

LET'S GO APP

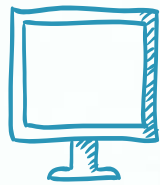
Vue.js – Bootstrap – Ionic – Swift

Anthony Loroscio

Alicia Marin

Thomas Monzein

Thibaut Terris



WEB COMPONENT

Vue.js

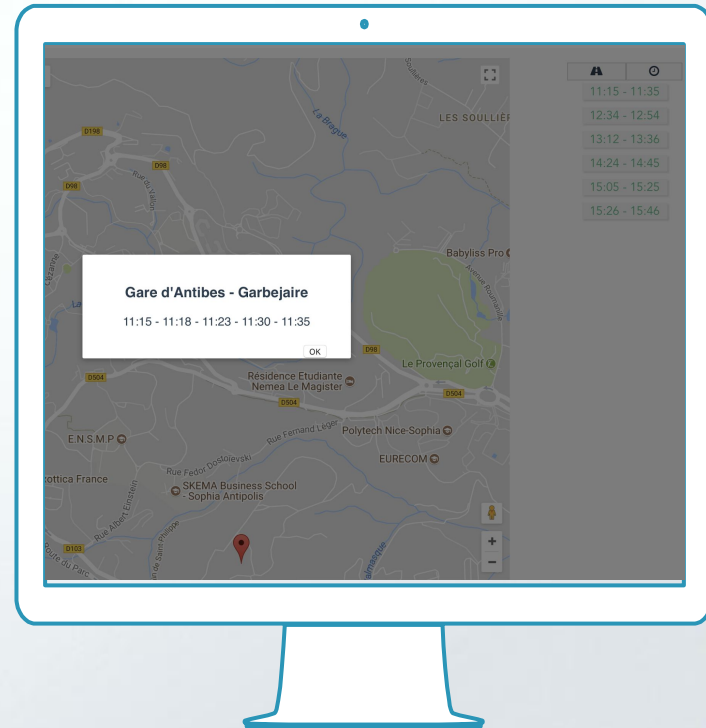
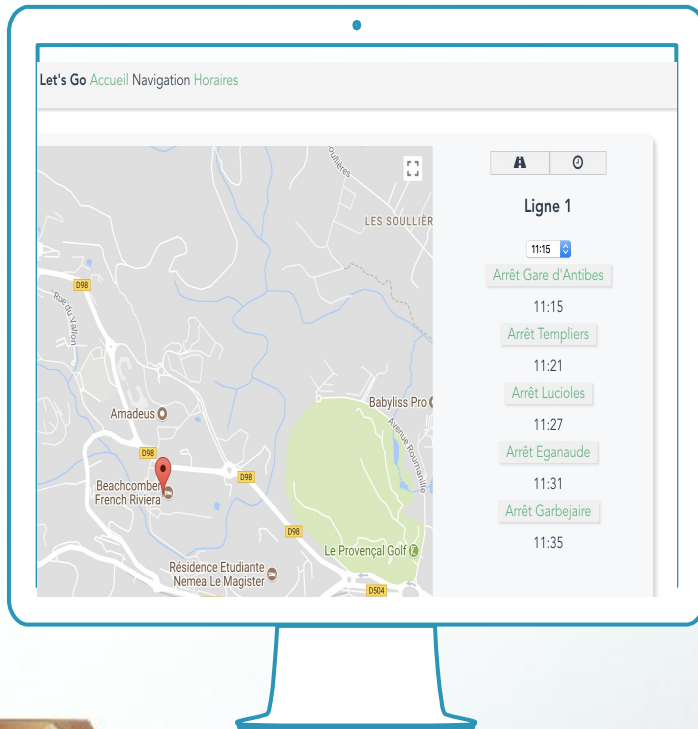
VUE.JS

- + Réutilisation dans différents contextes
- + html/css/js dans le même fichier

- + 3 pages
 - + Accueil, Navigation et Horaires
 - + Non-responsive

NAVIGATION - ONGLETS ET MODAL

“Je souhaite consulter les horaires d'un trajet”



HORAIRES - ONLGETS ET MODAL

“Je désire consulter les détails des lignes”

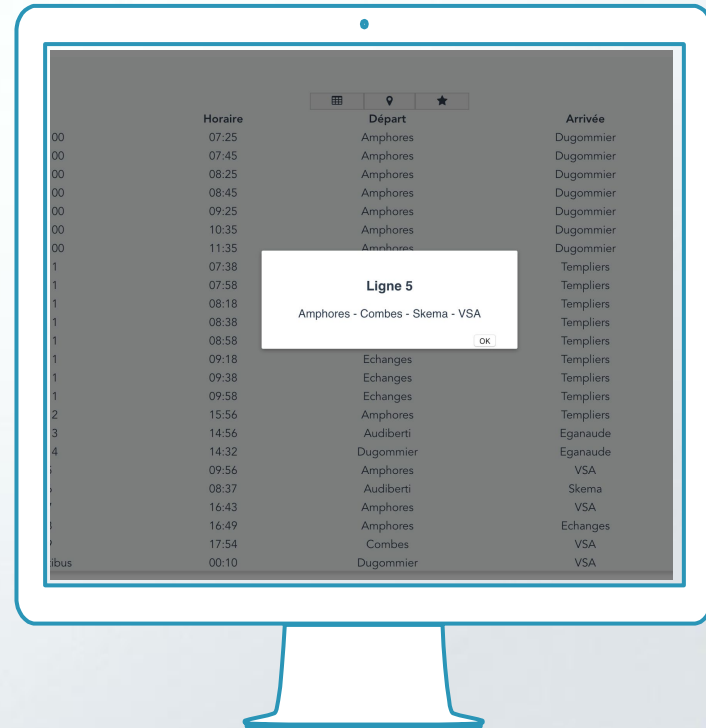
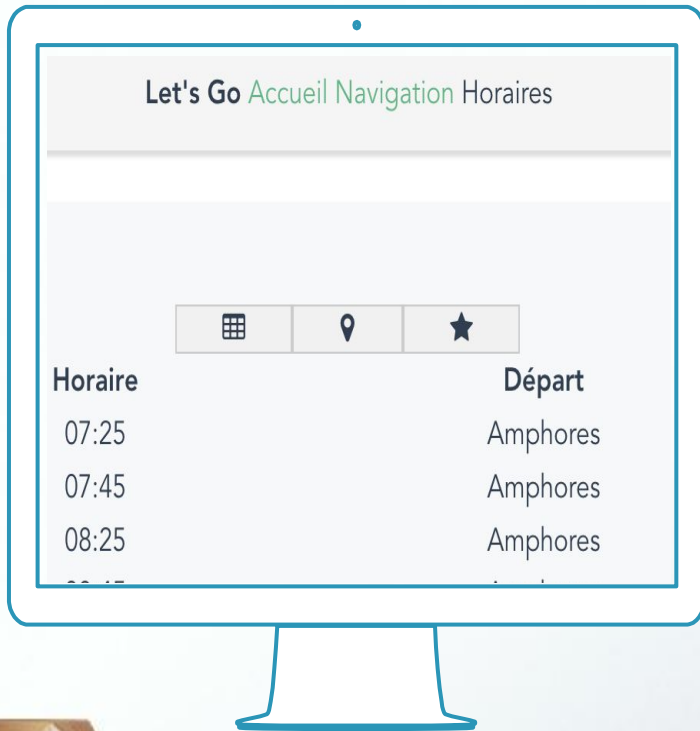


TABLEAU – UTILISATEUR PONCTUEL

“Je désire consulter les détails des lignes”

Ligne	Horaire	Départ	Arrivée
Polybus 100	07:25	Amphores	Dugom
Polybus 100	07:45	Amphores	Dugom
Polybus 100	08:25	Amphores	Dugom
Polybus 100	08:45	Amphores	Dugom
Polybus 100	09:25	Amphores	Dugom
Polybus 100	10:35	Amphores	Dugom
Polybus 100	11:35	Amphores	Dugom
Polybus 1	07:38	Echanges	Templi
Polybus 1	07:58	Echanges	Templi
Polybus 1	08:18	Echanges	Templi
Polybus 1	08:38	Echanges	Templi
Polybus 1	08:58	Echanges	Templi
Polybus 1	09:18	Echanges	Templi
Polybus 1	09:38	Echanges	Templi
Polybus 1	09:58	Echanges	Templi
Polybus 2	15:56	Amphores	Templi
Polybus 3	14:56	Audiberti	Eganau
Polybus 4	14:32	Dugommier	Eganau
Ligne 5	09:56	Amphores	VSA
Ligne 6	08:37	Audiberti	Skerr
Ligne 7	14:43	Amphores	VSA

TABLEAU - UTILISATEUR REGULIER

“Je désire consulter mes lignes”

Ligne	Horaire	Départ	Arrivée
lybus 2	15:56	Amphores	Templiers
lybus 3	14:56	Audiberti	Eganaude

TABLEAU - ENVIRONNEMENT

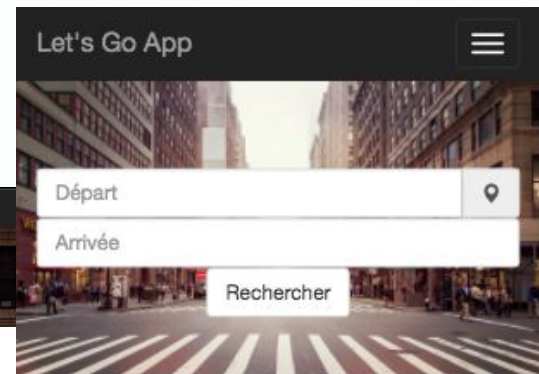
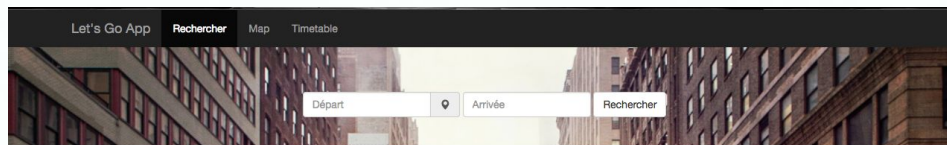
“Je désire consulter les lignes autour de moi”

Ligne	Horaire	Départ	Arrivée
Polybus 100	07:25	Amphores	Dugommi
Polybus 100	07:45	Amphores	Dugommi
Polybus 100	08:25	Amphores	Dugommi
Polybus 100	08:45	Amphores	Dugommi
Polybus 100	09:25	Amphores	Dugommi
Polybus 100	10:35	Amphores	Dugommi
Polybus 100	11:35	Amphores	Dugommi
Polybus 2	15:56	Amphores	Templier
Ligne 5	09:56	Amphores	VSA
Ligne 7	16:43	Amphores	VSA
Ligne 8	16:49	Amphores	Echange

BOOTSTRAP (v 3.3.7)

- + Framework Web
- + Documentation très complète
- + Importante communauté
- + Fonctionne avec un système de grille

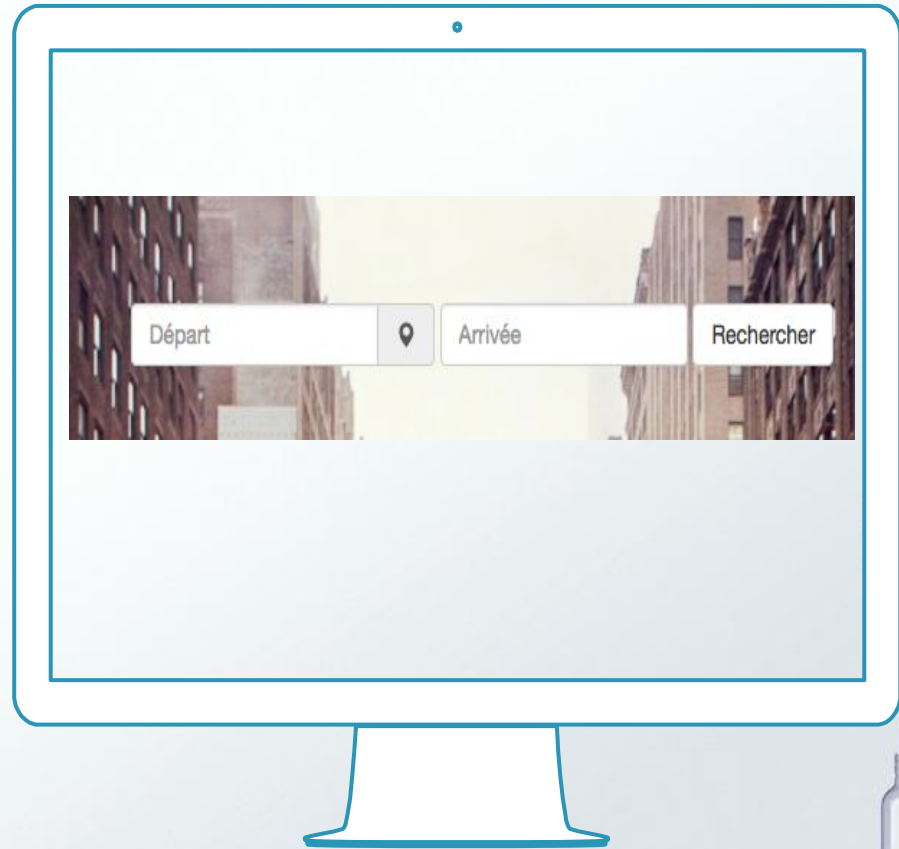
RECHERCHER UN TRAJET



ADAPTATION À L'ENVIRONNEMENT

Trajet de ma position à un
point d'arrivé

Géolocalisation par le
navigateur (JS)

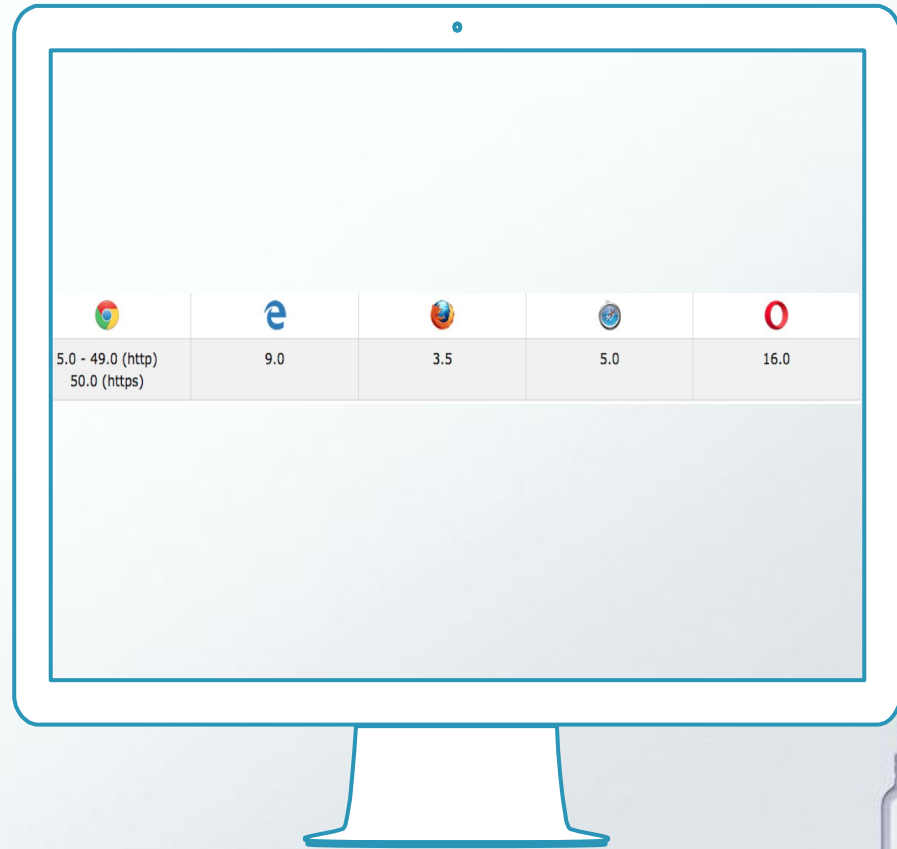


ADAPTATION À L'ENVIRONNEMENT

Trajet de ma position à un
point d'arrivé

Géolocalisation par le
navigateur (JS)

Compatible navigateur
PC/Smartphone



VOIR LA CARTE

Let's Go App Rechercher **Map** Timetable

Liste des lignes

- 1 Sophia-Valbonne
- 2 Sophia-Valbonne
- 3 Sophia-Valbonne



Liste des horaires ligne 1

Choisissez heure de départ

Arret	Heures
Arret1	8h30
Arret2	9h00
Arret3	9h30
Arret4	10h00
Arret5	10h30
Arret6	11h00

Let's Go App



Liste des lignes



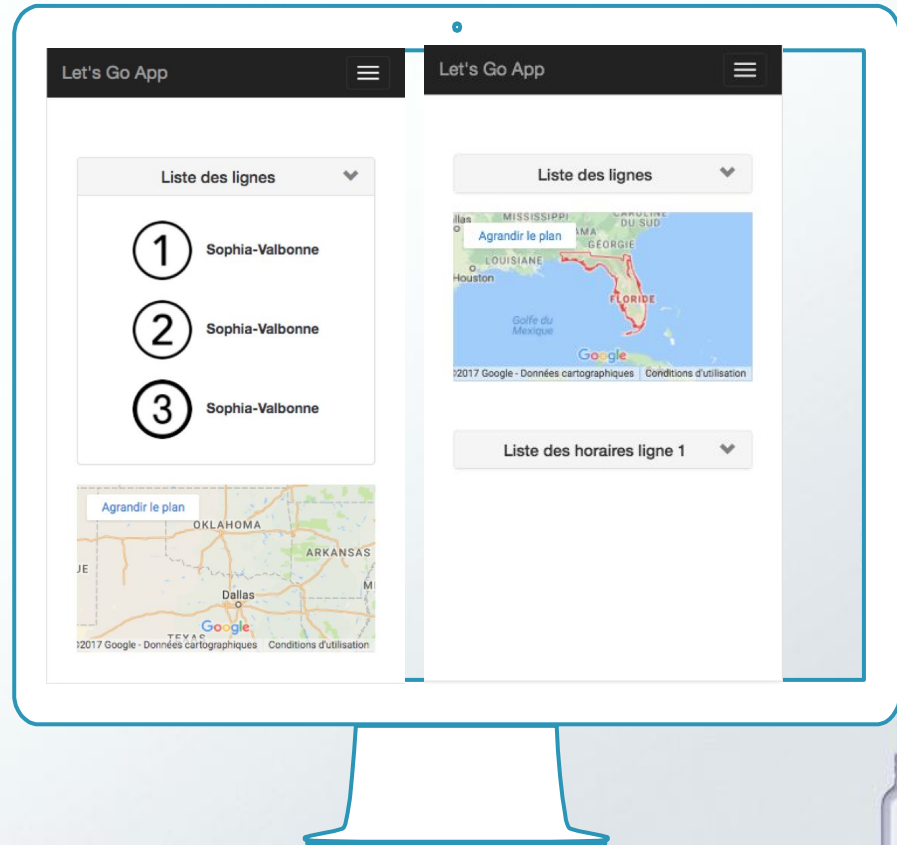
Liste des horaires ligne 1

ADAPTATION AU DISPOSITIF

Voir les différentes
section

Utilisation panel-box +
collapse

Collapse fermé/ouvert
selon taille écran = JQuery



ADAPTATION AU DISPOSITIF

Sélectionner une heure

Combobox cache le
tableau = Perte
d'information

Solution surcharger le
CSS

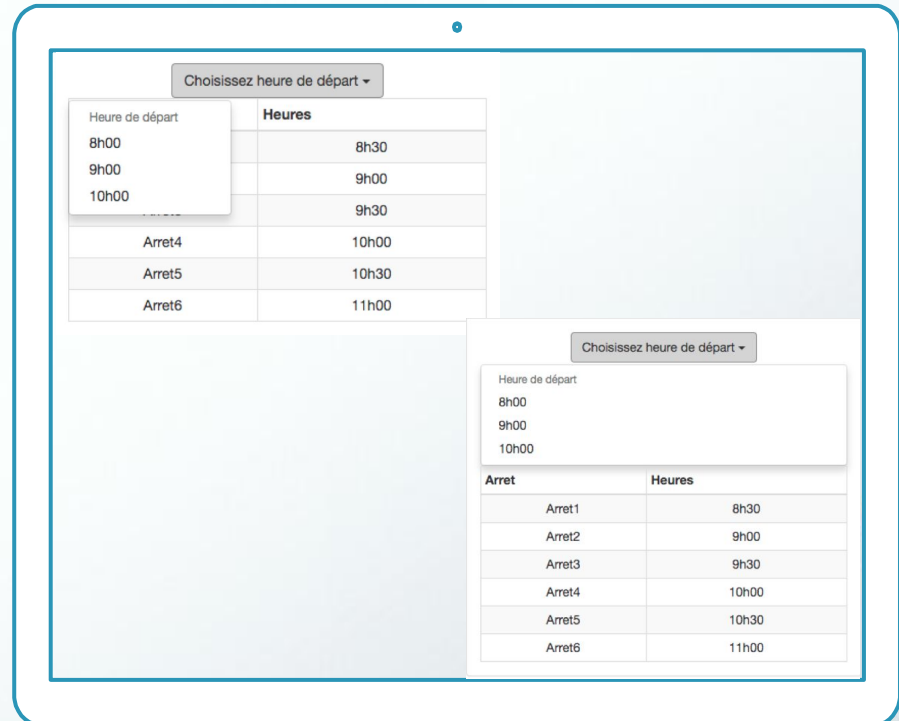


TABLE HORAIRE

Let's Go App



Let's Go App Search Map **Timetable**

Liste des lignes



Liste des lignes



ADAPTATION AU DISPOSITIF

Perte des entêtes du tableau sur téléphone

Media-query qui répète les entetes à chaque ligne

Let's Go App			
	05:53 06:38	06:38 07:23	06:38 07:23
	07:23 08:13	08:13 08:58	08:13 08:58
	08:58 09:43	09:43 10:28	09:43 10:28
	10:28 11:13	11:13 11:58	11:13 11:58
	11:58 12:43	12:43 13:28	12:43 13:28
	13:28 14:13	14:13 14:58	14:13 14:58
Arret5	05:08 05:38	05:20 05:53	05:20 05:53
	05:53 06:38	06:38 07:23	06:38 07:23
	07:23 08:13	08:13 08:58	08:13 08:58
	08:58 09:43	09:43 10:28	09:43 10:28
	10:28 11:13	11:13 11:58	11:13 11:58
	11:58 12:43	12:43 13:28	12:43 13:28
	13:28 14:13	14:13 14:58	14:13 14:58
Arret6	05:08 05:38	05:20 05:53	05:20 05:53
	05:53 06:38	06:38 07:23	06:38 07:23
	07:23 08:13	08:13 08:58	08:13 08:58
	08:58 09:43	09:43 10:28	09:43 10:28
	10:28 11:13	11:13 11:58	11:13 11:58
	11:58 12:43	12:43 13:28	12:43 13:28
	13:28 14:13	14:13 14:58	14:13 14:58
Arret7	05:08 05:38	05:20 05:53	05:20 05:53
	05:53 06:38	06:38 07:23	06:38 07:23
	07:23 08:13	08:13 08:58	08:13 08:58
	08:58 09:43	09:43 10:28	09:43 10:28
	10:28 11:13	11:13 11:58	11:13 11:58
	11:58 12:43	12:43 13:28	12:43 13:28
	13:28 14:13	14:13 14:58	14:13 14:58

Liste des lignes	
	Arret1
Lundi-Vendredi	05:08 05:38 05:53 06:38 07:23 08:13 08:58 09:43 10:28 11:13 11:58 12:43 13:28 14:13
Samedi	05:20 05:53 06:38 07:23 08:13 08:58 09:43 10:28 11:13 11:58 12:43 13:28 14:13 14:58
Dimanche	05:20 05:53 06:38 07:23 08:13 08:58 09:43 10:28 11:13 11:58 12:43 13:28 14:13 14:58
	Arret2
Lundi-Vendredi	05:08 05:38 05:53 06:38 07:23 08:13

CONCLUSION WEB

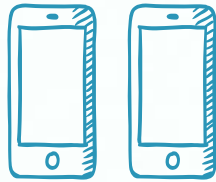
Bootstrap

- + Excellent outil de **mise en page**
- + Facilement personnalisable
- Ne se suffit pas à lui même pour des détails d'adaptation et animation
- Code monobloc

Vue.js

Prendre le temps de définir les composants

- + Gain de temps
- + Meilleur contrôle
- Mise en place
- Communication entre composants fastidieuse



CROSS PLATFORM ET NATIVE

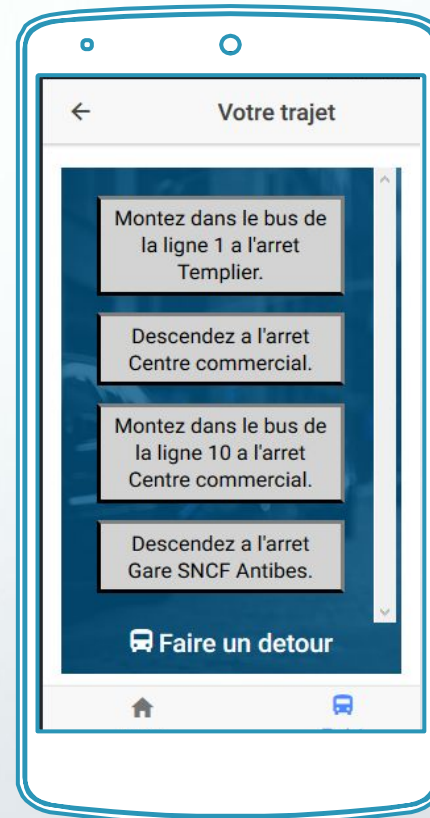
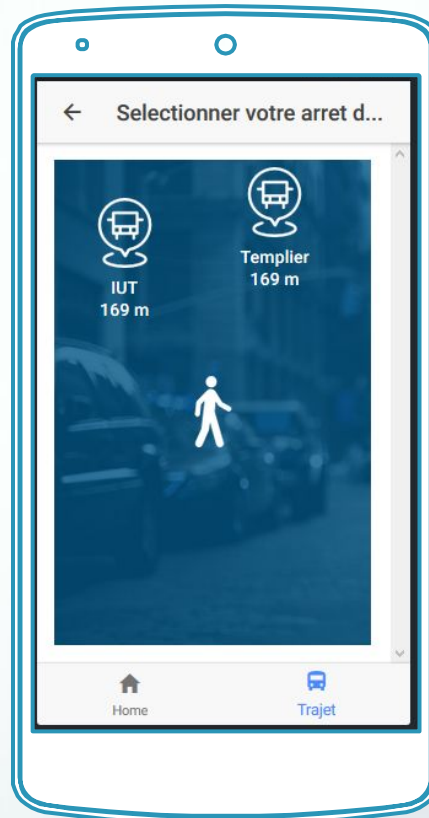
Ionic et Swift

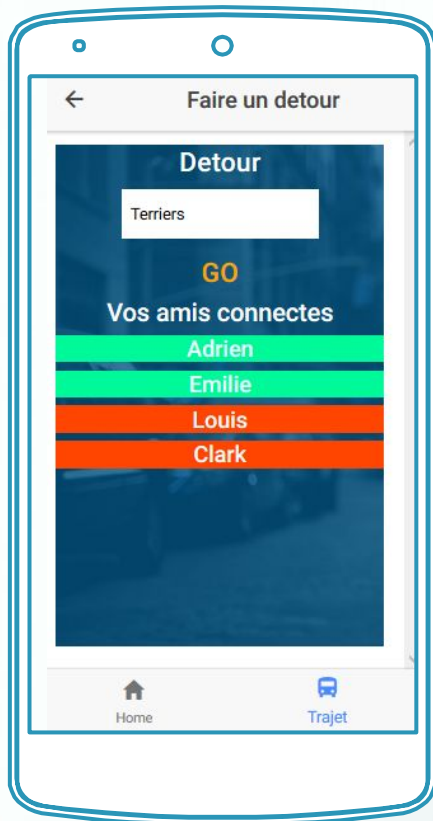
TROUVER UN TRAJET EN BUS

- + Guider l'utilisateur tout au long de son trajet
- + S'adapter aux dispositifs : capteurs
- + S'adapter à l'utilisateur : changement du trajet de l'utilisateur

CROSSPLATFORM : IONIC 3

- + Développement d'un site web
- + Utilise HTML, CSS, TypeScript et Angular 4
- + Un seul code pour toutes les plateformes
- + Ajout de module pour l'accès au capteur





IONIC : BIEN OU MAL ?

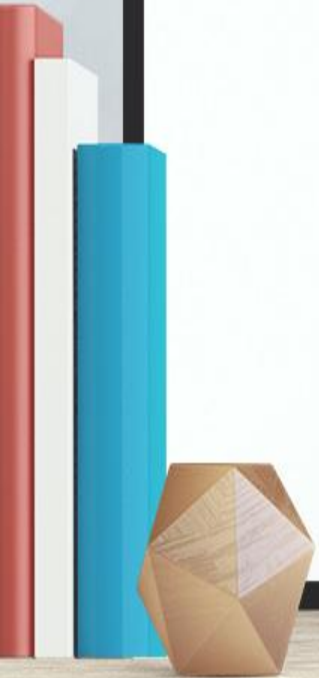
- + Un seul code pour toutes les plateformes
- + Développement Web
- + Accès aux capteurs possible

- Capteur pas très performant en temps (GPS)
- Consomme de la batterie
- Développement Web

NATIF : SWIFT

- + Créé par Apple en 2014
- + Rendu open source en 2015
- + Actuellement : SWIFT 4
- + Chaque version n'est pas retrocompatible

DEMO



SWIFT : ACCÈS AU GPS

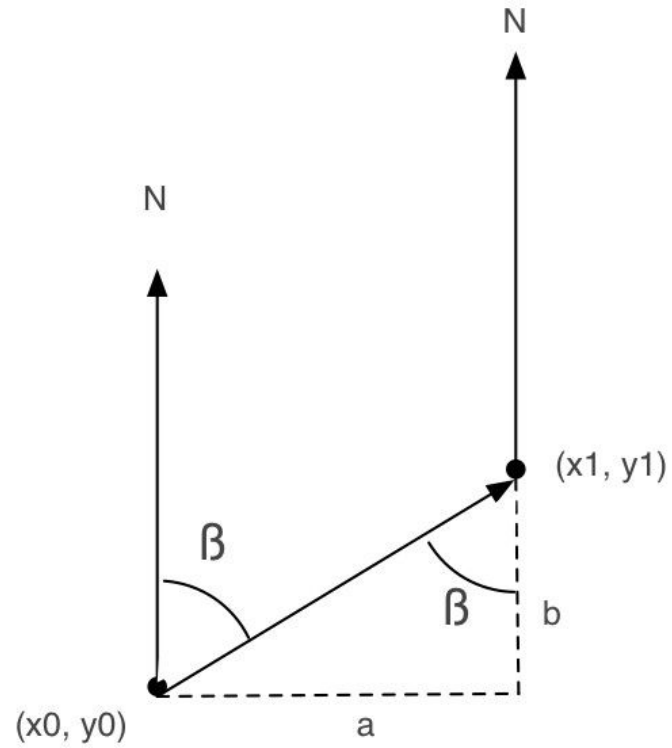
- + Très rapide (en tout cas le plus rapide possible, accès à un tableau)
- + Suffit d'appeler une fonction (`CLLocationManager`)
- + On peut faire varier la précision (jusqu'à 10 décimales)
- + Rafraîchissement en temps réel (l'OS s'en occupe)

SWIFT : LE CAPTEUR GPS ET NOTRE APPLICATION

- + La rapidité est un plus
- + La simplicité est pratique (pour les développeurs)
- + La précision est plus que suffisante
- + Le rafraîchissement en temps réel est bénéfique

SWIFT : BOUSSOLE

- + Assez fluide (dépends du processeur)
- + Indique notre orientation par rapport au nord
- + On doit calculer le palier manuellement



$$\beta = \text{arc tan} (a , b)$$

SWIFT : LA BOUSSOLE DANS NOTRE APPLICATION

- + La fluidité est un plus
- + Calcul manuel : à la fois bien et pas bien
- + Peut être un peu compliqué

SWIFT CONCLUSION

- + Une application, un seul code, pour toutes les devices apple
- + Open Source
- + Grande possibilités d'évolution
- + Beaucoup plus agréable que l'Objective-C
- + Contrôle total sur l'application

- Limité à apple, faut un mac
- Un langage à apprendre
- Xcode et story board à apprendre
- Un SDK à manier

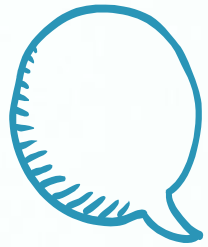
SWIFT PRISE DE RECUL

Bien évaluer les besoins pour ne pas complexifier l'application.

Ai-je besoin des avantages du natif ?

Quel est mon niveau d'exigence ?

A qui ça s'adresse ?



DES QUESTIONS ?

Merci pour votre attention !

Anthony Loroscio Alicia Marin

Thomas Monzein

Thibaut Terris