**Voici la description complète des fonctionnalités du contrat LotTracker :**

**Résumé Général**

Le contrat **LotTracker** est un suivi de chaîne d'approvisionnement basé sur des NFT (ERC-721).  
Voici les principales caractéristiques et fonctionnalités :

* **Fonction principale** : Il crée (minte) un NFT unique (token ERC-721) pour chaque "lot de médicament" ajouté au système.
* **Propriétaire & Transfert** : Le propriétaire du NFT (ex : fabricant, distributeur, pharmacie) est le propriétaire actuel du lot. Le transfert du NFT via la fonction push représente un transfert de la garde dans la chaîne d'approvisionnement.
* **Stockage des données** : Toutes les données critiques (nom, fabricant, emplacement, statut) sont stockées directement sur la blockchain et liées à ce NFT unique.
* **Modèle économique** : Les utilisateurs (mais pas le propriétaire du contrat) doivent payer des frais en **CreditToken** (ERC-20) pour effectuer des actions.

**Système de Frais et Qualification à Deux Niveaux**

C'est la logique métier la plus complexe et importante du contrat, destinée à garantir que les utilisateurs aient une participation minimale avant de pouvoir interagir avec le contrat et à leur faire payer une petite commission par transaction.

1. **Le CreditToken (ERC-20)**  
   Le contrat est lié à un token ERC-20 externe (le **CreditToken**). Tous les frais sont payés en **CreditToken**, et non en HBAR ou ETH. Cela signifie que toutes les fonctions d'écriture (comme insert) ne sont pas payables en HBAR/ETH.
2. **Le Modificateur hasCredit (Qualification Initiale)**  
   Ce modificateur est le "gardien" de toutes les fonctions principales. Il applique une vérification de "participation" une seule fois :
   * **Première Transaction** : Lorsqu'un utilisateur (qui n'est pas le propriétaire) essaie d'effectuer sa première action (par exemple insert), le modificateur vérifie si l'utilisateur a au moins **100 CreditToken** (minCreditBalance).
   * **Qualification** : Si l'utilisateur remplit la condition des 100 tokens minimum, le contrat définit isQualifiedUser[msg.sender] = true.
   * **Transactions Subséquentes** : Pour toutes les transactions futures, le modificateur vérifie que l'utilisateur est qualifié et passe outre la vérification des 100 tokens.
3. **La Fonction \_takeFee (Frais par Transaction)**  
   Cette fonction est appelée par toutes les fonctions principales des utilisateurs (comme insert, push, updateLotInfo) après que la vérification **hasCredit** ait été validée :
   * **Vérification des frais** : Elle vérifie si l'utilisateur dispose des tokens nécessaires pour payer le **fixedFee** (1 CreditToken).
   * **Approbation Requise** : Elle utilise creditToken.transferFrom(). Cela signifie que l'utilisateur doit avoir d'abord envoyé une transaction d'approbation au contrat **CreditToken**, autorisant le contrat **LotTracker** à retirer au moins 1 token.
   * **Exemption pour le Propriétaire** : Le propriétaire du contrat est exempté de la fonction \_takeFee, ce qui lui permet de réaliser des actions sans payer de frais ni avoir besoin d'une approbation.

**Résumé Fonctionnel Complet**

**Fonctions Principales (Transactions d'Écriture)**

Ce sont les principales actions qu'un utilisateur qualifié peut effectuer. Toutes ces fonctions nécessitent le modificateur **hasCredit** et, sauf pour le propriétaire, appellent la fonction \_takeFee.

| **Fonction** | **Description** |
| --- | --- |
| insert(...) | Crée un nouveau lot. Minter un nouveau NFT (ID de Token) pour l'appelant (msg.sender) et stocke toutes les données fournies (nom, fabricant, date, emplacement, URI) dans le mapping medicationLots. |
| push(...) | Transfère un lot. Transfère le NFT du propriétaire actuel (msg.sender) à un nouveau propriétaire et met à jour currentLocation et currentHandler. Cela met aussi à jour ownershipHistory. |
| updateLotInfo(...) | Met à jour les informations du lot. Permet au propriétaire actuel d'un NFT de modifier ses métadonnées (nom, fabricant, emplacement, etc.). |
| setInactive(...) | Définit le statut du lot sur "Inactif". |
| activateLot(...) | Réactive un lot, modifiant son statut de "Inactif" à "Actif". |
| discardLot(...) | Définit le statut du lot sur "Discaré" (état final permanent). |

**Fonctions de Lecture (Vue)**

Ces fonctions peuvent être appelées gratuitement pour récupérer des données depuis la blockchain.

| **Fonction** | **Description** |
| --- | --- |
| pull(tokenId) | Récupère toutes les données du lot de médicament correspondant au tokenId (retourne toute la struct MedicationLot et l'historique des propriétaires). |
| getLotStatus(tokenId) | Retourne le statut du lot sous forme de chaîne (ex. : "Actif", "Inactif"). |
| isQualifiedUser(address) | Retourne **true** ou **false** selon si un utilisateur a rempli la condition des 100 tokens. |
| medicationLots(tokenId) | Récupère la struct **MedicationLot** du mapping medicationLots. |
| ownershipHistory(tokenId, index) | Récupère l'historique des propriétaires d'un lot en fonction du tokenId et de l'index. |
| tokenIdCounter() | Retourne le nombre total de lots créés. |
| creditToken() | Retourne l'adresse du contrat **CreditToken**. |
| FEE\_RECIPIENT() | Retourne l'adresse du portefeuille qui reçoit tous les paiements des frais fixes. |
| fixedFee() | Retourne le montant des frais pour chaque transaction (en wei). |
| minCreditBalance() | Retourne le solde minimum nécessaire pour la qualification (en wei). |

**Fonctions Administratives (Uniquement pour le Propriétaire)**

Ces fonctions peuvent uniquement être appelées par le propriétaire du contrat (l'adresse qui a déployé le contrat).

| **Fonction** | **Description** |
| --- | --- |
| setFixedFee(newFee) | Permet au propriétaire de modifier les frais fixes (1 token). |
| setMinCreditBalance(newMin) | Permet au propriétaire de modifier le solde minimum nécessaire pour la qualification (100 tokens). |
| setFeeRecipient(newAddress) | Permet au propriétaire de modifier l'adresse du portefeuille recevant les paiements des frais. |
| setCreditTokenAddress(newAddress) | Permet au propriétaire de mettre à jour l'adresse du contrat **CreditToken**. |

Cela résume les fonctionnalités principales et administratives du contrat **LotTracker** ainsi que la logique des frais et qualification des utilisateurs.