

2026	PROJET SIO JANVIER
BTS SIO 1 SIOA Groupe A	Auteurs : BALAN—HAYASHI Aïto & Orieux Adrien
	Date de rédaction : 05/01/2026
	Groupe : 05

## Compte-rendu d'étape

### Objectif de la phase

L'objectif de cette phase est de renforcer la sécurité des comptes en mettant en place une génération automatique de mots de passe conformes à des règles strictes. L'idée est d'éviter les mots de passe faibles ou générés manuellement, tout en produisant un fichier CSV exploitable pour la suite du projet.

### Méthode de génération des mots de passe

#### - Principe général

Les mots de passe sont générés automatiquement en respectant une structure précise. Ils ne contiennent ni espace, ni lettre **O** majuscule, ni chiffre **0**, afin d'éviter toute confusion visuelle. Le résultat final est exporté dans un fichier CSV contenant une seule colonne nommée *password*.

#### - Règles appliquées

Chaque mot de passe respecte les contraintes suivantes :

- longueur minimale de 12 caractères ;
- au moins 2 majuscules, 2 minuscules, 2 chiffres et 2 caractères spéciaux ;
- caractères spéciaux autorisés limités à une liste définie ;
- aucun caractère spécial en début ou en fin ;
- aucun doublon dans le fichier généré.

Pour garantir la robustesse, une vérification simple est également appliquée afin d'éviter les mots trop proches de mots courants (français ou anglais).

### Logique de construction

Le script commence par sélectionner le nombre minimum de caractères requis pour chaque catégorie (majuscules, minuscules, chiffres et spéciaux). Les caractères restants sont ensuite ajoutés de manière aléatoire à partir des ensembles autorisés.

Une fois le mot de passe construit, les caractères sont mélangés afin d'éviter toute structure prévisible, puis les contraintes de position et de validité sont vérifiées. Si une règle n'est pas respectée, le mot de passe est rejeté et régénéré.

### Vérification des mots de passe

Un programme de contrôle permet de vérifier automatiquement chaque mot de passe présent dans le fichier CSV. Pour chaque ligne, le script teste la longueur, la présence de caractères interdits, le respect des quotas et la position des caractères spéciaux.

Les doublons sont également détectés afin de garantir l'unicité des mots de passe.

### Résultats et livrables

Deux fichiers sont générés :

- un fichier *validation\_report.csv* indiquant, pour chaque mot de passe, s'il est valide ou non et les éventuelles erreurs ;
- un fichier *validation\_summary.txt* présentant un résumé global (nombre total de mots de passe, valides, invalides et doublons).

Cette phase permet de s'assurer que tous les mots de passe générés sont sécurisés, conformes aux règles et prêts à être utilisés.

2026	PROJET SIO JANVIER
BTS SIO	Auteurs : BALAN—HAYASHI Aïto & Orieux Adrien
1 SIOA	Date de rédaction : 05/01/2026
Groupe A	Groupe : 05

```
password
tY5qf59DtN
nR4ed97CsP
cS1ff19QpF
uF5ss75FkD
bG2cs24LnR
zD6ug31CgF
jE2ee62VMD
iM6bu96RjT
mL1vs92EjZ
dY8ve12XmU
gI9cl83GbU
wX1jr85LlZ
eD9nc31NtR
hZ8xg48NaZ
eL6ma81GqI
lE5nr11TuH
oE3tm17RiH
lF2vd86YbR
oB7cx16BpB
mN5hq67ZnU
iW5oc43EvW
oR2hg75HpX
hF6xe33NnC
uC3sd66ExF
tS8ac95CgK
aI5mw74QgK
iM6aj11JaV
wH3dl195nX
mY1tu83JdZ
```

```
def generate_password(length=12):
    if length < 10:
        return "Error: Password must be at least 10 characters long."
    chars = "abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*~.-_+=`{|}~"
    allowed_pos = [0] * length
    for i in range(length):
        if i % 2 == 0:
            allowed_pos[i] = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
        else:
            allowed_pos[i] = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]
    password = ""
    for i in range(length):
        if i % 2 == 0:
            password += chr(random.choice(allowed_pos[i]))
        else:
            password += chr(random.choice(allowed_pos[i]))
    return password

def create_password_list(length=12, count=10):
    passwords = []
    for i in range(count):
        password = generate_password(length)
        passwords.append(password)
    return passwords

def main():
    passwords = create_password_list(12, 10)
    for password in passwords:
        print(password)
```