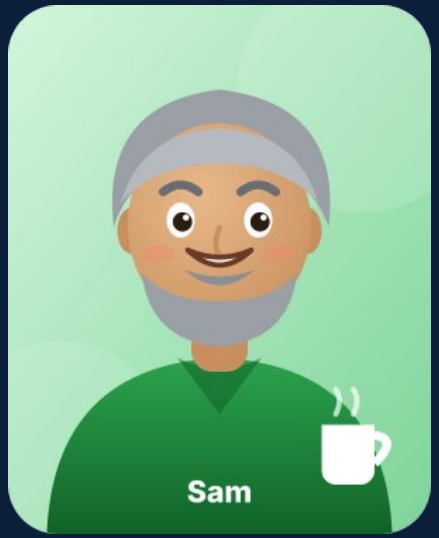
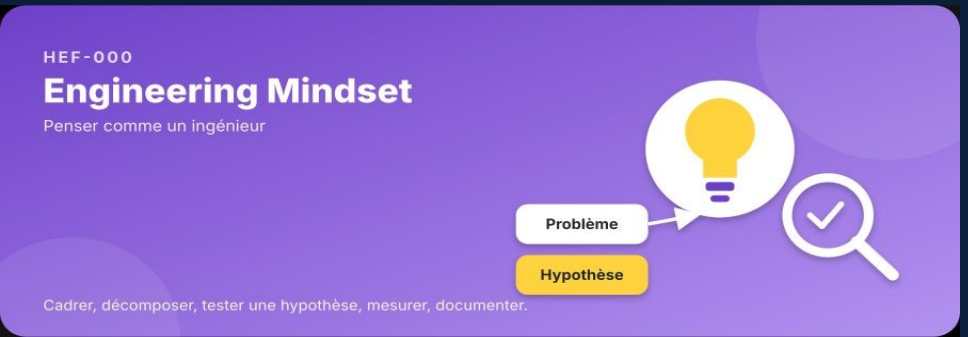
HEF-000 · MISSION 03 / 9

Penser juste et cacher le superflu

Raisonner sans se tromper, et savoir ce qu'on peut ignorer.





Badge : Logicien·ne

70 min · débutant

L'histoire

Un jeudi, le site tombe en panne. Quelqu'un dit :

> *« Il y a eu un déploiement à 14 h, la panne a commencé à 14 h 05. C'est le déploiement ! »*

Sam fronce les sourcils.

> *« Peut-être. Mais « juste après » ne veut pas dire « à cause de ». »*

Plus tard, un nouveau développeur demande comment fonctionne le chiffrement des mots de passe. Sam répond : « Utilise la fonction `hashPassword`, tu n'as pas besoin de savoir ce qu'il y a dedans. »



L'histoire (suite)

Deux idées dans une mission : bien raisonner (la logique) et savoir ce qu'on peut cacher (l'abstraction).

Ta mission

À la fin, je serai capable de…

1. utiliser ET, OU, NON pour écrire des conditions précises ;
2. distinguer une corrélation d'une cause ;
3. expliquer ce qu'est une abstraction avec mes mots ;
4. choisir un bon niveau de détail selon la décision à prendre ;
5. reconnaître une « fuite d'abstraction ».

Défi d'échauffement

Défi d'échauffement

Dans une ville, on constate : **les jours où les ventes de glaces augmentent, les noyades augmentent aussi.**

Question : les glaces causent-elles les noyades ? Écris ta réponse et **une autre explication possible**. Ne cherche pas à être parfait·e.

Le concept

Partie 1 — La logique

Une **condition** est une phrase qui est soit vraie, soit fausse.

Trois outils pour les combiner :

Opérateur	Sens	Exemple
ET (AND)	les deux doivent être vrais	« Connecté **ET** abonné » → accès au cours
OU (OR)	au moins un vrai	« Admin **OU** propriétaire » → peut modifier
NON (NOT)	inverse	« **NON** connecté » → afficher la page de connexion

Table de vérité du ET :

Le concept

Partie 1 — La logique (suite)

A	B	A ET B
vrai	vrai	vrai
vrai	faux	faux
faux	vrai	faux
faux	faux	faux

🧠 ***Astuce*** *: une condition vague (« l'utilisateur est actif ») est une source de bugs. Précise toujours : actif depuis quand ? sur quoi ?*

Corrélation et cause

Deux choses qui arrivent **en même temps** ne sont pas forcément liées par une cause.

Le concept

Corrélation et cause (suite)

Les glaces et les noyades augmentent toutes les deux **l'été** : c'est la chaleur, une troisième cause cachée.

Pour la panne de 14 h : le déploiement est une **hypothèse** forte, mais il faut la **tester** (annuler le déploiement, regarder ce qui a changé…).

Partie 2 — L'abstraction

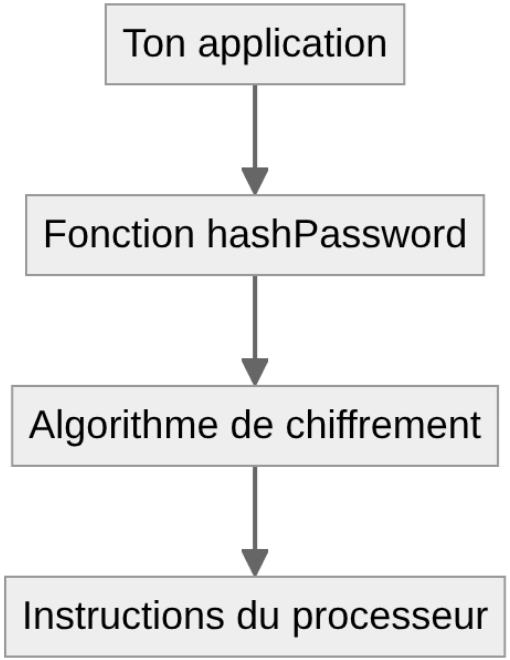
Une **abstraction**, c'est une façade qui **cache les détails** pour que tu puisses utiliser quelque chose sans tout comprendre.

Analogie : une voiture. Tu utilises le volant et les pédales. Tu n'as pas besoin de comprendre l'injection du carburant pour aller au travail.

En logiciel, c'est pareil :

Le concept

Partie 2 — L'abstraction (suite)



Le concept

Partie 2 — L'abstraction (suite)

Description du schéma : quatre couches empilées. Ton application utilise une fonction ; la fonction utilise un algorithme ; l'algorithme utilise les instructions du processeur. Chaque couche cache celle du dessous.

Quel niveau de détail ?

Pose cette question : « **De quel détail ai-je besoin pour prendre cette décision ?** »

- Pour choisir un forfait d'hébergement : le prix, la capacité. Pas les composants des serveurs.
- Pour trouver pourquoi un serveur est lent : là, il faut descendre dans les détails.

Les fuites d'abstraction

Parfois, ce qui était caché **réapparaît**. Exemple : une fonction « envoyer un fichier » marche très bien… jusqu'à ce que le réseau soit lent. Tout à coup, il faut comprendre ce qui se passe dessous.

Le concept

Les fuites d'abstraction (suite)

🧠 ***À retenir*** *: une abstraction est un* ***prêt de confiance****.*
Elle tient jusqu'au jour où elle ne tient plus.

Exemple pas à pas

Affirmation : « La panne est arrivée juste après le déploiement, donc c'est le déploiement. »

Étape 1 : relever les faits.

- Déploiement à 14 h 00.
- Premières erreurs à 14 h 05.

Étape 2 : lister les autres explications possibles.

- Un pic de visiteurs à 14 h 05 (ce jour-là, une newsletter est partie).
- Un partenaire externe en panne.

Étape 3 : concevoir un test.

- Regarder les journaux : les erreurs viennent-elles du code déployé ?

Exemple pas à pas (suite)

- Regarder le trafic : y a-t-il un pic ?

Étape 4 : conclure uniquement ce qui est prouvé.

Résultat : on n'accuse pas trop vite. On passe d'une opinion à une **enquête**.

À toi, avec aide

Écris les conditions suivantes avec ET, OU, NON.

1. Un utilisateur peut accéder au tableau de bord s'il est connecté et que son compte n'est pas suspendu.
2. Un message d'erreur s'affiche si le mot de passe est vide ou s'il fait moins de 8 caractères.

À toi, avec aide · réponse

Indices progressifs

Indice 1

Dans la phrase 1, cherche les deux choses qui doivent être vraies en même temps.

Indice 2

« N'est pas suspendu » se dit : NON suspendu.

À toi, avec aide · réponse

Réponse

1. connecté ET (NON suspendu)
2. (mot de passe vide) OU (longueur < 8)

À toi, en solo

Voici trois affirmations. Pour chacune, dis si c'est un **fait**, une **corrélation** ou une **cause prouvée**, puis propose un test.

1. « Les utilisateurs qui lisent les rappels terminent plus de missions. »
2. « Depuis qu'on a changé la couleur du bouton, les inscriptions ont baissé. »
3. « Quand on désactive le cache, la page devient lente. »

Pistes de correction

1. Corrélation. Peut-être que les plus motivés lisent *et* terminent. Test : envoyer des rappels à un groupe au hasard et comparer.
2. Corrélation temporelle. Test : montrer l'ancien bouton à une moitié des visiteurs et le nouveau à l'autre.
3. Cause plausible, car on l'a provoquée. Test : réactiver le cache et mesurer.

Pièges classiques

Piège	Exemple	Comment l'éviter
Condition vague	« si l'utilisateur est actif »	Écrire exactement la règle
Corrélation = cause	« le déploiement a tout cassé »	Chercher une autre explication, puis tester
Mauvais niveau d'abstraction	Parler de bits à un client	Adapter au décideur
Trop d'abstraction	Couche sur couche sans raison	Se demander : *cette couche m'aide-t-elle vraiment ?*

Mini-boss

Mini-boss

Tu reçois ce rapport : *« Après la mise à jour de la page de profil, les téléchargements de certificats échouent. »*

1. Écris la **condition précise** décrivant quand l'échec se produit (avec ET/OU/NON).
2. Liste 3 explications possibles autres que « la mise à jour ».
3. Dis à quel **niveau de détail** tu travailles pour chacune (utilisateur, application, serveur, réseau).
4. Propose un test pour départager.

Mini-boss · réponse

Exemple

1. Échec = utilisateur connecté ET demande un certificat ET formation terminée après la mise à jour.
2. Un changement de droits ; une panne du générateur de PDF ; un volume de demandes inhabituel.
3. Application, serveur, serveur.
4. Annuler la mise à jour sur un environnement de test et observer si l'échec disparaît.

Quiz éclair

Quiz éclair

1. A ET B est vrai quand :

A. au moins un des deux est vrai
B. les deux sont vrais
C. aucun n'est vrai

2. Deux événements se produisent en même temps. On peut conclure :

A. que l'un cause l'autre
B. qu'on a une corrélation à tester
C. qu'il n'y a aucun lien

3. Une abstraction sert à :

A. cacher des détails inutiles pour la décision

Quiz éclair (suite)

B. rendre le code plus long
C. supprimer les bugs

4. 🔁 *Révision mission 02* : décomposer, c'est :

A. couper au hasard
B. séparer en parties en gardant les liens
C. recommencer de zéro

5. 🔁 *Révision mission 01* : un fait est :

A. une information vérifiée
B. une opinion
C. une solution

Quiz éclair · réponse

Réponses

B. 2. **B**. 3. **A**. 4. **B**. 5. **A**.

Je réfléchis

Je réfléchis

1. Reprends ton défi d'échauffement. Quelle autre explication as-tu trouvée ?
2. Cite une situation où tu as pris une corrélation pour une cause.

À retenir

À retenir

1. Une condition est vraie ou fausse : précise-la.
2. ET, OU, NON se combinent pour exprimer une règle.
3. Corrélation ne veut pas dire cause.
4. Une abstraction cache le détail inutile, jusqu'à ce qu'il refasse surface.
5. Le bon niveau de détail dépend de la décision à prendre.

MISSION ACCOMPLIE

+100 XP

Badge débloqué : Logicien·ne

Prochaine mission : Déboguer avec méthode

