**Recursos útiles:**

* Archivo **TutorialPython3.pdf**.Página 33

**Actividades**:

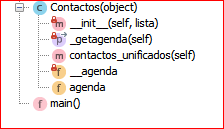
1. A partir de una lista de personas con sus e-mail, mostrar para cada una de ellas todos sus e-mails

lISTA ORIGINAL: [('Ana', 'correo1@ana.es', 'correo2@ana.es'), ('Maria', 'correo1@maria.es'), ('Iker', 'correo1@iker.es'), ('Ana', 'correo3@ana.es', 'correo4@ana.es'), ('Iker', "correo'2@iker.es"), ('Ana', ['correo5@ana.es](mailto:'correo5@ana.es)')]

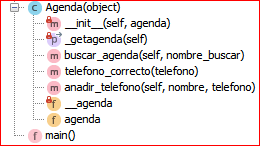
Maria [('correo1@maria.es',)]

Iker [('correo1@iker.es',), ("correo'2@iker.es",)]

Ana [('correo1@ana.es', 'correo2@ana.es'), ('correo3@ana.es', 'correo4@ana.es'), (['correo5@ana.es](mailto:'correo5@ana.es)',)]

Para ello utiliza una clase Contactos cuyo único atributo será la lista (como atributo oculto):

* **\_\_agenda**: atributo oculto o privado.
* **agenda**: es propiedad de sólo lectura.
* método **contactos\_unificados :** devuelve un diccionario en el que la clave es el nombre de una persona y el valor es una lista con todos sus teléfonos.

1. **Continuación de agenda simplificada**. Realiza la siguiente clase:

* **\_\_agenda:** diccionario que tiene como clave nombre y como valor el teléfono de una persona.
* **telefono\_correcto**: es un método estático que devuelve True si el teléfono es correcto o un error de ejecución si no lo es. Correcto si todos sus dígitos son número y tiene longitud 9 dígitos.
* **buscar\_agenda**: método que devuelve el teléfono de una persona.
* **anadir\_telefono**:método quemodifica el **telefono** deuna persona, **nombre**

El programa que gestiona la agenda debe:

* Solicitar un nombre. Si no existe pedir su número de teléfono (validar con telefono\_correcto). Si existe mostrar su número de teléfono y pedir el nuevo número (validarlo). Modificar la agenda si todo ha ido correctamente.

{'Adela': '444444444', 'Pedro': '222222222', 'Bertoluchi': '777777777', 'Paco': '555555555', 'Berta': '666666666', 'Alba': '333333333', 'Ana': '111111111'}

Nombre a buscar: *Pedro*

Pedro. Modificar 222222222 (Enter:no modificar:) *000000000*

Agenda actualizada

{'Adela': '444444444', '*Pedro': '000000000*', 'Bertoluchi': '777777777', 'Paco': '555555555', 'Berta': '666666666', 'Alba': '333333333', 'Ana': '111111111'}

{'Alba': '333333333', 'Berta': '666666666', 'Paco': '555555555', 'Bertoluchi': '777777777', 'Adela': '444444444', 'Ana': '111111111', 'Pedro': '222222222'}

Nombre a buscar: *Berta*

Berta. Modificar 666666666 (Enter:no modificar:) *999999999*

Agenda actualizada

{'Alba': '333333333', '*Berta': '999999999*', 'Paco': '555555555', 'Bertoluchi': '777777777', 'Adela': '444444444', 'Ana': '111111111', 'Pedro': '222222222'}

{'Adela': '444444444', *'Paco': '555555555',* 'Berta': '666666666', 'Alba': '333333333', 'Pedro': '222222222', 'Ana': '111111111', 'Bertoluchi': '777777777'}

Nombre a buscar: *Paco*

Paco. Modificar 555555555 (Enter:no modificar:) *gt666*

*Número erróneo*

{'Adela': '444444444', 'Paco': '555555555', 'Berta': '666666666', 'Alba': '333333333', 'Pedro': '222222222', 'Ana': '111111111', 'Bertoluchi': '777777777'}

1. Amplía el ejercicio de la **máquina expendedora** de forma que sea una máquina para diferentes artículos: café, agua, palomitas, aquarius,.....

De cada producto tendremos su precio y cantidad de unidades que hay en la máquina (en un diccionario **producto: (precio,cantidad), … }**. Esto será un atributo de clase de tipo diccionario.

El usuario tecleará un producto a comprar e irá introduciendo dinero hasta pagarlo. Saca el ticket y recoge la vueltas (como en el ejercicio original). A medida que compra productos se descontarán de su total de unidades.

Deberá controlar si quedan unidades del artículo solicitado.

Que sirva para más de un producto, hasta apagar la máquina, que en ese momento mostrará el total recaudado y las unidades que le quedan a cada artículo.

Decide qué atributos y métodos añadir o modificar.

<https://www.lawebdelprogramador.com/foros/Python/1629211-ayuda-diccionario.html>