



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN  
IIC2233 - PROGRAMACIÓN AVANZADA

# Actividad 03

2º semestre 2017  
24 de Agosto de 2017

## Estructuras de Datos: Built-ins

### Introducción

¡PrograBanner está en problemas! Después de una difícil toma de ramos, los genios detrás de este sistema se encuentran en apuros, luego de que un virus creado por el **El Rico y Mafioso Nebil** borrara toda la toma de ramos y el sistema para siempre! D:. En su desesperación, el equipo pide ayuda a ustedes, futuros programadores, arreglar parte de este tremendo problema.

### PrograBanner

El nuevo sistema PrograBanner debe consistir de dos etapas de toma de ramos. En la primera etapa cada estudiante puede tomar hasta 3 ramos distintos, sin repetir. En la segunda etapa, cada uno puede tomar ramos hasta completar un total de 5 cursos distintos como máximo. Entre las dos etapas existe un ajuste de vacantes en que el sistema otorga vacantes adicionales a algunos cursos. Luego de la segunda etapa, se revisan los horarios de los alumnos de las unidades académicas que controlan tope de horario y se les botan los ramos con tope.

Para lograr levantar este sistema deben considerar las siguientes entidades relevantes:

- **PrograBanner:** Sistema que mantiene la oferta de cursos y a los alumnos que esperan para tomar ramos, quienes son atendidos según orden de llegada. Asigna y ajusta las vacantes de los cursos y además controla el tope de horario una vez terminada la toma de ramos.
- **Curso:** Cada curso tiene sigla, nombre, un horario predefinido y una cantidad limitada de cupos que además están numerados de acuerdo al orden en que se van asignando. La numeración permite a la universidad rastrear las vacantes. Como ejemplo, si el alumno 15436789 tiene el cupo con número de cupo 3 de IIC2233, significa que fue el tercer estudiante que tomó un cupo de Avanci3n Programada.
- **Alumno:** Posee número de alumno, unidad académica y un registro de los cupos asignados. Este registro debe tener los cupos clasificados de acuerdo a la sigla del curso para facilitar posibles consultas. El alumno no puede tomar dos veces el mismo ramo y tiene un límite de 5 cursos.
- **Cupos:** Cada cupo posee un número de cupo y el horario de la sección a la que pertenece.

### Asignación de cupos

El sistema escoge aleatoriamente los cursos para el siguiente alumno en espera y revisa si estos tienen un cupo disponible. En tal caso se le asigna la vacante del curso al alumno y esta deja de estar disponible

para futuros alumnos. Si no existen cupos disponibles en un curso, simplemente el alumno **quedará sin ese curso**. En la primera etapa de la toma de ramos este proceso se ejecuta con 3 ramos por alumno y en la segunda, con suficientes ramos para que el alumno quede con hasta 5 cursos asignados.

## Ajuste de vacantes

Entre las dos etapas de toma de ramos, el sistema crea nuevas vacantes a ciertos cursos. Por ejemplo, si a un curso con 10 vacantes originales se deben agregar 2 nuevos cupos, estos tendrán la numeración 11 y 12 respectivamente, e independiente de la cantidad de cupos restantes en el curso al término de la primera etapa de toma de ramos.

## Tope de horarios

Una vez finalizadas las dos etapas de toma de ramos, se controlará el tope de horario para los alumnos de las unidades académicas que lo controlan. Si un alumno de tales unidades tiene dos o más ramos con el mismo horario, el sistema le eliminará el cupo de todos ellos, es decir, el alumno ya no tendrá dichos cursos en su registro de ramos y **las vacantes serán restituidas en el registro de cupos de los ramos**.

## Archivos

Se les entregan tres archivos de texto para poblar su sistema. En `cursos.txt` se encuentran las asignaturas con sus respectivos cupos y capacidad de cupos extra, además del horario asignado, de la forma `sigla,cupos,cupos_extra,horario`. En `alumnos.txt` se encuentran los datos de los estudiantes y sus unidades académicas, de la forma `numero_alumno,unidad_academica`. En `unidades.txt` se encuentra el detalle de las unidades académicas que controlan tope de horario, de la forma `unidad_academica,control` donde `control` es 1 cuando se controla el tope y 0 cuando no.

## Consultas al sistema

Para probar su sistema se pide que implementen los siguientes métodos a `PrograBanner` y que entreguen un ejemplo para cada una.

- `alumno_en_curso(num_alumno, sigla)`: Imprime en consola el número del cupo del alumno `num_alumno` en el curso `sigla` o, en su defecto, un mensaje de que el alumno no está en el curso.
- `alumnos_en_curso(sigla)`: Imprime en consola un listado de los alumnos inscritos en un curso, indicando su número de alumno y el número del cupo asignado.
- `cursos_comunes(num_alumno_1, num_alumno_2)`: Imprime en consola la sigla y el horario de todos los ramos comunes de los estudiantes con números de alumno dados. Pueden ser cupos de horarios distintos para el mismo ramo.

A continuación se muestra un ejemplo de output de los tres métodos requeridos.

```
13202388 no tiene cupo en EDP8575

Alumnos inscritos en EDP8575
15621945 - cupo 5
14631681 - cupo 7
13631277 - cupo 1
15201814 - cupo 6
13201433 - cupo 4

Cursos comunes de 13201645 y 13202388:
IIC9145
EDP8575
```

## Notas

- Para esta actividad deben escoger las estructuras de datos de python que sean más adecuadas para almacenar la información del sistema en sus diferentes partes y funciones. **¡Se penalizará si se utilizan listas para todo!**
- La utilización de `globals()` esta prohibido y será fuertemente penalizado.
- La selección de ramos es completamente aleatoria. Para tomar `k` elementos de una lista o tupla `c`, se les recomienda el uso de `random.sample(c,k)` de la librería `random`.
- Los alumnos se ubican de forma aleatoria para esperar su turno en la toma de ramos. Este orden es distinto para cada etapa de toma de ramos.
- Solo hay dos posibilidades de horarios, el `LWX` y `MJX` donde `X` puede ser un número del 1 al 6 simbolizando el módulo. (Así es, en esta universidad, todos los fines de semana son largos :D)
- Es posible que un alumno quede sin ningún curso.
- Al restituir cupos a los cursos en la etapa de tope de horario, no importa que estos queden ordenados de acuerdo a su número de cupo. (Pueden estar desordenados)
- Ejemplo de una opción como podría leerse un archivo:  

```
with open(filename, 'r') as f:
    for line in f:
        # line sera el string de la linea que estan leyendo de su archivo.
```

## Requerimientos

- (5.00 pts) **PrograBanner**
  - (0.40 pts) Se eligió la estructura de datos (EDD) adecuada para el almacenamiento de alumnos por orden de llegada.
  - (0.60 pts) Se implementó de manera adecuada la EDD para el almacenamiento de los alumnos.
  - (0.40 pts) Se eligió la EDD adecuada para almacenar los cupos en los cursos.
  - (0.60 pts) Correcta implementación del almacenamiento de cupos de acuerdo a numeración.
  - (1.00 pts) Correcto modelamiento y uso de estructuras de datos para guardar las unidades académicas.
  - (1.00 pts) Correcta asignación de cupos: Asignar al alumno y eliminar el cupo del curso (que deje de estar disponible).

- (1.00 pts) Correcta implementación en el control de topes de horario: eliminación de los cupos con tope en el registro del alumno y devolución del cupo al curso.
- (1.00 pts) **Consultas**
  - (0.20 pts) `alumno_en_curso()` correctamente implementada.
  - (0.20 pts) `alumnos_en_curso()` correctamente implementada.
  - (0.60 pts) `cursos_comunes()` correctamente implementada con la EDD adecuada.

## Entrega

- **Lugar:** En su repositorio de Github en la **carpeta** `Actividades/AC03/`
- **Hora:** 16:55
- Si está trabajando en pareja, basta con que un miembro suba la actividad. Si se suben actividades distintas, se corregirá una de las dos al azar.