



HAL
open science

Sustainability of mobile educational resources in electrical engineering

Jean Paul Guillet

► **To cite this version:**

Jean Paul Guillet. Sustainability of mobile educational resources in electrical engineering. *Études & Pédagogies*, 2024, 10.20870/eep.2024.7913 . hal-04741482

HAL Id: hal-04741482

<https://hal.science/hal-04741482v1>

Submitted on 17 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Durabilité des ressources pédagogiques sur support mobile en génie électrique

Sustainability of mobile educational resources in electrical engineering

Jean-Paul Guillet^{1,2,3}

¹ Université de Bordeaux, Laboratoire CeDS EA 7440 3 ter, place de la Victoire 33076 Bordeaux, France

² Université de Bordeaux, Laboratoire IMS UMR CNRS 5218, 351 Cours de la Libération 33405 Talence, France

³ IUT de Bordeaux, Département GEII, 15 rue de Naudet 33175 Gradignan, France

Auteur pour la correspondance : Jena-Paul Guillet, jean-paul.guillet@u-bordeaux.fr

Date de soumission : 10/12/2023

Date de publication : 23/02/2024

Mots-clés : Durabilité dans l'éducation de l'électronique – Applications mobiles pour l'éducation – Applications mobiles l'enseignement supérieur – Éducation basée sur smartphone – Applications web progressives

Keywords: *Sustainability in electronics education – Mobile apps for education – Higher education mobile apps – Smartphone-based education – Progressive web apps*

Résumé

L'éducation durable (Sterling, 2009) est un enjeu aux multiples dimensions, tant concernant l'impact social et humain, le rôle de l'électronique dans la transition liée aux enjeux climatiques et mondiaux, que des thématiques telles que l'obsolescence matérielle et logicielle. L'utilisation des applications mobiles dans l'enseignement supérieur se développe dans le monde entier dans de nombreuses disciplines telles que l'apprentissage des langues, la physique, le droit, les mathématiques, etc. Les solutions commerciales sont nombreuses, mais deux questions se posent : tout d'abord, quelle place l'enseignant peut-il prendre dans le développement de ces solutions ? Et quelle est la pérennité de ces solutions ? Cet article apporte un retour d'expérience de huit ans avec trois solutions de création d'applications pédagogiques qui ont été comparées et mises en œuvre avec des étudiants de l'enseignement supérieur.

Abstract

Sustainable education is an issue with many dimensions, both concerning the social and human impact, the role of electrical engineering on the transition linked to climate and global issues, as well as themes such as hardware and software obsolescence. The use of mobile applications in higher education is growing worldwide in many disciplines such as language learning, physics, law, mathematics, etc. There are numerous commercial solutions, but two questions arise: first of all, what place can the teacher take in the development of these solutions? And what is the sustainability of these solutions? This article provides eight years of experience feedback with three solutions for creating educational applications that have been compared and implemented with students in higher education.

Introduction

L'éducation durable est un enjeu aux multiples dimensions, tant concernant l'impact social et humain, le rôle de l'électronique sur la transition liée aux enjeux climatiques et mondiaux, que des thématiques telles que l'obsolescence matérielle et logicielle. L'interopérabilité, la durabilité et la diffusion des contenus éducatifs est donc un enjeu important, à la fois pour donner l'exemple mais aussi pour rendre plus efficace l'enseignement supérieur par une réutilisation des ressources, une accessibilité élargie et la possibilité pour les enseignants d'intégrer ces ressources dans leur séquence pédagogique.

Les enseignants du supérieur développent des ressources pédagogiques en ligne depuis l'apparition du web (Benbunan, 1997 ; Del Pilar García-Chitiva, 2021 ; Hiltz, 1997). Si au début ces ressources étaient souvent des page web figées (contenu basé sur du code HTML) , certaines technologies comme les applets Flash et Java ont ensuite apporté l'interactivité (Chen et al., 2010 ; Reed & Afjeh, 1998). Ces technologies sont ensuite devenues obsolètes en raison du déclin de l'utilisation de Java dans les navigateurs (problématiques de sécurité, lourdeur) et de l'apparition d'appareils mobiles ne prenant pas en charge Java et le rendu Flash (Kuusisto, 2018). Depuis 2015, on considère que le web est devenu *mobile first*, ce qui signifie qu'il y a plus d'utilisateurs mobiles que sur ordinateurs (Margea & Margea, 2017). Ces évolutions conduisent naturellement à l'émergence de ressources éducatives sur de nouveaux supports comme les mobiles ou les systèmes en réalité augmentée (Furió et al., 2017). Aujourd'hui, les magasins Google Play (Android) et Apple (iOS) comptent plusieurs centaines de milliers d'applications mobiles éducatives.

Même s'il existe des applications éducatives disponibles sur les marchés, elles ne sont parfois pas totalement en phase avec les besoins des étudiants ni contextualisées à l'aune du programme pédagogique local, d'un programme national ou d'une ressource dans une certaine langue. Dans cette situation, il peut être intéressant de développer localement des applications avec un contenu adapté aux besoins spécifiques des étudiants, à leur niveau, aux compétences attendues et au contexte d'une formation. Ces développements peuvent être faits soit par les enseignants, soit par d'autres étudiants dans le cadre de projets (Anni, 2021 ; Klimova, 2019 ; Mella-Norambuena et al., 2021 ; Michalakakis et al., 2020), et être intégrés dans le cadre de cours ou travaux pratiques, pour le tutorat d'étudiants en difficultés ou pour illustrer des notions parfois abstraites sous un autre angle. Alors que les enseignants ont su pendant des siècles créer des contenus contextualisés, à travers des livres, des supports pédagogiques ou des sujets de travaux pratiques, comment concilier l'émergence de ces nouvelles technologies avec la nécessaire transmission de savoirs ou de méthodes de l'enseignant à l'élève alors qu'il devient plus difficile de créer des contenus numériques ? Au-delà des professeurs d'informatique qui sont naturellement plus aptes à développer de telles ressources, ces nouveaux développements posent la question de la capacité des enseignants d'autres disciplines comme les langues (Kacatl & Klimova, 2019), la physique (Hochberg et al., 2018) ou les mathématiques (Drigas & Pappas, 2015) pour produire du contenu mobile.

Cet article présente une étude sur huit ans autour du développement de ressources mobiles par l'enseignant ou par les étudiants, dans le but d'accompagner l'enseignement ou via le tutorat des étudiants rencontrant des difficultés notamment en première année de l'enseignement supérieur. Les principales interrogations tournent autour de la diffusion de ces ressources, de la pérennité (Napal et al., 2020) des contenus numériques éducatifs ainsi produits, de la fragmentation de l'environnement logiciel (Zhang & Yu, 2017) et enfin des techniques récentes avec les applications web progressives (*progressive web apps*, PWA), avec le but de donner des indications utiles aux enseignants qui souhaiteraient se lancer dans de tels développements.

1. Méthodes

Ce travail s'appuie sur l'analyse du développement, de l'usage et du cycle de vie de trois groupes d'applications pédagogiques réalisées selon trois méthodes différentes. Cette partie présente les méthodes de développement associées, ainsi que la motivation pédagogique qui a poussé à ces développements.

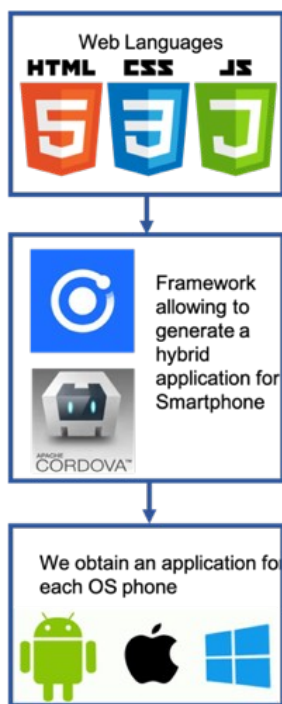
1.1. Applications natives

Le développement mobile natif permet le développement d'applications offrant toutes les fonctionnalités possibles des systèmes d'exploitation mobiles. Chaque système d'exploitation a son langage de programmation dédié : si vous cherchez à créer une application iOS, vous devrez utiliser Objective-C ou Swift. Les langages natifs d'Android sont Java ou Kotlin. Le développement d'une application sur deux systèmes nécessitera donc deux développements parallèles, chaque langage ayant ses spécificités. L'avantage de cette méthode est l'accès à des fonctions permettant d'utiliser au mieux les ressources du téléphone, mais l'inconvénient est la plus grande complexité de programmation qui nécessite l'apprentissage de nouveaux langages.

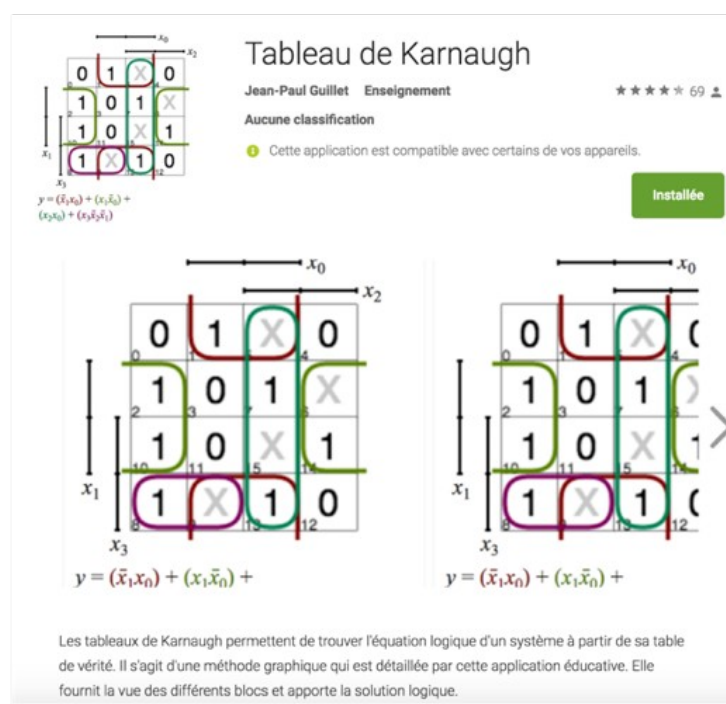
1.2. Développement hybride multiplateforme

Les environnements (ou *frameworks*) de développement hybrides sont des solutions permettant de créer des applications web hybrides pour les appareils mobiles utilisant les technologies Web (c'est-à-dire les langages CSS3, HTML5 et JavaScript) au lieu de s'appuyer sur des environnements de développement natifs spécifiques aux systèmes Android ou iOS (Malavolta et al., 2015). Cette solution permet d'avoir une application mobile qui est construite par l'intégration de code CSS, HTML et JavaScript comme présenté dans la Figure 1(a). Parmi les différentes solutions comme Xamarin, React Native, Ionic, Apache Cordova, Sencha, etc., nous avons utilisé dans notre travail principalement Apache Cordova (anciennement PhoneGap). Cette solution permet de réutiliser du code déjà développé pour des applications web, ce qui simplifie le développement.

Pour illustrer cette technique, une application a été développée pour aider les étudiants dans l'apprentissage des tableaux de Karnaugh. Un étudiant en électronique numérique apprend dans ses premiers cours de logique combinatoire la manière de simplifier une équation logique. Pour ce faire, nous pouvons utiliser le tableau de Karnaugh (Holder, 2005 ; Malavolta, 2016), qui est une manière particulière de représenter la table de vérité (tableau à deux dimensions décrivant les entrées et la sortie d'un système logique). En respectant certaines règles de présentation, ils permettent directement d'obtenir la forme la plus simple possible d'une fonction logique. Chaque cellule du tableau correspond à une ligne du tableau de vérité d'une fonction logique : une fonction à n variables est donc représentée par un tableau à 2^n cellules disposées de telle sorte qu'une seule variable change de valeur en passant d'un carré à un carré adjacent. Il faut que les groupes soient les plus grands possibles, la même case peut être utilisée plusieurs fois si nécessaire pour plusieurs groupes différents mais il faut faire le moins de cases possible. Dès que toutes les cases font partie d'au moins un groupe, nous avons terminé. Pour chaque groupe, seules les variables qui ne changent pas d'état sont conservées pour l'équation logique. L'une des principales difficultés d'apprentissage de cette partie du programme réside dans la difficulté pour un élève de travailler seul tout en s'entraînant, car la méthode consiste à créer des tables de vérités aléatoires puis à s'exercer, ce qui nécessite ensuite de vérifier la réponse, ce qui prendra du temps et expose l'élève au risque d'avoir fait une erreur sans le savoir. L'application présentée sur la Figure 1(b) propose donc une résolution de la table de Karnaugh en offrant une vue graphique des différents blocs. Par conséquent, une application permettant de trouver la bonne réponse et affichant une représentation graphique identique à celle effectuée manuellement permet de voir immédiatement si la solution trouvée est la bonne. Ainsi, une application a été développée en 2014 à l'aide de PhoneGap (alors Apache Cordova) pour proposer une représentation de la solution simplifiée d'une table de vérité utilisant la méthode des tables de Karnaugh.



(a)



(b)

Figure 1 – Générations d'applications hybrides à l'aide d'un framework multiplateforme (a) Flux de développement des technologies Web à l'application mobile, qui part des langages web, passe par des *frameworks* comme Cordova et permet d'obtenir une application (b) Copie d'écran de l'application cartographique Karnaugh développée sur Google Play Store, construite à l'aide d'Apache Cordova, qui permet aux étudiants de vérifier leurs résultats lors d'un travail personnel de résolution de tableaux de Karnaugh.

1.3. Application web progressive

Les applications web progressives sont à la fois un site web et une application mobile. Il n'est pas nécessaire de compiler une application et de l'installer car c'est lorsque l'on accède pour la première fois au site qu'il suggère la possibilité d'installer une version portable sur le mobile. Ces applications fonctionnent partout et offrent quelques fonctionnalités qui donnent à l'utilisateur les mêmes avantages que les applications natives. Un site web devient une PWA s'il possède trois caractéristiques techniques principales :

- (1) L'application web doit être servie sur un réseau sécurisé avec le protocole HTTPS.
- (2) Un Service worker doit être disponible, offrant la possibilité d'accéder aux ressources y compris hors ligne. L'application devrait fonctionner sur le téléphone même si vous n'avez pas accès au site Web.
- (3) Un fichier manifeste doit être disponible. Il s'agit d'une description du nom de l'application, de son adresse et de divers éléments qui permettront au système d'exploitation mobile de pouvoir présenter la PWA comme une application sur le téléphone.

Les applications qui ont été développées avec cette technologie PWA ont été réalisées dans le cadre de tutorat pour améliorer la réussite des étudiants de première année. L'idée est de proposer des applications qui seront utilisées par les élèves en dehors des cours (transport, au domicile, n'importe où) avec des contenus pour renforcer la compréhension des élèves sur des thématiques qui ont été identifiées comme problématiques pour certains élèves en difficulté. Par rapport aux autres solutions de transmission de contenus, l'avantage est d'avoir un accès facile au support, de bénéficier de notifications pour informer les étudiants des nouveaux contenus et d'une plus grande liberté dans le type de ressources par rapport aux solutions MOOC proposées par l'Université. Ces applications sont développées soit par l'enseignant, soit par des étudiants de troisième année (Licence Professionnelle SARI, IUT de Bordeaux, Département GEII). Plus d'une dizaine d'applications ont été créées au fil des années, avec une présentation en Figure 2 de quelques exemples avec du contenu lié au cours sur le microcontrôleur, avec des ressources et des vidéos sur

Arduino. Il existe également une application sur la carte de Karnaugh avec le même solveur que celui de la Figure 1(b) mais en version PWA, ainsi que l'intégration de vidéos réalisées par d'autres étudiants de deuxième année. Ensuite, une autre implémentation a été faite en utilisant le gestionnaire de contenu WordPress associé à Lifter LMS qui est un système de gestion de l'apprentissage (*Learning Management System, LMS*) qui a été converti avec succès en PWA, permettant de construire des quiz et des activités interactives.

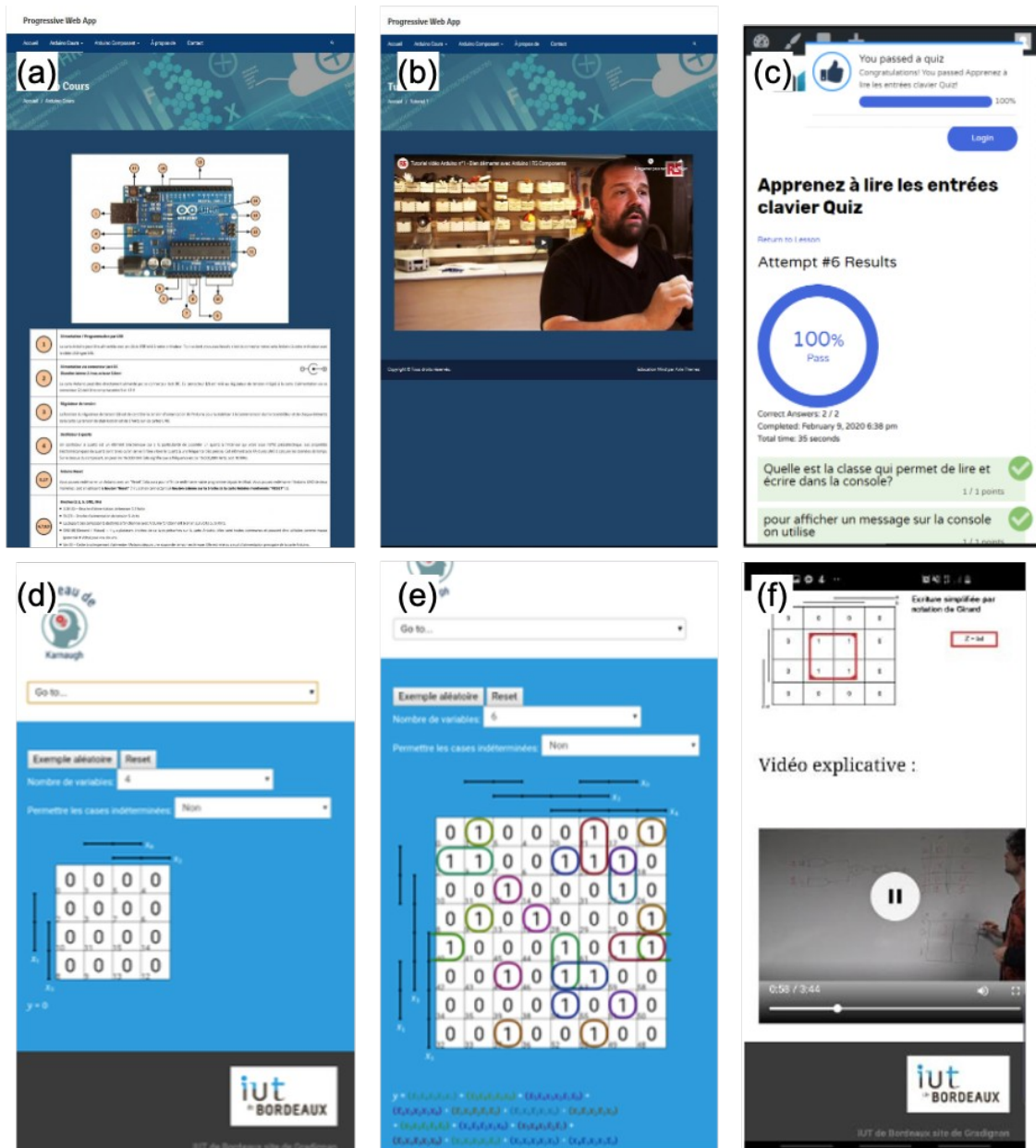


Figure 2 – Exemples d'applications basées sur des PWA réalisées au département GEII pour des étudiants en DUT GEII ou Lience pro SARI (a) Tutoriel Arduino (b) Contenus videos (c) LMS encapsulé dans une PWA (d) Tableaux de Karnaugh à 4 variables (e) Tableau de Karnaugh à six variables (f) Tutoriel pour vidéo pour la résolution des tableaux de Karnaugh réalisés par des étudiants de 2e année

2. Résultats

La distribution des applications et le nombre d'installations dépendent fortement du type d'application et de l'optimisation des moteurs de recherche dans les magasins d'applications. L'application Karnaugh Map construite avec Apache Cordova a été soumise en 2014 au Google Play store et cela a permis d'obtenir des statistiques d'installation, le nombre d'installations selon les pays, ainsi que des statistiques techniques sur les bugs selon les caractéristiques des smartphones. Une grande partie des installations ont été réalisées hors de France, montrant ainsi l'intérêt des étudiants du monde entier pour cette solution. Alors que cette application a été initialement développée pour les étudiants du GEII de l'IUT de Bordeaux Département, elle a été installée sur plus de 40000 smartphones hors de France, et 2000 installations en France. Bien que l'application soit en anglais, elle a donc été utilisée dans de nombreux pays et on peut noter en 2016 une augmentation du nombre de nouvelles installations suite à la rédaction de fiches descriptives de l'application sur le Google Play Store en espagnol et portugais, ce qui a entraîné une augmentation de la colonisation en provenance d'Amérique latine.

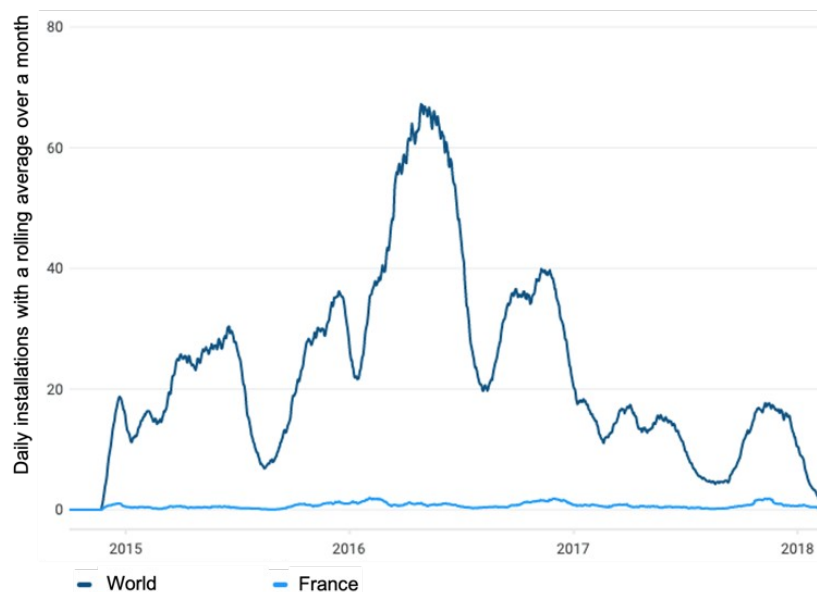


Figure 3 – Nombre d'installations quotidiennes avec une moyenne mobile sur un mois dans le monde et en France sur la base des statistiques du Google Play store d'octobre 2014 à février 2018. On constate que cette application initialement destinée aux étudiants de l'IUT de Bordeaux a été installée sur plus de 40000 smartphones hors de France, et 2000 installations en France.

Ensuite, les applications PWA ont été diffusées uniquement via le site Web personnel ou les MOOC universitaires associés aux projets. Il en résulte une dissémination beaucoup plus faible (quelques dizaines d'installations) correspondant aux utilisateurs qui ont été informés de la présence de ce contenu dans le cadre d'un cours. Le référencement des PWA est donc beaucoup moins apte à diffuser des contenus pédagogiques, même si les récentes annonces sur la possibilité de les référencer sur le Play Store vont sans doute changer la donne à l'avenir. Quoiqu'il en soit, sur la base de cette expérience, le développement d'une application native ou hybride permet donc un référencement dans le Play Store, ce qui induit automatiquement une large diffusion du contenu, au-delà du périmètre de l'université ayant développé le contenu.

Le cycle de développement d'une application mobile est séquencé en quatre étapes principales qui sont (1) la conception, (2) le développement, (3) les tests et (4) le déploiement et la maintenance. Dans le contexte éducatif, la principale difficulté vient de la maintenance, car les projets financés peuvent permettre des développements, mais la maintenance est souvent difficile en raison du manque de fonds récurrents dans le temps pour assurer la pérennité de ces projets. Dans cette étude, je propose une analyse dans le cadre de développements applicatifs réalisés soit par des étudiants dans le cadre d'un projet, soit par l'enseignant. Par conséquent, la comparaison entre les méthodes de développement dépendra principalement des besoins de

mise à jour induits par les différentes technologies. Dans ce contexte, les solutions natives et hybrides nécessiteront des mises à jour fréquentes en raison des failles de sécurité logicielles régulièrement constatées dans les différents frameworks. Si elle n'est pas mise à jour, l'application est automatiquement supprimée du Play Store comme ce fut le cas pour l'application Karnaugh en 2018. En revanche, les PWA font beaucoup mieux car elles reposent sur des solutions purement web qui nécessitent généralement moins de mises à jour, même si les choses dépendent de la méthode de développement choisie. Par exemple, une PWA développée en HTML5, CSS JavaScript nécessite peu de mises à jour, alors qu'une PWA de WordPress avec un plugin de génération PWA nécessitera les mises à jour inhérentes à WordPress. Néanmoins, cette solution PWA nécessitera des mises à jour qui peuvent être automatiques ou généralement plus simples, ce qui rend le travail plus facile et plus durable pour un développement réalisé par un enseignant qui n'est pas informaticien.

Le niveau de compétences informatiques requis pour un enseignant souhaitant développer des ressources pédagogiques sur mobile sera le plus élevé pour les applications natives, puisque les acquis du web technologies ne peuvent pas être directement réutilisés, et parce qu'il faudra un double développement pour obtenir des applications fonctionnant sur iOS et Android. L'approche consistant à utiliser des frameworks hybrides comme Cordova est plus accessible et permettra la production d'applications multiplateformes plus rapidement pour quelqu'un qui n'est pas un expert en programmation. Des bases sur les technologies web seront tout de même nécessaires pour mener à bien un projet. Enfin, la voie d'accès la plus simple pour les enseignants sera certainement la PWA, car il s'agit simplement d'un site web avec quelques fichiers supplémentaires. Il est intéressant de noter qu'il devient possible de créer des applications mobiles sans avoir à coder avec des solutions de gestion de contenu comme WordPress, ou des outils d'intelligence artificielle générative comme ChatGPT, ouvrant ainsi la voie à de nombreux enseignants qui ont un site web personnel et qui peuvent ainsi le transformer en PWA via ajout d'un plugin pour la formation. Par exemple, mon site web personnel fonctionnant sous WordPress a été transformé en PWA et peut donc être installé sur un mobile en tant qu'application. Cette dernière solution sera limitée si vous souhaitez créer des animations interactives, mais largement suffisante pour diffuser du contenu pédagogique.

Conclusion

Sur la base de ces développements sur huit ans, il est possible de comparer les trois techniques de développement d'applications mobiles dans un contexte d'enseignants et d'étudiants créant des ressources mobiles sans être informaticiens. Il en résulte globalement une préférence pour les PWA qui offrent le meilleur accès aux développements mobiles. La pérennité est également bien meilleure du fait du moindre besoin de mise à jour pour des applications simples, ou de la possibilité de bénéficier d'une mise à jour automatique dans le cas d'un déploiement de CMS comme WordPress. Cette solution est donc privilégiée pour les enseignants qui souhaitent débiter. Cependant, des limites peuvent apparaître en termes de performance, qui pourront alors guider l'enseignant vers des solutions hybrides voire vers un développement natif.

De même, dans le cadre d'un projet pédagogique financé, il est important de prendre en compte ces éléments pour assurer une plus grande ampleur aux ressources après la période de développement. Le maintien des ressources étant rarement financé, le choix d'une solution qui rend l'enseignant autonome est judicieux.

L'usage du mobile dans le cadre de l'enseignement supérieur est amené à se développer, avec peut-être aussi une baisse de l'usage de l'ordinateur. Si des outils de qualité existent sur les app stores, il est important de donner aux enseignants la possibilité de créer, agréger ou reformater des contenus et des applications mobiles, pour correspondre à la formation qu'ils souhaitent mettre en place. Pour cela, il est important d'avoir à la fois des solutions accessibles techniquement pour les enseignants non informaticiens, des techniques accessibles aux enseignants de différentes disciplines et aussi d'assurer la pérennité des ressources. Ce travail a pu montrer que, dans ce contexte, les ressources mobiles éducatives les plus pérennes sont celles qui sont ouvertes, avec des solutions basées sur le web. Les conclusions de cet article s'appuient sur des travaux dans les années 2014-2022 et il conviendra d'actualiser régulièrement les conclusions présentées ici, notamment

concernant les solutions techniques de développement mobile. Néanmoins, la pérennité des solutions, leur ouverture doivent être prises en compte par les enseignants pour construire des ressources durables et réutilisables.

Remerciements

Je voudrais remercier plusieurs collègues pour des discussions intéressantes comme Serge Bouter, Ulysse Delabre et François Augereau de l'université de Bordeaux et Thorsten Thormählen de Philipps-Universität Marburg.

Liste des abréviations

- PWA Progressive Web App : Application web qui peut apparaître à l'utilisateur de la même manière que les applications natives ou les applications mobiles.
- HTML5 HyperText Markup Language : Language de representation de pages web
- CSS Cascading Style Sheets : Language de description des éléments de style d'une page web
- MOOC Massive Open Online Course : Plateforme de cours en ligne.
- LMS Learning Management System : Plateforme web de gestion des apprentissages, utilisable pour créer un MOOC ou pour de l'enseignement hybride.
- CMS Content Management System : Système informatique qui permet de créer des sites web en ligne.

Ressources

Contenus et applications disponibles sur <https://terahertz.fr>

Références

- Anni, M. (2021). Quantitative comparison between the smartphone based experiments for the gravity acceleration measurement at home. *Education Sciences*, 11(9). <https://doi.org/10.3390/educsci11090493>
- Benbunan, R. (1997). *Effects of computer-mediated communication systems on learning, performance and satisfaction: A comparison of groups and individuals solving ethical scenarios*. Rutgers The State University of New Jersey-Newark.
- Chen, X., Song, G., & Zhang, Y. (2010). Virtual and remote laboratory development: A review. *Earth and Space 2010: Engineering, Science, Construction, and Operations in Challenging Environments*, 3843–3852.
- Del Pilar García-Chitiva, M. (2021). Collaborative learning in higher education processes mediated by internet. *Revista Electronica Educare*. <https://doi.org/10.15359/ree.25-2.23>
- Drigas, A. S., & Pappas, M. A. (2015). A review of mobile learning applications for mathematics. *International Journal of Interactive Mobile Technologies*, 9(3). <https://doi.org/10.3991/ijim.v9i3.4420>
- Furió, D., Fleck, S., Bousquet, B., Guillet, J.-P., Canioni, L., & Hachet, M. (2017). Hobit: Hybrid optical bench for innovative teaching. *Proceedings of The 2017 CHI Conference on Human Factors in Computing Systems*.
- Hiltz, S. R. (1997). Impacts of college-level courses via Asynchronous Learning Networks: Some preliminary results. *Journal of Asynchronous Learning Network*, 1(2). <https://doi.org/10.24059/olj.v1i2.1934>
- Hochberg, K., Kuhn, J., & Müller, A. (2018). Using Smartphones as Experimental Tools—Effects on Interest, Curiosity, and Learning in Physics Education. *Journal of Science Education and Technology*, 27(5). <https://doi.org/10.1007/s10956-018-9731-7>
- Holder, M. E. (2005). A modified karnaugh map technique. *IEEE Transactions on Education*, 48(1). <https://doi.org/10.1109/TE.2004.832879>
- Kacetl, J., & Klimova, B. (2019). Use of smartphone applications in english language learning—A challenge for foreign language education. *Education Sciences*. <https://doi.org/10.3390/educsci9030179>

- Klimova, B. (2019). Impact of mobile learning on students' achievement results. *Education Sciences*, 9(2). <https://doi.org/10.3390/educsci9020090>
- Kuusisto, R.-H. (2018). Transition from java applet to modern web application.
- Malavolta, I. (2016). Beyond Native Apps: Web Technologies to the Rescue! (Keynote). *Mobile! 2016 - Proceedings of the 1st International Workshop on Mobile Development, Co-Located with SPLASH 2016*, (October 2016), 1–2. <https://doi.org/10.1145/3001854.3001863>
- Malavolta, I., Ruberto, S., Soru, T., & Terragni, V. (2015). Hybrid Mobile Apps in the Google Play Store: An Exploratory Investigation. In *Proceedings - 2nd ACM International Conference on Mobile Software Engineering and Systems, MOBILESoft 2015*. <https://doi.org/10.1109/MobileSoft.2015.15>
- Margea, R., & Margea, C. (2017). Considerations for the Mobile Web. Paradigm Shift. *Informatica Economica*, 21(1/2017). <https://doi.org/10.12948/issn14531305/21.1.2017.02>
- Mella-Norambuena, J., Cobo-Rendon, R., Lobos, K., Sáez-Delgado, F., & Maldonado-Trapp, A. (2021). Smartphone use among undergraduate stem students during COVID-19: An opportunity for higher education? *Education Sciences*, 11(8). <https://doi.org/10.3390/educsci11080417>
- Michalakakis, V. I., Vaitis, M., & Klonari, A. (2020). The development of an educational outdoor adventure mobile app. *Education Sciences*, 10(12). <https://doi.org/10.3390/educsci10120382>
- Napal, M., Mendióroz-Lacambra, A. M., & Peñalva, A. (2020). Sustainability teaching tools in the digital age. *Sustainability (Switzerland)*, 12(8). <https://doi.org/10.3390/SU12083366>
- Reed, J. A., & Afjeh, A. A. (1998). Developing interactive educational engineering software for the World Wide Web with Java. *Computers and Education*, 30(3–4). [https://doi.org/10.1016/S0360-1315\(97\)00062-6](https://doi.org/10.1016/S0360-1315(97)00062-6)
- Sterling, S. (2009). *Sustainable Education: Re-Visioning Learning and Change*. eric.ed.gov.
- Zhang, S. J., & Yu, G. H. (2017). Mobile learning model and process optimization in the era of fragmentation. *Eurasia Journal of Mathematics, Science and Technology Education*, 13(7). <https://doi.org/10.12973/eurasia.2017.00750a>