

Entrega Parcial N°1

Tema: Planificación de las actividades del área de Extensión de una Universidad

Integrantes: Felix Lazcano - Rocio Ramirez - Kevin Rojas

EP1.1:

Utilizaremos en un principio 3 clases principales "Activity", "Student" y "Manager", que hacen referencia a actividades a ser planeadas, estudiantes asistentes a actividades y encargado de actividad respectivamente. La idea general es tener los datos necesarios para el funcionamiento de la aplicación en archivos CSV y que se cargaran al inicio de la aplicación y se actualizaran al salir de esta. Por otro lado tenemos 2 clases que funcionaran como la conexión entre estos componentes "EntryPoint" y "Application"

Funcionalidades Planteadas:

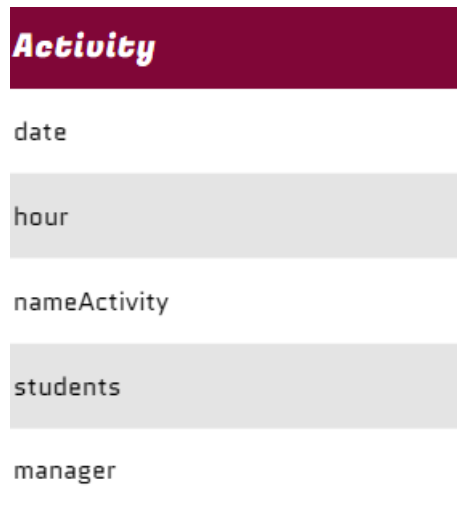
- Agregar actividades a una lista o mapa de actividades.
- Agregar o eliminar estudiantes participantes a una actividad especifica.
- Asignar al encargado (Manager) de una actividad.
- Planificar la actividad para una fecha y hora determinada.
- Mostrar actividades de un cierto día especifico.
- Mostrar estudiantes asistentes a una actividad.
- Mostrar datos del encargado de una actividad.

Funcionalidades A Considerar:

- Calendario que muestre visualmente las actividades en un mes.
- Posibilidad de que un estudiante pueda ser un encargado de actividad.

EP1.2 y EP1.3:

- Clase Activity:



```
1  package com.proyectorplanu.proyectorplanu;
2
3  import java.util.ArrayList;
4
5  public class Activity
6  {
7      private String date;
8      private float hour;
9      private String nameActivity;
10
11     private ArrayList<Student> students;
12     private Manager manager;
13
14     public Activity()
15     {
16         date = "01-01-2022";
17         hour = 08.00f;
18         nameActivity = "Presentarse";
19         students = new ArrayList();
20         manager = new Manager();
21     }
22
23     public String getDate() { return date; }
24     public float getHour() { return hour; }
25     public String getNameActivity() { return nameActivity; }
26     public Manager getManager() { return manager; }
27
28     public void setDate(String date) { this.date = date; }
29     public void setHour(float hour) { this.hour = hour; }
30     public void setNameActivity(String nameActivity) { this.nameActivity = nameActivity; }
31     public void setManager(Manager manager) { this.manager = manager; }
32 }
```

- Clase Manager:

Manager

age

name

job

```
1  package com.proyectoplanu.proyectoplanu;
2
3  public class Manager
4  {
5      private int age;
6      private String name;
7      private String job;
8
9      public Manager()
10     {
11         age = 18;
12         name = "Juan Perez";
13         job = "Encargado";
14     }
15
16     public int getAge() { return age; }
17     public String getName() { return name; }
18     public String getJob() {return job; }
19
20     public void setAge(int age) { this.age = age; }
21     public void setName(String name) { this.name = name; }
22     public void setJob(String job) {this.job = job; }
23 }
24
```

- Clase Student:

Student

age

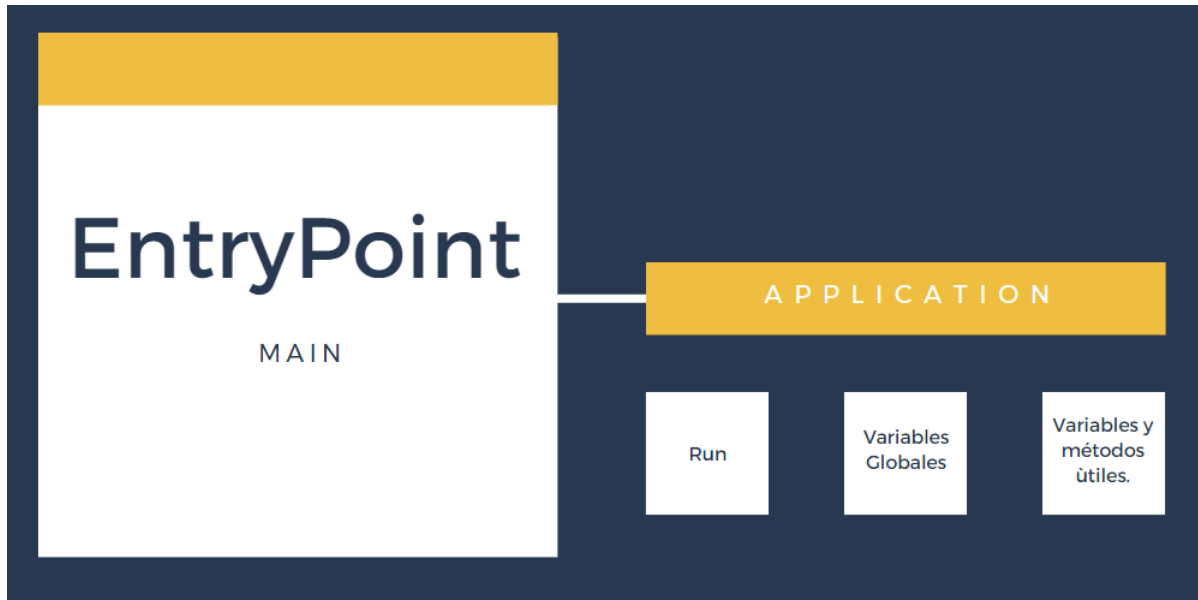
name

years

```
1  package com.proyectoplanu.proyectoplanu;
2
3  public class Student
4  {
5      private int age;
6      private String name;
7      private int years;
8
9      public Student()
10     {
11         age = 18;
12         name = "Julio Martinez";
13         years = 1;
14     }
15
16     public int getAge() { return age; }
17     public String getName() { return name; }
18     public int getYears() { return years; }
19
20     public void setAge(int age) { this.age = age; }
21     public void setName(String name) { this.name = name; }
22     public void setYears(int years) {this.years = years; }
23 }
24
```

EP1.4 y EP1.5:

Cada constructor de clase se inicializa por ahora con valores iniciales por defecto, estos valores pueden ser sobrescritos con los getter y setter de cada clase. Se utiliza una clase temporal de lectura de archivos CSV para comprobar el correcto funcionamiento de nuestra aplicación.



- Clase EntryPoint:

```
1 package com.proyectoplanu.proyectoplanu;
2
3 import java.io.IOException;
4
5 public class EntryPoint
6 {
7     public static void main(String[] arg) throws IOException
8     {
9         Application app = new Application();
10        app.Run();
11    }
12 }
13
```

La clase "EntryPoint" solamente contiene al método main en el cual se crea un objeto (clase) de tipo "Application", esto para tener a simple vista donde inicia nuestra aplicación y sea mas sencillo el debug de problemas.

- Clase Application:

```
7 public class Application
8 {
9     private ArrayList<Activity> activities;
10
11     public void Run() throws FileNotFoundException, IOException
12     {
13         activities = new ArrayList();
14
15         CSV csv = new CSV("Activities");
16         String line = csv.firstLine();
17         while (line != null)
18         {
19             Activity activity = new Activity();
20             activity.setDate(csv.get_csvField(line, 0));
21             activity.setHour(Float.parseFloat(csv.get_csvField(line, 1)));
22             activity.setNameActivity(csv.get_csvField(line, 2));
23
24             Manager manager = new Manager();
25             manager.setName(csv.get_csvField(line, 3));
26             activity.setManager(manager);
27
28             activities.add(activity);
29
30             line = csv.nextLine();
31         }
32
33         for (Activity activity : activities)
34         {
35             System.out.println("Actividad: " + activity.getNameActivity());
36             Manager manager = activity.getManager();
37             System.out.println("Encargado: " + manager.getName());
38             System.out.println("Fecha - Hora: " + activity.getDate() + "/" + activity.getHour());
```

Se esta utilizando como una clase de pruebas, se espera que sirva como la clase principal donde estarán los mapas o listas a utilizar durante el programa.