

Devs

**COMMENT DÉCHIRER À TON
ENTRETIEN D'EMBAUCHE ?**



[sfɛir]



Préface	5
Chapitre 1. Typologies d'entreprise	7
Editeurs	8
Startups	8
Sociétés de Services	9
Clients finaux	9
Chapitre 2. Processus de recrutement	11
Processus de recrutement chez Google	12
Processus de recrutement chez Amazon	13
Processus de recrutement chez SFEIR	13
Chapitre 3. Prépare ton entretien	15
Ton process vu du ciel (La Carto)	17
Optimise ton CV	18
Les mots pour les RH	18
La description pour les managers et les ingénieurs	18
La grille de tes projets	19
Le Dress code	20
Chapitre 4. L'évaluation de ta personnalité	21
Structure tes réponses avec le modèle SAR :	22
Situation/Action/Résultat	23
Comment répondre aux questions de comportement	23
Dis-m'en plus à propos de toi	24
Focus sur toi	24
Décris tes qualités, forces et faiblesses	24
Ton empathie client	24
Vérification de tes références	25

Chapitre 5. L'évaluation de tes compétences techniques	27
Le QCM	29
La conversation avec un expert	29
Au clavier	29
Test en remote	29
Test en pair programming	29
Détrompeur en ligne	29
Impératifs pour réussir un test en pair programming	29
Le Challenge	30
La Culture Générale	30
Chapitre 6. Comment bien choisir sa société	31
Chapitre 7. Questions d'Algo	35
Algorithme	36
Le test d'algo	36
Exemples de questions	40
Bibliographie	42
Pour aller plus loin	44
Remerciements	50



SFEIR est une société de conseil spécialisée dans le développement d'applications sous toutes leurs formes (web, mobile, objets connectés, cloud, data etc.). Sa vocation est d'accompagner ses clients dans leur transformation numérique en créant pour eux les meilleures produits possibles.

Trouver de bons développeurs est un vrai défi pour une entreprise comme la nôtre. Il y a de nombreux acteurs qui cherchent à attirer les développeurs talentueux. En tant que société dont la notoriété se limite au cercle des initiés, il est difficile de les rencontrer pour les recruter.

Les développeurs sont au cœur de la transformation numérique. Ils construisent les produits qui vont aider les entreprises à affronter la mutation en cours. Recruter les meilleurs éléments est crucial. C'est pourquoi le processus de recrutement est un élément essentiel de la santé économique de SFEIR, qui doit être capable de séduire aussi bien les étudiants que les développeurs confirmés.

Les juniors que nous recrutons sortent directement de l'école : il s'agit principalement de coups de cœur. Il nous faut alors plusieurs mois pour en faire de bons développeurs. Nous n'avons pas d'école favorite. Les nouveaux Sfeiriens viennent d'établissements divers : Epita, Epitech, Mines de Nantes, Centrale Lille, Ecole 42, Universités, etc.

Si l'investissement pour leur formation est important, c'est toujours un plaisir de les voir grandir et acquérir leur métier.

Pour attirer des développeurs expérimentés, c'est plus compliqué. La première manière de recruter se base sur la cooptation. Les Sfeiriens sont contents d'être chez SFEIR et invitent naturellement leurs connaissances à les rejoindre. Les autres sources de recrutement sont les sites d'annonces classiques, les réseaux sociaux tel que LinkedIn, ou, plus ciblé, des plateformes proposant des contenus basés sur des témoignages authentiques de collaborateurs comme Welcome To The Jungle.

Il est difficile de bien connaître une entreprise avant de la rejoindre, notre rôle est donc de communiquer notre vision aux collaborateurs potentiels. Il est important qu'ils partagent nos valeurs, car une fois chez SFEIR, ils deviendront les ambassadeurs de notre marque. C'est pourquoi nous avons créé les PlayOffs, un moment particulier où les développeurs qui veulent nous rejoindre vont participer à des expériences de développement avec des Sfeiriens.

Ce document vise à aider les candidats qui veulent nous rejoindre à bien préparer ce moment. Nous pensons qu'il peut être utile à d'autres, c'est pourquoi nous le rendons libre d'accès.

Bonne aventure technologique.

Chapitre 1

Typologies d'entreprise





Éditeurs

Les GAFA (pour Google, Amazon, Facebook, Apple ou autres gros acteurs du web) et les SaaS providers recherchent des développeurs très compétents qui en plus ont un très gros potentiel. Les processus d'embauche dans ces structures sont longs. La relation employeurs / candidats est aussi atypique. Ils reçoivent beaucoup de candidatures spontanées. L'effectif des GAFA est constitué de 50% de développeurs.

Les éditeurs Open Source aiment les pointures technologiques, le recrutement se fait souvent par « cooptation ». Rejoindre ce type de structures est une vocation, les conditions de travail sont souvent très particulières : télétravail, petites équipes, gros ego. Ce sont sans doute les structures les plus amicales pour les développeurs. L'effectif de ce type d'éditeurs est composé de plus de 70% de développeurs, avec de grandes différences entre les gros éditeurs et les petits.

Les éditeurs traditionnels sont quant à eux friands de profils sachant participer à l'analyse et à la rédaction des spécifications techniques des logiciels, développer et réaliser des évolutions sur des applications métiers et rédiger des documents nécessaires à l'utilisation du logiciel (manuels d'utilisation, d'installation, d'exploitation). Ils recherchent aussi des développeurs capables de se projeter sur le long terme. L'effectif des éditeurs est constitué de 30% de développeurs.

Startups

Les startups sont une catégorie bien spécifique, ce sont des entreprises qui se caractérisent par critères : disruption, croissance et incertitude. Elles se divisent en deux types: celles qui sont peu financées (niveau « Seed ») et celles qui sont financées (niveau « Series »). Ce type d'entreprise n'a pas le temps de former ses nouveaux arrivants. Les startups préfèrent embaucher des candidats qui apprennent vite et qui sont déjà familiarisés avec les outils utilisés au sein de leur entreprise, par exemple leur stack technique.

Les startups niveau « Seed » sont à la recherche d'ingénieurs proactifs qui savent prendre des initiatives et proposer des idées novatrices permettant de développer le produit. L'effectif des startups « Seed » est constitué de +70% de développeurs.

Les startups niveau "Series" sont à la recherche d'ingénieurs qui sont capables de rentrer dans un existant et de le désendetter tout en continuant à assurer une grosse productivité. Ces startups ont déjà un gros patrimoine de code qu'il faut faire bouger. L'effectif de ces startups est composé de 50% de développeurs.

Sociétés de Services

Les cabinets de développeurs sont des « pures players » du développement. Ils recrutent uniquement des développeurs, ils recherchent les meilleurs, sont très sélectifs, que ce soit sur la personnalité, les compétences, le potentiel et le leadership. Ces cabinets forment de très bonnes « écoles » pour progresser. L'effectif des cabinets de développeurs est constitué de 80% de développeurs. C'est dans cette catégorie d'entreprises que SFEIR se positionne.

Les cabinets de conseil délivrent des avis ou des recommandations sur ce qu'il convient de faire. Il s'agit d'une activité de services qui vise à améliorer le fonctionnement et la performance, actuels ou futurs d'une entreprise. L'effectif des cabinets de conseil est constitué de 20% de développeurs.

Les Agences Web sont aux services des Directions de la Communication, elle recherche principalement des développeurs débrouillards, qui gèrent bien la pression et les délais. L'effectif des agences web est constitué de 20% de développeurs.

Les ESN (SSII) sont à la recherche de profils développeurs de tous niveaux. Ce qui les intéresse ce sont les profils qui majoritairement feront du développement fonctionnel.

Clients finaux

Les entreprises « CAC 40 » souhaitent avoir des développeurs qui savent s'approprier rapidement les technologies émergentes en gardant un oeil attentif sur ce qu'elles pourront apporter à l'utilisateur final. La majeure partie de ces entreprises ont leur propre système d'information développé en interne. Elles ont également le temps, les ressources et les moyens logistiques nécessaires pour former leurs nouvelles recrues. Une grande partie du temps, le développeur fera de la maintenance. L'effectif des entreprises matures est constitué de moins de 5% de développeurs.

Les entreprises publiques apprécient les spécifications. Elles sont à la recherche d'ingénieurs connaissant les processus spécifiques liés à ce type d'entreprise. Une grosse partie du développement est externalisé, en général vers des SSII contractualisant au forfait. L'effectif des entreprises publiques est constitué de moins de 5% de développeurs.

La première chose pour toi est donc de trouver l'entreprise qui te correspond le mieux. Oui, nous allons nous tutoyer pour simplifier les explications !

Chapitre 2

Processus de recrutement



Processus de recrutement



Voici un processus de recrutement typique :

- 1. Premier entretien téléphonique** : quelques questions te seront posées afin de valider que tu ne te trompes pas,
- 2. Échange par Skype ou Hangout avec un ingénieur** : tu vas, à cette occasion, présenter du code dans un document partagé ou un site de tests en ligne. Ne prends pas cet exercice à la légère,
- 3. Entretiens sur place en une ou plusieurs fois** : tu auras des « tête-à-tête » avec des RH, des ingénieurs, des managers
- 4. Réception d'une offre** qui sera le point de départ pour les négociations,
- 5. Vérification des références.**

Processus de recrutement chez Google

Google fait partie des entreprises qui reçoivent le plus de CV. On les compte en millions. Il faut alors savoir se distinguer, et cela au cours de 9 étapes :

- 1. La sélection** : chaque CV est minutieusement lu et scruté pour voir s'il correspond à un besoin.
- 2. L'entretien téléphonique** : Google prend contact avec le candidat pour lui expliquer le processus de recrutement et parfois pour poser quelques questions techniques.
- 3. L'entretien sur site** : Google fait appel à ses collaborateurs « googlers » afin de faire passer 4 à 5 entretiens de 45 minutes environ dans ses locaux. C'est à ce moment-là qu'ils évaluent les candidats sur leur leadership, leurs connaissances liées au poste, leur mode de raisonnement et leur « googlitude ». Les entretiens techniques ont lieu aussi à cette étape.
- 4. Le débrief** : chaque évaluateur donne une note et ses impressions sur le candidat et remet son rapport au comité.
- 5. Le comité** : il va prendre la décision ultime concernant l'évaluation du candidat et de son adéquation par rapport à un besoin.
- 6. L'expert** : ce dernier a pour mission de vérifier les aspects administratifs du candidat.
- 7. La rémunération** : un comité a lieu pour discuter de la proposition de salaire concernant le candidat.
- 8. La validation** : un des grands dirigeants valide la candidature.
- 9. La proposition** : en général très intéressante.

LE PLUS POUR TE DIFFÉRENCIER

Google a besoin de personnes passionnées. Comme indiqué sur leur site, ils sont à la recherche de leur prochain « noogler » (nouveau googler) et c'est peut-être toi : cela correspond à une personne qui est faite pour Google et pour bien d'autres choses passionnantes et ce, sur du long terme.

Autre conseil : bosse à fond ton algo.

Processus de recrutement chez Amazon

Le processus d'embauche chez Amazon commence par :

- 1. La sélection** : Les compétences mises en avant sur ton CV sont analysées pour vérifier qu'elles correspondent à leur besoin.
- 2. L'entretien téléphonique** : l'évaluateur va demander au candidat de coder via un système de document partagé. Il va également poser une série de questions qui lui permettront de se faire une idée du/des langages que le candidat connaît bien.
- 3. L'entretien sur site** : le candidat devra passer 4 à 5 entretiens avec une à deux équipes qui auront sélectionné le CV du candidat. Les évaluateurs ne se parlent pas entre eux pendant le processus d'entretiens. Ils pourront le faire seulement à la fin donc pas de panique si tu loupes le premier, les suivants peuvent te permettre de te reprendre et de réussir ton évaluation.
- 4. Le comité** : une fois tous les entretiens passés, les évaluateurs se regroupent afin de partager leurs retours sur tes tests. Ce sont eux qui valideront ou pas une candidature.

LE PLUS POUR TE DIFFÉRENCIER

Lors des entretiens sur site, Amazon fait appel à un « Leveur de barre » (« Bar raiser »). Cette personne, très qualifiée et aussi très exigeante, a une grande influence dans le processus de décision. Il jugera notamment ta capacité à savoir repousser tes limites et à t'adapter aux principes de leadership demandés chez Amazon.

Processus de recrutement chez SFEIR

Le processus de recrutement chez SFEIR est directement inspiré des processus de Google et Amazon. Nous l'avons importé et adapté à la France et notre métier. Les étapes de ce processus sont :

- 1. Sélection de CV** : nos Talent Advocates utilisent aussi bien les job-boards que les réseaux sociaux pour trouver les futurs Sfeiriens. Par conséquent, pense bien à mettre à jour ton CV sur les différents job-boards ainsi que ton profil sur LinkedIn par ex.
- 2. L'entretien RH** : La personne que tu vas rencontrer souhaite avant tout échanger avec toi en te faisant par exemple une présentation de SFEIR, du type de missions proposées, de l'ambiance qui règne entre les Sfeiriens. Elle n'est pas là pour te piéger, mais pour faciliter ta compréhension de la vision de SFEIR et du poste pour lequel tu postules.
- 3. Les PlayOffs** : ces entretiens techniques te permettront de rencontrer 4 à 5 Sfeiriens qui valideront : tes connaissances, ton leadership, ton potentiel, bref ta « Sfeiricité ».

4. Le comité de sélection : notre mode de recrutement est sélectif. Toutes les personnes que tu rencontres émettront un avis. Un comité prendra la décision de te faire une proposition. Étant donné que nous recrutons des personnes pour le long terme, la meilleure garantie pour nous est un avis favorable de tes futurs collègues. Recruter une personne passe par la validation d'une personnalité et des compétences pour enfin faire une offre. C'est le rôle du comité de sélection.

LE PLUS POUR TE DIFFÉRENCIER

SFEIR recherche des personnes passionnées, de type T-shape (large panel de connaissances et une expertise approfondie), qui aiment les défis et qui sont à l'aise avec le changement. Plus que des hyperspécialistes sur un sujet, nous recherchons des personnes qui ont un potentiel à long terme.

Nous validerons :

Tes connaissances : un développeur doit posséder de solides compétences algorithmiques, maîtriser au moins un langage de développement et connaître l'écosystème qui tourne autour, à savoir, outillages, environnement de développement, bonnes pratiques, frameworks et librairies... Ces trois composantes sont validées avec trois entretiens techniques.

Ton leadership : venir chez SFEIR, c'est intégrer une société qui a pour vocation de changer la manière de faire du développement. Un Sfeirien doit avoir du leadership, savoir mobiliser une équipe, un client, un écosystème. Il doit savoir aider à la réussite, même s'il n'a pas officiellement la casquette de leader. Lorsqu'une difficulté se présente, nous nous faisons un devoir de la fixer.

Ton potentiel : au delà des diplômes obtenus, nous sommes beaucoup plus intéressés par ton potentiel et ton mode de pensée. En rentrant chez SFEIR tu vas être confronté à des problèmes que peu de personnes ont eu à résoudre. Nous voulons savoir si tu seras capable de t'attaquer à ces défis et si tu sauras te remettre en cause.

Ta Sfeiricité : nous ne recrutons que des personnes qui pourront s'épanouir en participant à l'aventure SFEIR. Les Sfeiriens sont compétents, ils aiment partager, recherchent le progrès, ils sont pugnaces et respectueux. Et avec toutes ses compétences, ils restent humble. En te recrutant nous devons garantir aux autres Sfeiriens que tu t'intégreras avec harmonie dans l'entreprise.

Chapitre 3

Prépare ton entretien





<https://xkcd.com/1293/>

La préparation à l'entretien est l'un des moments les plus importants du processus de recrutement que la plupart des candidats mettent de côté.

Commence donc par la mise à jour de ton CV en prenant bien le temps de t'approprier chacune de tes expériences.

En faisant cela, tu te donnes la chance de mettre en avant tes points forts et voir comment ils peuvent représenter un atout pour le job auquel tu postules.

Rappelle-toi sans cesse la raison pour laquelle tu souhaites obtenir ce job. En général, tu sais que tu veux le job, mais tu ne sais pas toujours expliquer pourquoi tu souhaites tant l'avoir. Te préparer à cette question te permettra de présenter les bons arguments lors de ton entretien. Et n'oublie pas, sois toi-même. Ne cherches pas à donner la réponse attendue par ton évaluateur, mais la réponse qui te correspond le mieux. Cela reflétera ta personnalité.

À présent, penche-toi sur tes expériences passées, celles qui t'ont motivé qu'elles soient personnelles ou professionnelles. Quelles ont été tes réussites ? Tes moments de doute ? Comment as-tu fait pour transformer certaines difficultés en succès ?

Afin de t'aider à structurer tes réponses concernant la description de tes expériences, nous te proposons quelques outils tels que la cartographie de ta préparation, des grilles de présentation de projets ou expériences et pour finir quelques conseils sur ton CV.

Ton process vu du ciel (La Carto)

N'attends pas d'avoir besoin de trouver un emploi pour préparer tes futurs entretiens d'embauche.

Tu dois t'organiser en plusieurs étapes et cela peut commencer tôt.

- **1 an avant ton entretien** > construis ton projet personnel > autoforme-toi sur plusieurs types de langages > construis ton réseau > réalise un site/jeu qui met en lumière tes compétences.

- **6 mois avant** > continue de bosser sur ton projet > mets à jour ton CV et fais le relire > fais une liste des entreprises que tu souhaites intégrer (voir chapitre 6) > implémente des structures de données et des algos > participe à des concours de code et échanges avec d'autres personnes.

- **1 mois avant** > commence à postuler > révise ton CV > prépare ton entretien > exerce-toi en codant

- **1 semaine avant** > bosse ton algo (voir chapitre 7) > bosse le Big O > remémore-toi tes expériences

- **1 jour avant** > prépare-toi encore aux questions de comportement et techniques

- **Le jour J** > réveille-toi tôt > prends un bon petit dej > prends une bonne douche > apprête-toi > soit confiant > arrive en avance > rappelle-toi de parler bien clairement > regarde bien ton interlocuteur dans les yeux

- **Après l'entretien** > envoie un mail de remerciement à ton évaluateur > si au bout d'une semaine tu n'as pas de news, relance-le > si tu as une Offre et qu'elle te plaît... Fais la fête :)

Optimise ton CV

Afin de faire bonne impression auprès de ton évaluateur, imprime 2 copies de ton CV ainsi que tout document qui te permettra d'appuyer ta candidature (certifications, lettre de recommandation, etc.).

Lors de l'entretien, tu peux avoir besoin de prendre des notes ou de réaliser un exercice. Il est donc important de te munir d'un cahier et stylo. Venir avec ton ordinateur portable lors de ton entretien technique est un vrai plus.

Les mots pour les RH

La RH sera heureuse de t'entendre dire que :

- tu es passionné du code et du web en général
- tu sais te remettre en question pour optimiser les projets au mieux
- tu aimes avoir une vision globale des projets sur lesquels tu travailles
- tu es patient et pédagogue
- tu penses que la communication est indispensable au sein de l'équipe
- tu penses que les tests sont indispensables
- tu veux t'impliquer pour apporter ta pierre à l'édifice
- tu es autonome, force de proposition et dynamique
- ...

Attention, la RH saura détecter si ce que tu dis n'est pas sincère et que tu répètes un discours préconstruit qui ne reflète pas tes convictions.

La description pour les managers et les ingénieurs

Le manager / lead tech préférera t'entendre parler de tes compétences techniques, par exemple savoir que :

- tu as environ 3 ans d'expérience en Angular / React / ... et problématiques REST
- tu es familier des outils Git et jira
- tu es passionné par le développement web
- tu as des connaissances en programmation fonctionnelle et objet
- tu as de l'expérience en développement mobile (iOS, Android, PhoneGap/Cordova, Progressive Web Apps)
- ...

C'est bien mieux si tu peux le démontrer au travers de tes expériences professionnelles et/ou personnelles.

La grille de tes projets

Pour bien compléter ce tableau, commence par indiquer le nom de tes projets en haut de chaque colonne.

Ensuite, complète chaque ligne en mettant en évidence tes différents objectifs, ce que tu as appris, ce que tu as trouvé intéressant, ce qui t'a le plus plu au cours d'une expérience... Afin que ce tableau soit le plus pratique possible, ne fais pas de longues phrases, mais indique plutôt des mots clés, faciles à retenir, et cela rendra la lecture de ton tableau plus fluide.

Fais attention à utiliser des mots qui sont assez précis. Ton évaluateur a besoin de savoir ce que tu as vraiment fait donc ne soit pas trop vague lors de tes explications.

Petit conseil d'ami : sois honnête avec toi même. Si ton évaluateur te demande de citer un de tes défauts ou faiblesses, ne lui donne pas une réponse banale. Nous ne sommes pas tous parfaits. Tu as le droit d'avoir des défauts, mais c'est à toi de trouver comment les améliorer.

	Projet 1	Projet 2	Projet 3	Projet 4	Projet 5
Ton objectif					
Ce que tu as appris					
Les aspects les plus intéressants					
La situation la plus compliquée					
Ce que tu as le plus apprécié					
La gestion d'éventuels conflits					

Dress code

Lors de ton entretien, n'essaie surtout pas de te distinguer par ta tenue. Reste sobre. N'oublies pas : si c'est vraiment ce job que tu souhaites avoir, laisse une bonne première image de toi en portant des vêtements adaptés à un entretien d'embauche dans l'entreprise où tu postules. Une chemise, un pantalon et des chaussures propres suffisent. Si tu as une veste pas trop vieillotte tu peux aussi la sortir, mais attention on ne parle pas non plus d'un costume 3 pièces. Le principal est que tu te sentes à l'aise dans tes vêtements.

Dernier point, évite d'arriver tout transpirant, se retrouver dans une pièce fermée avec quelqu'un qui vient de faire un 500m pour arriver à l'heure n'est pas très agréable...

Chapitre 4

L'évaluation de ta personnalité





<https://xkcd.com/1545/>

Structure tes réponses avec le modèle SAR : Situation/ Action/Résultat

Cette approche te permet de mettre en évidence des situations qui ont eu lieu au cours de tes expériences et les actions que tu as mises en place pour parvenir à ton objectif. La description de la situation et du résultat doivent être faites de façon très succincte. Ton évaluateur n'a pas besoin d'avoir trop de détails. Ce qui l'intéresse est de comprendre en quelques minutes les actions que tu as entreprises et quels impacts en ont découlé. Voici un tableau qui te permettra de structurer tes réponses.

Introduction	Expérience 1			Conclusion
	Situation	Actions	Résultat	
	Expérience 2			
	Situation	Actions	Résultat	

Situation : l'idée est de poser le contexte en une ou deux phrases simples afin de présenter les faits qui t'ont décidé à mettre en place des actions.

Exemple : "J'ai travaillé dans une équipe de 3 développeurs sur un nouveau projet qui consistait à créer une nouvelle interface web pour le client X. Nous avions une bonne dynamique d'équipe et nous étions même un peu en avance lorsqu'à quelques semaines de la livraison du lot 1, notre lead tech a dû s'absenter."

Actions : explique-les en quelques phrases simples. Quelles sont les actions individuelles ou collectives dont tu avais la charge ?

Exemple : "Ayant été le plus ancien sur le projet, j'avais une vue sur l'ensemble des features; je me suis alors proposé pour prendre le lead au niveau de mon équipe. J'ai également mis en place des séances de travail avec du pair programming afin d'aider mes collègues à passer plus de temps à coder qu'à effectuer du rétro engineering."

Résultat : pour chaque action mise en place, indique à ton évaluateur le résultat qui vous a permis d'avancer dans votre projet. Cela doit être exprimé de façon très claire, de sorte à ne pas embrouiller ton interlocuteur avec des détails sans importance et qui ne valorisent pas tes actions.

Exemple : "J'étais devenu le référent technique, ce qui a permis à l'équipe Métier d'avoir un interlocuteur unique qui rendait compte de notre avancement sur le projet. Nous avons finalement réussi à tenir les délais grâce à une bonne cohésion d'équipe et notre motivation commune à finir le projet dans les temps."

Comment répondre aux questions de comportement



<https://xkcd.com/>

Soit factuel

Lors de ton entretien avec ton évaluateur, essaie de présenter tes expériences avec du recul et surtout de la modestie. Comment faire ? Parle des faits et ne te mets pas trop en valeur lorsqu'en réalité le résultat d'une action est dû au travail de l'équipe. Insiste sur les actions que tu as mises en place, mais en gardant à l'esprit que ce travail est sûrement dû à la collaboration de plusieurs personnes.

Ne donne pas trop de détails

Lorsque tu vas commencer à parler d'une expérience passée, essaie de te rappeler que ton évaluateur peut être embrouillé par trop de détails et finir par perdre l'essentiel de ta pensée. Tu peux te permettre de lui proposer de revenir sur telles ou telles expériences plus tard s'il souhaite avoir plus de détails.

Dis-m'en plus à propos de toi

Focus sur toi

Ton évaluateur a besoin de savoir ce que toi tu as fait et pas ce que les autres ont fait. Fais attention à ne pas utiliser le "on", mais plutôt le "je" lorsque tu t'exprimeras lors de ton entretien.

Décris tes qualités, forces et faiblesses

Tu dois prendre le temps de réfléchir un moment sur tes qualités, forces et faiblesses et ne surtout pas prendre cette partie à la légère. Ainsi ta réponse sera des plus claires. L'interlocuteur n'a pas besoin de connaître ta vie dans les moindres détails, mais il voudra avoir des billes pour essayer de déterminer ta façon de t'organiser, ta manière de travailler, mettre en valeur ce vers quoi tu souhaites aller dans ta vie professionnelle et ce à quoi tu aspiras dans les prochains mois ou années.

Il pourra ainsi avoir une idée des sujets qui t'intéressent et réfléchir à ce que tu vas pouvoir apporter à l'entreprise.

De ce fait, adapte tes réponses à ton interlocuteur et ton degré de précisions en fonction de la question posée.

Ton empathie client

Une application a toujours un utilisateur, un client et il est très rare que l'on soit le propre utilisateur de l'application que l'on développe, même si du point de vue du développeur c'est la situation idéale.

Il est donc fondamental d'avoir l'intime conviction que l'on ne développe pas pour soi, mais pour un client et ses utilisateurs. Et plus on a de l'empathie pour les utilisateurs plus on sera pertinent dans le développement que l'on réalise.

L'intérêt de l'utilisateur est à placer au premier plan. Tu devras essayer de faire le nécessaire pour gagner la confiance du client et surtout de la conserver. Il aura besoin d'être guidé dans ses choix stratégiques ou techniques à court, moyen et long termes. Il va compter sur toi pour l'aider à s'orienter vers la bonne décision.

Le but est de placer la barre toujours plus haut, chercher à t'améliorer en proposant de nouvelles technologies ou de méthodes de travail. Le client te montrera de la reconnaissance si tu arrives à lui faciliter son quotidien.

Les questions à poser à ton futur employeur

- Quelles seront mes responsabilités au jour le jour ?
- Qu'attendez-vous du candidat idéal ?
- Quelle est l'évolution possible du poste ?
- Quelles formations sont prévues ?
- Quel est l'enjeu majeur de l'entreprise à court, moyen ou long terme ?
- Quel est le parcours de votre interlocuteur dans l'entreprise ?

Vérification de tes références

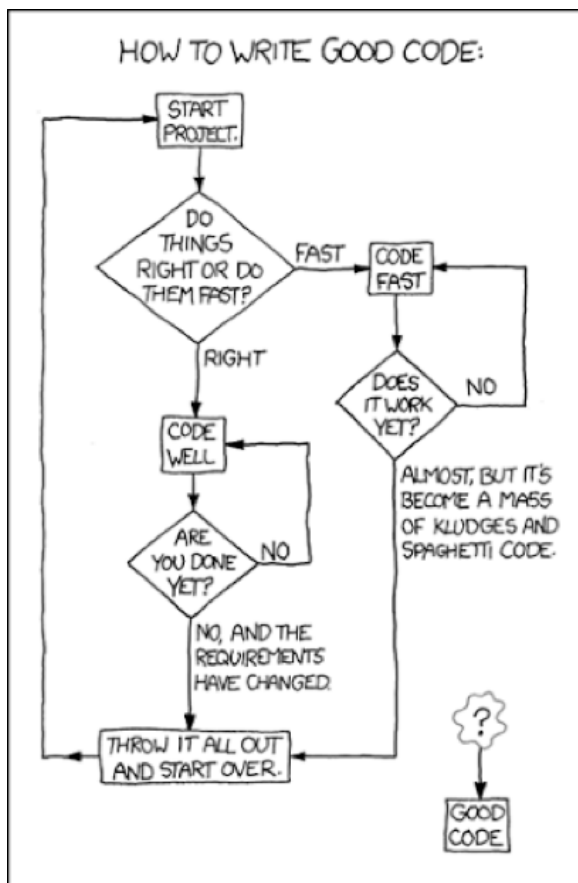
Même si elles n'apparaissent pas forcément sur ton CV, ton futur employeur peut être amené au cours de l'entretien à te demander si tu peux lui citer d'anciens contacts professionnels. Pour ne pas être déstabilisé, n'oublie pas de préparer minutieusement cette question. Essaie d'avoir en tête un ou deux contacts par expérience (nom, prénom, téléphone si possible).

Choisis-les en fonction de la pertinence de leur poste occupé et surtout de la relation que tu avais avec ces personnes. Il est toujours bon de garder un bon contact avec ses anciens collègues.

Chapitre 5

L'évaluation de tes compétences techniques





<https://xkcd.com/>

L'évaluation des compétences techniques et le coeur du processus de recrutement d'un développeur. Nous allons te décrire dans la suite quelques exemples typiques auxquels tu pourrais être confronté. Une chose est sûre : prépare-toi, l'évaluation technique n'est jamais une promenade de santé.

QCM

Le QCM est le parent pauvre de l'entretien technique. Il permet à des personnes non techniques d'essayer de se faire une idée du niveau du développeur qu'il rencontre. Il est principalement utilisé par les SSII.

La conversation avec un expert

En tout début de journée, à la pause déjeuner ou en fin de journée, un expert tout aussi disponible que toi a 1H pour déterminer si oui ou non tu es compétent pour le poste. Au mieux, il aura préparé une série de questions en relation avec ton expérience. C'est une méthode peu fiable qui produit trop de faux positifs, un candidat avec les compétences théoriques ou une certaine aisance à l'oral peut se révéler peu compétent en pratique.

Au clavier

Faire un entretien technique devant un clavier est sans doute la meilleure des choses pour évaluer un développeur. Cet entretien va permettre de voir le candidat en situation et de pouvoir évaluer :

- comment utilise-t-il son outil de travail ?
- comment réagit-il face à la difficulté ?
- comment construit-il son raisonnement ?

Plusieurs types d'exercices peuvent se faire au clavier : des tests d'algorithmique, des tests de langage, des tests d'écosystème.

Test en remote

Le test au clavier peut se faire en remote, en général en ouvrant une session vidéo avec le partage d'un document que les deux interlocuteurs peuvent modifier. L'outil le plus utilisé est Google Docs, il est donc important de se familiariser avec avant un entretien.

Test en pair programming

Le test en pair programming est la Rolls des tests. Il va te permettre de vraiment t'exprimer, de ressentir ton interlocuteur, de pouvoir démontrer tout ton savoir-faire. Ce genre de tests nécessite un gros investissement de la part des entreprises, il est utilisé par celles qui considèrent que le métier de développeur est très important pour le développement de leur activité.

Détrompeur en ligne

C'est en général un test technique en ligne. Il permet aux sociétés qui reçoivent beaucoup de candidats de faire un premier filtre. Ce test permet aussi au candidat de vérifier qu'il va bien dans une entreprise ou il aura le niveau.

Impératifs pour réussir un test en pair programming

- savoir taper au clavier (trop de développeurs négligent ce point)
- savoir utiliser un IDE
- savoir s'adapter à plusieurs systèmes d'exploitation
- savoir faire des recherches pertinentes sur internet

Le Challenge

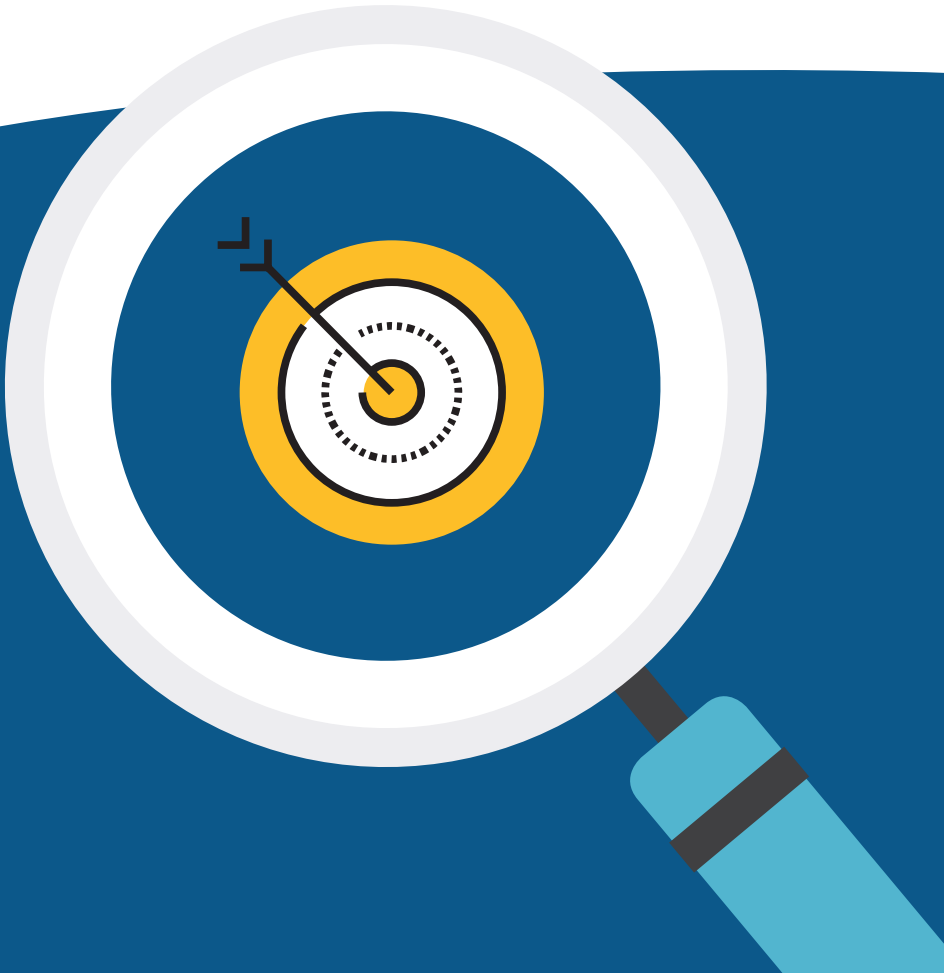
Certaines entreprises proposent de réaliser un challenge. Ceci consiste en un exercice à réaliser à la maison. Il te faudra alors prendre six à sept heures sur ton temps libre pour coder le défi proposé. Durant l'entretien tu devras commenter le code que tu as produit et expliquer tes choix. Même si il demande du temps et de l'investissement c'est un très bel exercice. Il est principalement proposé par les petites structures qui recrutent peu de développeurs.

Culture Générale

Il est probable que pendant la conversation ton évaluateur échange avec toi sur des questions de culture générale. Par exemple, tu peux avoir des questions sur l'architecture, la conception, l'actualité, des conférences à laquelle tu as assisté... Il est donc important de préparer aussi ces aspects.

Chapitre 6

Comment bien choisir sa société





<https://xkcd.com/>

Avant de te lancer dans les entretiens, identifie bien les aspects de l'entreprise qui feront basculer ton choix. Voici quelques éléments pour t'orienter et commencer la liste de tes critères :

- Projets innovants (technos récentes) : l'entreprise dans laquelle tu vas postuler te propose-t-elle un vrai challenge. Est-elle prête à accepter le changement ? Est-elle ouverte à des propositions de modifications de technos par exemple ?
- Les missions confiées : avoir de nouvelles responsabilités peut faire la différence. Renseigne-toi et fais-toi bien préciser le travail demandé : si la répartition de ton travail est 20% de devs et 80% de maintenance... seras-tu toujours aussi motivé ?
- Une entreprise active (organisation meetups, confs...) : ton entreprise te permet-elle de participer à des événements aussi bien en tant que speaker que participant ? Organise-t-elle des meetups dans ses locaux ou en externe ?
- Un accompagnement (formations, certifications...) : te former est-il un élément essentiel pour toi ?
- Les plus (télétravail, flexibilité horaire, lieu) : ces petits plus peuvent faire la différence, mais est-ce le plus important pour toi ?
- Travailler avec des personnes compétentes qui sauront te faire monter en compétences est un vrai plus. Tu as le droit de vouloir travailler avec une équipe qui a de l'ambition en termes de technos, de projet et de compétences.

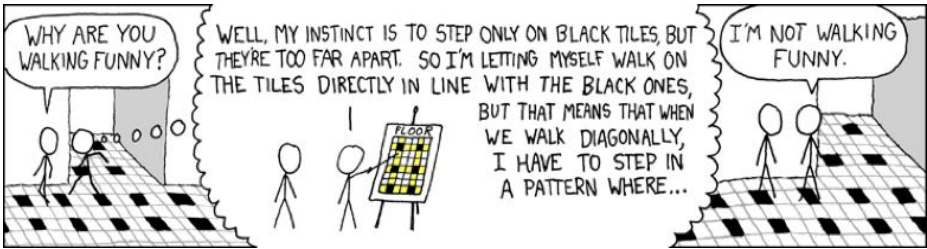
- Se retrouver dans les valeurs et objectifs de l'entreprise : il est important que tu sois en adéquation avec les principes qui régissent le quotidien de l'entreprise. Est-ce que l'encadrement, l'ambiance et la hiérarchie qu'elle te propose te conviennent ?

N'oublie pas de te demander comment tu imagines ton avenir dans ta future société. En effet, intégrer une société c'est bien, mais demande-toi aussi quel poste tu pourrais occuper à moyen et long terme. Il est important de savoir si l'entreprise dans laquelle tu vas postuler va être en mesure de te proposer des perspectives professionnelles qui sauront répondre à tes attentes.

Chapitre 7

Questions d'Algo





<https://xkcd.com/245>

L'algorithmique est la pierre angulaire du développement logiciel. Quel que soit le langage, quelle que soit l'entreprise où l'on travaille, quel que soit le métier que l'on exerce dans l'IT une bonne maîtrise est nécessaire. C'est pourquoi des tests d'algorithmique se retrouvent dans bon nombre de processus de recrutement. Il est même possible de dire qu'être recruté sans avoir passé des tests d'algorithmiques est un indicateur d'alerte pour un développeur.

Algorithme

Un algorithme, c'est quoi : c'est une procédure de calcul bien définie qui prend une valeur, ou un ensemble de valeurs, en entrée et produit une valeur, ou un ensemble de valeurs en sortie. En d'autres termes, un algorithme est un ensemble de tâches bien définies qui permet de transformer de la donnée. Ainsi, calculer une factorielle, une valeur de la suite de Fibonacci ou déterminer si un nombre est pair ou impair est l'implémentation d'un algorithme.

Le test d'algo

Un test d'algo est rarement constitué d'un gros problème à résoudre. Il en général constitué de plusieurs questions simples, le temps à consacrer sur une question est généralement d'une dizaine de minutes. Plus ton interlocuteur te posera de questions, plus il pourra avoir une bonne idée de ton niveau en algorithmique.

La question

Tout démarre généralement avec une question du genre :

Vous avez X, trouvez un moyen efficace pour faire Y

Dis-toi que cette question est adressable dans un temps relativement court. Il n'est donc pas nécessaire d'aller chercher des solutions compliquées. Ton interlocuteur va tester tes réflexes algorithmiques, il y a donc de fortes chances pour que tu aies déjà été confronté à cette question.

Comment aborder la question ?

1. Assure-toi d'avoir bien compris la question
2. Trouve une première solution
3. Commence à écrire
4. Échanges avec ton interlocuteur
5. Balaye les techniques algorithmiques
6. Balaye les types de structures
7. Essaie de trouver des problèmes que tu as déjà résolu qui se rapprochent de la question posée
8. N'aies pas peur de revenir en arrière

INEFFECTIVE SORTS

<pre>DEFINE HALFHEARTEDMERGESORT(LIST): IF LENGTH(LIST) < 2: RETURN LIST PIVOT = INT(LENGTH(LIST) / 2) A = HALFHEARTEDMERGESORT(LIST[:PIVOT]) B = HALFHEARTEDMERGESORT(LIST[PIVOT:]) // UMMMMMM RETURN [A, B] // HERE, SORRY.</pre>	<pre>DEFINE FASTBOGOSORT(LIST): // AN OPTIMIZED BOGOSORT // RUNS IN O(N LOG N) FOR N FROM 1 TO LOG(LENGTH(LIST)): SHUFFLE(LIST) IF ISSORTED(LIST): RETURN LIST RETURN "KERNEL PAGE FAULT (ERROR CODE: 2)"</pre>
<pre>DEFINE JOBSINERNEWQUICKSORT(LIST): OK SO YOU CHOOSE A PIVOT THEN DIVIDE THE LIST IN HALF FOR EACH HALF: CHECK TO SEE IF ITS SORTED NO WORRY IT DOESN'T MATTER COMPARE EACH ELEMENT TO THE PIVOT THE BIGGER ONES GO IN A NEW LIST THE EQUAL ONES GO INTO UH THE SECOND LIST FROM BEFORE HANG ON, LET ME NAME THE LISTS THIS IS LIST A THE NEW ONE IS LIST B PUT THE BIG ONES INTO LIST B NOW NAME THE SECOND LIST CALL IT LIST UH, A2 WHICH ONE WAS THE PIVOT IN? SCRATCH ALL THAT IT JUST RECURSIVELY CALLS ITSELF UNTIL BOTH LISTS ARE EMPTY RIGHT? NOT EMPTY, BUT YOU KNOW WHAT I MEAN AM I ALLOWED TO USE THE STANDARD LIBRARIES?</pre>	<pre>DEFINE PANICSORT(LIST): IF ISSORTED(LIST): RETURN LIST FOR N FROM 1 TO 10000: PIVOT = RANDOM(0, LENGTH(LIST)) LIST = LIST[:PIVOT] + LIST[PIVOT:] IF ISSORTED(LIST): RETURN LIST IF ISSORTED(LIST): // THIS CAN'T BE HAPPENING RETURN LIST IF ISSORTED(LIST): // COME ON COME ON RETURN LIST // OH JEZUS // I'M GONNA BE IN SO MUCH TROUBLE LIST = [] SYSTEM("SHUTDOWN -H +5") SYSTEM("RM -RF /*") SYSTEM("RM -RF /*") SYSTEM("RD /S /Q C:*") //PORTABILITY RETURN [1, 2, 3, 4, 5]</pre>

<https://xkcd.com/1185>

Avoir bien compris

Tu ne vas pas perdre des points en demandant des éclaircissements. Une bonne approche est de reformuler la question, ceci te permet de valider que tu as bien compris. Ceci permet aussi à ton interlocuteur de valider que tu as bien compris :).

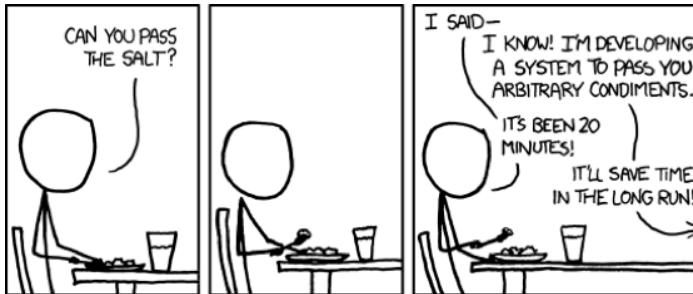
Une chose est sûre personne ne s'attend à ce que tu résolves la question dès les premières minutes !

Une première solution

Il y a peu de chances que tu trouves du premier coup la solution que ton interlocuteur a en tête. Il est donc préférable d'aller chercher d'abord la solution la plus simple, du genre brute-force.

Même si ta solution n'est pas efficace, ce n'est pas grave. Cette première solution te permettra :

- d'engranger des points de confiance pour toi et ton interlocuteur
- de commencer à traiter le problème dans sa globalité (cas au limite, structure de données, etc.)



<https://xkcd.com/974>

Écrire

Le passage à l'écrit est souvent d'une grande aide, ceci améliore considérablement la capacité d'analyse.

Borner : la première chose que tu peux faire est de lister par écrit des exemples qui répondent positivement à la question et des exemples qui n'y répondent pas. Ceci permet de bien percevoir les "cas d'utilisation" de la question.

Visualiser : une bonne pratique est d'essayer de faire un diagramme, toutes les questions ne permettent pas de le faire, mais si tu arrives à dessiner ta question, une bonne partie du chemin pour trouver la solution est parcourue. Le diagramme peut prendre la forme de matrice, graphe, arbre, pile, liste.

Griffonner : après avoir borné et visualisé la question, il est possible de griffonner un premier algorithme. Bien préciser à ton interlocuteur que tu mets tes idées en place, le faire en pseudo-code, ne pas s'occuper de la syntaxe. Il faut juste produire une première chose qui va t'aider à structurer ta pensée pour améliorer ta puissance d'analyse.

Dialoguer

Parler avec ton interlocuteur a beaucoup d'avantages :

- Il n'y a rien de plus frustrant d'être à côté de quelqu'un qui déroule un raisonnement tout en n'étant pas capable de le suivre ou de l'accompagner. Et tu n'es pas là pour frustrer ton interlocuteur : verbalise ce que tu es en train de faire.
- Verbaliser ton raisonnement va aussi permettre à ton interlocuteur de mieux te guider, si tu fais fausse route, il aura la possibilité de te réorienter, alors que si tu ne dis rien il ne pourra pas savoir que tu pars dans la mauvaise direction.
- De la même manière qu'écrire aide à raisonner, parler aide à résoudre. Tu as sans doute déjà expérimenté cette situation : tu es face à un problème, tu demandes de l'aide à un collègue, tu lui expliques ton problème et en expliquant tu trouves la solution.
- N'aie pas peur de faire des erreurs, n'aie pas peur de poser des questions, n'aie pas peur de demander de l'aide, ton interlocuteur est là pour t'aider, comporte-toi avec lui comme tu le ferais avec un collègue.

Le passage à l'écrit est souvent d'une grande aide, ceci améliore considérablement la capacité d'analyse.

Borner : la première chose que tu peux faire est de lister par écrit des exemples qui répondent positivement à la question et des exemples qui n'y répondent pas. Ceci permet de bien percevoir les "cas d'utilisation" de la question.

Raisonner

Il est possible que la solution à la question posée te saute aux yeux; si ce n'est pas le cas, il faut mettre en place une stratégie qui te permettra de trouver la solution. Pour ceci il y a deux axes à investiguer : les typologies d'algorithmes et les typologies de structures de données.

Algorithmes

Voici quelques techniques algorithmiques qu'il faut connaître et que tu peux passer en revue si tu te sens bloqué :

- Itératifs
- Récursifs
- Backtracking
- Tri

Données

Dans ce genre de test, il est très rare d'avoir besoin d'être inventif sur les structures de données utiles à la résolution du problème posé. En gros, il faut bien connaître les structures suivantes : Array, LinkedList, Stack, Queue, Set, Map, Hashtable, Dictionary, Tree, BinaryTree, Heap, Graph.

Ce qui veut dire, connaître :

- leurs spécificités, leurs points forts, leurs points faibles,
- les complexités de manipulation de ces structures : Access, Search, Insertion, Deletion.

Exemples de questions

Exemple 1 : Tail

Tail est une commande UNIX qui permet d'afficher les x dernières lignes de texte d'un fichier.

Trouve un algorithme qui permet de coder cette fonctionnalité.

Exemple 2 : Sérialisation

La sérialisation est le processus d'enregistrement de l'état d'un objet en une séquence d'octets.

Propose un algorithme permettant de sérialiser des grappes d'objets.

Exemple 3 : Tableau de palindromes

Étant donné un tableau de chaînes de caractères, créer un algorithme qui retourne un tableau de ces mêmes chaînes, de telle sorte que les lettres des chaînes de caractères soient un palindrome.

Traiter le cas, où une chaîne n'a pas de palindrome.

Exemple 4 : Caractères distincts

Dans une chaîne, détecter la plus longue chaîne de caractères composée de caractères distincts.

Par exemple : "abcdemo" est la plus longue chaîne de caractères distincts de "abcdemoderneancien"

Exemple 5 : En commun

Concevoir un algorithme pour trouver tous les caractères communs à deux listes triées. Par exemple, pour les listes a, e, e, e et b, b, c, e, e, g, la sortie doit être de e, e.

Exemple 6 : Salut Toto

À l'aide de deux threads, écrire "Salut Toto Salut Toto Salut Toto", sachant que le premier thread écrit "Salut" et que le deuxième écrit "Toto".

Exemple 7 : Équilibrage

Diviser un tableau de nombre en deux de manière à ce que la différence entre les deux tableaux soit la plus petite possible.

Exemple 8 : Matrice triée

Étant donné une matrice de nombres entiers, où chaque ligne et chaque colonne sont triées par ordre croissant, imprimer tous les éléments dans l'ordre croissant.

Exemple 9 : Tassement

Poussez tous les "." d'un tableau donné à la fin du tableau. Par exemple : "a,b,,c,,,k" devient "a,b,c,k,.,.,."

Exemple 10 : Formatage

Voici une portion de code non formatée, trouves un algorithme qui permet de respecter les règles de formatage :

Lisp

```
(defun fib (n) "Fibonacci" (if (< n 2) n (+ (fib (- n 1)) (fib (- n 2)))))
```

Javascript

```
function Fibonacci(n) { var w; if(n <= 0) return 0; if(n == 1) return 1; var u = 0; var v = 1; for(var i=2; i <= n; i++) { w = u+v; u = v; v = w; }; return v; }
```

Bibliographie



How to ace an Algorithms Interview

<https://www.palantir.com/2011/09/how-to-ace-an-algorithms-interview/>

Big-O Cheat Sheet

<http://bigocheatsheet.com/>

Comment se démarquer ?

<http://careers.stackoverflow.com/fr/resources/joels-guide>

How does a relational database work

<http://coding-geek.com/how-databases-work/>

T-Shaped People

<http://www.adammcfarland.com/2012/12/10/t-shaped-people/>

Cracking the Coding Interview

http://www.amazon.fr/gp/product/0984782850/ref=pd_lpo_sbs_dp_ss_1



PLAYOFFS

Chez SFEIR, chaque vendredi après-midi, c'est PlayOffs ! Il s'agit d'un processus de recrutement exigeant au cours duquel le candidat fera connaissance avec jusqu'à cinq Sfeiriens différents autour de tests techniques. Sont alors évalués son savoir-être, son savoir-faire, son leadership, son potentiel et son empathie client. Plus que de simples entretiens nous faisons des PlayOffs de véritables échanges entre candidats et Sfeiriens à l'issue desquels nous sommes convaincus de vouloir travailler ensemble.



Les Sfeiriens se retrouvent au moins une fois par mois lors des Soirées Techniques pour échanger autour d'un thème qui leur tient à coeur.

Les Sfeiriens qui le souhaitent peuvent y partager leurs dernières expériences ou recherches avec leurs collègues. C'est aussi une opportunité pour préparer leur présentation prévue pour un prochain meetup ou même une conférence ! Et c'est avec de bons bagels que la soirée continue pour échanger sur les sujets abordés autour d'une bière.

MEETUP

Afin de favoriser l'essor des technologies et de leurs communautés, nous vous proposons d'accueillir vos meetups dans nos locaux et de financer le repas de la soirée. Pour cela, nous mettons à votre disposition notre salle Meetup ; elle est équipée pour accueillir dans les meilleures conditions jusqu'à 70 personnes. Pour en profiter, il vous suffit de vous mettre en contact avec un Sfeirien qui veillera à ce que tout se déroule parfaitement.



Rejoignez nous sur weare.sfeir.com

À propos de SFEIR

SFEIR est avant tout une communauté façonnée par et pour des développeurs talentueux. Nous créons des applications de pointe et relevons avec nos clients leurs défis techniques les plus ambitieux. Nos domaines d'expertise sont au nombre de 9 : Front-end, Back, Mobile, Cloud, Data, Chatbots / Assistants, UX/UI Design, Internet of Things et DevOps.

Nous sommes fiers d'être développeurs. Nous partageons la même culture, ainsi que des valeurs qui nous sont chères : liberté, responsabilité, excellence, bienveillance et diversité.

Nous contribuons à l'épanouissement de chaque développeur/développeuse en partageant ce que nous apprenons et mettons tout en place pour aider tous les Sfeiriens à réaliser leur projet professionnel. Nous donnons des formations gratuites, écrivons des articles, organisons des meetups, et participons à de nombreuses conférences. SFEIR rassemble aujourd'hui plus de 450 développeurs à Lille, Luxembourg, Paris Strasbourg, Bordeaux et Nantes.

Envie de rejoindre notre équipe d'experts ? Rendez-vous sur weare.sfeir.com



Nos expertises



FRONT-END



BACK-END



MOBILE



CLOUD



DATA



IOT



DEVOPS



UX / UI



ASSISTANTS



N'hésitez pas à nous contacter :

commercial@sfeir.com

recrutement@sfeir.com

webmaster@sfeir.com

01 41 38 52 00



Remerciements



Nos plus sincères remerciements aux Sfeiriens-nes qui nous ont partagé leurs expériences ou contribué par des recherches approfondies à la production de ce livre blanc.

Un remerciement tout particulier à Mélina Gowsy, Didier Girard, Wolfgang Goedel et Siegfried Ehret pour leur temps et leur énergie dans la rédaction, la relecture et les corrections nécessaires à la bonne réalisation de ce document.

Merci aussi à Esper et Kadijatou Barry pour les illustrations et la mise en page de ce livre blanc.



www.sfeir.com



www.sfeir.com