



# Dev Blockchain - Evaluation finale

## A partir du smart contract de vote, ajouter les fonctionnalités suivantes au sein d'un projet *Foundry*

1. Ajouter un rôle d'administrateur à l'aide d'*OpenZeppelin*
2. La fonction qui permet d'enregistrer des candidats ne pourra être exécutée que par les administrateurs.
3. Actuellement le *smart contract* ne définit pas d'ordre dans lequel exécuter les fonctions. Il est possible de voter avant même que les candidats soient enregistrés. Ce qui n'est pas correct.  
Faites en sorte de définir un statut de workflow pour le *smart contract* qui peut prendre 4 valeurs : *REGISTER\_CANDIDATES*, *FOUND\_CANDIDATES*, *VOTE*, *COMPLETED*.  
De fait, quand le statut du workflow sera à l'état *REGISTER\_CANDIDATES*, seule la fonction d'enregistrement des candidats pourra être exécutée.  
Quand le statut sera à l'état *VOTE*, seule la fonction de vote pourra être exécutée. etc ....  
Toutes les fonctions devront donc respecter leurs phases correspondantes  
Seul un *administrateur* pourra modifier le statut du workflow
4. Ajouter un rôle de *FOUNDER* qui permettra à chaque candidat de recevoir des fonds. Donc seuls les *founders* pourront envoyer des fonds aux candidats
5. Faites en sorte que la fonction de vote qui n'a pas de restriction de rôle puisse être exécutée uniquement 1 heure après que le statut *VOTE* ait été set par l'*administrateur*
6. Créer un SC de NFT simple qui sera donné à toute personne ayant voté. Pour pouvoir voter il faudra donc s'assurer que le votant ne possède pas de NFT de vote
7. Ajouter une fonction qui désigne le vainqueur du vote en fonction du bon statut de workflow
8. Réaliser les test unitaires dans Foundry pour ce smart contract
9. Déployer le smart contract sur un vrai testnet Sepolia à l'aide d'un script de déploiement.
10. Commit/Push dans un repo public en mettant dans le readme quelques URL de transactions sur le réseau Sepolia