

UNIVERSIDAD
AUSTRAL



Universidad Austral
Ingeniería Biomédica

Introducción a la Programación II

EventIt

Alumnos:

Fernandez Josefina

Juarez Melina

Sipowicz Martina

Villar Martina

Vischi Felix

Grupo 7

30/11/2021

Introducción

Se realizó, a través de Python, un programa que ofrece el servicio de monitoreo de eventos para una ciudad. En el mismo, los usuarios (ciudadanos) podrán registrarse por única vez y tendrán acceso a un mapa virtual donde se señalarán los eventos que están ocurriendo en ese momento. Asimismo, los usuarios podrán compartir los eventos con otros, a quienes previamente deberán aceptar como contactos, por lo que el programa también incluye interacción entre usuarios.

El control de los usuarios es llevado a cabo por administradores, capaces de bloquearlos - una vez que las solicitudes del sujeto fueron rechazadas cinco veces -, desbloquearlos, crear otros administradores, editarlos, borrarlos. A su vez, son los encargados de habilitar distintas categorías, las cuales se les son asignadas a los eventos. Estos últimos son reportados tanto por usuarios como sensores, donde los primeros cuentan con la posibilidad de asistir al mismo junto con sus contactos.

No sólo es posible acceder a un mapa donde puede ver los sucesos que están ocurriendo, sino que también puede solicitar la visualización de un tablero de eventos. Mediante esta opción, se obtiene, a su vez, un ranking respecto al impacto del acontecimiento sobre la zona y la concurrencia de éste.

A continuación, se explicará brevemente la lógica detrás de la elaboración del programa. Además, se mencionará el funcionamiento de ciertas clases y métodos empleados.

De ciudadanos a usuarios

Todos los datos de los ciudadanos se encuentran en la “base de datos”¹ “datasetAnses.csv”. Esto significa que cada ciudadano posee un CUIL y un número celular asociado con el cual se los identifica. Cuando el ciudadano desea registrarse, el programa crea un ciudadano usuario, el cual contará con sus datos personales y se le sumará un nombre de usuario y una clave, que le permitirán el acceso al sistema. En caso de que los datos que se ingresen para lograr el registro no se encuentren en la base de datos, se levantará una excepción la cual evita que se frene el programa y permite que se vuelva a intentar.

Una vez ingresado en el sistema, puede acceder a las diferentes opciones de usuario. Las acciones que se pueden realizar son:

- Enviar solicitudes de “amistad” a otros usuarios del programa que se encuentren en la aplicación. Esto funciona ingresando el CUIL o celular del

¹ Para este trabajo se trabajó con archivos de textos para lograr una permanencia de datos.

ciudadano al que se desee añadir. Se decidió que la mejor forma de representar estos movimientos entre usuarios sería con listas (lista de solicitudes recibidas, lista con solicitudes aceptadas y un contador para las veces que el usuario fuera rechazado²), ya que son elementos de colección fáciles de acceder y de modificar.

- Revisar las solicitudes recibidas. El usuario podrá seleccionar cada una de ellas y decidir si aceptarla o no:
 - Aceptar solicitud. Quien recibe la solicitud es agregado a la lista de contactos de quien la envía y viceversa.
 - Denegar solicitud. Aumenta el contador de solicitudes rechazadas de quien envió la solicitud.
- Reportar evento. Mediante esta acción se crea un evento que contempla la categoría seleccionada por el usuario entre las habilitadas por el/los administrador/es, una pequeña oración que describe el acontecimiento, los involucrados y la hora en la cual fue reportado³.
- Ver contactos. El usuario puede acceder a la lista de contactos asociada.
- Ver mapa. El usuario puede acceder al mapa que contiene todos los eventos del día, diferenciados por colores de acuerdo a la concurrencia de los mismos.
- Ver tablero de estadísticas de eventos. Se imprime en pantalla una “tabla” que cuenta con un registro de los eventos creados junto con dos rankings: de impacto (de acuerdo a la ubicación del evento) y de concurrencia (de acuerdo a la cantidad de personas involucradas).

Administradores

Los administradores son quienes poseen mayor control sobre el programa y, a diferencia de los ciudadanos, no atraviesan un proceso de registro. Esto se debe a que un administrador sólo puede ser creado por medio de otro, razón por la cual existe en el programa un administrador por defecto (*Nombre de usuario: Admin. Clave: 000*).

Para ingresar al programa, el administrador debe ingresar una clave y un nombre. Esto permite identificar los diferentes administradores de una manera más sencilla. En el momento en que son instanciados, se agregan a una lista destinada a los mismos permitiendo, luego,

² De esta manera, el contador registrará cuando el usuario alcance las 5 solicitudes rechazadas, habilitando un bloqueo.

³ Para esta etapa fue crucial el uso del módulo `datetime` y el método `now()`.

que puedan ser creados, modificados o eliminados por cualquier otro administrador. Si no se encuentra el administrador deseado en la lista de administradores a la hora de intentar eliminarlo o editarlo, se lanzará una excepción para que el programa siga corriendo. Lo mismo ocurre si se desea crear un administrador con el mismo nombre y clave de usuario.

Sus funciones dentro del programa son:

- Bloquear usuarios. Se despliega una lista de ciudadanos usuarios que cumplen con el requisito para ser bloqueados. El administrador puede seleccionar a uno de ellos y bloquearlo.
- Desbloquear usuarios. Se despliega una lista de ciudadanos usuarios que se encuentran bloqueados. El administrador puede seleccionar a uno de ellos y desbloquearlo.
- Crear administrador. Se ingresa el nombre de usuario que se le asignará al nuevo administrador, junto con una clave, que se utilizan para instanciar un Administrador y ser agregado a la lista de administradores usuarios.
- Borrar administrador. Al desplegar la lista de administradores usuarios, se selecciona aquel que desea eliminarse y es removido de la lista de administradores usuarios, al igual que en el registro.
- Editar administrador. Al desplegar la lista de administradores usuarios, se selecciona aquel que desea editarse, se ingresan el nuevo usuario y la nueva clave que se le asignarán y se reemplazan por los valores anteriores. Esta modificación también es considerada en el registro de administradores.
- Crear categoría de evento. El administrador ingresa el nombre de una posible categoría o tipo de evento y luego es instanciado un tipo de evento, agregado a la lista de tipos de eventos.
- Ver mapa. Mismo propósito que en el usuario ciudadano.
- Ver tablero de estadísticas de eventos. Mismo propósito que en el usuario ciudadano.

Sensor

El programa también cuenta con sensores, considerados como dispositivos cuya única función es reportar un evento. Cada sensor tiene asociado un tipo de evento y un par de coordenadas, correspondientes a la ubicación del mismo. En el instante en que es reportado el

evento por el sensor, se registra al mismo con la categoría asignada, la localización del dispositivo y la hora en que es reportado. Cada sensor creado se añade a una lista de sensores.

Evento

Esta clase sirve para crear e instanciar eventos con los datos del reporte efectuado por un ciudadano o sensor. Por cada evento hay un contador al cual se le suma la cantidad de personas que confirman la asistencia al mismo y, cada vez que se crea un nuevo evento, este se añade a la lista de eventos la cual se almacena en la clase Login. Tanto las coordenadas como el índice de asistencia se reservan para la formación del mapa.

El propósito detrás de la implementación del contador es la posibilidad de elaborar el indicador de calor de ocurrencia de eventos (implementado mediante un barra gráfica). Al confirmar la asistencia al evento, el contador aumenta un valor. En caso de tener que compartir el evento, el contador de asistencia aumenta varias unidades dependiendo de las personas que asistan⁴.

Zonas

Para la representación de las zonas de la ciudad se implementó una clase “Barrio”. Mediante ella, se asignan las coordenadas en el mapa donde empieza y termina una zona, logrando así delimitar sectores del mapa lo más definidos posible. Gracias a estas determinaciones, fue posible identificar en qué zona se encuentran los eventos creados, lo cual fue esencial para crear el ranking de impacto.

Es importante mencionar que cada ciudadano tiene asignado, dentro de la dataset de Anses, el barrio al cual pertenece. En este también se indican las coordenadas del mismo.

Sistema de monitoreo

Se decidió implementar una clase en donde se encontrarán los métodos que permitieran “organizar” los eventos creados y delimitar las zonas correspondientes a la ciudad. Métodos dentro de Sistema de monitoreo:

- Ordenar eventos de barrio. Se indica el barrio cuyos eventos desean enlistarse y, luego, aquellos eventos con coordenadas comprendidas dentro de las que conforman la zona se agregan a una lista, que es retornada por el método.

⁴ Desarrollado dentro del método *reportarEvento()* en *CiudadanoUsuario*.

- Zonas delimitadas. Se enlistan las zonas/sectores de la ciudad, instanciando barrios mediante la información brindada por los usuarios en la dataset de Anses, donde cada elemento de esta lista se encuentra formado por otra que contiene al barrio instanciado y la cantidad de habitantes.
- Eventos más impactantes. Haciendo uso de los métodos anteriores, se enlistan ordenadamente de mayor a menor impacto los eventos, según la cantidad de personas a quienes impactan. El evento más impactante será aquel que se encuentre en la zona con mayor población.
- Mapeo de eventos. Crea y muestra un mapa⁵ que contiene los eventos del día y un indicador de calor el cual referencia los colores de los puntos - que representan a los distintos eventos- que corresponden al porcentaje de concurrencia⁶. Para su formación se empleó un rectángulo dividido en 4 zonas - equivalente a la cantidad de barrios- y una barra gráfica vertical dividida en porcentajes.

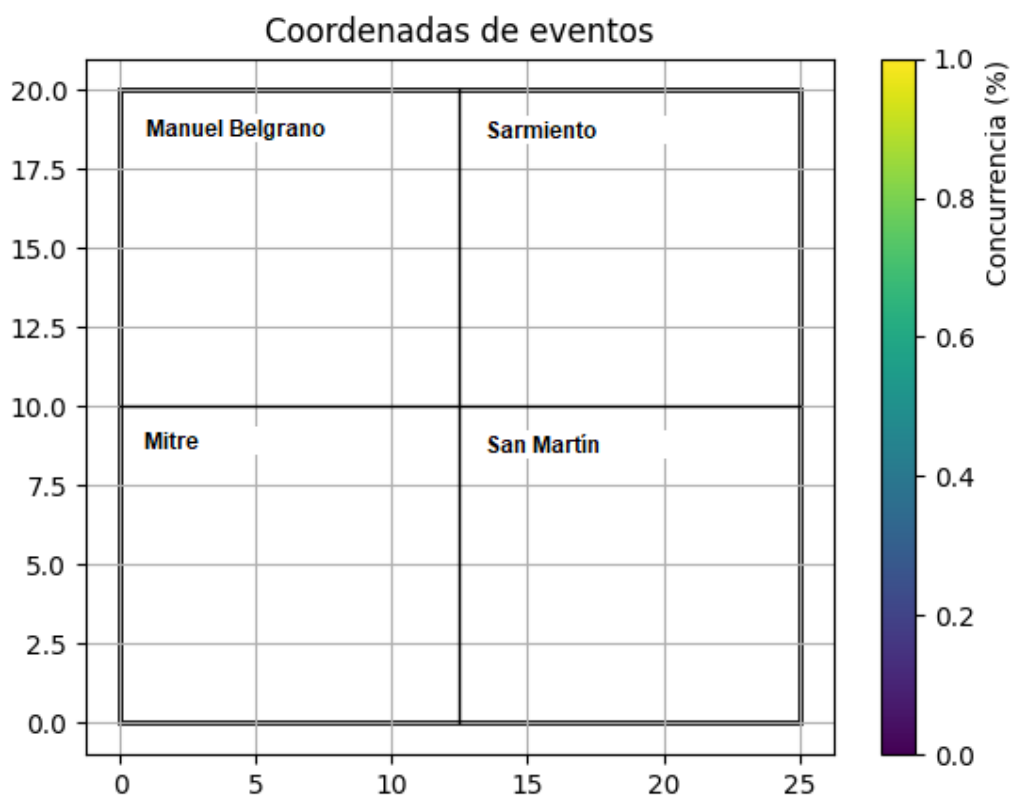


Gráfico correspondiente al mapa de la ciudad, delimitado por sus barrios.

⁵ Realizado con la biblioteca matplotlib y las sugerencias del profesor Rodrigo Peralta.

⁶ El porcentaje corresponde a la cantidad de personas que asistieron al evento sobre la cantidad de usuarios ciudadanos registrados.

InfoBase

La implementación del presente módulo fue la solución a los *circular imports* ocasionados entre los diversos módulos y LogIn. Dentro de la misma, encontramos listas que contienen a los objetos que se instancian a medida que se corre el programa.

Módulo Error

El propósito de este módulo es bastante sencillo. Para una mejor organización, se decidió agrupar todas las clases derivadas de Exception correspondientes a los distintos errores contemplados a lo largo del programa.

LogIn

Con la finalidad de elaborar la parte de interfaces y unificar todas las clases, se creó el módulo LogIn. En el mismo es posible encontrar las diversas listas que contienen a los objetos del programa y que acceden a la información en el InfoBase. Recordar que fue la solución a los *circular imports*.

La clase Login es aquella que se halla constantemente en contacto con el usuario, es la forma de acceder a todas las funciones y operaciones del sistema. Mediante el uso de inputs, el usuario puede avanzar en el programa, accediendo a las distintas funcionalidades de forma ordenada. Cuando se llama a un método, se comunica inmediatamente con el siguiente, como si estos se encontrasen unidos por un hilo.

Desde el primer momento el LogIn presenta opciones a seguir al usuario, el cual tiene la posibilidad de elegir mediante el marcado de números o letras señaladas por el programa. De esta manera, uno es capaz de acceder a la aplicación en forma de administrador, usuario ciudadano o sensor, y, a partir de la elección, se despliegan una serie de opciones a seguir según lo seleccionado.

Manejo de archivos de texto

Se utilizaron archivos de tipo CSV (valores separados por comas), el cual permite el almacenamiento de datos en un formato accesible. Los archivos creados fueron:

- “Administradores.csv”: almacena los nombres y claves de usuarios administradores. Contiene, inicialmente, la información del Admin inicial. A medida que se van creando, editando y eliminando otros administradores, el archivo es modificado.

- “Categorías.csv”: almacena las categorías de los eventos que son avaladas por los administradores.
- “Tablero_estadistica.csv”: almacena los datos del ranking de impacto, ranking de concurrencia, detalle del evento, tipo de evento, sus coordenadas, horario y concurrencia. Estos datos luego se usan para imprimir una “tabla” una vez solicitada la opción “Ver tablero de estadísticas de eventos”.
- “Tablero_eventos.csv”: almacena el detalle del evento, tipo de evento, sus coordenadas, horario y concurrencia.
- “Usuarios.csv”: almacena los datos de usuarios ciudadanos, a partir del registro de ciudadanos, con la clave y nombre de usuario elegido, número de celular, número de CUIL, y su estado, correspondiente a si se encuentra bloqueado o no. Por defecto, se encuentran como desbloqueados, pero cuando un usuario ciudadano es bloqueado por un administrador, el estado cambia a bloqueado y en el momento de iniciar sesión esto es notificado y es enviado a la opción inicial de inicio de sesión, por si desea cambiar de cuenta, dado que no podrá acceder a las opciones de usuario.
- “DatasetAnses.csv”: es un archivo de texto donde se encuentran los nombres de cada ciudadano acompañados del CUIL, el celular, su barrio de residencia, y las coordenadas delimitadoras de dicho barrio. La información de cada ciudadano que desee registrarse en el programa debe coincidir con la información presentada en esta base de datos.

Conclusión

A lo largo del trabajo se lograron desarrollar distintas habilidades. Se logró aprender acerca del manejo de archivos de texto, la implementación de un diseño óptimo, la escucha activa acompañada de un debate fructífero, la capacidad de búsqueda de información, entre otras cosas.

El desarrollo del trabajo fue un desafío para el grupo en conjunto que logró su resolución óptima por los aportes individuales y las puestas en común de los fuertes de cada uno. La solución final logró enorgullecer a todo el grupo, al lograr vincular las clases en un programa de interfaz por consola, pudiendo cumplir con todas las consignas pedidas en tiempo y forma.