

UNIVERSIDAD
AUSTRAL



EventIt

TPF

Universidad Austral: Ingeniería Biomédica

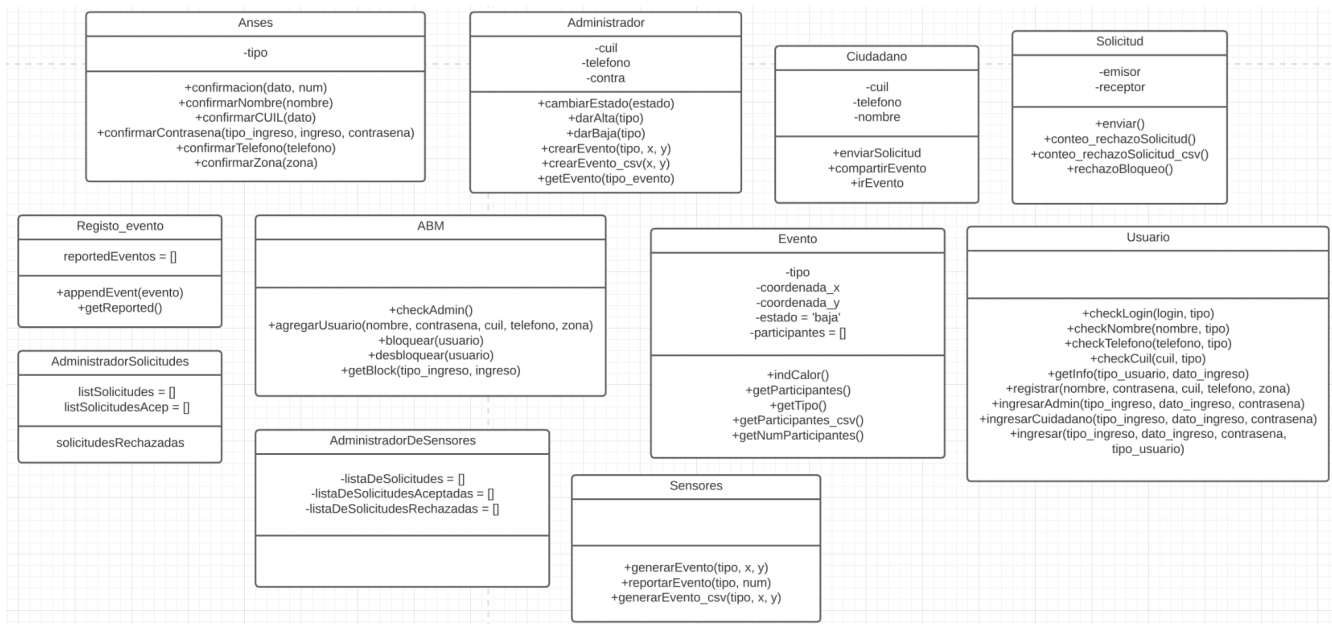
Materia: Introducción a la programación 2

Profesores: Ezequiel Lamonica, Rodrigo Peralta.

Alumnos: Kishkill Ana, López Valentina, Nader Capra Bianca, Sánchez Facundo, Sole Valentina, Viglione Clementina.

Grupo: 13

Fecha de entrega: 15/11/2021



Para comenzar, empezamos planteando un diagrama que nos ayudó a empezar a plantear el diseño para poder después plasmarlo en nuestro sistema. Lo primero que se hizo fue crear una clase de administradores y otra de ciudadanos. A ambos se les pasó como atributo un cuil, teléfono y contraseña. Luego, se implementó un ABM para los usuarios administradores, una base de datos, donde ellos ingresan nuevos usuarios administradores y realizan sus correspondientes tareas, ya sea agregar usuarios, bloquear/desbloquear, etc.

Los administradores tienen la responsabilidad de bloquear y desbloquear usuarios si fueron rechazados más de 5 veces, esto funciona con un contador que se le da a cada usuario como atributo donde, cada vez que alguien rechaza su solicitud se le suma uno al contador. Cuando se llega a 5, un método en el ABM de administradores se encarga de cambiar el atributo que está dentro de usuario que indica su estado (bloqueado/desbloqueado); si está bloqueado aparecerá una “y” de “yes” y si no lo está, aparecerá una “n” de “no”.

Para la funcionalidad del programa se agregó también una clase usuario el cual puede chequear el login, nombre, teléfono, cuil de un tipo de usuario (administrador o ciudadano). Todos estos datos son cruzados con el dataset del ANSES el cual valida la información. Además puede obtener la información de un CSV por tipo de usuario. También registra por nombre, contraseña, cual, telefono y coordenadas en X y en Y en este método chequea que estos datos ya no estén ingresados en el CSV y luego los registra. Y por último ingresa tipos de usuario, el cual requiere un tipo de usuario, de ingreso y un dato de ingreso, en el cual se fija que se hayan completado todos los datos (se confirma con el dataset del ANSES) y también verifica que el usuario no esté bloqueado en el ABM entonces se permite el ingreso y si alguno de los datos anteriores falla se lanza una excepción. Existen

tres métodos de ingreso, el método “ingreso” es el método base donde son ejecutadas las funciones, mientras que “ingresoAdmin” e “ingresoCiudadano” ejecutan las funciones de “ingreso” pasándoles el parámetro “tipo_usuario”, ya se Admin o Ciudadano según corresponda.

Los administradores tienen el poder de dar de baja y alta distintos tipos de eventos, para esto también se le dio la responsabilidad al administrador de crear dichos eventos con coordenadas en X y en Y, lo cual indica una zona, además de poder cambiar el estado del mismo (baja o alta). Para dar de alta un evento, se llama al método “darAlta” lo cual modifica el atributo “estado” que se encuentra dentro de evento, que por default es “baja”; para darlo de baja, se utiliza el método “darBaja” que funciona exactamente igual al otro.

Con respecto a los sensores, lo que se implementó fue una clase Sensores, con tres métodos. Los tres modifican una base de datos CSV, la cual recolecta los datos de cada evento que se cree con su correspondiente sensor. Los métodos son “generarEvento” (el cual recibe como atributo coordenadas en X y en Y, que indican la zona) y “reportarEvento” (que recibe como atributo un número, el cual representa la cantidad de acompañantes que irán al evento).

Como los ciudadanos tienen que poder indicar contactos de interés con otro ciudadano, aceptando mutuamente dicha relación, se les dio la responsabilidad a estos de enviar solicitudes a otros ciudadanos además de compartir un evento y así vincular a algunos de sus contactos de interés. Para manejar el tema de las solicitudes entre ciudadanos se creó la clase Solicitud que cuenta con un emisor y un receptor, la cual interactúa con el ciudadano al enviar la solicitud. Además para resolver otro problema también se le dio la responsabilidad a esta clase de hacer el conteo de las solicitudes rechazadas por un ciudadano. Todos estos datos son guardados en las listas de la clase Administrador de solicitudes que cuenta con una lista de solicitudes y una de solicitudes aceptadas.

Dentro del ciudadano se encuentran métodos que permiten enviar solicitudes a cualquier contacto de interés a través de un receptor, lo cual crea una solicitud a un usuario a través de su CUIL. También se encuentran los métodos de tanto aceptar como rechazar la solicitud. Si se acepta, el CUIL de la persona se appendea a la lista de amigos del ciudadano, si se rechaza se le agrega uno al contador de rechazados de cada ciudadano. Como se mencionó anteriormente, al alcanzar las 5 solicitudes rechazadas el ciudadano es bloqueado por los administradores.

Un ciudadano puede reportar o vincular un evento, eso significa que, si uno quiere asistir al evento puede hacerlo tanto individualmente como grupalmente. Si se elige la opción de “reportarEvento”, el ciudadano será appendeado a la lista de participantes del evento, lo cual está vinculado al CSV donde se ven modificados los datos de participantes. Si se desea invitar amigos (vincularlos), se le pasa una lista de ciudadanos al método “vincular_evento”, lo cual hace exactamente lo mismo que

“reportarEvento”, es decir se agregan los participantes, tanto el que vincula a sus amigos como a sus amigos, al CSV.

A la hora de plantear el punto 3 se utilizaron las librerías pandas, csv (ya utilizada anteriormente) y seaborn (la cual al ser una librería que sobrescribe la de matplotlib, tuvo que ser importada). Para resolver el primer ítem se utilizó específicamente la librería heatmap de seaborn y database del tipo csv. Heatmap de seaborn permite representar de manera de gráfico rectangular el database o dataset pasado como argumento (el principal de los argumentos que admite la función heatmap, los demás). Del archivo csv que contienen las coordenadas, el tipo y la concurrencia del evento se utilizaron los datos de coordenadas y concurrencia para generar el mapa. Para seguir con el segundo ítem, se hizo uso especialmente de la librería pandas , la cual fue utilizada para filtrar y tabular la información de los eventos en las zonas y lograr visualizar los picos.