

IES Milà I Fontanals

# Pràctica de documentació i testing amb Visual Studio 2015

Visual Studio, VSDocMan



## TESTING

I FIND YOUR LACK OF TESTS DISTURBING.

BERNAT  
17/01/2017

## PRÀCTICA DE DOCUMENTACIÓ I TESTING UNITARI

### PRIMERA PART: IMPLEMENTACIÓ

**IMPORTANT:** Les classes que crearem a continuació han d'estar al namespace **"practicaTestDoc.[elvostrenom]"**

Donada una classe **Persona**, amb les propietats següents:

- NIF
- Nom
- Cognom 1
- Cognom 2
- Data de naixement

Cal implementar-la seguint les següents pautes de validació:

- Constructor:
    - Una persona té obligatòriament nom i primer cognom, NIF i data de naixement. Altrament es llança una excepció.
- NOTA:** El nom i cognoms s'entenen buits tan si són una cadena buida, nul·la, o conté únicament espais en blanc.
- Validacions i accions en els setters de les propietats
    - El NIF s'ha de validar completament (lletra inclosa)
    - S'eliminaran espais en blanc al voltant de les cadenes del nom i cognoms.
    - El nom i el cognom 1 han de tenir obligatòriament 3 caràcters. Si s'intenta posar un valor erroni per aquests dos camps es llançarà una excepció.
    - La data de naixement serà qualsevol data vàlida no nul·la, posterior a l'any 1950. Altrament es llançarà excepció.

Farem una segona classe anomenada **Classe**, que consistirà en una llista de persones. La classe població tindrà els mètodes:

- *afegirPersona(p:Persona):boolean*, que admetrà qualsevol persona nova a la llista. La clau primària de persona és NIF. Si no pot afegir la persona, retorna fals.
- *getPersona(idx : int):Persona*, que ens retornarà una persona a un índex concret. Si l'índex és erroni, ens donarà una excepció de tipus *IndexOutOfRangeException*.
- *getNumPersones()*: retorna el nombre total de persones a la llista.
- *eliminarPersona(idx:int)* : elimina la persona a l'índex indicat. Llança una excepció si l'índex està fora de rang.

Es desitja també crear un formulari per visualitzar les persones dins de la **Classe**, que ens permeti també crear noves persones. El formulari tindrà la següent estructura:

NIF (\*)

Nom (\*)

Cognom 1 (\*)

Cognom 2

Data. Naix (\*)

Classe

Maria Irla  
Joan Gómez  
Pere Pau Vázquez  
Jaume Aregall

esborrar seleccionat

desar afegir

La validació de cada camp es fa en calent, els errors es mostren amb una icona d'error i un camp de text adjacent a cada control. A més, es canviarà el *background* a color vermell.

La validació de persones repetides a la classe es fa al desar. Si hi ha error, es mostra un text d'error sota la llista de classe.

## SEGONA PART: DOCUMENTACIÓ

Documenteu la classe persona de forma completa, tant a nivell de classe com de propietats, constructors, etc. Usant *VSDocMan*, cal generar documentació en forma de pàgina web.

Es demana que la documentació inclogui com a mínim:

- Una referència usant <see>
- Un exemple d'utilització d'un mètode, propietat o constructor, que continguin 2 fragments de codi en visual bàsic i c# a la documentació.
- Una <note>
- Una <list>
- Creeu la documentació del namespace, que contindrà una imatge amb el diagrama de classes general.

## TERCERA PART: PROVES

Posteriorment cal crear les proves de l'aplicació:

- 1) El testing unitari de la classe Persona, de forma que garanteixi el 100% de cobertura de la classe.
- 2) El testing de la interfície gràfica, considerant els casos següents:

- a) Cas 1: L'usuari afegeix una nova persona. Tot va bé, l'usuari introdueix tots els valors correctes i al prémer el botó desar, la persona s'afegeix a la llista.
- b) Cas 2: NIF erroni. Verificar que per un NIF erroni, apareix un error adjacent al textbox.
- c) Cas 3: L'usuari afegeix una nova persona, amb un NIF existent. Mostra error de "persona ja existent" al desar.
- d) Cas 4: L'usuari selecciona un persona de classe. El test verifica que es mostren totes les seves dades al formulari.