

Universidad de San Carlos de Guatemala

Facultad de Ingenieria

Escuela de Ciencias y Sistemas

Estructuras de Datos

PROYECTO 1
MANUAL TECNICO

Integrantes:

Obed Bosbely Fuentes Eredia 200516309

Pablo Josue Ortiz Linares 202200231

Sebastian Levi Velasquez Valle 202300580

Karvin Armaldo Lago Perez 202200254

Indice

Contenido

Indice.....	2
Introducción.....	2
Comunidades	3
Estructura lista de listas.....	3
Pablo Ortiz	3
Obed Fuentes	4
Levi Velasquez.....	4
Kevin	4
Diseño Visual.....	5
Obed Fuentes	5

Introducción

Este manual muestra las integraciones hechas por el equipo de desarrollo para el proyecto del curso de estructuras de datos en su fase 1.

Comunidades

Se requiere realizar la gestión de comunidades, esta debe almacenar el nombre de la comunidad en una lista enlazada, también debe permitir agregar usuarios a las comunidades la cual también estará en una lista enlazada y ambas formaran una lista de listas.

Estructura lista de listas

Esta estructura contiene las clases necesarias para almacenar usuarios y comunidades para ellos se creo un archivo pas que contiene las estructuras y clases necesarias. A continuación, se muestra las estructuras hechas por cada miembro del grupo y como se integraron entre sí.

Como equipo nos dividimos partes de la estructura para luego unificarlo y probarlo

Pablo Ortiz

ListList: Contiene la estructura que unifica las listas en una sola entidad lista de listas

```
ListList = class
  private
    head : zNode;
    countI : integer;
  public
    property aHead: zNode read head write head;
    constructor create;
    procedure add(comunityName: String);
    procedure insert(u: CUser; comunityName: String);
    function findByEmail(eMail,comunityName: String): CUser;
    function findByName(comunityName: String): CLinkedList;
    procedure comunityReport;
end;
```

Obed Fuentes

CLinkedList: contiene la implementacion de una lista enlazada donde se almacenan la información de comunidades como de usuarios.

```
bNode = ^rNode;  
rNode = record  
  data: CUser;  
  next: bNode;  
end;  
  
CLinkedList = class  
  private  
    cHead: bNode;  
  public  
    constructor create;  
    destructor Destroy; override;  
    procedure add(aUser: CUser);  
    function findEmail(aEmail: string): CUser;  
end;
```

Levi Velasquez

Reporte: Genera el script dot para crear el reporte usando la herramienta graphviz

```
procedure ListList.comunityReport;  
var  
  dotFile: TextFile;  
  currentCom: zNode;  
  currentUser: bNode;  
  comNodeName, userNodeName: string;  
  userCount: Integer;  
  folderPath, dotPath, pngPath: string;  
  proc: TProcess;  
begin  
  // Crea carpeta  
  folderPath := 'Reporte de comunidades';  
  if not DirectoryExists(folderPath) then  
    CreateDir(folderPath);  
  
  dotPath := folderPath + PathDelim + 'comunidades.dot';  
  pngPath := folderPath + PathDelim + 'comunidades.png';  
  
  // Genera archivo de Graphviz  
  AssignFile(dotFile, dotPath);  
  Rewrite(dotFile);  
  
  Writeln(dotFile, 'digraph G {');  
  Writeln(dotFile, '  rankdir=LR;');  
  Writeln(dotFile, '  node [shape=record, style=filled, fontname="Helvetica"];');
```

Kevin

findByName: rutina para buscar por nombre

findByEmail: rutina para buscar por email

```
function CLinkedList.findByEmail(aEmail: string): CUser; //encuentra el usuario dentro de la lista enlazada
var
  current: bNode;
begin
  current := cHead;
  while current <> nil do
  begin
    if current^.data.Email = aEmail then
    begin
      Result := current^.data;
      Exit;
    end;
    current := current^.next;
  end;
  Result := nil;
end;
```

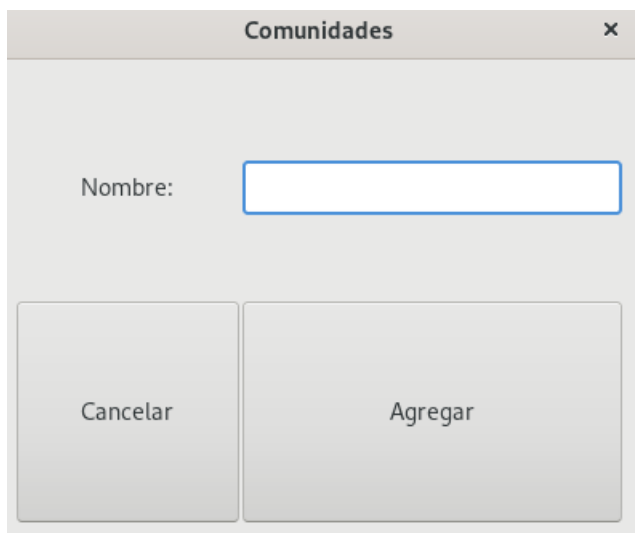
Diseño Visual

Para almacenar el diseño visual cada integrante del grupo tomo la estructura ya realizada he implemento distintas ventanas usando la misma estructura de lista de listas

Obed Fuentes

Diseño 2 pantallas, una para crear una nueva comunidad y la otra para asignar una comunidad a un usuario específico.

Estas pantallas utilizan la misma estructura de lista de listas creada en grupo previamente



The image shows a screenshot of a software window titled "Comunidades" with a close button (X) in the top right corner. Inside the window, there is a label "Nombre:" followed by a rectangular text input field. Below the input field, there are two buttons: "Cancelar" on the left and "Agregar" on the right. The window has a light gray background and a thin black border.

Comunidades

×

Nombre:

Usuario:

Cancelar

Agregar