?" (list "dios" "religion" "extraterrestres") (list "respuesta1" "respuesta2" "respuesta3"))
```

; TDA = Respuesta

;Representacion

; La respuesta está representada por una lista que contiene el id de respuesta, un numero que va a

; ID Respuesta: Int, identificador unico para cada respuesta

; Fecha de creación: Representado por un Int , con el formato AñoMesDiaHoraMinuto, ej 202011051221 -

> 05 Noviembre del año 2020 hora: 12:21.; Autor: String que representa quien formula la pregunta.

; Votos Favor: Int que representa la cantidad de votos a favor.

; Votos Contra: Int que representa la cantidad de votos en contra.

; Estado de Aceptacion: representa si la respuesta es aceptada o no, int, 1 aceptada 0 no aceptada.

; Reportes de Ofensa: int, que representa la cantidad de veces que ha sido reportada la respuesta.

; Respuesta: String que representa la respuesta dada a la pregunta.

; Etiquetas: lista de string con etiquetas relativas a la pregunta.

; TDA = Stack

;Representacion

;El stack está representado por una lista compuesta por una lista de preguntas, que representa todas las preguntas

;realizadas en la plataforma stack, con sus respectivas etiquetas y respuestas, los usuarios son representados por una lista,

;con todos sus atributos y estado.

;Dominio: preguntas, usuarios

;Recorrido: stack

;preguntas: lista (TDA pregunta)

;usuarios: lista (TDA usuario)

; TDA = Usuario

;Representacion

;El usuario es representado por una lista de Int y Strings, que detallan el id del usuario, un numero unico, el nombre del usuario,

; la contraseña del usuario, y la reputacion.

;ID Usuario: Int, identificador unico para cada usuario.

;Nombre de Usuario: String, nombre del usuario en la app.

;Password: String, contraseña del usuario.

;Reputacion: Int, numero que representa la reputacion del usuario, todo usuario nuevo inicia con 50 puntos de reputación.

;Estado: Int que representa si el usuario está activo o no, 1 si está activo, 0 si no lo está.

```
;Dominio: Int, String, String, Int, Int  
;Recorrido: Un usuario, representado por una lista,  
;con su id usuario, nombre usuario, contraseña y reputacion.
```

Conclusiones

Personalmente la realización de este trabajo utilizando el paradigma funcional y el lenguaje de scheme fue un verdadero desafío, la representación a través de funciones me ayudó al desarrollo de tareas cotidianas en mi trabajo, ya sea creando scripts de automatización como en la configuración de algunos servicios de monitoreo en REDES, el único inconveniente que tuve fue el tiempo, una mala planificación