

SAÉ 3.01

# PLANIFICATION DU PROJET



BOB PARTY

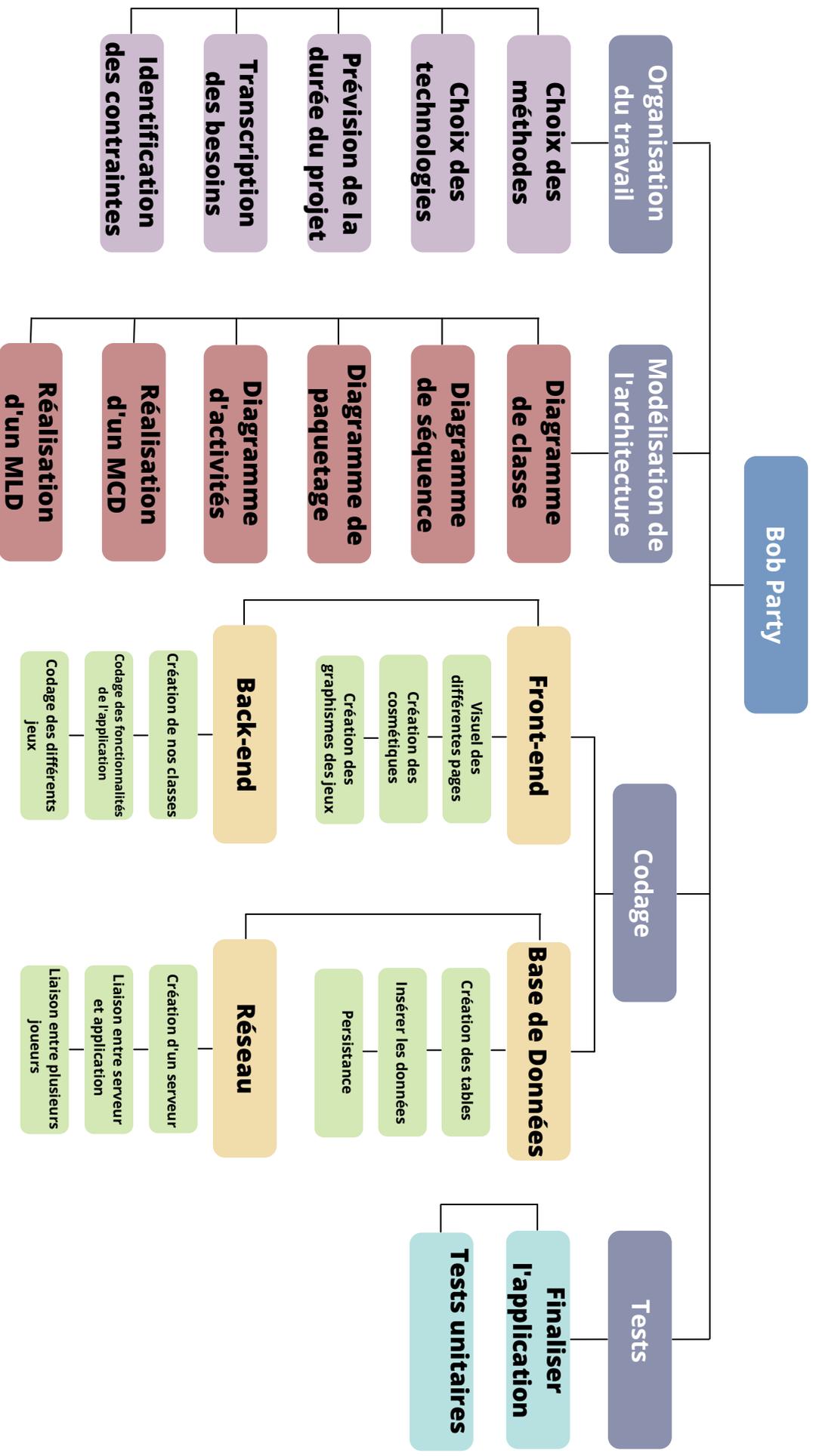


**CHAZOT Thomas, BRETON Lilian, GUILHOT Alban,  
BEDOURET Lucie, JEAN Mathilde**

# SOMMAIRE

- 01** Décomposition du projet - WBS
- 02** Estimation des tâches
- 03** PERT, chemin critique et GANTT prévisionnel
- 04** Estimation des coûts prévisionnels
- 05** Indicateurs de suivi de projet et de qualité
- 06** Conclusion

# WORK BREAKDOWN STRUCTURE



# EXPLICATION DU WBS

Pour ce projet, nous avons réalisé le WBS ci-dessus. Selon nous, notre projet sera découpé en 4 grandes étapes, dans lesquelles nous aurons différentes tâches à accomplir.

Tout d'abord, il fallait organiser nos idées pour pouvoir commencer le projet sur les meilleures bases possibles. Pour cela, nous avons trouvé 5 tâches distinctes. Nous devons en premier nous mettre d'accord sur le choix des méthodes ainsi que des technologies que nous allons utiliser. Par cela, nous entendons la méthode d'organisation que nous allons appliquer ainsi que le choix de tous les langages de programmation et des outils que nous souhaitons utiliser tout au long de notre projet. Ensuite, nous avons dû réaliser la prévision de la durée de notre projet, en prenant en compte toutes les tâches que nous avons identifiées plus tôt. Cela nous a permis de transcrire nos besoins, et donc d'identifier les différentes contraintes que nous posait ce projet.

Après cela, nous avons commencé la deuxième étape de notre projet. Cette étape consistait à la modélisation de notre application via différents schémas et diagrammes. En premier, nous avons besoin d'un diagramme de classes, pour identifier ce que nous allons avoir à coder pour notre application. Nous avons aussi besoin de diagrammes de séquences, de paquetages et d'activités. Ces 3 diagrammes sont utiles pour avoir une bonne visualisation des dépendances entre nos classes définies plus tôt, et ainsi savoir exactement ce que nous allons avoir à coder.

Enfin, nous trouvons les MCD et MLD. Ces 2 diagrammes sont très utiles et importants pour nous permettre de coder notre base de données dans le futur, car ils permettent de schématiser le code que nous allons devoir produire de manière très précise.

Ces 2 étapes sont une sorte de mise en place de notre projet, elles permettent de définir tout ce dont nous avons besoin, tout ce que nous avons à faire, et aussi la façon de le faire.

# EXPLICATION DU WBS

Passons maintenant à la troisième étape, et celle qui comporte le plus de tâches. En effet, nous arrivons au codage. Pour cette grande étape, nous avons décidé de la découper en 4 sous-tâches plus précises. La première, le Front-end, consiste au codage de tout le visuel de notre application. Nous allons donc devoir créer les différentes pages qui la constitueront, ainsi que les cosmétiques et les graphismes des différents jeux qu'elle comprend. Ensuite, nous avons la partie Back-end. Cette partie définit les tâches à réaliser pour que la sublime application codée précédemment devienne fonctionnelle. Pour cela, nous avons besoin de coder les différentes classes que nous avons définies lors du diagramme de classe au cours de l'étape précédente. Une fois les classes en place, nous avons besoin de coder toutes les fonctionnalités présentes dans l'application, faire en sorte de pouvoir naviguer librement dessus. Enfin, nous devons coder nos jeux, le point central et donc le plus important de l'application.

Cependant, le code n'est pas encore fini à cette étape. En effet, nous avons encore besoin de créer une base de données, afin de stocker les données traitées par notre application, comme par exemple les données des joueurs. Nous allons donc devoir créer la base de données et la relier à notre application de façon à pouvoir insérer les données directement à l'intérieur, et de façon automatisée.

Enfin, la dernière tâche de la partie codage se porte sur le réseau. Comme vous le savez sûrement, notre application nous permettra de pouvoir jouer en multijoueur en affrontant des amis, et nous avons donc besoin de créer un serveur, qui hébergera ces parties. Une fois le serveur créé, nous devons le lier à notre application, afin de pouvoir créer une liaison entre les joueurs qui souhaitent s'affronter.

Pour finir, nous avons une dernière tâche, qui consistera aux différents tests. Évidemment, nous allons réaliser des tests tout au long du développement, cependant, nous pensons qu'il est très important de réaliser des tests à la fin. Cela nous permettra d'être sûrs que toute notre application fonctionne, qu'il n'y a aucun bug. Pour cela, nous allons donc avoir besoin de créer des tests, et ensuite de les réaliser. Si notre application réussit à passer tous ces tests, alors nous estimons qu'elle sera terminée, et notre projet s'achèvera donc dans le même temps.

# ESTIMATION DES TÂCHES

Ici, nous avons estimé la durée de chacune des tâches décrites au dessus. Pour cela, nous nous sommes basés sur l'expérience acquise lors de nos précédents projets, ainsi que sur nos compétences à tous. Ainsi, nous arrivons à un total de plus de 700 heures de travail à se répartir à 5 personnes, ce qui fait environ 140h par personnes.

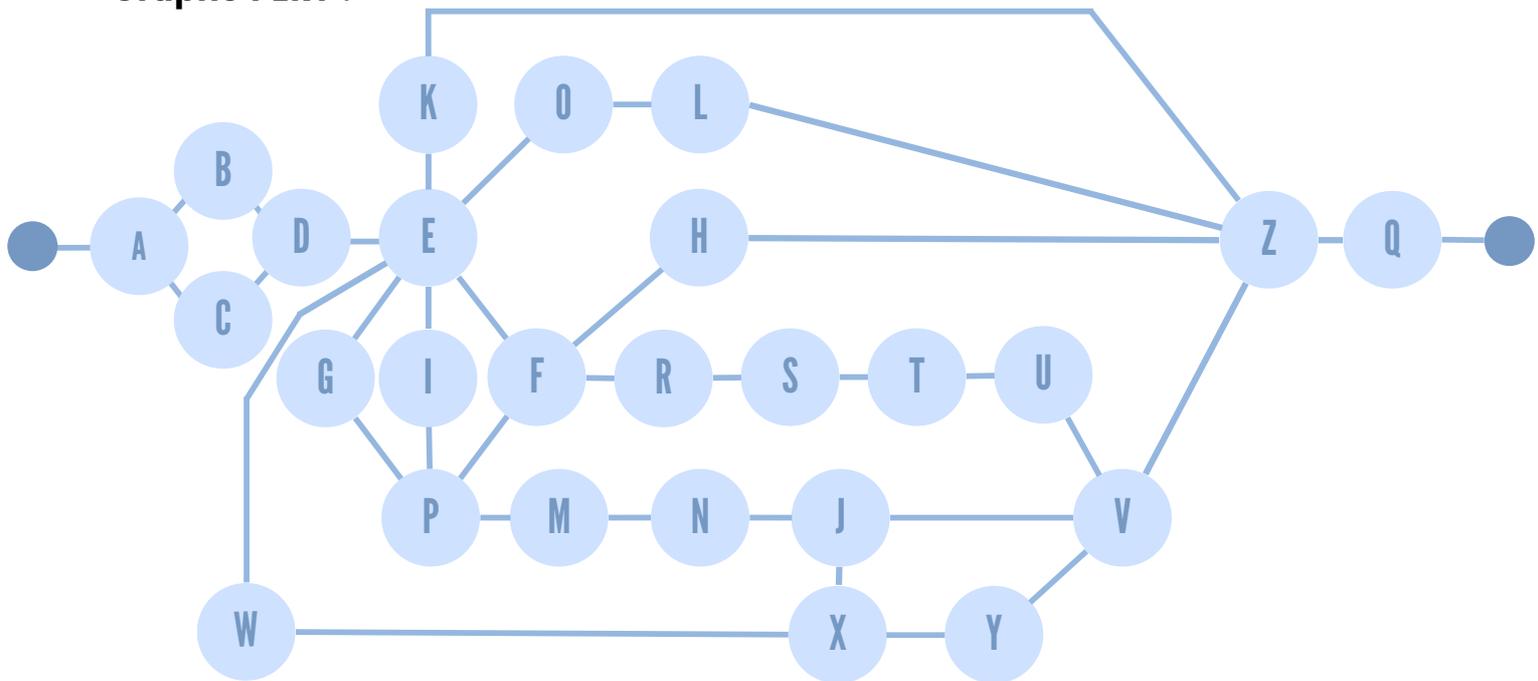
<b>Tâches</b>	<b>Durée</b>
<b>Choix des méthodes</b>	<b>1</b>
<b>Choix des technologies</b>	<b>15</b>
<b>Transcription des besoins</b>	<b>20</b>
<b>Identification des contraintes</b>	<b>5</b>
<b>Prévision de la durée du projet</b>	<b>10</b>
<b>Diagramme de classe</b>	<b>10</b>
<b>Diagramme de séquence</b>	<b>10</b>
<b>Diagramme de paquetage</b>	<b>1</b>
<b>Diagramme d'activités</b>	<b>6</b>

<b>Tâches</b>	<b>Durée</b>
<b>Visuel des différentes pages</b>	30
<b>Création des cosmétiques</b>	3
<b>Création des graphismes des jeux</b>	30
<b>Création des classes</b>	50
<b>Codage des fonctionnalités de l'application</b>	50
<b>Codages des différents jeux</b>	200
<b>Tests unitaires</b>	50
<b>Déploiement de l'application</b>	2

<b>Tâches</b>	<b>Durée</b>
<b>Réalisation d'un MCD</b>	5
<b>Réalisation d'un MLD</b>	2
<b>Création des tables</b>	2
<b>Insérer les données</b>	6
<b>Persistence</b>	30
<b>Création d'un serveur</b>	40
<b>Liaison du serveur avec l'application</b>	60
<b>Liaison entre plusieurs utilisateurs (jeux/chat)</b>	80
<b>Finaliser l'application</b>	6

# PERT & GANTT

Graphe PERT :



## Tâches :

- |   |  |
|---|--|
| A : Choix des méthodes                          | N : Visuel des différentes pages                     |
| B : Choix des technologies                      | O : Codage des différents jeux                       |
| C : Transcriptions des besoins                  | P : Tests unitaires                                  |
| D : Identification des contraintes              | Q : Déploiement de l'application                     |
| E : Prévision de la durée du projet             | R : Réalisation d'un MCD                             |
| F : Diagramme de classe                         | S : Réalisation d'un MLD                             |
| G : Diagramme de séquence                       | T : Création des tables                              |
| H : Diagramme de paquetage                      | U : Insérer les données                              |
| I : Diagramme d'activités                       | V : Persistance                                      |
| J : Codage des fonctionnalités de l'application | W : Création d'un serveur                            |
| K : Création des cosmétiques                    | X : Liaison du serveur avec l'application            |
| L : Création des graphismes des jeux            | Y : Liaison entre plusieurs utilisateurs (jeux/chat) |
| M : Création des classes                        | Z : Finaliser l'application                          |

# DESCRIPTION DU PERT

Afin d'avoir une estimation plus précise du temps nécessaire à la réalisation de notre projet, et surtout de l'ordre dans lequel réaliser chaque tâche, nous avons réalisé un graphe PERT. Ce graphe nous permet de visualiser les liaisons et les dépendances entre les tâches que nous avons définies, et ainsi d'avoir une vision précise du déroulement du projet.

En toute logique, ce projet commence par toutes les tâches définies dans la première étape de notre WBS. Il nous faut impérativement terminer ces tâches afin de pouvoir continuer le projet. Après cela, nous devons réaliser l'estimation de la durée de chaque tâche. C'est seulement à ce moment-là que nous pouvons réaliser plusieurs tâches indépendamment et simultanément.

Commençons par le haut de notre graphe, et donc la tâche K. Elle représente la création des cosmétiques de l'application. Cette tâche est à part, nous n'avons pas besoin de finir d'autres tâches pour la réaliser, et nous pouvons donc la faire parallèlement à d'autres.

Puis, nous trouvons les tâches O et L. Ces tâches sont celles du codage de nos jeux. Elles n'ont pas spécialement besoin de l'application pour avancer, cependant il faudra les relier à cette dernière à la fin du projet afin d'avoir une application finalisée.

Ensuite, les tâches G, I et F. Ces trois tâches représentent des diagrammes, nous en avons donc besoin afin de commencer tout ce qui correspond au codage de l'application. Il y a aussi la tâche H, correspondant aussi à un diagramme mais qui n'est pas obligatoire pour avancer dans le projet. En effet, ce diagramme nous sert simplement à avoir une vision plus claire de ce que nous codons, mais il n'est en aucun cas indispensable pour coder.

Nous retrouvons aussi les tâches R, S, T et U. Ces tâches correspondent à toute la partie de base de données. Elles sont donc importantes pour notre application finale, et nous devons les relier à elle pour que tout fonctionne. Elles nous permettent de gérer la partie persistance, représentée par la tâche V, et qui sert à ce que notre application conserve des données et n'ait pas besoin de tout retélécharger à chaque démarrage.

Les tâches M, N et J sont quant à elles les tâches restantes pour le codage de notre application. Ces tâches peuvent aussi être réalisées à part des autres, et devront aussi être bien reliées à la fin. Nous pouvons voir cela sur notre graphe, via la liaison avec la tâche X.

Cette tâche est la liaison de notre serveur avec notre application. Il faudra donc évidemment avoir créé le serveur au préalable. Une fois cela effectué, il faudra terminer la partie réseau, en permettant aux joueurs de se connecter au serveur pour pouvoir jouer ensemble. Il ne nous restera plus qu'à relier tout ce qui compose la persistance, et de relier tout cela au reste de nos tâches pour finaliser l'application.

Une fois cela fini, il ne restera plus que la tâche Q, qui est le déploiement de l'application, et qui correspond donc à la fin de notre projet.

# PERT TEMPS

Activités	Antériorité	Durée (heures)	Dates au plus tôt		Dates au plus tard		Marges
			Début	Fin	Début	Fin	
A	-	1	0	1	0	1	0
B	A	15	1	16	6	21	5
C	A	20	1	21	1	21	0
D	B/C	5	21	26	21	26	0
E	D	10	26	36	26	36	0
F	E	10	36	46	46	56	10
G	E	10	36	46	46	56	10
H	F	1	46	47	265	266	219
I	E	6	36	42	50	56	14
J	N	50	176	226	186	236	10
K	E	3	36	39	263	266	227
L	O	30	236	266	236	266	0
M	P	50	96	146	106	156	10
N	M	30	146	176	156	186	10
O	E	200	36	236	36	236	0
P	G/I/F	50	46	96	56	106	10
Q	Z	2	272	274	272	274	0
R	F	5	46	51	225	230	179
S	R	2	51	53	230	232	179
T	S	2	53	55	232	234	179
U	T	2	55	57	234	236	179
V	U/Y/J	30	226	256	236	266	10
W	E	40	36	76	56	96	20
X	W	60	76	136	96	156	20
Y	X	80	136	216	156	236	20
Z	V/H/L/K	6	266	272	266	272	0

**Chemin critique : A-C-D-E-O-L-Z-Q**

# DESCRIPTION DU PERT TEMPS

Le PERT temps est un graphe qui nous permet d'estimer la durée nécessaire au minimum pour la réalisation de notre projet dans son intégralité.

Dans notre cas, il nous permet de repérer plusieurs choses. Tout d'abord, la durée minimale de ce projet est de 274 heures. C'est évidemment une estimation minimale, et nous sommes conscients que cela durera en réalité bien plus de temps que cela, mais l'utilité du graphe PERT n'est pas seulement là.

En effet, ce graphe nous permet aussi de repérer le chemin critique de notre projet, c'est à dire les tâches pour lesquelles nous devons absolument respecter la durée prévue, sans quoi notre application prendrait rapidement beaucoup de retard. Ce chemin est ici le suivant : A-C-D-E-O-L-Z-Q. En regardant notre PERT, on remarque que ce chemin correspond principalement aux tâches de codages de nos jeux. Nous savons donc que ces tâches seront importantes, et qu'il ne faudra surtout pas prendre de retard durant leur avancée.



# GANTT PRÉVISIONNEL





# ESTIMATION DES COÛTS

Au cours d'un projet, il est important d'estimer ce que ce dernier va nous coûter. Dans notre situation d'étudiants, il est évident que cela ne va rien nous coûter du tout. Cependant, si nous étions en entreprise, il y aurait de nombreux coûts, que nous allons vous détailler ici.

En premier lieu, il y a évidemment le salaire. Ce projet est estimé à 724 heures de travail au total, en comptant l'entièreté des heures des 5 membres de l'équipe. Nous avons pris comme référence, un salaire de 14€/heure. Ainsi, le total des coûts des salaires pour l'entièreté du projet s'élève à un peu plus de 10 000€.

Ensuite, il faut rajouter le coût des licences des logiciels utilisés, s'il y en a. Dans notre cas, nous utilisons VSCode et Code#0. Ces deux outils sont gratuits, et nous ne prendrons donc pas ce type de coûts en compte.

Enfin, il faut penser au déploiement de notre application. Il nous faut donc déjà le poster sur les différentes plateformes (AppStore et Google Play Store). Pour l'AppStore, il faut payer un abonnement, qui s'élève à 99€ par an. Pour le Play Store, il suffit de payer 25€ une seule fois. Il faut aussi payer un serveur qui hébergera notre application. Ici, le prix est estimé entre 30 et 100€ par mois. Pour l'exemple, nous prendrons la moyenne, donc 65€.

En additionnant tout cela, cela monte les coûts prévisionnels de notre projet à environ 10 200€, auxquels il faudra rajouter 65€ par mois pour le serveur, et 99€ par an pour l'abonnement à l'AppStore.

# INDICATEURS DE SUIVI DE PROJET ET DE QUALITÉ

## Indicateurs de suivi de projet :

Afin de toujours avoir un œil sur notre progression et de ne pas tomber dans le piège du retard, nous allons utiliser différents outils de suivi de projets.

- Le premier est le diagramme de GANTT que nous avons déjà mis en place. C'est un diagramme très clair qui nous permettra de gérer notre temps grâce à ses échéances réalistes et donc de nous motiver avec sa vue d'ensemble sur l'avancement du projet. Nous l'avons fait sur Excel de manière à pouvoir l'adapter au fur et à mesure de notre avancée tout au long du projet en changeant les valeurs directement.
- Comme le diagramme de GANTT est un outil qui couvre tout le projet, nous aurons besoin d'un outil qui nous permettra de gérer nos tâches au jour le jour. Pour ceci, nous utiliserons des To-Do Lists. À chaque début de séance de SAE nous prenons 5 minutes en groupe pour désigner les tâches que nous devons traiter avant la prochaine séance et nous nous les répartissons. Nous reprenons donc le concept de la méthode SCRUM avec ses sprints planning meeting et ses sprints à notre manière avec des planning meeting et des sprints bien plus courts afin de pouvoir être très réactifs en cas de retards ou de changements inattendus dans le projet. Ces To-Do lists sont au format markdown et sont stockées sur notre dépôt code# dans la branche doc.
- Pour avoir une visualisation plus concrète de l'avancée de notre projet, nous allons également utiliser un KanBan, pour pouvoir suivre l'avancée de chaque user story individuellement et ainsi s'assurer qu'aucune fonctionnalité ne soit oubliée.

## Indicateurs de qualité :

Pour tester la qualité de notre projet nous utiliseront des tests unitaires à la manière de la méthode XP, le but est que chaque fonctionnalité ait au moins un test attitré créé avant le développement de l'application qui nous assure que notre travail correspond à nos attentes préalables en termes de qualité de notre projet.

# CONCLUSION

En conclusion, nous pouvons dire que selon notre planification, nous arriverons à finir le projet à temps, avec une qualité prouvée grâce aux tests du code. Si nous faisons bon usage des indicateurs de suivi que nous avons choisis, nous arriverons à être dans les temps. Il faudra cependant être consciencieux dans notre utilisation de ces outils et faire attention au chemin critique pour ne pas prendre de retard à cause d'un manque d'organisation. L'évolution de notre GANTT en fonction des ajustements sur les durées des tâches sera répertoriée dans les prochains dossiers, ainsi que des rapports sur notre organisation collective et individuelle.