



## Programación—Certamen Recuperativo (CV) - Jueves 10 de Julio de 2014

Nombre:

Rol:

2. [35%] Pytopia, una nueva empresa de desarrollo de software, necesita ayuda para coordinar las agendas de sus empleados. Para ello, se cuenta con la información de todas las actividades en que participan los empleados de Pytopia en una lista de tuplas, donde cada tupla tiene la siguiente estructura: persona, nombre del evento, lugar, día y una tupla con hora de inicio y fin de la actividad. Más de una persona puede participar en una actividad, por ejemplo, Pedro y Diego participarán en una reunión de coordinación en la Sala de Reunión el 1 de Abril, 09:00 - 10:00. Asuma que las actividades están bien definidas, o sea, para una misma persona, no hay traslapo de horario entre distintas actividades.

```
actividades = [  
#(persona, evento, lugar, dia, (hora inicio, hora fin))  
( 'Juan', 'Reunion con cliente', 'Falaferia', (2014, 4, 15), ('14:00', '15:30')),  
( 'Diego', 'Reunion de coordinacion', 'Sala de Reunion', (2014, 4, 1), ('09:00', '10:00')),  
( 'Juan', 'Inicio de actividades', 'SII', (2014, 3, 3), ('10:00', '10:30')),  
( 'Pedro', 'Reunion de coordinacion', 'Sala de Reunion', (2014, 4, 1), ('09:00', '10:00')),  
( 'Pedro', 'Reunion con abogado', 'Oficina abogados', (2014, 3, 14), ('12:30', '13:30')),  
( 'Juan', 'Entrevista con programador', 'Sala Reunion', (2014, 4, 15), ('11:30', '12:30')),  
( 'Diego', 'Entrega prototipo 1', 'Falaferia', (2014, 5, 14), ('11:00', '12:00')),  
( 'Pedro', 'Reunion con cliente', 'Falaferia', (2014, 4, 15), ('14:00', '15:30')),  
# ...  
]
```

- a) Desarrolle la función `disponible(persona, fecha, hora)`, la cual recibe como parámetro el nombre de una persona, una tupla (año, mes, día) y una hora en formato `hh:mm` (24 hrs), e indique si esa persona esta o no disponible en ese horario.

```
>>> disponible('Juan', (2014, 4, 15), '11:45')  
False  
>>> disponible('Juan', (2014, 4, 15), '16:00')  
True
```

- b) Desarrolle la función `agenda_dia(persona, fecha)`, la cual recibe como parámetro el nombre de una persona y una fecha en formato tupla (año, mes, día). Esta función debe retornar una lista de tuplas con las actividades agendadas por la persona para la fecha dada. La estructura de la tupla a retornar es: `((hora inicio, hora fin), nombre de actividad, lugar)`. Las actividades deben estar ordenadas por hora de inicio.

```
>>> agenda_dia('Juan', (2014, 4, 15))  
[(('11:30', '12:30'), 'Entrevista con programador', 'Sala Reunion'),  
(('14:00', '15:30'), 'Reunion con cliente', 'Falaferia')]
```

- c) Desarrolle la función `verificar_agenda(persona1, persona2, fecha)` la cual recibe como parámetro 2 string correspondientes al nombre de personas y una fecha en formato tupla (año, mes, día). Esta función retorna **True** si las personas tienen algún evento en común o **False** en caso contrario.

```
>>> verificar_agenda('Juan', 'Pedro', (2014, 4, 15))  
True
```

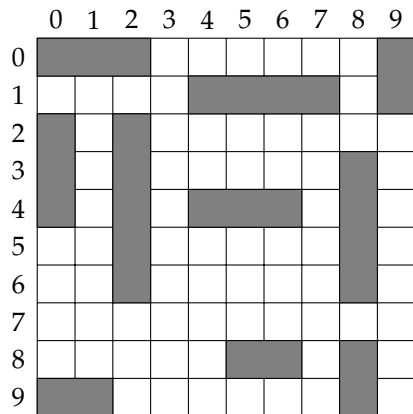
# Programación—Certamen Recuperativo (CV) - Jueves 10 de Julio de 2014

Nombre:

Rol: -

3. [40%] El *Combate Naval* es un juego de papel y lápiz en que cada jugador posiciona una flota de barcos en una grilla de  $10 \times 10$ . La flota consta de portaaviones, acorazados, submarinos y cruceros. Los barcos pueden ponerse horizontal o verticalmente y no deben traslaparse.

En la siguiente figura vemos un ejemplo de tablero que podría llenar un jugador, a su derecha cuál es el formato de archivo que usaremos para representarlo y finalmente un diccionario que especifica el tamaño de cada barco.



a.txt

```

Portaaviones 2 2 V
Acorazado 3 8 V
Acorazado 1 4 H
Submarino 2 0 V
Submarino 0 0 H
Submarino 4 4 H
Crucero 0 9 V
Crucero 9 0 H
Crucero 8 5 H
Crucero 8 8 V
    
```

```

barcos = {
    'Portaaviones': 5,
    'Acorazados': 4,
    'Submarinos': 3,
    'Cruceros': 2,
}
    
```

Cada línea del archivo representa un barco de la flota, y contiene los siguientes datos: tipo de barco, fila de su primera celda, columna de su primera celda y dirección del barco (V si es vertical y H si es horizontal).

- a) Escriba la función `obtener_celdas_barco(linea, barcos)` que reciba como parámetro una línea del archivo de la flota y el diccionario de tamaños. La función retorna un conjunto con las celdas que ocupa ese barco:

```

>>> obtener_celdas_barco('Submarino 0 0 H', barcos)
set([(0, 0), (0, 1), (0, 2)])
>>> obtener_celdas_barco('Acorazado 3 8 V', barcos)
set([(3, 8), (4, 8), (5, 8), (6, 8)])
    
```

- b) Escriba la función `validar_flota(nombre_archivo, barcos)` que indique si los barcos de la flota contenida en el archivo no se traslapan (es decir, si no hay varios barcos compartiendo una misma celda):

```

>>> validar_flota('a.txt', barcos)
True
    
```

- c) Escriba la función `crear_tablero(nombre_archivo, barcos)` que cree un nuevo archivo con un mapa del tablero, siguiendo el formato ilustrado en la figura de la derecha:

```

>>> crear_tablero('a.txt', barcos)
>>>
    
```

