

Sistemes de gestió empresarial

# UD 05.(Semipresencial)

## Activitats avaluables 02

---



Autor: Sergi García Barea

Actualitzat Octubre 2021

## Llicència



**Reconeixement - No comercial - CompartirIgual (BY-NC-SA):** No es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original.

## Nomenclatura

Al llarg d'aquest tema s'utilitzaran diferents símbols per distingir elements importants dins del contingut. Aquests símbols són:

**Important**

**Atenció**

**Interessant**

## ÍNDEX DE CONTINGUT

<b>Data d'entrega</b>	<b>3</b>
<b>Observacions prèvies a la realització de la tasca avaluable</b>	<b>3</b>
<b>Citació demostració d'autoria dels exercicis i castic per còpia</b>	<b>3</b>
<b>Entrevista tècnica</b>	<b>4</b>
<b>Pregunta entrevista 01</b>	<b>4</b>
<b>Pregunta entrevista 02</b>	<b>4</b>
<b>Pregunta entrevista 03</b>	<b>5</b>
<b>Pregunta entrevista 04</b>	<b>5</b>
<b>Pregunta entrevista 05</b>	<b>5</b>
<b>Pregunta entrevista 06</b>	<b>6</b>
<b>Pregunta entrevista 07</b>	<b>7</b>
<b>Pregunta entrevista 08</b>	<b>8</b>
<b>Pregunta entrevista 09</b>	<b>8</b>
<b>Pregunta entrevista 10</b>	<b>9</b>
<b>Pregunta entrevista 11</b>	<b>9</b>
<b>Pregunta entrevista 12</b>	<b>10</b>
<b>Pregunta entrevista 13</b>	<b>10</b>
<b>Pregunta entrevista 14</b>	<b>10</b>
<b>Pregunta entrevista 15</b>	<b>11</b>

## UD05. ACTIVITATS AVALUABLES 02 (SEMIPRESENCIAL)

### 1. DATA D'ENTREGA

**Data límit d'entrega: Dissabte 20 de novembre a les 23:55.**

L'activitat serà avaluada quan haja passat la data límit d'entrega.

**! Atenció:** la data d'entrega no és prorrogable. Si no entregues en temps, la qualificació serà 0.

### 2. OBSERVACIONS PRÈVIES A LA REALITZACIÓ DE LA TASCA AVALUABLES

- Per a lliurar les imatges dels contenidors, has d'utilitzar el repositori de "Docker Hub".
  - Les imatges dels contenidors han de ser privades.
- Has d'utilitzar el repositori Github creat per lliurar treballs del mòdul. Recordeu que el repositori de Github:
  - És un lloc de treball. Es valorarà si s'ha utilitzat correctament (pujant versions i no actuant únicament com un lloc on lliurar la tasca).
  - És molt important que el repositori siga privat, per tal d'evitar una hipotètica còpia d'altres alumnes.
  - El repositori ha d'estar organitzat en carpetes i fitxers, amb noms descriptius tant en fitxers com de carpetes, de forma que siga fàcil navegar, no a soles per mi, sinó per un hipotètic contractador que veja el teu repositori com un portfolio.
  - El repositori és un portfolio i no "un lloc on entregar la pràctica". Faciliteu la lectura pujant el document en PDF mínim (a més, podeu pujar en ODT, DOCX, etc.). No fiqueu el document en un ".zip" o ".rar".
- Quan s'entreguen documents, aquests han de tindre una bona presentació. Encara que siga xicotet, ha de tindre portada, índex, capçalera, peu de pàgina (amb nombre de pàgina), a més de ser coherent en estil.
- Quan s'entregue fitxers amb codi han d'estar correctament formatats i abundantment comentats.
  - **Nota: podeu fer els comentaris dels programes a la llengua que vulgueu (Castellà, Valencià o Anglès).**
- Les parts de redactat de l'activitat han de fer-se en valencià. No serem molt estrictes en gramàtica i ortografia, però almenys heu d'intentar fer-ho bé.

**! Atenció:** no complir aquestes consideracions pot reduir la nota fins a 3 punts.

### 3. CITACIÓ DEMOSTRACIÓ D'AUTORIA DELS EXERCICIS I CASTIC PER CÒPIA

Per a considerar un exercici realitzat, no basta únicament amb lliurar el codi amb comentaris. L'alumnat ha de SER CAPAÇ d'entendre i defensar el seu exercici a petició del professor i SER CAPAÇ de realitzar modificacions relacionades amb el mateix, amb el fi de demostrar l'adquisició de coneixement i evitar qualsevol sospita de còpia.

**Es ficara una nota provisional, però posteriorment tot l'alumnat serà citat per a defensar en persona o videoconferència l'autoria completa d'aquest avaluable.**

Si l'alumnat no pot fer aquesta defensa i aquestes modificacions, es considerarà que ha copiat l'exercici. **La còpia es castiga amb el suspens del mòdul complet.**

## 4. ENTREVISTA TÈCNICA

Aquesta activitat consisteix en la preparació per una entrevista tècnica per un lloc de treball on ens demanen coneixements de Python 3.

Resolent aquestes preguntes, a més de preparar l'entrevista tècnica, estem preparant un portfolio. Aquestes preguntes no són només un element a resoldre, sinó que seran la nostra imatge de cara a l'empresa.

**! Atenció:** en aquesta activitat no volem acabar exercicis. Volem demostrar que sabem Python 3. Per aquest motiu **tot exercici sense abundants comentaris, serà qualificat amb un 0.**

**Els comentaris podran fer-se en valencià, castellà o anglès.**

### 4.1 PREGUNTA ENTREVISTA 01

Explica i posa un exemple senzill d'una funció lambda en Python 3.

Després, posa un exemple que utilitze les funcions de Python:

Utilitzant "map()" (i "list()" e "int()" com a recolzament per convertir a llista i a nombres sencers) heu de llegir de teclat una cadena de text formada per números separats amb espai i transformar-la en una llista formada per nombres sencers.

Utilitzant "filter()", elimina de la cadena anterior els números menors que 10.

Amb la cadena resultant i utilitzant "reduce()", retorna la multiplicació dels elements de la llista.

### 4.2 PREGUNTA ENTREVISTA 02

Realitza una aplicació que comprove si una solució d'un Sudoku és correcta o no. L'explicació dels regles del Sudoku els teniu ací <https://es.wikipedia.org/wiki/sudoku>

El Sudoku serà llegit d'un fitxer "Sudoku.in", format per 9 línies on cadascuna d'aquestes estarà formada per 9 números separats per espais. El programa ens mostrarà per consola i el resultat serà mostrat per consola si és correcte o no.


Internament el programa realitzarà la comprovació de si el Sudoku és o no correcte en una funció definida com "def esSudokuCorrecto(miArrayBi)" que retornarà true si és correcte, false en cas contrari.

#### Exemple de "Sudoku.in"

```
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
```

#### Exemple eixida:

Incorrecte


 **Consell 01:** intenta implementar les funcions demanades tu a soles. No copies la solució d'Internet ni dones una mirada a la d'un company, perquè a l'entrevista personal es demanaran modificacions i s'ha de comprendre que estan fent "esSudokuCorrecto".

### 4.3 PREGUNTA ENTREVISTA 03

Realitza una aplicació que sol·licite una cadena de text. El programa ha de dir quantes vegades ocorre cadascun d'aquests patrons sense distingir majúscules i minúscules: "00" "101", "ABC", "HO". Un caràcter pot formar part de més d'un patró oposat. Per exemple:

En la cadena "000" el patró "00" apareix dues vegades (una comença en la posició 0 i una altra comença en la posició 1).

Internament el programa realitzarà el compte de patrons amb una funció definida com "def numeroPatrones(text)" que retornarà un nombre enter amb el nombre de patrons oposats.

 **Consell 01:** intenta implementar les funcions demanades tu a soles. No copies la solució d'Internet ni dones una mirada a la d'un company, perquè a l'entrevista personal es demanaran modificacions i s'ha de comprendre que estan fent "numeroPatrones".

### 4.4 PREGUNTA ENTREVISTA 04

Realitza una aplicació que reba en codi mitjançant un array bidimensional (de longitud variable) un escenari de "Buscaminas", on hi haja un 0 on no hi ha mines i un -1 on si hi ha. Per a cada casella que no tinga una mina, diga quantes mines adjacents hi ha (en diagonal, horitzontal i vertical).

Internament el programa realitzarà les accions amb una funció definida com a "def contandoMinas(miCampo)" que retornarà un array bidimensional amb el nombre de mines adjacents en cada posició.

#### Exemple Entrada (llegit de teclat)


0 0 -1 0

0 -1 -1 0

#### Exemple Eixida (mostrat per pantalla)

1 3 -1 2

1 -1 -1 2

 **Consell 01:** intenta implementar les funcions demanades tu a soles. No copies la solució d'Internet ni dones una mirada a la d'un company, perquè a l'entrevista personal es demanaran modificacions i s'ha de comprendre que estan fent "contandoMinas".

### 4.5 PREGUNTA ENTREVISTA 05

Dissenya la classe "Escola" per a un programa que permeti la gestió completa de diverses escoles. Per cada escola, s'instanciara la classe "Escola" que definim ara:

- **Classe Escola:** contindrà la informació de les escoles (nom, localitat, responsable...) així com dels diferents professors i com els diferents grups d'alumnes, utilitzant les següents classes proposades.
  - Ha de tindre mètodes per afegir/eliminar alumnes, professors.

De forma auxiliar a la classe "Escola", s'han de crear les següents classes:

- **Classe Professor:** contindrà informació dels professors que treballen allí (nom, tipus

(ciències, lletres o mixt)).


- **Classe Alumne:** contindrà la informació dels alumnes de l'escola (nom, curs, professor responsable (només un professor)).

No hi ha un alumne que vaja a dues escoles diferents, ni dos professors que treballen en diferents escoles. Per tant trieu l'estructura d'objectes més adequada per al seu correcte emmagatzematge.

Les classes han d'estar declarades en un fitxer separat del programa. A més:

- S'han de representar les relacions entre classes.
- Han de tindre mètodes per gestionar correctament tots els elements, amb les seues corresponents operacions (inserció, modificació, esborrat, visualització, etc.).

**Finalment, fes un codi d'exemple que permeta provar l'has capacitats del teu disseny.**

 **Consell 01:** aquest tutorial pot ajudar-te a organitzar les classes Python en diversos fitxers


<https://www.delftstack.com/es/howto/python/python-import-class-from-another-file/>

## 4.6 PREGUNTA ENTREVISTA 06

SQLAlchemy és una biblioteca de Python amb diverses funcions i drivers per tractar amb bases de dades. Una de les funcions més destacades és la inclusió d'un ORM (Object Relational Mapping).

Partint del disseny de classes de la pregunta anterior i el seu codi, fes un nou programa que afegisca persistència a les classes, mantenint les seues relacions utilitzant l'ORM de Python 3 inclòs a la biblioteca SQLAlchemy amb **el driver de SQLite**.

Fes un codi d'exemple que permeta provar la persistència adquirida en implementar l'ORM amb objectes de la classe "Alumne", indicant clarament en comentaris que ha de fer-se per provar-la.


 **Consell 01:** Pot ajudar-te en aquesta tasca aquests tutorials:


a) Per posar en marxa SQLAlchemy ORM:

<https://j2logo.com/python/sqlalchemy-tutorial-de-python-sqlalchemy-guia-de-inicio/>

b) Per entendre relacions 1 a molts <https://programmerclick.com/article/38091329500/> i <https://www.hectormansilla.cl/2020/05/14/relaciones-uno-a-muchos-en-sqlalchemy/>

**IMPORTANT:** Llegiu i entengueu els tutorials de manera crítica.

 **Consell 02:** una bona estratègia per provar la persistència, és fer un codi/funcions que creen/modifiquen/eliminen alguns objectes, tancar el programa, comentar eixos fragments i comprovar que pese a no executar-se els canvis, aquests estan reflectits en la persistència i en el programa.

 **Consell 03:** per instal·lar SQLAlchemy, pots ajudar-te de "***pip3 install SQLAlchemy***".

## 4.7 PREGUNTA ENTREVISTA 07

Realitza un programa que rebrà per paràmetres de consola dos noms: el del fitxer d'entrada i el del fitxer d'eixida.

A més, per tal de tindre aquest programa en un executable i poder distribuir-ho haurem d'empacar el programa en un executable utilitzant la biblioteca <https://www.pyinstaller.org/>.

El programa implementarà dues funcions: “esPalindromo” i “esPrimo”.

**Important:** aquestes funcions hauran d'estar comprovades amb un test unitari. Recomanem per aquest cas “doctest” <https://docs.python.org/es/3/library/doctest.html> però val qualsevol altra utilitat per fer tests unitaris.

El programa ha de llegir el fitxer d'entrada. Aquest fitxer conte un número per línia (no s'especifica quantes línies, ha de funcionar per qualsevol número).

El nostre programa ha d'imprimir en el fitxer d'eixida dues línies i després un llistat amb un número per línia:

- La primera línia ha de dir quants números eren “palíndroms” (capicua)
- Una altra línia dient quants eren “cosins”.
- Finalment, “n” línies indicant quins “n” números eren al mateix temps cosins i palíndroms.

Per exemple:

**Fitxer “exempleEntrada.txt”**


5  
7  
11  
13  
14  
15  
22


**Fitxer “exempleEixida.txt”**

Hi han 4 números palíndroms.


Hi han 4 números cosins.

5  
7  
11

 **Consell 01:** intenta implementar les funcions demanades tu a soles. No copies la solució d'Internet ni dones una mirada a la d'un company, perquè a l'entrevista personal es demanaran modificacions i s'ha de comprendre que estan fent “esPrim” i “esPalindrom”.

 **Consell 02:** aquest tutorial pot ajudar-te amb “Doctest”

<https://www.digitalocean.com/community/tutorials/how-to-write-doctests-in-python>

 **Consell 03:** aquest tutorial pot ajudar-te amb “PyInstaller”

<https://www.devdungeon.com/content/pyinstaller-tutorial>

#### 4.8 PREGUNTA ENTREVISTA 08

Crea un programa Python 3 que ens ajude a organitzar els fitxers de la carpeta actual. El programa tindrà una llista amb extensions de programes (ha de funcionar amb qualsevol).

El programa en executar-se, per cada extensió, mourà tots els fitxers amb eixa extensió a una carpeta amb eixe nom.


Per exemple, amb llista=["png","mp4", "doc"], el programa mouria tots els fitxers amb extensió “.png” de la carpeta actual a una carpeta anomenada “png”. El mateix amb els fitxers “.mp4” a la carpeta “mp4” i els fitxers “.doc” amb la carpeta “doc”.

A més, per tal de tindre aquest programa en un executable i poder distribuir-ho haurem d’empacar el programa en un executable utilitzant la biblioteca <https://www.pyinstaller.org/>.

 **Consell 01:** aquest tutorial i cheatsheet poden ser útils per solucionar aquest problema

<https://unipython.com/operaciones-con-archivos-y-carpetas-en-python/>

<https://github.com/sergarb1/CheatSheetsCastellano/raw/master/Cheatsheet%20-%20Python%20para%20crear%20Scripts%20de%20Sistemas%20operativos.pdf>

 **Consell 02:** aquest tutorial pot ajudar-te amb “PyInstaller”

<https://www.devdungeon.com/content/pyinstaller-tutorial>

#### 4.9 PREGUNTA ENTREVISTA 09

Utilitzant la biblioteca <https://pypi.org/project/python-barcode/>, hem de llegir un fitxer “.csv” passat per paràmetres de consola.


Aquest fitxer tindrà un format en el qual cada línia usará “nomalumne,ID”.

Per exemple

“Alumne 01”,“1”

“Alumne 02”,“2”

Haurem de fer un programa, que llegint eixe fitxer “.csv” genere per cada alumne genere un fitxer “nomalumne.png” amb un codi de barres en format “EAN 13” que tinga com informació el “ID” de l’alumne.

 **Consell 01:** per llegir fitxers “.csv” pots utilitzar la biblioteca “csv” o la biblioteca “pandas”. Ací tens exemples:

<https://code.tutsplus.com/es/tutorials/how-to-read-and-write-csv-files-in-python--cms-29907>

<https://pythonbros.com/leer-csv-con-pandas/>



**Consell 02:** pots instal·lar Python Barcode amb *'pip3 install python-barcode'* i *'pip3 install "python-barcode[image]'*".

**Consell 03:** tens exemples de funcionament de Python Barcode que són útils directament en <https://pypi.org/project/python-barcode/>. Records comprendre que estàs fent. Has de tindre clar que és EAN13 abans de codificar [https://es.wikipedia.org/wiki/European\\_Article\\_Number](https://es.wikipedia.org/wiki/European_Article_Number)

#### 4.10 PREGUNTA ENTREVISTA 10

Complementant la pregunta anterior i utilitzant com a base el següent exemple de codi <https://www.geeksforgeeks.org/how-to-make-a-barcode-reader-in-python/> fes un programa que llegisca tots els ".png" d'un directori i ens diga el nom d'alumne i el seu ID. (Recorda, el nom de l'alumne és el nom del ".png" i el ID has d'extraure-lo del codi de barres).

**Consell 01:** per usar aquesta biblioteca, a més d'instal·lar-la amb *"pip3 install pyzbar"*, has d'instal·lar la utilitat "zbar" que es pot instal·lar amb *"apt install zbar-tools"*

**Consell 02:** per fer servir la biblioteca "cv2" (les sigles CV son de Computer Vision), has d'instalar *"pip3 install opencv-python"* i *"pip3 install opencv-contrib-python"*.

#### 4.11 PREGUNTA ENTREVISTA 11

Crea un programa que tinga una interfície gràfica" senzilla fent servir "Kivy", un entorn multiplataforma per Python <https://kivy.org>.

Aquest programa, en arrancar, carregara un fitxer ("users.csv") que per cada línia tindrà un usuari i el SHA1 del password, amb el següent format:

"user": "passwordinSHA1"

La interfície gràfica que tindrà només dos camps de text, "usuari" i "contrasenya" (en text pla, però oculta en teclejar) i un botó "comprovar". Si l'usuari existeix i la contrasenya és correcta, mostrarà un missatge de "OK". En cas contrari, un missatge "ERROR".

**Consell 01:** per llegir fitxers ".csv" pots utilitzar la biblioteca "csv" o la biblioteca "pandas". Aquí tens exemples:

<https://code.tutsplus.com/es/tutorials/how-to-read-and-write-csv-files-in-python--cms-29907>

<https://pythonbros.com/leer-csv-con-pandas/>

💬 **Consell 02:** per comprovar si una contrasenya en SHA1 és correcta, aprofitant les propietats de les funcions “hash” [https://es.wikipedia.org/wiki/Funci%C3%B3\\_hash](https://es.wikipedia.org/wiki/Funci%C3%B3_hash) el que has de fer és obtenir el “hash” de la contrasenya en text pla i comparar eixe “hash” obtingut amb el “hash” emmagatzemar. Si són iguals, la contrasenya és correcta.

Un exemple de SHA1 en Python 3 <https://programmerclick.com/article/7363885029/>

💬 **Consell 03:** tens un exemple d’una interfície d’inici de sessió en Kivy en <https://gist.github.com/Cheaterman/812203a74f8c552a4918>

També aquest tutorial

<https://kivycoder.com/creating-a-login-screen-with-kivymd-python-kivy-gui-tutorial-44/> amb el vídeo <https://www.youtube.com/watch?v=G-Rp41BzGxg>

#### 4.12 PREGUNTA ENTREVISTA 12

Replica aquest exemple de Python 3 per fer reconeixement de text en imatges. Afegeix comentaris per explicar el seu ús.

<https://towardsdatascience.com/building-a-simple-text-recognizer-in-python-93e453ddb759>

Podeu provar amb alguna imatge senzilla (només text i un fons homogeni) com:

<https://www.facebook.com/MumukiOrg/photos/pcb.4322522981176358/4322522757843047/>

💬 **Consell 01:** per aquest prova, a més de les biblioteques Pillow y PyTesseract instal·lades amb “*pip3 install Pillow pytesseract*”, has d’instal·lar la utilitat “Tesseract” que es pot instal·lar amb “*apt install tesseract-ocr libtesseract-dev*”

#### 4.13 PREGUNTA ENTREVISTA 13

Utilitzant la biblioteca de Python 3 “requests” i basant-se en exemples com

<https://pythonizando.com/consumir-una-api-rest-con-python-3-usando-requests/>

Fes un programa que consumeix l’API de “Rick and Morty” <https://rickandmortyapi.com/>

El nostre programa ens permetrà, demanant per teclat una espècie (atribut “specie”), obtenir un llistat de quins personatges són d’eixa espècie.

💬 **Consell 01:** per instal·lar “requests” pots fer-lo amb “*pip3 install requests*”.

#### 4.14 PREGUNTA ENTREVISTA 14

El problema de les 8 Reines és àmpliament conegut

[https://es.wikipedia.org/wiki/problema\\_de\\_las\\_ocho\\_reinas](https://es.wikipedia.org/wiki/problema_de_las_ocho_reinas)

Anem a donar-li una volta i anem a proposar el problema de les N reines.


Volem que faces un programa, que per a un nombre N ( $4 \leq N \leq 10$ ) calcule de quantes formes **diferents** es poden col·locar N reines en un tauler de NxN dimensions.

**NOTA IMPORTANT:** no s'admeten precàlculs, el programa ha de calcular les solucions.

Per provar, els resultats haurien de ser:

Tamany N	Solucions diferents	Solucions totals (sense eliminar repeticions)
4	1	2
5	2	10
6	1	4
7	6	40
8	12	92
9	46	352
10	92	724
11	341	2,680
12	1,787	14,200

**Per simplificar el problema, el vostre programa només haurà de calcular les “Solucions totals” del problema de les N reines (sense eliminar repeticions).**

 **Consell 01:** intenta implementar les funcions demanades tu a soles. No copies la solució d'Internet ni dones una mirada a la d'un company, perquè a l'entrevista personal es demanaran modificacions i s'ha de comprendre que estan fent el programa.

#### 4.15 PREGUNTA ENTREVISTA 15

Fes un programa que llegisca d'un fitxer que es passe com a paràmetre, un llistat de paraules (una paraula per línia). Aquestes paraules serà un llistat de “paraules prohibides”.

Utilitzant la biblioteca <https://pyperclip.readthedocs.io/en/latest/>, realitzeu un programa que quan copies alguna cosa del porta-retalls, analitzi el text copiat i substituïska de les paraules prohibides (sense distingir majúscules i minúscules) els caràcters de les paraules prohibides al llistat per \* i el torne a copiar al porta-retalls.

#### Exemple de funcionament:


Cridem al programa ***“programa.py llista.txt”***

El fitxer “llista.txt” content les paraules:

Hola

pYThon

Amb el programa en marxa, si copie al porta-retalls la frase “HOLA Sergi viva PYTHON”, hauria de modificar al porta-retalls aquesta frase i deixar-la com “\*\*\*\*\* Sergi viva \*\*\*\*\*”, de forma que quan peguem, es pegue aquesta segona versió.

 **Consell 01:** Podeu instal·lar-la amb ***“pip3 install pyperclip”***.

Si vos dona "Not implemented Error", instal·leu els següents programes:

***“sudo apt-get install xsel”***

***“sudo apt-get install xclip”***