

1º Cuatrimestre 2019

El objetivo del trabajo práctico es realizar simulaciones del sistema BAEcoBici disponible en la ciudad de Buenos Aires que permite a los ciudadanos registrados hacer uso de bicicletas para desplazarse entre distintos puntos de la ciudad.

El problema cuenta con las siguientes entidades:

- **Estación de bicicletas**

- Identificador: numérico empezando en 1
- Dirección: string
- Latitud y longitud: números con decimales, pueden ser negativos
- Capacidad: entre 15 a 30 anclajes
- Bicicletas: arreglo de bicicletas ancladas, los anclajes comienzan en 1

- **Bicicletas**

- Identificador: numérico empezando por el número 1000
- Estado: En condiciones, Necesita reparación
- Ubicación: Anclada en estación, en circulación, en reparación

- **Usuario**

- DNI: numérico entre 7 y 8 dígitos
- PIN: numérico de 4 dígitos, en blanco si el usuario está bloqueado
- Nombre y apellido: string
- Celular: string de dígitos y puede permitir los siguientes caracteres: (,), + y -. Validar un mínimo de 8 dígitos numéricos.

- **Viajes en curso**

- DNI del usuario
- Identificador de la bicicleta
- Estación de salida
- Hora de salida (string en formato hhmmss)

- **Viajes finalizados:** análogo a la anterior, agregando:

- Estación de llegada
- Hora de llegada

La simulación se hará con viajes dentro de un mismo día.

Se requiere:

1) Generar una función de carga de datos automática que al invocarse cree 10 estaciones, 250 bicicletas y 5 usuarios, ubicando las bicicletas ancladas en estaciones predefinidas (ejemplo: 15 en la estación 1, 25 en la estación 2, 0 en la estación 3, etc.). 10 de ellas deben estar en reparación. Los datos de estas estaciones, bicicletas y usuarios tienen que estar en el código fuente y desde allí cargarse en estructuras en memoria cada vez que esta función es invocada. Múltiples invocaciones borran todo y empiezan de nuevo.

- 2) Generar otra función que haga lo mismo que el ítem anterior, pero distribuya las bicicletas en las estaciones de forma aleatoria hasta agotar las 250 y sin sobrepasar la cantidad de anclajes por estación.
- 3) Generar una función que permita dar de alta a un nuevo usuario. Validar todos los datos ingresados de acuerdo a lo especificado en la sección de entidades. Bonus: permitir modificar a un usuario existente.
- 4) Permitir que un usuario se ingrese al sistema con DNI y PIN, y pueda modificarlo (solicitar ingresarlo 2 veces para evitar errores de tipeo)
- 5) Permitir desbloquear a un usuario listando los usuarios bloqueados del sistema, permitir elegirlo y generar un PIN aleatorio que se mostrará por pantalla. Para esto, se requiere el ingreso de la palabra secreta "shimano".
- 6) Bajo la opción de "Retirar Bicicleta" permitir que un usuario:
 - a. Seleccione un número de estación
 - b. Ingrese su DNI y PIN (3 reintentos en caso de error y cancela la operación, bloqueando al usuario)
 - c. El sistema automáticamente le asignará una bicicleta sana y le indicará al usuario el número de anclaje donde tiene que retirar su bicicleta
- 7) Permitir que un usuario ingrese su DNI y número de estación donde está devolviendo una bicicleta, y registrar esta acción. La duración del viaje será una cantidad de minutos aleatoria entre 5 y 75 minutos. Aquellos usuarios que sobrepasan la hora son bloqueados. Permitir que el usuario que acaba de devolver una bicicleta indique que necesita reparación antes de finalizar la operación.
- 8) Generar una función que
 - a. Seleccione un usuario que no esté actualmente en viaje ni bloqueado y una estación al azar
 - b. Intente retirar una bicicleta para el usuario
 - c. Genere una duración aleatoria de viaje del usuario menor a 90 minutos
 - d. Devuelva la bicicleta en otra estación (no la misma)
 - e. Vaya informando de estas acciones por pantalla, por ejemplo:
 - i. Juan Pérez retiró la bicicleta 1022 de la estación 3 de Las Heras 3255 a las 12:35:25 h.
 - ii. Juan Pérez devolvió la bicicleta 1022 en la estación 15 de Gallo 224 a las 13:30:12 h. Al exceder los 60 minutos de uso ha sido bloqueado.
- 9) Generar otra función que permita realizar lo mismo que en el punto anterior, pero preguntando cuántas veces repetir la operación entre 1 y 100.
- 10) Se solicitan los siguientes informes:
 - a. Top 10 de usuarios de bicicleta, por cantidad de viajes
 - b. Top 5 de usuarios de bicicleta, por duración acumulada de los viajes
 - c. Listado de estaciones que tienen bicicletas que necesitan reparación, incluyendo esta cantidad
 - d. Top de estaciones más activas: se entiende por "actividad" a la extracción y la devolución de bicicletas

Estructuras de datos

Utilizar arreglos de diccionarios basándose en las entidades descritas anteriormente. Pueden agregarse campos a estas entidades para contabilizar información relevante. Se sugiere que el grupo genere un diagrama con estas entidades y lo presente al ayudante asignado en una etapa temprana del desarrollo.

Menú

El siguiente es un menú para el programa. Se aceptarán variaciones a criterio del grupo, a validar con cada ayudante:

1. Carga de datos
 - a. Carga automática
 - b. Carga automática aleatoria
2. Usuarios
 - a. Listado
 - b. Alta
 - c. Modificación
 - d. Desbloquear
3. Retiros automáticos
 - a. Viaje aleatorio
 - b. Viajes aleatorios múltiples
4. Informes
 - a. Usuarios con mayor cantidad de viajes
 - b. Usuarios con mayor duración acumulada de viajes
 - c. Bicicletas en reparación
 - d. Estaciones más activas
5. Uso del sistema

El ingreso al sistema muestra un nuevo menú:

1. Modificar PIN
2. Retirar Bicicleta
3. Devolver Bicicleta

Modalidad de entrega

Entregar a través del campus el código fuente con comentarios en cada función (pre y post condiciones). El archivo debe estar comprimido en un zip que lleve el formato “\${nombreGrupo} - TP1” (deben reemplazar \${nombreGrupo} por el nombre de su grupo). Dentro del zip debe haber un archivo TXT donde se debe incluir nombre, apellido y padrón de los miembros del equipo. También se deben realizar todas las aclaraciones que a su criterio sean necesarias, explicando la funcionalidad del código cuando sea relevante. Validar con cada ayudante asignado si requerirán además una versión impresa del código fuente.

Fecha de entrega

El límite para la entrega a través del campus es el jueves 23 de mayo de 2019 a las 22 hs.