



TASCA GUIADA TDD – GIT 4. VERIFICACIÓ PASSWORD

Validació del camp d'entrada de la contrasenya

Creeu una funció que es puga utilitzar com a validador per al camp de contrasenya d'un formulari de registre d'usuari. La funció de validació **pren una cadena com a entrada i retorna un resultat de validació**. El resultat de la validació hauria de contenir un booleà que indique si la contrasenya és vàlida o no, i també un camp amb els possibles errors de validació.

Requisits

1. La contrasenya ha de tenir *almenys 8 caràcters*. Si no es compleix, s'ha de retornar el missatge d'error següent: **"La contrasenya ha de tenir almenys 8 caràcters"** i el booleà a fals.
2. La contrasenya ha de contenir *almenys 2 números*. Si no es compleix, s'ha de retornar el missatge d'error següent: **"La contrasenya ha de contenir almenys 2 números"** i el booleà a fals.
3. La funció de validació hauria de gestionar diversos errors de validació.

Per exemple, "qwert" hauria de tenir un missatge d'error: **"La contrasenya ha de tenir almenys 8 caràcters\nLa contrasenya ha de contenir almenys 2 números"**

4. La contrasenya ha de contenir *almenys una majúscula*. Si no es compleix, s'ha de retornar el missatge d'error següent: **"La contrasenya ha de contenir almenys una lletra majúscula"** i el booleà a fals.
5. La contrasenya ha de contenir *almenys un caràcter especial*. Si no es compleix, s'ha de retornar el missatge d'error següent: **"La contrasenya ha de contenir almenys un caràcter especial"** i el booleà a fals..

1. Usar repositori GITHUB privat per allotjar el projecte de tasques
2. Connectar IntelliJ amb el vostre GITHUB
3. Crear mòdul **PasswordVerify**
4. Crear Classe **PasswordVerify** buida
5. Crear tests per a dita classe.
6. Test Entrada: "" Salida: Objecte {false, "La contrasenya ha de tenir almenys 8 caràcters\nLa contrasenya ha de contenir almenys 2 números\nLa contrasenya ha de contenir almenys una lletra majúscula\nLa contrasenya ha de contenir almenys un caràcter especial"}
7. Test Entrada: "hola" Salida: Objecte {false, "La contrasenya ha de tenir almenys 8 caràcters\nLa contrasenya ha de contenir almenys 2 números\nLa contrasenya ha de contenir almenys una lletra majúscula\nLa contrasenya ha de contenir almenys un caràcter especial"} → Commit
8. Test Entrada: "holacaracola" Salida: Objecte {false, "La contrasenya ha de contenir almenys 2 números\nLa contrasenya ha de contenir almenys una lletra majúscula\nLa contrasenya ha de contenir almenys un caràcter especial"} → Commit



9. Test Entrada: "hola12" Salida: `Objecte {false, "La contrasenya ha de tenir almenys 8 caràcters\nLa contrasenya ha de contenir almenys una lletra majúscula\nLa contrasenya ha de contenir almenys un caràcter especial"}` → Refactor && Commit
10. Test Entrada: "holA" Salida: `Objecte {false, "La contrasenya ha de tenir almenys 8 caràcters\nLa contrasenya ha de contenir almenys 2 números\nLa contrasenya ha de contenir almenys un caràcter especial"}` → Commit
11. Test Entrada: "hola!" Salida: `Objecte {false, "La contrasenya ha de tenir almenys 8 caràcters\nLa contrasenya ha de contenir almenys 2 números\nLa contrasenya ha de contenir almenys una lletra majúscula"}` → Refactor && Commit
12. Test Entrada: "holaA!" Salida: `Objecte {false, "La contrasenya ha de tenir almenys 8 caràcters\nLa contrasenya ha de contenir almenys 2 números"}` → Refactor && Commit
13. Test Entrada: "hola12!" Salida: `Objecte {false, "La contrasenya ha de tenir almenys 8 caràcters\nLa contrasenya ha de contenir almenys una lletra majúscula"}` → Refactor && Commit
14. Test Entrada: "hola12A" Salida: `Objecte {false, "La contrasenya ha de tenir almenys 8 caràcters\nLa contrasenya ha de contenir almenys un caràcter especial"}` → Refactor && Commit
15. Test Entrada: "holacaracola12" Salida: `Objecte {false, "La contrasenya ha de contenir almenys una lletra majúscula\nLa contrasenya ha de contenir almenys un caràcter especial"}` → Refactor && Commit
16. Test Entrada: "holacaracola!" Salida: `Objecte {false, "La contrasenya ha de contenir almenys 2 números\nLa contrasenya ha de contenir almenys una lletra majúscula"}` → Refactor && Commit
17. Test Entrada: "holacaracolaA" Salida: `Objecte {false, "La contrasenya ha de contenir almenys 2 números\nLa contrasenya ha de contenir almenys un caràcter especial"}` → Refactor && Commit
18. Test Entrada: "ho12A!" Salida: `Objecte {false, "La contrasenya ha de tenir almenys 8 caràcters"}` → Refactor && Commit
19. Test Entrada: "holacaracolaA!" Salida: `Objecte {false, "La contrasenya ha de contenir almenys 2 números"}` → Refactor && Commit
20. Test Entrada: "holacaracola12!" Salida: `Objecte {false, "La contrasenya ha de contenir almenys una lletra majúscula"}` → Refactor && Commit
21. Test Entrada: "holacaracola12A" Salida: `Objecte {false, "La contrasenya ha de contenir almenys un caràcter especial"}` → Refactor && Commit



22. Test Entrada: "holacaracola12A!" Salida: Objecte {true, ""}
23. Test Entrada: "#P4blit0cl4v0uncl4vit0#" Salida: Objecte {true, ""}

Nota: S'ha de crear una classe amb nom PasswordResult per exemple que tinga el booleà i el missatge d'error, que és el que verificarem amb el assertEquals.