

**ÉCRIT PAR**

**Charles DUPAIX  
Paul GUIBERT  
William TRUONG**

**DATE  
2024**

**Digital Green x TheCodingMachine**

**Livre Blanc - L'Eco-conception de services numériques**

**Comment développer une application web ou mobile de manière écologique  
et s'inscrire dans une logique de numérique responsable ?**

## **Note sur cette version :**

Ce Livre Blanc s'adresse à l'ensemble des profils contribuant à n'importe quel niveaux à la création et production d'un service numériques : développeurs, concepteurs de produits numériques (UX/UI Designer, Product Owner...), chefs de projet mais aussi aux décideurs d'entreprises soucieux de s'inscrire dans une démarche écoresponsable.

Il est particulièrement pertinent pour ceux qui aspirent à concilier innovation technologique et respect de l'environnement, en intégrant les principes de l'éco-conception dans le développement de leurs applications web et mobiles.

Alors, que vous soyez à la tête d'une start-up innovante, membre d'une équipe de développement dans une grande entreprise, ou acteur dans le secteur public désireux de promouvoir le numérique responsable, ce livre blanc offre des clés de compréhension, des méthodologies et des exemples pratiques pour vous guider dans la réalisation de projets numériques plus verts.

En adoptant les conseils et meilleures pratiques présentés, vous contribuerez non seulement à réduire l'empreinte écologique de vos services numériques, mais aussi à sensibiliser et inspirer votre entourage professionnel à adopter une approche plus durable du numérique.

Si vous avez des remarques, des questions ou même des projets dans ce domaine, n'hésitez pas à prendre contact avec nous !

# INTRODUCTION

## Le numérique, bien loin d'être "Green" :

Les chiffres du numérique qui datent déjà de 2019 (Étude de GreenIT.fr « Empreinte environnementale du numérique mondial ») sont clairs, le secteur est loin d'être « green » :

- 34 milliards d'équipements informatiques (smartphones, ordinateurs, enceintes connectées, etc.) pour 4,1 milliards d'utilisateurs, soit 8 équipements par utilisateur ;
- 3,8 % des émissions de gaz à effet (GES) de serre au niveau mondial, ce qui équivaut à peu près à l'empreinte carbone mondiale de l'aviation ;
- 4,2 % de la consommation d'énergie primaire ;
- 5,6 % de la consommation d'électricité ;
- Entre 59 % et 84 % des impacts du numérique mondial viennent des équipements des utilisateurs.
- Si le numérique était un pays, il aurait environ 2 à 3 fois l'empreinte environnementale de la France.

De plus, selon les différentes projections, le numérique pourrait représenter rapidement 8 % des émissions de gaz à effet de serre en 2025. En particulier car les nouvelles technologies, en particulier l'IA et la Blockchain, sont extrêmement gourmandes en ressources. En termes d'ordre de grandeur, 8% représente à peu près les émissions mondiales du secteur automobile.

Il devient donc crucial d'avoir une approche plus rationalisée du numérique en particulier là où c'est possible et pas trop compliqué, à savoir en priorité son usage !

# L'éco-conception, une simple tendance ou un vrai changement ?

L'éco-conception est souvent perçue comme une tendance émergente, portée par la vague grandissante de l'écologie et de la prise de conscience environnementale. Cependant, réduire l'éco-conception à une simple mode serait sous-estimer son potentiel révolutionnaire. Bien que son adoption actuelle puisse sembler timide ou dérisoire, il est impératif de reconnaître l'éco-conception pour ce qu'elle devrait réellement être : un changement profond et durable dans notre manière de concevoir, produire, et consommer. Chaque individu, entreprise, et institution a un rôle crucial à jouer dans cette transformation.

Il ne s'agit pas seulement d'adopter des alternatives écologiques lorsque cela est pratique ou en vogue, mais de s'engager dans une démarche de responsabilité et de durabilité à long terme. L'éco-conception doit évoluer d'une tendance passagère à un pilier fondamental de notre société, guidant nos choix et nos innovations vers un avenir plus responsable. Ce passage à une réelle éco-conception exige de tous un engagement actif, une remise en question de nos habitudes, et une volonté d'adopter des pratiques qui soutiennent la santé de notre planète pour les générations actuelles et futures.

Cette prise de conscience s'est d'ailleurs déjà traduite en norme à travers la loi du 15 novembre 2021 (dite loi REEN) visant à réduire l'empreinte environnementale du numérique en France. Cette loi interdit entre autre :

- Aux fabricants de rendre impossible la restauration de l'ensemble des fonctionnalités d'un terminal réparé ou reconditionné ;
- Les techniques empêchant le consommateur d'installer les logiciels ou les systèmes d'exploitation de son choix sur son appareil au bout d'un délai de deux ans.

La loi REEN oblige toutes les collectivités de plus de 50.000 habitants à établir une stratégie numérique responsable. L'éco-conception est donc devenu un des critères de décision des appels d'offres publics.

# Table des matières

<b>INTRODUCTION</b>	<b>3</b>
Le numérique, bien loin d'être "Green" :	3
L'éco-conception, une simple tendance ou un vrai changement ?	4
<b>1. C'EST QUOI UNE APPLICATION WEB, UN SITE, "ÉCOLOGIQUE" ?</b>	<b>7</b>
1.1 Comment analyser l'empreinte écologique du numérique	7
1.2 Une application qui vise la simplification des parcours utilisateurs	9
1.3 Une application qui est éco...nome !	10
<b>2. COMMENT INCLURE L'ÉCO-CONCEPTION DÈS LA GENÈSE DE SON PROJET</b>	<b>11</b>
2.1 L'approche MVP - Minimum Viable Product	11
2.2 L'importance de l'étude des parcours utilisateurs	12
2.3 Les aspects graphiques	13
2.4 Le choix des technologies pour votre projet	13
<b>3. RÉFÉRENTIELS ET BONNES PRATIQUES</b>	<b>14</b>
3.1 Dix optimisations simples et impactantes	14
3.2 Pour aller plus loin : les sources de références en matière d'éco-conception	19
<b>4. QUELS INDICATEURS POUR ÉVALUER L'ÉCO-CONCEPTION ?</b>	<b>21</b>
4.1 Les indicateurs techniques	21
4.2 Les indicateurs d'utilisation	25
4.3 Les outils de mesure	25
4.4 Tableau récapitulatif des indicateurs :	31
<b>5. LES BIENFAITS DE L'ÉCO-CONCEPTION !</b>	<b>31</b>
5.1 La simplicité d'utilisation et l'accessibilité	31
5.2 La rapidité	32
5.3 Le référencement naturel ou sponsorisé	32
5.4 La durabilité et la maintenabilité	32
<b>6. L'ÉCO-CONCEPTION, UNE DÉMARCHE DE LONG-TERME</b>	<b>33</b>
6.1 Bien gérer sa dette technique	33
6.2 Avoir une feuille de route définie	34
6.3 Anticiper la fin de vie	34
<b>7. QUELQUES PIÈGES À ÉVITER</b>	<b>35</b>
7.1 Ne pas tomber dans la sur-optimisation	35

7.2 Comparer ce qui n'est pas comparable	35
7.3 Ne pas oublier les autres aspects de l'écologie	35
<b>CONCLUSION</b>	<b>36</b>
<b>ANNEXE</b>	<b>37</b>
Annexe 1 : Langage et consommation d'énergie :	37

# 1. C'EST QUOI UNE APPLICATION WEB, UN SITE, "ÉCOLOGIQUE" ?

## 1.1 Comment analyser l'empreinte écologique du numérique

De manière générale, quand il s'agit de calculer l'empreinte écologique d'un service numérique (une application web ou mobile), il y a trois étapes majeures du cycle de vie à analyser :

- L'impact écologique lié à la fabrication , à l'extraction et la transformation des matériaux qui composent toutes les infrastructures (réseaux, serveurs, terminaux,...) nécessaires au fonctionnement du service numérique.
- L'impact écologique lié au fonctionnement de ce service numérique, essentiellement une consommation d'électricité.
- L'impact écologique lié à la fin de la vie des terminaux et leur recyclage ou non. Puis l'impact écologique lié à la gestion de la fin de vie du service numérique (l'application) en lui-même.

Dans le numérique, et en particulier pour la partie applicative, cette approche semble souvent très abstraite. En effet, les solutions numériques utilisent des machines (processeurs, carte graphique, disque dur, ...) et des infrastructures (le réseau) qui sont souvent mises à disposition sous la forme de services partagés (ex : le Cloud, le Web).

Ainsi, une même machine va faire tourner un grand nombre d'applications et l'impact de sa fabrication est tellement divisé qu'il devient difficile de l'apprécier. Pour donner un exemple, il semble très difficile d'évaluer quelle part de l'empreinte totale d'un datacenter va représenter telle ou telle application.

De plus, toutes les dimensions de l'empreinte écologique d'un datacenter ne sont d'ailleurs pas facilement mesurables (et donc facile à répartir). Si la consommation d'électricité est quantifiable par nature, il est plus difficile d'évaluer par exemple l'impact d'un datacenter sur la biodiversité aux alentours et de répartir cet impact entre les différentes applications que celui-ci fait tourner.

Cette complexité est aussi renforcée par le fait qu'une application est en réalité elle-même constituée d'une multitude de petites applications et d'usages.

On arrive ainsi à un niveau de granularité pour lequel les ordres de grandeur ne sont pas nécessairement faciles à apprécier.

Ainsi, par souci de simplification, **l'impact écologique d'une application web est souvent réduit à l'énergie nécessaire pour faire fonctionner les différents composants de la machine nécessaire à son fonctionnement.** Cela permet d'ensuite faire des comparaisons via des clés de répartition.

Le seul moyen d'avoir une analyse réelle de l'impact écologique global du numérique serait d'apprécier en théorie tout l'écosystème dans sa globalité ou d'analyser l'ensemble du cycle de vie sur des ressources dédiées à une seule application (ce qui concerne seulement les géants comme Facebook, Amazon, Netflix,...). **Bref, tout est multiple et connecté !**

Imaginez : si pour envoyer un seul message, il fallait envoyer votre propre satellite, déployer un câble, installer des serveurs,... Heureusement, que le web est une infrastructure partagée ! Bref, ce qui est impressionnant c'est la disproportion entre l'insignifiance apparente d'un usage (faire une recherche sur le web par exemple) et la quantité des infrastructures, de matériels partagés, que cela nécessite.

Toutefois, pour vous aider, voici une première approche des ressources matérielles que mobilise une application web :

- Toutes les ressources côté serveur (l'hébergement de vos données, data center) : processeurs, mémoires mais aussi climatisation, refroidissement,...
- Toutes les ressources côté client (terminaux des utilisateurs) : votre ordinateur (téléphones, tablettes, ...), tout ce qui le compose (processeur, écran, carte graphique, ...) et les matériaux pour produire tout cela.
- Toutes les ressources côté réseau : serveurs, câbles, routeurs, antennes, satellites ...

Pour rester pragmatique, il nous reste donc à déterminer comment limiter la consommation de ressources (essentiellement électrique) d'une application ce qui est le but de ce livre blanc ! Les autres points, en particulier les infrastructures, ne peuvent être analysés que d'un point de vue global et systémique.

## 1.2 Une application qui vise la simplification des parcours utilisateurs

Toute action sur une application consomme des ressources. A partir de ce constat, il y a deux axes de travail à actionner quel que soit la technologie utilisée :

- Optimiser la consommation de ressources pour chaque action.
- Limiter et simplifier au maximum le nombre d'actions.

Les optimisations de performance technique sont traitées dans une partie dédiée de ce livre blanc. TheCodingMachine a aussi rédigé un autre livre blanc uniquement dédié à la quête de performance pour une application web car il s'agit d'un sujet riche et complexe.

Dans le cas de la diminution du nombre d'actions, il faut **distinguer les actions automatisées de celles directement effectuées par l'utilisateur**. Pour cela, il est fortement recommandé de définir des workflows qui vont retracer toute la succession d'actions déclenchées par exemple par le clic d'un utilisateur.

**Ensuite, il faut se concentrer en priorité sur les usages clés**, notamment pour la première version de votre application. Il ne s'agit pas de prévoir une solution à l'ensemble des cas limites.

Par exemple, dans le cas de traitement de démarche en ligne, imaginez qu'une seule démarche représente 90 % des demandes, les autres 10 % étant des exceptions nécessitant chacune une démarche spécifique. Dans ce cas, il peut être légitime de se poser la question s'il ne vaut pas mieux encore passer par un traitement manuel de ces demandes plutôt que de développer un grand nombre de parcours spécifiques.

Cette question est d'autant plus vraie s'il s'agit de la première version de votre application car il est difficile d'anticiper la réalité des comportements utilisateurs et comment seront traitées certaines demandes spécifiques.

La même analyse est possible par rapport à vos canaux de revenus. Bien souvent, il est possible d'envisager de nombreux canaux de revenus différents pour un site ou une application web. Toutefois, il y aura forcément un canal et un marché stratégique pour votre entreprise : un canal qui doit marcher pour la rentabilité de votre entreprise.

**Avant d'envisager de nombreux cas d'usages liés à différents canaux, il s'agit de tester uniquement ce canal stratégique car si celui-ci ne marche pas, votre application sera difficilement rentable.**

Si vous ne trouvez pas votre marché, il s'agira alors de pivoter, de changer votre business model, ce qu'il est plus facile de faire tant que votre application est relativement simple et que vous n'avez pas investi trop d'argent, de temps et d'énergie pour la développer.

**La nécessité d'investir de manière rentable sur les bonnes fonctionnalités et la connaissance du besoin utilisateur sont loin d'être des problématiques nouvelles dans le web.**

Il s'agit du problème à l'origine de toutes les méthodologies dites "agiles". Il existe donc de nombreuses méthodologies de projet web qui sont là vous aider à développer le strict nécessaire au bon moment.

## **1.3 Une application qui est éco...nome !**

Et oui , consommer moins de ressources et prévoir moins de fonctionnalités permet de faire des économies sur les coûts de développement, d'hébergement et de maintenance de votre application ! Cela rend aussi votre application plus performante, résiliente et durable.

L'éco-conception est une démarche spécifique qui nécessite de réorienter vos réflexions en amont de votre projet. Ainsi, on pourrait penser qu'une réelle démarche d'éco-conception pourrait venir augmenter vos coûts lors de la phase de conception et de recherche de votre projet.

**En réalité, il s'agit plutôt d'un transfert de coût.** En effet, dans une approche traditionnelle, il y a très souvent beaucoup de temps passé à chercher comment faire fonctionner une fonctionnalité technique spécifique (les 10% de cas d'usage "limite" exprimés plus haut). Dans une approche d'éco-conception, il y a un gain de temps car on va potentiellement renoncer à certaines fonctionnalités en amont du projet. En revanche, ce gain de temps est utilisé pour trouver des alternatives d'éco-conception aux fonctionnalités classiques restantes.

Par ailleurs, l'éco-conception est aussi l'un des quatre critères permettant de justifier d'une approche innovante dans le cadre du Crédit Impôt Innovation (CII).

Dans le cadre du CII, il n'est pas nécessaire de justifier des quatre critères d'innovation, un seul suffit. Pour rappel, **adressé aux PME**, le CII permet de bénéficier d'un crédit d'impôt à hauteur de 30 % des dépenses engagées (depuis le 1er janvier 2023) pour la conception et le développement d'un produit innovant par votre entreprise, dans la limite d'un plafond de dépenses d'innovation de 400 000 € par an et par entreprise.

Ces dépenses peuvent être engagées en interne (le temps passé par vos équipes) ou **auprès d'un prestataire extérieur agréé** (comme TheCodingMachine !). Cet agrément est desservi par la Direction Générale des Entreprises et atteste de la capacité d'un prestataire à effectuer des missions d'innovation pour le compte de ces clients.

Bref, **une démarche d'éco-conception, documentée et justifiée, peut vous permettre de récupérer sous la forme de crédits d'impôt jusqu'à 30 % des dépenses engagées sur votre projet web.**

Remarque : pour prévenir tout risque de redressement, TheCodingMachine vous recommande fortement d'être accompagné par un expert en financement de l'innovation.

## 2. COMMENT INCLURE L'ÉCO-CONCEPTION DÈS LA GENÈSE DE SON PROJET

Avant de démarrer un projet avec une démarche d'éco-conception, la première étape est de se questionner sur la sobriété. Quelle est la pertinence du service numérique que l'on souhaite développer ? Dans cette réflexion, il est important d'étudier son écosystème, de dresser une cartographie de l'existant et de donner un rôle à chacun des services numériques. Une fois cette analyse faite, il est alors possible de mettre en place une démarche d'éco-conception pour produire un nouveau support ou refondre un ancien.

### 2.1 L'approche MVP - Minimum Viable Product

L'approche MVP provient du Design Thinking, c'est une philosophie qui existait déjà avant que l'éco-conception devienne une véritable préoccupation.

A l'origine, le principe d'un MVP est de développer le strict nécessaire, la fonctionnalité clé d'un produit pour tester rapidement un marché. Pour cela, il s'agit de confronter le plus rapidement possible une application à ses utilisateurs (un groupe pilote ou d'early adopters) et donc de s'assurer que la proposition de valeur trouve bien son marché.

Cette approche est totalement alignée avec l'éco-conception car **elle place la compréhension du besoin utilisateur au centre du développement produit tout en ayant une approche minimaliste.**

L'approche MVP a aussi l'avantage de mettre en avant la vertu morale de l'éco-conception en luttant contre les dark patterns dès la conception du service. En effet, il s'agit d'adresser le plus directement et simplement possible le besoin utilisateur.

## 2.2 L'importance de l'étude des parcours utilisateurs

C'est probablement la meilleure approche pour ne pas développer des fonctionnalités inutiles qui ne viennent pas répondre au besoin terrain.

En effet, on a souvent tendance à vouloir développer trop de fonctionnalités par faute d'informations sur le comportement réel des utilisateurs. Le manque d'information concrète incite à vouloir couvrir tout l'éventail théorique des possibles.

Pour combler ce manque, il existe plusieurs solutions : avoir un recours à un expert extérieur en la matière (freelance ou agence UX), mettre en place des outils permettant de recueillir et d'analyser de la donnée utilisateur, s'inscrire dans une approche de Design Thinking avec des ateliers internes,... Pour des applications qui représentent un investissement important et ont une importance stratégique, TheCodingMachine recommande fortement le recours à un prestataire spécialisé en matière d'analyse UX et d'étude terrain de l'expérience utilisateur. Cela permet en général de faire des économies (en ne développant que ce qui est nécessaire) et de limiter le risque de ne pas répondre au besoin.

Sinon à votre propre échelle, il existe de nombreux outils vous permettant d'avoir une première approche d'observation comme par exemple Google Tag Manager ou Contentsquare. Vous pouvez aussi organiser vos propres tests utilisateurs en interne et des ateliers d'idéation.

## 2.3 Les aspects graphiques

L'aspect visuel de votre site est nécessairement lié à l'expérience utilisateur que vous visez. Toutefois, il y a des aspects purement graphiques qui vont affecter la consommation énergétique de votre site, il s'agit notamment :

- Des polices de texte choisies : une police personnalisée, non native aux navigateurs, devra être chargée à chaque visite de votre site.
- Des caractéristiques globales de votre charte graphique et de ses composants : plus votre charte graphique possède de couleurs différentes, de formes différentes, plus le code CSS associé sera conséquent. Le choix de vos couleurs est aussi important : afin de réduire la consommation électrique, privilégiez le "dark mode" de plus en plus plébiscité par les utilisateurs.
- De limiter et d'optimiser au maximum les contenus visuels, en particulier les images et les vidéos qui s'affichent dès l'arrivée sur votre site.
- De bien penser à ce que les animations soient au service du parcours utilisateurs et non un simple gadget UI.

Aujourd'hui, l'éco-conception a défini des tendances de web design qui lui sont propres et de nombreux designers y sont sensibles. Ces tendances étant d'actualité, n'ayez pas peur d'avoir un site minimaliste, c'est dans l'ère du temps !

## 2.4 Le choix des technologies pour votre projet

C'est une question bien plus complexe susceptible d'engendrer de long débat au sein de la communauté technologique. Mais les langages, les frameworks et autres outils, que vous choisirez pour développer votre application auront tous un impact sur la consommation de votre application.

La question est difficile. Certaines solutions techniques, par exemple des langages, sont par construction (ou "par nature") plus légères ou efficaces en termes de consommation de ressources numériques. Toutefois, un langage "plus efficace" par défaut ne sera peut être pas celui le plus adapté à votre besoin, à la fonctionnalité que vous souhaitez développer.

Ainsi, le langage en théorie le plus optimisé ne sera peut être pas celui qu'il vous faut. Il faut aussi prendre en compte tous les autres critères susceptibles d'affecter votre projet : courbe d'apprentissage, écosystèmes techniques et compatibilités,... Pour conclure, il semble difficile de faire du critère écologique, le principal critère pour choisir la future stack technologique de votre application. D'autant que la plupart des optimisations significatives pouvant être menées pour réduire l'impact de votre site peuvent être effectuées quelque soit la technologie utilisée.

Si vous souhaitez toutefois satisfaire votre curiosité, vous trouverez en annexe de ce livre blanc les résultats d'une étude menée par des chercheurs pour évaluer la consommation d'énergie des différents langages de programmation.

## 3. RÉFÉRENTIELS ET BONNES PRATIQUES

### 3.1 Dix optimisations simples et impactantes

#### #1 Optimiser vos images

Sur un site vitrine classique, les images représentent souvent 50 % du poids d'une page. Il y a trois moyens de diminuer l'impact des images :

- Les compresser et adapter leur format, les format WebP pour les images bitmap et SVG pour les images vectorielles sont recommandés. Pour cela, vous pouvez utiliser des plug-ins dédiés comme [WebP Express](#) pour Wordpress mais cela va alourdir le socle WP. Il est donc encore plus préférable de compresser les images sur Photoshop ou d'utiliser Illustrator ou Figma pour exporter vos illustrations au format vectoriel.
- Ne pas charger toutes les images systématiquement et éviter les composants inutiles de type slider ou carrousel. Il est plus pertinent de conditionner le chargement de certaines images à l'interaction, au clic, de l'utilisateur. Si dans son parcours, il interagit pour afficher l'image alors on lui affiche, sinon c'est que cela n'est pas pertinent.
- Limiter tout simplement le nombre d'images sur votre site.

## #2 Mettre en place du cache

Les systèmes de cache permettent d'éviter de recharger des éléments déjà téléchargés (ex : des images, des données, ...). Ceci permet d'alléger ou d'éviter les requêtes auprès des serveurs.

Les navigateurs des visiteurs gèrent leur propre cache. A part quelques options configurables par l'utilisateur, le fonctionnement de ce cache est défini par les paramètres du serveur. Pour des sites vitrines, de nombreuses solutions et plug-ins existent.

Toutefois, mettre en place un système de cache peut être technique et cela dépend du niveau de complexité de votre application : calculs et algorithmes avancés, traitement et envoi d'un grand nombre de données, ... Dans ce cas, il s'agira d'un sujet à traiter à part avec votre prestataire dans une logique d'arbitrage coût / bénéfice.

## #3 Mettre en place du lazy loading

Le lazy loading ou chargement différé est une technique qui permet de charger les ressources d'un site web en différant le chargement des ressources non critiques, non essentielles, au début du temps de chargement d'une page.

Il s'agit d'une technique qui peut être utilisée pour empêcher un navigateur de charger des ressources non critiques sur une page web lorsqu'il commence à générer la page. Par exemple, cela ne sert à rien de charger à l'arrivée de l'utilisateur une image tout en bas de la page, elle sera téléchargée lorsque l'utilisateur descendra.

Le lazy loading est aussi à rapprocher de la notion de First Contentful Paint. Le First Contentful Paint (FCP) mesure le temps entre le clic sur le lien d'un site Internet et le moment où il affiche le premier bit de contenu de la page. Le FCP a été défini par Google comme une donnée de référence de son algorithme de référencement. Mettre en place du lazy loading est donc aussi important pour votre référencement.

Mettre en place du lazy loading n'est pas toujours évident, il s'agit de faire des arbitrages entre ce qui est considéré comme du contenu critique ou non.

Sur Wordpress, il existe néanmoins des plug-ins pour mettre en place un lazy loading simple comme par exemple [LazyLoad](#).

## #4 Minimiser le code HTML, CSS et JS

Minimiser du code signifie optimiser du code pour gagner de l'espace sans en altérer la fonctionnalité. Concrètement il s'agit de supprimer tous les espaces blancs inutiles, les commentaires, les caractères de nouvelle ligne et tout ce qui rendait initialement le code lisible. Ainsi, une fois qu'il a été minifié, le code devient presque illisible à l'œil nu.

La minification du code est possible dans toutes les technologies de programmation de base, notamment HTML, CSS et JavaScript. Cependant, le processus n'est pas instantané. Un certain travail est nécessaire pour rendre le code plus compact tout en le gardant fonctionnel.

Pour vous aider, il existe des outils de minification spécifique à chaque langage. Vous trouverez une première liste d'outils [dans cet article](#).

## #5 Limiter et optimiser les scripts sur votre application

Chaque fois qu'un élément du DOM est modifié après le chargement initial de la page, cela requiert un effort de calcul supplémentaire de la part du terminal de l'utilisateur, augmentant ainsi la consommation énergétique.

Ce phénomène est particulièrement notable lors de l'utilisation intensive de bibliothèques telles que jQuery, qui, malgré leur facilité d'usage pour manipuler le DOM, peuvent conduire à une surcharge de modifications inutiles et, par conséquent, à une consommation excessive d'énergie.

Ainsi, pour minimiser l'impact environnemental d'une application, il est conseillé d'éviter l'utilisation excessive de bibliothèques comme jQuery pour des manipulations simples du DOM qui peuvent être efficacement gérées par du JavaScript natif.

L'adoption de JavaScript pur, notamment grâce à des fonctionnalités modernes telles que les "fetch API" pour les requêtes réseau, permet de contrôler plus finement l'efficacité du code et réduire l'empreinte énergétique de l'application. Cela souligne l'importance d'une réflexion approfondie sur l'architecture et le comportement des applications numériques, dans l'objectif de concilier performance technique et responsabilité écologique.

## #6 Éviter ou optimiser l'usage de la vidéo

Dans la mesure du possible, il est préférable de ne pas utiliser de vidéo. En revanche si son usage est nécessaire du point de vue de la compréhension du message alors il est très important de l'optimiser.

Au-delà de la possibilité de compresser cette vidéo pour avoir une qualité adaptée au web, il est également possible de questionner la nature de la vidéo. A-t-elle besoin de faire plus d'une minute alors que de nombreux utilisateurs ont tendance à scroller sans regarder la vidéo dans son intégralité ? Bref, **une vidéo plus courte et mieux ciblée permet une meilleure attention de l'utilisateur et réduit la consommation énergétique.**

Il est aussi important de se demander si la vidéo doit se lancer de manière automatique. En effet, il est préférable pour l'expérience utilisateur et l'impact écologique de la page d'insérer une fonction 'play' pour que l'utilisateur puisse la lancer seulement si cela est pertinent pour lui.

## #7 Désactiver ou optimiser les CRON

CRON est une abréviation de crontab, elle-même une abréviation de chrono table qui signifie table de planification.

Issu du monde Unix, le programme CRON, lancé au démarrage d'un système d'exploitation, permet de planifier l'exécution automatique de tâches avec une certaine fréquence ou à un moment donné. Parmi les tâches les plus régulièrement exécutées via des CRON :

- Planification d'un job de sauvegarde de votre site (extension BackWPup pour Wordpress par exemple).
- Planification de l'optimisation de la base de données (extension WP-Optimize pour Wordpress par exemple).
- Mise à jour automatique de votre version (de WordPress ou autre),

Bien que la plupart de ces actions restent recommandées, l'exécution de chacune de ces tâches consomme des ressources. Il est donc important de se demander quelle doit en être la bonne fréquence ou s'il est préférable de gérer cela manuellement.

## **#8 Eviter les iFrames qui s'affiche directement au chargement de la page**

Une iframe est un élément HTML qui permet d'intégrer une page web dans une autre page web. Cette méthode permet par exemple d'intégrer une vidéo hébergée sur une plateforme tierce en l'affichant directement sur votre site web. L'exemple le plus répandu est celui d'une vidéo Youtube.

Fonctionner à travers des Iframes ne permet pas d'avoir le contrôle total sur le contenu de l'autre page qui sera affichée. Il est donc possible de se retrouver à charger des fonctionnalités et du contenu inutiles.

Dans le cas des vidéos, il est ainsi recommandé de les héberger vous même sur votre propre site comme expliqué dans [cet article](#). Enfin, il est important de faire en sorte que les Iframes ne soient affichés et chargés qu'à la demande de l'utilisateur (dans la mesure du possible).

## **#9 Limiter le recours direct à des services tiers complexes pour des usages simples (ex : Google Map)**

GoogleMap est l'exemple le plus répandu. Bien souvent, une simple image / capture d'écran avec votre adresse sur une carte suffira amplement à votre utilisateur. D'autant que, désormais, de nombreux téléphones permettent la reconnaissance automatique d'une adresse dans un texte et l'ouverture automatique de Google Map via un simple clic.

De manière plus générale, pour vous inscrire dans une démarche d'éco-conception, il est nécessaire de vous demander si toutes les fonctionnalités d'un service tiers sont utiles pour le parcours utilisateur visé.

Pour réutiliser l'exemple de GoogleMap : est-il nécessaire d'avoir les fonctionnalités de calcul d'itinéraire, de gestion des avis ou de géolocalisation si le but est simplement d'afficher une adresse ?

La multiplication des services tiers, des solutions SAAS et la démocratisation des API a néanmoins rendu complexe une telle approche. Il n'est pas toujours évident de savoir et de maîtriser toutes les fonctionnalités d'une application tierce d'autant qu'elles évoluent potentiellement à chaque mise à jour.

Bref, dans une démarche d'éco-conception, une des meilleures solutions semble être encore de limiter au maximum l'usage de ces solutions.

Enfin, certains types d'API, comme GraphQL par exemple, permettent de personnaliser les requêtes afin de ne récupérer que les données nécessaires au besoin de votre utilisateur.

## **#10 Choisir un hébergeur éco-responsable et bien dimensionner son hébergement**

Comme expliqué précédemment, la performance écologique de votre service numérique va en grande partie se traduire par une consommation d'électricité côté client (téléphone et PC de vos utilisateurs) et côté serveur. Pour connaître les émissions associées à celle-ci, il est donc nécessaire d'analyser l'origine de l'électricité consommée.

Si il est difficile de savoir si vos utilisateurs chargent leur téléphone avec des panneaux solaires, il vous est en revanche possible d'agir côté serveur en choisissant un hébergeur "green".

Ces hébergeurs utilisent de l'électricité d'origine renouvelable et effectue différentes actions à impact positif (planter des arbres par exemple). Attention toutefois au Green Washing, vous trouverez une approche du sujet dans [cet article](#). Les différentes normes ISO en lien avec l'environnement (ISO 14 001 et ISO 50 001) sont de bonnes preuves de sérieux de la part d'un hébergeur.

## **3.2 Pour aller plus loin : les sources de références en matière d'éco-conception**

### **#1 L'ADEME et son Référentiel Général d'Écoconception de Services Numériques (RGESN)**

Ce référentiel est mené dans le cadre de la mission interministérielle numérique responsable et co-pilotée par la Direction interministérielle du numérique (DINUM), le Ministère de la Transition Écologique, l'ADEME et l'Institut du Numérique Responsable.

Ce référentiel adresse de manière exhaustive toutes les dimensions possible de votre projet :

- Stratégie
- Spécifications
- Architecture
- UX/UI
- Contenus
- Frontend
- Backend
- Hébergement

Vous pouvez y accéder via [ce lien](#).

## **#2 Les best practices du Collectif Conception Numérique Responsable**

Le collectif Green IT est le premier à avoir vulgariser les différents concepts de l'éco-conception de services numériques. Encore aujourd'hui, le collectif continue de promouvoir les meilleures pratiques.

Ces bonnes pratiques sont disponibles sur [Github](#) et existe aussi au format papier sur le site officiel de l'éditeur (Eyrolles) accessible via [ce lien](#).

## **#3 Le label Numérique Responsable**

Le numérique responsable reste un sujet récent et en constante évolution. Il peut donc être difficile de s'y retrouver.

Le label Numérique Responsable est le label le plus officiel existant en matière de pratiques numériques responsables comme expliqué sur [le site du gouvernement](#). Il se base sur un référentiel construit en 2019 par l'Institut du Numérique Responsable en partenariat avec le Ministère de la Transition écologique, l'ADEME, la FING et WWF.

Le processus de labellisation se déroule en 6 étapes :

- Entrée dans la Communauté NR
- Formation
- Auto-évaluation et préparation
- Audit de labellisation par un organisme indépendant
- Obtention du label NR
- Suivi et renouvellement tous les 3 ans

## 4. QUELS INDICATEURS POUR ÉVALUER L'ÉCO-CONCEPTION ?

### 4.1 Les indicateurs techniques

Pour faire une analyse pertinente, il est bien souvent nécessaire d'analyser l'ensemble des pages de votre site une par une ou dans le cadre de parcours utilisateurs prédéfinis liés à des blocs fonctionnels. Toutefois, certains outils permettent d'avoir rapidement une vision globale (cf partie 4.3). Il est aussi important de noter que la plupart des indicateurs techniques sont au moins en partie corrélés.

#### #1 – Le poids de votre page :

Le poids de votre page correspond à la quantité de données de votre page et se mesure donc en octet (Méga-octets la plupart du temps). Une page web est constituée :

- D'un DOM (Document Object Model) : c'est le guide qui va renseigner le navigateur sur le contenu et la structure de la page, avant même son rendu.
- De code statique : HTML et CSS.
- Des scripts (Javascript) permettant d'animer et de dynamiser votre site.

- Des fichiers : images, vidéos, ...
- Des plug-ins ainsi que des solutions tiers.

Le poids de votre page est essentiellement influencé par la quantité et le type de fichiers de votre page. Les fichiers les plus lourds sont les images et les vidéos qu'ils convient donc d'optimiser en premier (limiter leurs nombres, les compresser ou conditionner leur affichage).

Si vous avez besoin de vous donner un ordre de grandeur, vous pouvez consulter le site HTTP Archive accessible via ce lien : <https://httparchive.org/reports/page-weight>

## **#2 - Le nombre de requête :**

Lorsque vous construisez un site web, il comporte de nombreuses parties différentes. Vous avez les différents fichiers d'images que vous utilisez sur une page, les feuilles de style CSS qui contrôlent l'apparence du contenu, les fichiers JavaScript qui ajoutent toutes ces fonctionnalités intéressantes, etc.

Lorsqu'une personne parcourt votre site web, son navigateur doit pouvoir télécharger toutes les ressources nécessaires à cette page à partir de votre serveur. Pour ce faire, il adresse des requêtes HTTP au serveur pour chaque ressource individuelle.

Bref, pour faire simple, les requêtes est ce qui permet à votre site de récupérer toutes les informations dont il a besoin auprès de votre serveur (qui stocke vos données et fichiers).

Le nombre de requêtes est en partie corrélé au poids de votre page. En effet, l'affichage de chaque image nécessite par exemple une requête spécifique.

Toutefois, certains éléments spécifiques vont affecter particulièrement les requêtes, il s'agit par exemples :

- D'extensions (fréquentes en Wordpress notamment).
- Les polices personnalisées et les émojis.
- Des scripts liés à des applications tierces.

Pour optimiser les requêtes, il y a trois approches possibles :

- Supprimer les requêtes HTTP inutiles (ex : extension Wordpress non utilisée)
- Combiner les requêtes HTTP. Par exemple, vous pouvez combiner six petits fichiers CSS en un seul fichier CSS plus volumineux, qui se chargera quand même plus rapidement.
- Optimiser les requêtes les plus lourdes. L'exemple le plus simple est celui d'une image (il suffit de la compresser) mais c'est un sujet qui peut devenir complexe dans le cas de calcul notamment.

### **#3 – Nombre d'éléments du DOM :**

Le DOM, Document Object Model correspond à la structure de votre page. Si vous avez quelques notions en programmation web, vous savez que le contenu de votre page est écrit en HTML, langage de programmation de balise. Ainsi un paragraphe de texte correspond à une balise <p>. Disons qu'un article contient un titre et plusieurs paragraphes. Donc votre balise <article> va contenir une balise <h1> et plusieurs balises <p>. On parle de balise enfant et de balise parent. Ainsi, on arrive à reproduire le schéma d'une page web sous forme d'un arbre avec des branches.

Google PageSpeed Insight est très clair la dessus. La taille de votre DOM est considérée comme excessive quand elle contient plus de 1500 éléments, donc 1500 balises.

Par ailleurs, au niveau de l'arborescence, la profondeur de votre DOM doit être de 32 éléments maximum, avec moins de 60 éléments enfant / parent, un élément pouvant en effet avoir plusieurs enfants.

N'espérez pas que votre site se chargera beaucoup plus rapidement en réduisant la taille de votre DOM. Cela a bien sûr une influence sur le temps de chargement, mais cela reste sans comparaison avec l'optimisation des images ou des fichiers CSS ou JS.

Optimiser la taille du DOM vous permettra en revanche de rendre votre page interactive plus rapidement en réduisant la mémoire allouée par votre navigateur. On parle alors de ressenti utilisateur. Cela agit donc sur l'indice de vitesse, le délai avant interactivité, le premier processeur inactif et le Max Potential First Input Delay. Bref, 4 indicateurs sur les 6 majeurs des Core Vitals.

#### #4 – Proportion de code inutilisé ou “mort” :

Le code mort, c’est tout simplement du code qui n’est jamais exécuté. Il peut s’agir de fonctions obsolètes, de variables non utilisées, de branches conditionnelles toujours fausses... Bref, de tout ce qui ne sert à rien.

Pourquoi s’en débarrasser ? Le code mort augmente le poids de votre projet, le rend plus difficile à lire, à maintenir et peut même introduire des bugs.

Si vous utilisez Chrome, l’onglet DevTools possède un outil pratique qui vous permet de voir quel code est exécuté sur une page sur ce qui ne l’est pas. Pour accéder à cet outil, suivez les étapes ci-dessous :

- Ouvrir Chrome DevTools
- Ouvrir le menu de commande avec : cmd + shift + p (ou ctrl + maj + p)
- Taper « Coverage » et cliquez sur l’option « Instrument coverage ».
- Sélectionner un fichier CSS dans l’onglet Sources.

Tout CSS qui se trouve à côté d’une ligne verte continue signifie que le code a été exécuté. Une ligne rouge signifie qu’il n’a pas été exécuté. Une ligne de code qui est à la fois rouge et verte, signifie que seule une partie du code sur cette ligne est exécutée.

De nombreux outils d’analyse statique de code comme [ESLint](#), vous permettront d’automatiquement déterminer le code non exécuté (en tout cas une bonne part).

#### #5 – Vitesse d’affichage pour le chargement complet de votre page :

Comme expliqué plus haut dans la partie concernant le lazy loading, il existe différents types de temps de chargement qui sont mesurés pour l’affichage d’une page.

L’avantage du temps de chargement pour votre page complète est que cet indicateur, en plus d’être très important pour votre business, est une bonne synthèse de tous les autres. Ainsi, pour améliorer ce temps de chargement, il vous sera nécessaire d’optimiser les autres indicateurs.

## 4.2 Les indicateurs d'utilisation

### #1 - Nombre de pages consultées par visite :

Un indicateur simple qui peut être obtenu dans Google Analytics ou Matomo par défaut. Toutefois, il peut très rapidement devenir intéressant de segmenter les visites selon le besoin utilisateur, son origine ou d'autres critères. Dans ce cas, il s'agira le nombre de pages consultées par cohorte d'utilisateurs.

### #2 - Temps passé par visite :

Un indicateur toujours essentiel pour savoir s'il n'y a pas des anomalies au cours d'un parcours utilisateur (ex : difficulté à remplir un formulaire). De même que pour le nombre de pages visitées, il est souvent important d'analyser des parcours spécifiques.

Cela peut être aussi observé lors de sessions de test réel avec des utilisateurs. Par exemple, en disant à un utilisateur "Faites la démarche X ou Y" et en chronométrant. Cela permet d'identifier plus facilement les points de blocage.

### #3 - Terminaux utilisés pour les visites et versions des navigateurs:

Un élément essentiel à connaître afin de faire en sorte que votre application tourne le mieux possible et le plus longtemps possible sur la plupart des terminaux utilisés par vos utilisateurs. Cette information est en général facilement accessible dans Google Analytics ou Matomo.

## 4.3 Les outils de mesure

Le gouvernement a lui-même mis à disposition [une boîte à outil](#) pour vous inscrire dans une logique de numérique responsable. Voici, néanmoins, une rapide présentation des outils que nous trouvons les plus pertinents.

## #1 Pour l'analyse du code :

### a) SonarQube :

SonarQube est un outil d'analyse de qualité du code qui collecte et analyse le code source, fournissant des rapports sur la qualité du code de votre projet. Il combine des outils d'analyse statique et dynamique, permettant de mesurer la qualité de manière continue.

SonarQube inspecte et évalue chaque aspect de votre code, offrant un historique consultable du code pour analyser les problèmes : style, défaillances, duplications, manque de couverture de test ou complexité excessive.

Le logiciel analyse le code source sous différents angles, du niveau du module au niveau de la classe, fournissant des valeurs métriques et des statistiques révélant les zones problématiques nécessitant des améliorations.

SonarQube garantit également la fiabilité du code, la sécurité de l'application et réduit la dette technique en rendant votre base de code propre et maintenable. Il prend en charge 27 langues différentes, dont le C, le C++, Java, JavaScript, PHP, GO, Python, et bien plus encore. SonarQube propose également une intégration CI/CD et fournit des retours lors de la revue de code avec l'analyse des branches et la décoration des demandes d'extraction.

### b) EcoCode :

EcoCode est un outil Open Source basé sur un référentiel de règles d'éco-conception logicielle.

L'outil propose un catalogue d'analyseurs statiques de code pouvant s'intégrer à SonarQube permettant de lever les défauts de qualité du code qui peuvent avoir un impact écologique : consommation d'énergie et de ressources, obsolescence logiciel, durée de vie des terminaux,...

## #2 Pour l'analyse des performances techniques côté client :

### a) GTMetrix :

[GTmetrix](#) est un outil qui permet d'évaluer les performances web d'un site, notamment en tenant des comptes des critères Core Web Vitals liés à l'expérience utilisateur.

### b) Google Lighthouse :

Disponible depuis 2016 sous la forme d'une extension pour Chrome, Google lighthouse est maintenant disponible directement dans Chrome DevTools, via l'onglet « Audits ». Google lighthouse est une ressource très utile pour les développeurs qui s'intéressent à la performance et la qualité web.

Dès que [Google Lighthouse](#) a achevé l'évaluation de la page, vous obtenez un rapport d'audit, qui débute par plusieurs scores. Le Performance Score est obtenu à partir des performances obtenues par la page lors du test.

Obtenir un score de 100 signifie que la page testée est plus rapide que 98% des pages web référencées par l'outil. Un score de 50 indique que la page est plus rapide que 75% des pages

Après les scores généraux, vous trouverez les valeurs mesurées pour 6 indicateurs de performance, pour lesquels une description succincte est disponible via une infobulle.

- First Contentful Paint (FCP) : indique le délai à partir duquel le premier texte ou la première image est rendue.
- First Meaningful Paint (FMP) : mesure quand le contenu principal de la page est visible.
- Speed Index : Speed Index montre à quelle vitesse les contenus de la page sont rendus visibles.
- First CPU Idle : indique le délai à partir duquel la thread (fil d'exécution) principale est suffisamment disponible pour gérer les entrées.

- Time to Interactive : indique le délai à partir duquel la page sera pleinement interactive.
- Estimated Input Latency : ce score est une estimation du délai nécessaire à votre application pour répondre à une entrée de l'utilisateur, en millisecondes, durant la période de 5 secondes la plus active (busy) du chargement de la page. Si ce délai est supérieur à 50ms, l'utilisateur peut percevoir des lags (interactions en retard).

Pour aller plus loin sur le sujet, nous vous recommandons [cet excellent article](#).

### #3 Pour la consommation d'électricité et l'émission de GES :

a) [Ecoindex.fr](#) :

Vous entrez une URL dans [l'EcoIndex](#), qui calcule alors la performance et l'empreinte environnementale de la page testée :

- La performance environnementale est représentée par un score sur 100 et une note de A à G (plus la note est élevée, mieux c'est!).
- L'empreinte environnementale est matérialisée par les émissions de gaz à effet de serre et la consommation d'eau générées par la page.

Plusieurs critères sont pris en compte par notre méthode de calcul :

- La complexité de la page : le DOM (Document Object Model) représente la structure et les éléments d'une page web HTML. Plus le DOM comporte d'éléments, plus la page est complexe à déchiffrer, et donc à afficher pour le navigateur. Concrètement, tout ça signifie un effort plus important à fournir de la part du processeur de votre ordinateur pour afficher la page, ce qui diminue la durée de vie de votre équipement.
- Le poids des données transférées : avant d'apparaître sur votre écran, une page web est un ensemble de données stockées sur un serveur. Lorsque vous accédez à une page, votre navigateur envoie une requête au serveur pour qu'il lui communique ces données, afin de les mettre en forme et de les afficher sur votre écran. Seulement voilà : le transport de ces données, plus ou moins lourdes, du serveur au navigateur nécessite de l'énergie.

- Le nombre de requêtes HTTP : ce critère permet de prendre en compte l'effort fait par les serveurs pour afficher la page testée. Plus le nombre de requêtes est important pour une même page, plus il faudra de serveurs pour servir cette page.

Attention, EcoIndex ne donne aucune information sur les efforts faits en termes d'éco-conception. A l'instar d'un radar qui mesure une vitesse instantanée, EcoIndex mesure la performance environnementale instantanée. De manière générale, une analyse poussée de la performance et de l'empreinte environnementale de votre site nécessite l'intervention d'un expert.

b) Greenspector :

[Greenspector](#) est une suite logicielle dédiée à l'efficacité énergétique de vos services et à la performance côté client de vos applications mobiles et web. Il s'agit de l'artillerie lourde, essentiellement adaptée aux grands comptes devant piloter et analyser un grand nombre d'applications.

#### **#4 Pour l'analyse des usages fonctionnels :**

a) Par la donnée : Google Analytics, Matomo, ...

On ne présente plus ces outils et ils sont utilisés depuis bien plus longtemps que l'éco-conception.

Toutefois, un bon usage et un bon paramétrage de ces outils (définir des objectifs, des événements, des segmentations ou cohortes,...) permettent d'analyser via la donnée si l'expérience utilisateur est optimale par rapport à un certain nombre d'usages clés.

Par exemple, dans le cas d'un utilisateur ayant candidaté à une offre d'emploi, il s'agit de remonter son parcours pour savoir le nombre de pages consultées, le temps passé, le nombre d'événements déclenchés, ...

b) Via des outils spécialisés plus complet :

Des outils comme [Contentsquare](#) ou [Lucky Orange](#) permettent d'observer le parcours de vos utilisateurs via l'enregistrement et le visionnage de sessions et de très nombreux autres outils (heatmap, couche d'intelligence artificielle, ...).

A noter que l'utilisation de ces outils entraîne quand même la consommation de ressources supplémentaires. Il faut donc faire un arbitrage pertinent entre les gains d'énergie liés à un parcours plus optimisé et la consommation supplémentaire de ces outils.

## #5 Pour l'analyse des performances côté infrastructure :

### a) Hubblo :

Hubblo est un collectif proposant des services d'accompagnement et d'automatisation basés sur des méthodes, des données et des outils libres afin de réduire l'impact de vos services numériques et de vos systèmes d'information. Hubblo est notamment à l'origine des outils Open Source :

- [Scaphandre](#) pour la mesure de la consommation électrique.
- [Cabestan](#) pour l'inventaire des ressources numériques.

### b) Greenframe :

[GreenFrame](#) est une solution open source qui permet aux développeurs de décarboner leurs sites web en :

- Permettant de simuler des parcours utilisateurs via des scripts.
- Analysant en détails toutes les performances de tous les composants de votre architecture et infrastructure.
- Calculer la consommation électrique et l'empreinte carbone équivalente.



## 4.4 Tableau récapitulatif des indicateurs :

Les statistiques doivent idéalement être analysées pour chaque page de votre site. Voici notre proposition de tableau synthétique pour un suivi efficace de votre application.

<i>URL de la page</i>	<i>Unités</i>	<i>Source de la donnée</i>
Poids de la page	Mo	Lighthouse, GTMetrix
Nombre d'éléments du DOM		EcoIndex
Nombre de requête		Lighthouse, GTMetrix
Vitesse d'affichage*	ms	Lighthouse, GTMetrix
Estimation GES	mg eq CO2	Greenframe, EcoIndex
Électricité consommé	mWh	Scaphandre, Greenframe

*\* Pour un chargement complet de la page*

Pour conclure cette partie, il faut tout de même rappeler que malgré le nombre toujours croissant d'outils de mesure, aucun ne peut se vanter d'être parfaitement précis dans la prise en compte de l'ensemble de la chaîne numérique tant le sujet est complexe. Cela ne doit néanmoins pas empêcher la prise d'actions car "ce n'est pas la mesure qui pollue mais nos actions".

## 5. LES BIENFAITS DE L'ÉCO-CONCEPTION !

### 5.1 La simplicité d'utilisation et l'accessibilité

L'éco-conception a pour principal objectif la sobriété fonctionnelle (cf partie 1.2), cela limite le risque qu'un utilisateur se perde sur votre application, ne sache plus comment faire ce qu'il souhaite ou qu'il n'y parvienne pas à cause de contraintes physiques ou mentales spécifiques. L'expérience utilisateur se veut la plus directe possible et il ne doit pas exister trop de manière possible d'aller d'un point A à un point B.

De manière générale, une telle approche à tendance à aussi améliorer la conversion que vous visez sur votre site : consultation d'une page, récolte d'une adresse mail, abonnement, commande, ... Elle permet d'avoir des tunnels de conversion clairs et identifiés que vous pourrez suivre et améliorer au fil du temps.

De plus, une application éco-conçue à vocation à tourner sur des terminaux anciens, par exemple des anciennes versions de l'iPhone, ce qui est facilité par des performances techniques optimisées. Cela permet à votre application d'être accessible et de bien fonctionner auprès d'utilisateurs qui n'ont pas pu s'acheter le tout dernier modèle de téléphone ou qui n'ont pas installé la dernière version de leur navigateur (Google Chrome, Safari, ...). Vous atteignez donc un marché plus grand.

## **5.2 La rapidité**

Une page éco-conçue doit mettre peu de temps à charger, elle s'affiche donc vite pour votre utilisateur. C'est d'autant plus vrai si votre utilisateur dispose d'une mauvaise connexion internet ou d'un vieux téléphone. Cela permet de créer une expérience client optimale et donc un avantage concurrentiel. Pour rappel, plus de 50 % des visites de sites mobiles sont abandonnées si les pages prennent plus de 3 secondes à charger.

## **5.3 Le référencement naturel ou sponsorisé**

Les indicateurs de l'éco-conception (vitesse de chargement, poids de la page, ...) sont aussi analysés par les moteurs de recherche afin d'évaluer la pertinence de votre site et le classer dans les résultats de recherche (sponsorisés ou non).

Un site éco-conçu est un site performant qui sera donc apprécié des moteurs de recherche. Or, dans les résultats de recherche, une différence de quelques places a un impact considérable. Ainsi, une page en première position dans les résultats de recherche a un taux de clic aux alentours de 30 % pour seulement 15 % en 2ème position et 10% en 3ème position (Source : <https://backlinko.com/google-ctr-stats>).

## **5.4 La durabilité et la maintenabilité**

L'éco-conception force à mettre la durabilité et la maintenabilité au cœur de votre projet dès sa genèse ainsi qu'à mieux gérer la dette technique.

Or, bien souvent ces sujets sont habituellement traités quand ils sont déjà devenus problématiques. Il est donc particulièrement bénéfique d'avoir une approche plus transversale et continue de ces questions via l'éco-conception, ces outils et ses indicateurs.

Enfin, l'éco-conception a pour principe la sobriété fonctionnelle et technique. Cela facilite considérablement la gestion à long terme de votre application et la maîtrise de la connaissance.

## 6. L'ÉCO-CONCEPTION, UNE DÉMARCHE DE LONG-TERME

### 6.1 Bien gérer sa dette technique

Il s'agit d'un sujet vaste et qui constitue une bonne pratique même en dehors de l'éco-conception (ne serait-ce que pour des questions de sécurité). Pour rester simple, voici les différentes raisons pour lesquelles c'est écologiquement important :

- Au fil du temps, les technologies utilisées pour votre projet évoluent et en général deviennent plus performantes. Mettre à jour votre application permet donc de réduire sa consommation.
- Une mauvaise gestion de la dette technique va avoir tendance à entraîner un code peu efficace avec souvent de nombreux doublons ou des fonctions inutiles. Le poids de votre code augmente, votre application consomme plus de ressources et les performances diminuent.
- Une dette technique trop importante risque de rendre obsolète votre solution rapidement et de compliquer son utilisation sur le maximum de terminaux possibles (principe fondamental de l'éco-conception pour ne pas forcer vos utilisateurs à changer constamment leur terminaux).

Bref, la gestion de la dette technique est un sujet récurrent qui mériterait à lui tout seul un livre blanc !

## 6.2 Avoir une feuille de route définie

L'éco-conception est grandement liée à l'efficacité fonctionnelle de votre application auprès des utilisateurs.

Avoir un agenda de releases clair afin de correctement valider un maximum de features via des feedbacks utilisateurs est donc une composante clef.

Pour une démarche pertinente d'éco-conception à long terme, il est aussi nécessaire de faire en sorte que les évolutions de votre application soient compatibles avec un maximum de versions des différents systèmes d'exploitation. Cela nécessite une forme de vigilance afin de ne pas développer des features qui ne seraient seulement compatibles qu'avec les derniers systèmes d'exploitation.

Il s'agit aussi de surveiller si les versions des technologies que vous utilisez restent supportées par la plupart des systèmes existants (navigateurs et OS).

Ce sujet peut néanmoins rapidement devenir complexe en particulier à cause de la difficulté de tester votre application sur un grand nombre de versions différentes de systèmes d'exploitation et de navigateurs.

Il s'agit donc de rester dans une logique d'arbitrage raisonnable par rapport à l'évolution du parc actuel. Par exemple, il n'est aujourd'hui plus nécessaire de vérifier le fonctionnement d'une nouvelle application sur Internet Explorer qui n'est de toutes façons plus supporté par Microsoft.

## 6.3 Anticiper la fin de vie

Même si plus aucuns utilisateurs ne vont sur votre application, le code de celle-ci continue d'être stocké et nécessite donc potentiellement l'alimentation d'un serveur. Il faut donc penser à supprimer au maximum les données et le code inutilisés, ce qui s'inscrit d'ailleurs dans les meilleures pratiques liées au RGPD et à la sécurité.

# 7. QUELQUES PIÈGES À ÉVITER

## 7.1 Ne pas tomber dans la sur-optimisation

L'optimisation d'une application web est un sujet qui peut devenir rapidement complexe. Ainsi, si certaines optimisations permettent un retour rapide, il peut devenir rapidement coûteux et difficile d'optimiser une application déjà performante.

La complexité technique que cela peut venir ajouter viendrait alors en contradiction avec les objectifs visés initialement par l'éco-conception (par exemple la compatibilité avec les vieux terminaux). Il s'agit donc de faire les meilleures arbitrages possibles dans une logique d'efficacité.

## 7.2 Comparer ce qui n'est pas comparable

Beaucoup de facteurs font qu'une application est unique notamment les technologies qu'elle utilise ou les usages qu'elle vise.

Une application comme Netflix qui repose sur du contenu vidéo très consommateur en ressources ne peut pas être comparé à par exemple Wikipedia. L'exemple est délibérément caricatural mais la question de trouver un ordre de comparaison n'est pas nécessairement évidente.

Cela est vrai même pour des applications qui peuvent être similaires à l'usage mais construites très différemment et donc pas optimisables de la même manière.

Ce qui est important, c'est de réellement s'inscrire dans la démarche et d'ainsi piloter votre performance dans une logique d'amélioration continue.

## 7.3 Ne pas oublier les autres aspects de l'écologie

Bien qu'adopter une approche d'éco-conception peut permettre de réduire considérablement les émissions liées à votre site, il ne faut pas oublier qu'en valeur absolue cela représente potentiellement peu d'économie d'émissions par rapport à d'autres postes d'émissions de votre entreprise.

Ainsi, à l'échelle de votre entreprise, il est fort probable que d'autres actions aient un impact écologique plus important notamment en termes d'émissions. Il s'agit par exemple des déplacements pour lesquels la promotion du vélo ou même le télétravail entraînent aussi des économies conséquentes.

Cela ne veut pas dire toutefois qu'il faut négliger l'impact de votre site. En effet, comme c'est souvent le cas en matière d'écologie, si tous les acteurs adoptent cette démarche alors les émissions du secteur numérique pourraient être drastiquement réduites.

Comme pour les autres approches environnementales et écologiques, il est nécessaire de développer une pensée systémique et d'accumuler les gestes sur le maximum de leviers.

## CONCLUSION

À première vue, l'idée d'éco-concevoir une application web ou un site internet peut paraître dérisoire dans l'immensité des défis écologiques auxquels nous sommes confrontés. Pourtant, cette démarche s'inscrit dans la même logique que le tri des déchets ou le choix de consommer moins de vêtements : il s'agit de petites actions individuelles qui, collectivement, peuvent avoir un impact significatif sur notre environnement. Chaque bit d'information stocké, chaque requête envoyée, et chaque page visualisée consomment de l'énergie, principalement produite par des sources non renouvelables. En optimisant la conception des sites web et des applications pour qu'ils soient plus légers et plus efficaces, on peut réduire leur empreinte énergétique. De la même manière que recycler ou choisir de réduire sa consommation de biens matériels, l'éco-conception numérique est une étape nécessaire pour diminuer notre impact environnemental global. Ce n'est peut-être qu'une goutte dans l'océan, mais chaque goutte compte pour former un océan plus sain.

# ANNEXE

## Annexe 1 : Langage et consommation d'énergie :

	Energy
(c) C	1.00
(c) Rust	1.03
(c) C++	1.34
(c) Ada	1.70
(v) Java	1.98
(c) Pascal	2.14
(c) Chapel	2.18
(v) Lisp	2.27
(c) Ocaml	2.40
(c) Fortran	2.52
(c) Swift	2.79
(c) Haskell	3.10
(v) C#	3.14
(c) Go	3.23
(i) Dart	3.83
(v) F#	4.13
(i) JavaScript	4.45
(v) Racket	7.91
(i) TypeScript	21.50
(i) Hack	24.02
(i) PHP	29.30
(v) Erlang	42.23
(i) Lua	45.98
(i) Jruby	46.54
(i) Ruby	69.91
(i) Python	75.88
(i) Perl	79.58

Source : <https://greenlab.di.uminho.pt/wp-content/uploads/2017/10/sleFinal.pdf>