

# Chiffrage du projet

Partie	Tâches fonctionnelle	Charge de travail (en jour homme)
<b>Data Fetching</b>		<b>9</b>
	<b>Redux</b>	<b>3</b>
	Mise en place du store initial	0,5
	Mise en place du reducer: Pour chaque service -> 3 actions: REQUEST, SUCCESS et ERROR	1
	Application du Middleware (Redux Thunk)	0,5
	Arborescence des dossiers: Separation du reducer, des services et des actions.	1
	<b>Raider io</b>	<b>3</b>
	Creation du service de fetching	0,5
	Creation des actions	0,5
	Gestion d'erreur	1
	Remplissage du store Redux dynamiquement grace au reducer	1
	<b>Warcraft Log</b>	<b>3</b>
	Creation du service de fetching	0,5
	Creation des actions	0,5
	Gestion d'erreur	1
	Remplissage du store Redux dynamiquement grace au reducer	1
<b>Data Analysis</b>		<b>8</b>
	<b>Importation des informations venant du jeu</b>	<b>2</b>
	Recuperation de tous les joueurs du groupe et de tous les joueurs inscrits dans la file d'attente. Pour chaque joueur on veut son nom et le nom du serveur sur lequel il joue	1
	Construire la base de notre modele de donnée avec les informations récupérées. Les joueurs seront stockes dans un tableau, chaque joueur étant un objet JSON nous permettant plus tard d'aller chercher ses informations sur les sites externes	1
	<b>Traitement des données venant d'api:</b>	<b>6</b>
	Pour tous les combats durant les 30 derniers jours	
	Regrouper les combats par boss	1
	ROW_DATA: Calculer le nombre de dégâts mitigés, de point de vie soignés ou de dégât infligés selon le role du personnage pour chaque combat	1
	RANK: Pour chaque combat, y associer la performance du joueur en % par rapport aux autres joueurs, 0% étant le plus faible, 100% la meilleurs performance effectuée.	1
	Pour chaque personnage, et pour chacun des boss, garder la moyenne de ROW_DATA, le max de ROW_DATA, la moyenne de RANK, le maximum de RANK	1

Partie	Tâches fonctionnelle	Charge de travail (en jour homme)
	Algorithm de notation en fonction des elements au dessus donnant une note a un joueur allant de S+ a D-	2
<b>DevOps</b>		<b>8</b>
	<b>Norme de développement</b>	<b>3</b>
	Installation du linter airbnb	0,2
	Configuration du linter	0,8
	Application des techniques de développement propre au linter airbnb (PropTypes etc..)	2
	<b>Integration continue</b>	<b>2</b>
	Bloquer le git pour que avoir des branches protégées (une branche pour production, une pour staging)	0,5
	Lancement du Linter a chaque push sur toutes les branches	0,5
	Si il a un push sur la branche master ou develop, mettre en ligne le code de la branche sur l'url propre a la version en production ou a la version en staging	0,5
	Faire en sorte que les outils d'analytics soit mis que sur la version en production	0,5
	<b>Analytics</b>	<b>3</b>
	Mise en place Fullstory	1
	Mise en place de Google Analytics	1
	Mise en place de Sentry	1
<b>Front development</b>		<b>13</b>
	<b>Composant React</b>	<b>6,5</b>
	Composant principal (Container)	1
	Composant barre de texte pour importer les informations venant du jeu	0,5
	Graphique de repartition des classes	1
	Graphique de repartition des roles	1
	Tooltip sur les graph	2
	Composant buff manquant	1
	<b>Composant du groupe</b>	<b>6,5</b>
	Nom du joueur (Couleur en fonction de la classe + icône de role + serveur)	1
	Experience du joueur en Raid sous forme de ProgressBar	1
	Experience du joueur en donjon a l'aide d'un score	1
	Affichage du classement du joueur sous forme de % par rapport aux autres joueurs sur chacun des boss (avec un code couleur)	3
	Affiche du niveau d'objet équipé du joueur avec le code couleur qui va avec	0,5
<b>Total</b>		<b>38</b>