

Travail de Bachelor d'informaticien de gestion
Cycle d'études 2013 – 2016
Restitution du travail, juillet 2016

ODOOSIM

Simulation d'entreprise sur PGI

Candidat : **Anthony Tomat**

Directeur : **Boris Fritscher** | Assistant : **Alessio De Santo**

anthony.tomat | boris.fritscher | alessio.desanto | @he-arc.ch

Filière informatique de gestion, HEG ARC – Haute Ecole Arc – Gestion,
Neuchâtel, Suisse

Résumé

Ce travail constitue l'apogée de la formation d'informaticien de gestion au niveau Bachelor, des Hautes Ecoles de Suisse occidentale. Il conjugue les domaines de la gestion d'entreprise et de l'informatique.

Il apporte un certain lot de réponses quant aux démarches à entreprendre pour la réalisation d'un nouveau scénario de jeu sérieux. Il aborde les problématiques à résoudre et les résultats obtenus.

Ce jeu sérieux consiste à mettre en compétition des sociétés entre elles dans une économie de marché. Il gère la plupart des interactions entre les différents agents économiques simulés et plonge les participants dans un environnement simulé au plus proche de la réalité.

L'objectif est de sensibiliser les participants à la gestion d'entreprise et aux progiciels de gestions intégrés. En effet, toutes les opérations sont effectuées grâce à l'outil informatique. Ainsi, les participants capitalisent sur deux axes : métier et informatique. De nos jours, ces compétences sont très recherchées dans le domaine industriel. Savoir aligner la stratégie d'entreprise sur le système d'informations informatisé représente une aptitude de grande valeur.

Afin de mener à bien ce projet, une revue rigoureuse de littérature est effectuée en vue de faire un état de l'art dans le domaine des jeux sérieux. Puis, l'analyse d'un jeu existant est faite afin d'identifier les enjeux et de pouvoir proposer un scénario adapté à un contexte PME. Finalement, une preuve de faisabilité est réalisée sur une partie réduite du scénario développé.

Mots-clés

Jeux sérieux – Formation ludique - Apprentissage – PGI – ERP – Gestion d'entreprise – Processus – Scénarisation – Simulateur – Programmation Java – Gestion de projet – Business – Pédagogie – Odoo.

Partie Introductive

Tables des matières

1.	REMERCIEMENTS.....	1
2.	L'AUTEUR.....	2
3.	AVANT-PROPOS	3
4.	CONTEXTE	5
5.	PROBLÉMATIQUE.....	7
6.	DÉMARCHE.....	8
7.	ÉQUIPE	8

1. Remerciements

Des remerciements doivent être faits à toutes les personnes qui ont contribués de près ou de loin à la réalisation de ce travail.

Monsieur Boris Fritscher,

Pour son soutien quotidien, ses recommandations, ses nombreux retours et les pistes nécessaires à la bonne conduite du travail qui l'a fourni.

Monsieur Do Santo Alessio,

Pour ses retours constructifs sur la réalisation du Poster Session et sur la soutenance à blanc du mardi 14 juin 2016.

Monsieur Bertrand Loison,

Pour avoir apporté une vision statistique sur l'algorithme utilisé pour les ventes.

Monsieur Cédric Gaspoz,

Pour ses retours constructifs sur la soutenance à blanc du mardi 14 juin 2016 et pour avoir mis à disposition les chiffres financiers inhérents à l'acquisition de la solution ERPSim.

Madame Mélanie Donzé,

Pour son aide sur les aspects économiques et sa relecture et correction orthographique de l'ouvrage.

Aux experts,

Pour le temps qu'ils investissent à la lecture et l'évaluation du travail.

2. L'auteur

Anthony Tomat, 27 ans, a débuté l'informatique en obtenant son certificat fédéral de capacité en 2008. Puis, il a continué à se former pour obtenir un diplôme de technicien ES en informatique qu'il a acquis en 2010. Suite à cela, il travaille durant 3 années en tant qu'administrateur système pour un grand groupe horloger Suisse.

Durant son activité professionnelle quotidienne axée principalement en informatique technique, il prend conscience avec intérêt qu'autour de lui tout un monde lié à la gestion et au business existe et touche de près son domaine. Ces constatations le poussent à entreprendre une réorientation. Réorientation qu'il débute en 2013 à la Haute Ecole de Gestion Arc à Neuchâtel.

Puis trois années passent et l'amène à rédiger ce travail qui représente pour lui la dernière ligne droite d'un long chemin menant à l'obtention du titre de Bachelor en informatique de gestion.

3. Avant-propos

« L'opposé du jeu n'est pas le sérieux mais la réalité »

(Sigmund Freud, 1856 – 1939¹)

Cette citation est fortement liée à la thématique de ce travail, les jeux sérieux. Elle apporte beaucoup de sens. En effet, ce qu'il doit en être compris est que les jeux ne sont pas uniquement destinés à des activités dont l'unique but est de s'amuser. Lorsqu'ils sont employés pour la bonne cause, ils peuvent devenir source d'apprentissage et d'apports non-négligeables pour le développement d'une personne. « Un jeu, c'est sérieux ! Car lorsque nous sommes de petits enfants, tous les progrès et les apprentissages, nous les faisons au travers de méthodes ludiques. On découvre en s'amusant et on a envie d'en apprendre d'avantage »² repris d'une citation de Madame Chantal Barthélémy Ruiz lors du salon des solutions ressources humaines tenu à Paris en mars 2011.

Après cette courte introduction au domaine du jeu et à ces bénéfices, pourquoi ne pas introduire ce type de procédé ludique dans les formations académiques ? Elles qui, souvent pâtissent d'être trop théoriques, pas assez pratiques et par conséquent ennuyeuses. Attention toutefois à ne pas prendre ces dires comme des actes de discréditation envers les formations traditionnelles. Le but étant de soulever une problématique et par ce travail, lui apporter un élément de réponse.

Une partie de cette réponse peut être trouvée dans ERPSim. Ce jeu sérieux a été développé en 2004 au Canada. Plus précisément à la Haute Ecole de Commerce de Montréal, par une équipe de chercheur et professeur³. Ce jeu propose de rendre l'apprentissage à la gestion d'entreprise pilotée par PGI plus ludique qu'auparavant avec les formations traditionnelles. Ce qui semble être de grande importance quand on prend en considération qu'« un quart des sociétés utilise un progiciel de gestion intégré pour partager automatiquement l'information en interne »⁴. ERPSim se décline en plusieurs versions qui permettent d'exercer les participants au travers de différents scénarios. Le but de cette diversification est d'entrainer des profils disparates allant des économistes, aux analystes métier ainsi qu'aux informaticiens de gestion. Tous peuvent tirer les bénéfices de ce jeu et de ses scénarios.

Tout naturellement, la Haute Ecole de Gestion Arc à Neuchâtel s'est appropriée la solution pour former ses étudiants. Ses filières en informatique de gestion ou en économie d'entreprise disposant de besoins auxquels le jeu canadien peut répondre.

Les besoins du marché et ceux de l'école identifiés, pourquoi chercher une alternative à ERPSim, qui somme toute semble être la solution idéale ? Pour

¹ https://fr.wikipedia.org/wiki/Sigmund_Freud

² <https://www.youtube.com/watch?v=1zL1gqYJdqk>

³ <https://erpsim.hec.ca/en/team>

⁴ http://www.insee.fr/fr/themes/document.asp?ref_id=tic11

trois raisons. Tout d'abord, cette solution repose sur le système SAP qui ne reflète pas la demande des entreprises de la région. En effet, la démographie des sociétés de la région est majoritairement composée de petites, voire moyennes entreprises. Suite à cela, les scénarios simulent des entreprises avec de grands moyens et des business qui tendent vers ce que l'on trouve outre-Atlantique. Finalement, le point névralgique, celui qui touche l'aspect financier. Cette solution est très onéreuse. Pour accéder au service, il faut s'acquitter de divers coûts allant de la location d'un accès aux systèmes SAP, à l'utilisation des mandants SAP ERPSim et l'obtention des licences étudiantes.

Force est de constater qu'ERPSim est une solution mature et hautement fonctionnelle, elle possède aussi ses désavantages. Là, ce travail prend tout son sens. Dans un futur proche, OdooSIM doit devenir une alternative « Swiss Made » à ERPSim. Son but étant de palier aux désavantages identifiés chez son grand frère.

Cependant, développer une telle solution implique énormément d'analyses et d'aspects techniques à maîtriser. Pour cela, ce travail fait suite à un travail personnel effectué en amont par Anthony Tomat durant le mois de mars 2016. Ce précédent travail avait comme objectif de statuer sur les possibilités offertes par Odoo pour endosser le rôle principal dans le nouveau jeu. Mais aussi d'analyser ERPSim et son scénario Muesli. Le but était de comprendre les enjeux du scénario sur le jeu et d'en tirer de bonnes idées pour en créer un nouveau.

Les résultats ont été probants à l'égard d'Odoo et plusieurs bonnes idées ont pu être récupérées d'ERPSim.

En partant de ces résultats, ce travail doit résoudre des problématiques plus profondes. À savoir, comprendre les tenants et aboutissants de chacun des processus qu'une société met en œuvre et comprendre comment ces processus sont soutenus par l'outil informatique, plus précisément Odoo. Investiguer autour des alternatives de configuration et capitaliser sur des compétences Odooéistique⁵.

Tous les éléments de réponse couverts par ce travail doivent servir de support au futur développement d'OdooSIM.

⁵ Mot inventé

4. Contexte

Jusqu'ici, les formations sont passées de celles que l'on connaît, c'est-à-dire les traditionnelles, aux formations e-learning et jusqu'il y a peu, les jeux sérieux ont commencé à faire leur apparition dans le domaine académique.

Aujourd'hui, « Les jeux sérieux sont très répandus dans les écoles de commerce et permettent de supporter l'enseignement de nombreuses disciplines. Dans le domaine des PGI, le jeu ERPSim développé par HEC Montréal et basé sur SAP est utilisé dans de très nombreuses écoles dont la nôtre. Cependant, SAP n'est pas une solution envisageable pour la majorité des PME. Ce projet va permettre d'analyser si des solutions de PGI orientées PME permettraient de mettre en place un jeu sérieux basé sur un nouveau scénario. » (Fritscher et al., 2016).

Comme le cite le précédent paragraphe, la Haute Ecole de Gestion Arc à Neuchâtel utilise ce type de support pour former ses étudiants. Cependant, elle désire investir du temps dans le développement d'un nouvel outil plus adéquat à la réalité économique de la région de l'Arc-Jurassien.

Dès lors, il devient nécessaire d'effectuer un petit recadrage des besoins. En effet, tout ou partie du pilotage d'une société s'effectue sur la base des données opérationnelles qu'elle génère au quotidien. Ses données font partie intégrante de son patrimoine. Elles témoignent de ses activités et permettent d'en extraire de l'information pertinente pour supporter la prise de décision. Cependant, avant d'arriver à cela, il est nécessaire que ces informations soient traitées et enregistrées pour constituer un élément pérenne. Cet élément prend le nom de SII pour système d'information informatisé. Informatisé car de nos jours les sociétés tendent à l'être. Le but ? Etre réactif face à son environnement.

Le système d'information informatisé dissimule souvent plusieurs éléments que l'on appelle ses constituants. Dans le meilleur des mondes, ils sont interconnectés les uns aux autres, c'est ce que l'on appelle le degré d'intégration. Malheureusement, le travail qu'incombe aux informaticiens de gestion de mettre en place les mécanismes nécessaires qui mènent à un haut degré d'intégration et complexe, prend du temps et coûte de l'argent.

Des solutions plus simplistes que les environnements intégrés existent. Elles permettent d'atteindre un niveau d'intégration proche du 100%. Elles portent le nom de PGI pour progiciel de gestion intégré. Ces PGI sont des programmes informatiques qui intègrent de manière standardisée tous les processus de l'entreprise dans un seul et unique système, une seule base de données. Ceux-ci sont très régulièrement rencontrés dans les entreprises.

Suite à ces observations, l'objectif de ce nouveau jeu est de sensibiliser les étudiants à l'utilisation de ce type d'outil informatique pour gérer une entreprise. Ainsi, lorsqu'ils se retrouvent sur le marché du travail, ils peuvent témoigner d'une compétence reconnue et appréciée par les entreprises. Cette compétence est déjà acquise grâce à ERPSim. Malgré tout et comme il est fait état précédemment, ce PGI n'est malheureusement pas adapté à la demande à laquelle les étudiants doivent faire face en sortant de leurs études.

L'Office Fédérale de la Statistique propose une étude qui observe la démographie sociétale. Les chiffres observés donnent du crédit à l'affirmation précédente :

			Etablissements
2011	Neuchâtel	Micro-entreprises (0 à <10 EPT)	11 844
		Petites entreprises (10 à <50 EPT)	1 104
		Moyennes entreprises (50 à <250 EPT)	247
		Grandes entreprises (250 EPT et plus)	24
	Jura	Micro-entreprises (0 à <10 EPT)	5 751
		Petites entreprises (10 à <50 EPT)	443
		Moyennes entreprises (50 à <250 EPT)	87
		Grandes entreprises (250 EPT et plus)	7
2012	Neuchâtel	Micro-entreprises (0 à <10 EPT)	11 575
		Petites entreprises (10 à <50 EPT)	1 139
		Moyennes entreprises (50 à <250 EPT)	248
		Grandes entreprises (250 EPT et plus)	26
	Jura	Micro-entreprises (0 à <10 EPT)	5 833
		Petites entreprises (10 à <50 EPT)	454
		Moyennes entreprises (50 à <250 EPT)	84
		Grandes entreprises (250 EPT et plus)	8
2013	Neuchâtel	Micro-entreprises (0 à <10 EPT)	11 915
		Petites entreprises (10 à <50 EPT)	1 195
		Moyennes entreprises (50 à <250 EPT)	247
		Grandes entreprises (250 EPT et plus)	25
	Jura	Micro-entreprises (0 à <10 EPT)	5 768
		Petites entreprises (10 à <50 EPT)	472
		Moyennes entreprises (50 à <250 EPT)	83
		Grandes entreprises (250 EPT et plus)	9

Figure 001 : Structure démographique des sociétés de l'Arc Jurassien.

Le résultat est que le taux de petites et moyennes entreprises pour les cantons de Neuchâtel et du Jura est de 99.83% dont uniquement 1.70% sont des moyennes entreprises composées de 50 à 249 employés.

Cette réalité donne du sens à ce projet. En effet, affirmer que la solution SAP n'est pas envisageable pour les sociétés de la région est tout à fait admissible lorsque l'on se réfère aux résultats de cette étude.

5. Problématique

De manière générale, la problématique de ce travail consiste à définir clairement tous les éléments entrant dans la composition d'un jeu sérieux lié au domaine de la gestion d'entreprise sur PGI.

Ce premier paragraphe soulève d'emblée une seconde problématique. Qui dit plusieurs éléments, dit forcément relations. Par conséquent, il s'agit de comprendre et mettre en place les diverses relations qui peuvent et doivent exister entre les composants du système.

Ces deux problématiques ont déjà été partiellement résolues par le travail personnel effectué en amont par Monsieur Tomat durant un travail nommé TP. Cependant, vu le temps court qu'il lui était consacré, ces éléments de réponse reste hypothétiques et à vérifier de manière pratique.

Ce que l'on sait, c'est qu'il faut trouver les moyens de faire interagir des participants, un PGI et un simulateur dans un contexte donné par une histoire que l'on nomme dans le jargon des jeux sérieux, un scénario.

Il est nécessaire de répondre à l'interrogation de ce qu'un bon scénario ? Quels sont les facteurs clés de succès ? Comment est-ce qu'il capte l'attention des participants ? Sont autant de questions auxquelles des réponses doivent être trouvées.

Le terme gestion d'entreprise que l'on trouve dans le titre du travail soulève lui aussi quelques incertitudes et problèmes. Il faut trouver un scénario qui soit réaliste avec le domaine que l'on veut faire apprendre aux participants. Mais qu'elles sont les éléments qui font qu'un scénario lié à la gestion d'entreprise soit pédagogiquement valide ?

À ce moment-là, ce que l'on peut définir comme étant problématique, est la manière que l'on va avoir de résoudre les inconnues liées aux trois principaux aspects que met en relation ce travail. À savoir :

- L'aspect métier lié à la gestion d'entreprise.
- L'aspect pédagogique lié aux éléments que l'on veut faire apprendre aux participants.
- L'aspect technique lié à la mise en œuvre des interactions entre les trois composants majeurs du jeu sérieux.

C'est autour de ces trois aspects et leurs problématiques que les actions performées dans le cadre de ce travail ont à apporter des éléments de réponses.

6. Démarche

Le cheminement pour résoudre les problématiques citées prend la forme qui suit. Il faut de toute évidence débuter le travail en concrétisant les idées de scénarios émises durant le travail personnel pour définir celui d'OdooSIM.

Pour y arriver, il faut sélectionner le type de société que l'on veut faire piloter aux participants. Puis, il s'agit de décortiquer la chaîne de valeur pour définir clairement les processus opérationnels à mettre en œuvre pour créer la valeur pour laquelle la demande a de l'intérêt. Évidemment, tous les éléments de ces processus doivent être maîtrisés pour pouvoir les intégrer dans le PGI.

Un PGI qu'il s'agit de le sélectionner. À ce stade, tout semble se porter sur la solution Open Source Odoo. Cependant, il faut effectuer des tests plus approfondis pour valider les possibilités qu'il offre afin de manipuler son référentiel de données par programmation.

Une fois le scénario, ses impératifs métiers et l'outil définis, commence la phase de développement d'une version limitée d'OdooSIM, à la partie des ventes uniquement.

Pour pouvoir y arriver, la démarche est de créer un référentiel de données en relation avec le scénario et puis, dans un premier temps de jouer manuellement le jeu. Le but de cette action étant d'identifier l'ordre des étapes, les objets manipulés et les contraintes qu'Odoo impose, s'il y en a. Là, des problématiques plus profondes peuvent être identifiées. Il s'agit de leurs trouver des solutions. Une fois que cela est fait, un référentiel des objets Odoo peut être créé ainsi qu'un processus global qui représente les activités à effectuer dans Odoo pour atteindre l'objectif de la société.

Avec ce référentiel et les diverses problématiques réglées ou contournées, le développement doit pouvoir s'effectuer sans encombre. Parallèlement, la prise en main de l'API doit s'effectuer afin d'être productif durant ce temps somme toute limité à quatorze semaines.

7. Équipe

Les personnes qui œuvrent dans la réalisation de ce travail sont au nombre de trois.

La partie maître d'ouvrage est assurée par Monsieur Boris Fritscher dans le rôle de directeur de travail et Monsieur Alessio De Santo en tant qu'assistant.

La partie maître d'œuvre se compose de Monsieur Anthony Tomat uniquement. Dans le rôle de candidat à l'obtention du Bachelor. C'est lui qui doit apporter les éléments de réponse à toutes les problématiques citées.

Partie Théorique

Table des matières

1.	INTRODUCTION	1
2.	LES FORMATIONS LUDIQUES	1
2.1.	<i>Les jeux sérieux</i>	2
2.1.1.	Facteurs clés de succès	3
2.1.1.1.	Analyse des besoins et des attentes.....	3
2.1.1.2.	Définition des objectifs pédagogiques.....	3
2.1.1.3.	Favorisation d'une coopération entre pédagogue et développeurs	4
2.1.1.4.	Création d'un scénario immersif	4
2.1.1.5.	Prévoir une communication efficace.....	4
2.1.1.6.	Prévoir un moyen d'évaluation	4
2.1.2.	Pour quel public	5
2.1.3.	Jeux sérieux et PGI	6
2.1.4.	Un étalon de mesure.....	7
2.1.4.1.	Approche	8
2.1.4.2.	Scénarios	9
2.1.4.3.	Support pédagogique	9
3.	LE SCÉNARIO	10
3.1.	<i>Réalité d'entreprise</i>	10
3.1.1.	Plus-value	10
3.1.2.	Opérations	10
3.1.3.	Parties prenantes	11
3.2.	<i>Réalité des environnements économiques</i>	11
3.2.1.	Principe micro-économique	11
3.2.1.1.	L'élasticité.....	12
3.2.1.2.	L'élasticité prix.....	12
4.	PROGICIELS DE GESTION INTÉGRÉS.....	13
4.1.	<i>Principales caractéristiques</i>	13
4.2.	<i>Motivations d'acquisition</i>	13
4.3.	<i>Processus standardisés</i>	15
4.4.	<i>Accès au service</i>	16
4.4.1.	On-premises	16
4.4.2.	Application Service Provider	16
4.4.3.	Software as a Service	16
5.	COMPARATIF DES PGI POUR PME	17
5.1.	<i>Fonctionnalités</i>	17
5.1.1.	Vente	18
5.1.2.	Stock	18
5.1.3.	Production	19
5.1.4.	Approvisionnement	19
6.	CHOIX DU PGI	19
7.	ODOO	20
7.1.	<i>Couverture fonctionnelle</i>	20
7.2.	<i>Versions disponibles</i>	21
7.2.1.	Online	21
7.2.1.1.	Le programme éducatif	21
7.2.2.	Entreprise	22
7.2.3.	Community	22
7.2.3.1.	Licence LGPLv3	22
7.3.	<i>Pour les développeurs</i>	23
7.3.1.	Tutoriaux	23
7.3.2.	Configuration	24

7.3.3.	Références	24
7.3.4.	Interface de programmation.....	25
7.3.4.1.	Connexion et authentification	25
7.3.4.2.	Création	26
7.3.4.3.	Lecture.....	26
7.3.4.4.	Mise à jour.....	27
7.3.4.5.	Suppression	27
8.	BUSINESS PROCESS MODELING NOTATION (BPMN).....	28
8.1.	<i>La palette d'éléments BPMN 2.0.....</i>	28
9.	SYNTHÈSE THÉORIQUE	29

1. Introduction

Cette partie a pour objectif de poser le cadre théorique qui est utilisé pour atteindre les résultats du travail.

Elle éclaire les principes de formations ludiques et les jeux sérieux qui en sont une déclinaison. Elle présente un des principaux acteurs dans le domaine des jeux sérieux appliqués à la gestion d'entreprise sur PGI. Puis elle met en avant l'importance d'un scénario de jeu dans une telle démarche.

Les progiciels de gestion intégré sont présentés et analysés. Une étude comparative est faite entre divers PGI pour PME. Elle sert principalement à défendre le choix effectué dans ce travail. Ce choix est par la suite présenté.

Le langage de modélisation utilisé dans le cadre de ce travail est présenté afin de donner une base sur laquelle s'appuyer pour comprendre les diagrammes présents dans la partie pratique de cet article.

Finalement, le chapitre se clos en proposant une synthèse des éléments parcourus.

2. Les formations ludiques

Bien souvent, le mot formation rime avec les mots passivité, écoute, professeur et l'on se dit qu'une journée de formation peut très vite devenir longue. Comment tirer parti de ce qu'il est nécessaire d'apprendre si d'emblée, ce sont les premiers ressentis que l'on a ? Force est de constater que dans cette optique-là, l'assimilation des éléments étudiés en est rendu plus difficile.

Il existe pourtant un remède aux limites qu'ont les formations traditionnelles. Cette solution réside dans l'ajout d'une composante ludique. Par ludique, il faut comprendre le terme jeu. Même si le jeu suscite souvent une relation implicite avec des termes comme amusement, non sérieux, bêtise, l'intégration d'une approche ludique dans ses dispositifs pédagogiques permet de rendre l'apprentissage plus dynamique, plus impliquant.

Comme le cite Monsieur D. Vellut dans son article « Pourquoi vous devriez utiliser des jeux en formation et comment les intégrer de manière simple et efficace »⁶, voici deux bonnes raisons à retenir dans le cadre de ce travail. Premièrement, le jeu est le premier facteur d'engagement et de motivation pour les participants. En second lieu, la formation est revue pour intégrer le participant au cœur du processus et le rôle du professeur est celui de « devenir un coach » (Léger et al., 2011).

Suite à ces considérations, le candidat parfait pour faciliter l'intégration d'une ludification dans la formation est le support nommé jeu sérieux.

⁶ <http://www.formavox.com/pourquoi-comment-utiliser-integrer-jeux-formation>

2.1. Les jeux sérieux

Un jeu sérieux n'est autre qu'un « jeu dont la finalité est autre que le simple divertissement. Il s'agit avant tout de faire vivre des expériences enrichissantes, par l'immersion, la simulation et l'apprentissage par l'action. Les jeux sérieux proposent des environnements ludiques des plus immersifs derrière lesquels prennent forme des mises en situation à valeur pédagogique. Les jeux sérieux sont en fait une déclinaison utile du jeu vidéo au service des professionnels. Ils sont présents dans plusieurs secteurs d'activité comme l'éducation, l'administration, la santé, la défense, la formation en entreprise, la communication, l'environnement, l'écologie, l'humanitaire, la sécurité civile, l'art et les sciences. Suivant le public considéré, ils peuvent être utilisés pour la sensibilisation à des problèmes généraux de santé, de sécurité ou d'environnement. »⁷.

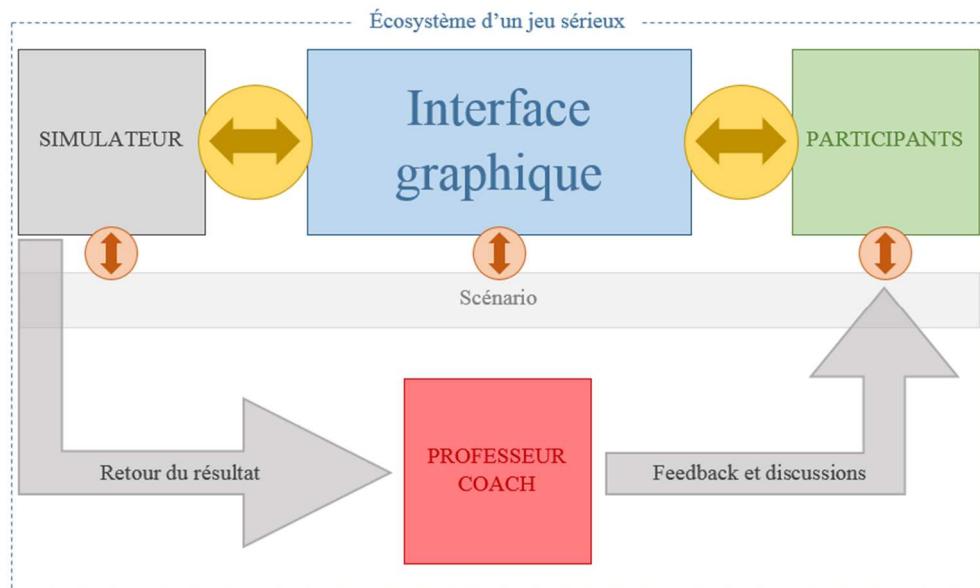


Figure 002 : Un jeu sérieux et son écosystème.

Sur l'illustration qui précède, les éléments composant un jeu sérieux sont illustrés distinctement. Les participants jouent grâce à une interface graphique. Cette IHM permet aux participants de visualiser des informations et de prendre des décisions. Le simulateur récupère les actions utilisateurs et effectue les opérations en conséquence. Tout est contextualisé à l'aide du scénario. Le professeur dans un rôle de coach récupère les résultats que le simulateur génère et donne des feedbacks aux participants afin d'en retirer un apprentissage.

Les jeux sérieux sont depuis quelques temps une solution innovante dans les modes de formation qui existe. En effet, « les professionnelles ont constaté un désengagement des personnes vis-à-vis de la formation classique à distance et se sont mis à rechercher des dispositifs plus attractifs. Le jeu est une réponse possible pour faire renaitre cette adhésion indispensable de

⁷ http://www.gdt.oqlf.gouv.qc.ca/ficheOqlf.aspx?Id_Fiche=26502971

l'apprenant à l'action de formation » (Paraître, 2010). Ces jeux sérieux sont créés par des groupes de personnes pluridisciplinaires. Ces acteurs viennent d'horizons complètement différents et mettent leur propre savoir à disposition. Le développement d'un jeu sérieux met en relation des personnes aux compétences différentes, comme des chercheurs, des informaticiens techniques et de gestions, des spécialistes métier et de clients (Baudet et al., 2015).

C'est le résultat de ce panachage de compétences qui rend un jeu et son scénario intéressants pour les participants et à haute valeur pédagogique.

L'environnement d'un tel produit est caractérisé par le fait que « Le plaisir des joueurs résulte de la sensation de pouvoir et de montée en compétence que leur procure le jeu. » (Paraître, 2010).

2.1.1. Facteurs clés de succès

Ces facteurs de succès sont repris et synthétisé à partir de l'article⁸ de Monsieur D. Vellut.

Tout d'abord, l'auteur reprend les trois raisons pour lesquelles il est envisageable d'utiliser les jeux sérieux. Le jeu est le facteur principal d'engagement pour des apprenants en formation. Il permet de prendre du recul par rapport à la réalité et prend une place de plus en plus importante dans la société actuelle.

L'auteur donne sept pistes afin de réussir un projet de création d'un nouveau jeu sérieux. Cependant, seul six sont réellement importantes dans le cadre de ce travail. Le facteur abandonné étant celui qui met en avant le fait qu'il faut disposer des moyens financiers adéquats car selon l'article, un budget de 50'000 à 500'000 Euros est nécessaire au développement d'une solution de jeu sérieux. Toutefois, le travail du candidat n'étant pas rémunéré, ce point-là n'est pas pris en considération.

2.1.1.1. Analyse des besoins et des attentes

Ce facteur de réussite dépend d'une vision claire des attentes du mandant.

Ce point est tout à fait précis dans le cadre de ce travail. En effet, la réalisation s'appuie sur un modèle qui doit servir de base et par conséquent de repère pour les résultats à fournir.

2.1.1.2. Définition des objectifs pédagogiques

Il est nécessaire de définir des objectifs pédagogiques à atteindre pour tracer un itinéraire précis afin de connaître les compétences nécessaires à développer auprès des participants.

Le titre de ce travail définit clairement les attentes pédagogiques envers la solution à développer.

⁸ <http://www.formavox.com/serious-game-cles-de-succes-formation-entreprise>

2.1.1.3. Favorisation d'une coopération entre pédagogue et développeurs

Pour assurer la cohérence entre la technique et la pédagogie dans OdooSIM, il est nécessaire d'intégrer deux équipes au projet. D'une part des pédagogues et de l'autre des développeurs.

Dans le cadre de ce travail, il n'est malheureusement pas possible directement d'introduire deux équipes de la sorte. Cependant, la communication entre le mandant et le mandaté doit pouvoir palier à cette recommandation.

2.1.1.4. Création d'un scénario immersif

Le scénario étant l'élément central dans la création d'un jeu sérieux, il est par conséquent très important. Principalement parce qu'il permet de capter l'attention des participants.

2.1.1.5. Prévoir une communication efficace

Réaliser un jeu sérieux n'est pas un projet commun. Bien qu'efficace, l'outil reste assez innovant dans le domaine de la formation et il est nécessaire d'investir des moyens en conséquence pour sensibiliser les futurs participants à son utilisation.

Les différents supports que les participants reçoivent doit permettre de donner les lignes directrices pour s'approprier la solution.

2.1.1.6. Prévoir un moyen d'évaluation

L'un des points essentiels dans une démarche visant à former des participants avec un jeu sérieux est de pouvoir leurs donner des feedbacks et de pouvoir les évaluer.

2.1.2. Pour quel public

Les jeux sérieux sont désormais présents dans la plupart des secteurs économiques et « Les domaines d'application des serious games sont à la fois très nombreux et très vastes. On les utilise dans le secteur militaire, aéronautique, pharmaceutique, médical, bancaire, médiatique... et pour la fonction commerciale, marketing, production, logistique, achats, finance... » (Michaud, L. & Alvarez, 2008).

C'est ainsi que « Depuis le milieu des années 2000, toutes les entreprises du CAC40 se sont lancées dans la co-conception d'un ou plusieurs serious games ». Cette affirmation expose l'importance que les grandes entreprises portent à ces nouvelles technologies de formation. Elles n'hésitent d'ailleurs pas à investir des sommes considérables dans le développement de tels outils « ...comme le jeu développé pour Randstad qui a couté plus de 200'000 Euro. » (Allal-Cherif et al., 2014).

Plus près de nous, ce sont nos hautes écoles et universités nationales qui travaillent avec des outils tels qu'ERPSim® pour former leurs étudiants. Il existe d'ailleurs des projets⁹ en cours de réalisation à la HES-SO Valais-Wallis, HES-SO et Haute Ecole de Santé (HEDS) de Genève.

⁹ Projets : « La chambre des erreurs » et « Tournoi « serious game interpro santé » » en lien avec Cyberlearn.

2.1.3. Jeux sérieux et PGI

L'application des concepts du jeu sérieux aux PGI est l'intérêt majeur de ce travail. Le but est de pouvoir ajouter une dimension ludique aux formations dispensées dans le domaine de la gestion d'entreprise à l'aide de ces systèmes informatiques.

À ce stade, il est important de se rendre compte que le développement d'un jeu sérieux pour le domaine sélectionné fait apparaître trois enjeux qu'il s'agit de maîtriser pour obtenir un résultat de qualité. Ils sont liés aux :

- Aspects métiers : Les participants doivent tirer des leçons et mettre en pratique des principes inhérents au domaine étudié. Par conséquent, l'élaboration du scénario doit être faite de sorte à ce qu'il soit le plus réaliste possible. Puis le fonctionnement du simulateur doit permettre de reproduire les interactions du monde réel.
- Aspects pédagogiques : Chaque action que les participants font dans le jeu doit apporter des réflexions et de la valeur pour l'apprentissage.
- Aspects techniques d'implémentation : Toutes les actions des participants doivent avoir un effet sur les éléments contenus dans le PGI. Par conséquent, il faut trouver le moyen de les identifier et de les manipuler par programmation.

Pour que ces enjeux puissent être maîtrisés, il devient nécessaire de connaître avec précision et clairvoyance ce que chaque action des participants implique sur les différents aspects.

2.1.4. Un étalon de mesure

La première étape de ce travail est de trouver un modèle sur lequel s'appuyer et prendre des informations. Ce modèle doit être un jeu sérieux qui a fait ses preuves dans le domaine visé.

Ce modèle est trouvé par le biais d'une solution canadienne nommée ERPSim. Ce jeu sérieux né en 2004, peut se targuer d'avoir capitalisé 17'000 simulations dans 255 universités du monde entier en à peine plus de six années d'exploitation. Plus récemment, durant l'année 2014, ce sont plus de 22'000 étudiants qui ont eu recours à cette solution¹⁰.

Qu'est-ce qu'il en a été fait ? Tout d'abord, ce modèle aide à identifier les éléments constituants un jeu sérieux de ce type-là, c'est-à-dire, les participants, le simulateur, le PGI et le scénario. Ensuite, il permet de comprendre les interactions qui existent entre eux et les responsabilités de chacun.

Sur la base des informations scénaristiques, les participants définissent des stratégies et prennent des décisions en rapport avec les contraintes qui s'exposent à eux. Puis, ils mettent en œuvre leurs résultats dans le PGI.

Le PGI stocke les informations du scénario dans son master data et suite aux indications données par les participants, met en œuvre les opérations pour soutenir les processus de l'entreprise. Ceci est réalisé en adéquation avec les décisions stratégiques prises en amont.

Le simulateur contient la logique interactive et la met en œuvre. Pour se faire, il affecte directement les données du master data du PGI. Afin d'y arriver, il communique grâce à une interface de programmation. En finalité, son but est de faire évoluer le scénario en tenant compte des actions des participants qui se traduisent par des modifications dans le PGI.

Tout ce fonctionnement peut être synthétisé par le schéma qui suit :



Figure 003 : écosystème d'un jeu sérieux lié au domaine de la gestion d'entreprise.

Ce qui doit être compris de ce schéma est que les interactions entre les participants et le PGI s'établissent grâce à une interface homme-machine (IHM). Les interactions entre le PGI et le simulateur s'exécutent avec une

¹⁰ Chiffres repris de « Fashion Game » (<http://fr.slideshare.net/PierreMajoriqueLger/new-erpsim-fashion-game-beta-version-by-erpsim-lab>).

interface de programmation (API). Le contexte est donné à partir du scénario qui impacte les trois parties.

2.1.4.1. *Approche*

ERPSim® propose deux approches complémentaires dans l'utilisation qui en est faite. Premièrement, l'approche d'apprentissage basée sur des problèmes « Problem-based learning PBL is an instructional strategy that focuses on guiding the learning process in a situated context through a series of realistic and potentially complex open-ended problems. »¹¹. Ce qu'il faut comprendre c'est que cette approche consiste à confronter les participants à des problèmes réalistes dans un contexte bien précis, en donnant la possibilité d'aborder des problématiques plus complexes.

Deuxièmement, l'approche « Simulation-based learning (SBL) » est une stratégie d'enseignement immersive où l'étudiant est confronté à des situations de la vie réelle pour un contexte donné. Elles sont structurées de manière à conduire les participants vers un résultat pédagogique attendu par le biais d'objectif. Le simulateur génère des problématiques et force les participants à prendre des décisions afin de les résoudre (Léger et al., 2011).

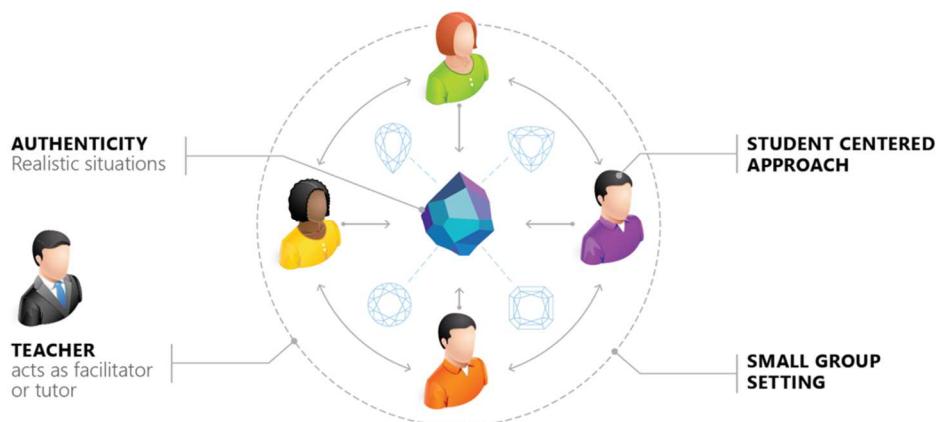


Figure 004 : Approche de formation mis en œuvre par ERPSim®¹²

Dans l'illustration précédente, « les apprenants sont au centre du processus d'apprentissage. Les professeurs prennent moins de place et sont enrôlés en tant que guide » (Léger et al., 2011).

En utilisant ce type d'approche dans les simulations, les apprenants sont plongés dans des situations où ils doivent trouver les solutions adéquates afin de réaliser certaines tâches. En tant que coach, les professeurs doivent donner un minimum d'informations afin de faire en sorte que les apprenants puissent achever les tâches avec un certain niveau de complexité. En utilisant les bonnes connaissances, les bons outils et les bonnes méthodologies, les apprenants identifient les options à disposition et peuvent ainsi choisir la bonne action à réaliser pour résoudre une problématique.

¹¹ Définition de PBL par ERPSim® : <https://erpsim.hec.ca/en/explore>

¹² <https://erpsim.hec.ca/en/explore>

Une observation intéressante de l'article fait état que « dans les premiers temps de la simulation, les apprenants cherchent des conseils plutôt axés techniques. Tandis qu'au fur et à mesure que la partie avance et qu'ils deviennent à l'aide avec l'outil informatique, ils cherchent des conseils plutôt tactiques et stratégiques liés à la gestion d'entreprise » (Léger et al., 2011).

2.1.4.2. Scénarios

Le jeu sérieux ERPSim peut se jouer dans divers scénarios. Actuellement, il se décline en trois versions différentes :



Figure 005 : Scénarios proposés par ERPSim®¹³

Récemment, un nouveau scénario a été développé, il se nomme **Fashion Game**. Pourquoi mettre à disposition des participants des scénarios différents ? Simplement, parce que chacun d'entre eux a des objectifs pédagogiques différents et un public cible qui n'est pas le même.

2.1.4.3. Support pédagogique

Pour rendre compréhensible le jeu et son scénario, la firme canadienne propose plusieurs supports pédagogiques. Ces supports sont dédiés aux professeurs et aux participants. En voici un extrait :



Figure 006 : Scénarios proposés par ERPSim®¹⁴

Professeurs et participants peuvent alors prendre connaissance du fonctionnement du jeu et de la composition du scénario.

¹³ <https://erpsim.hec.ca/en/explore>

¹⁴ <https://erpsim.hec.ca/en/explore>

3. Le scénario

Dans une démarche visant à développer un jeu sérieux au sens large, le scénario en est l'élément central.

Il doit donner du sens aux actions que les participants exécutent et accompagne les interactions des divers éléments de l'écosystème (se référer aux figures 002 et 003).

Pour qu'il soit qualifié de qualité, le principal critère repose sur sa capacité à immerger le participant dans le jeu. Cependant, l'équilibre entre le contenu pédagogique et l'aspect purement ludique doit se faire de bonne manière¹⁵.

Dans l'intérêt de ce travail, d'autres critères doivent entrer en ligne de compte. Par exemple, bien cerner le contexte d'une entreprise et son environnement. C'est ce qui est abordé dans les chapitres qui suivent.

3.1. Réalité d'entreprise

La position stratégique d'un informaticien de gestion se situe au beau milieu des frontières de l'informatique technique et de la gestion. Elle confère une responsabilité supplémentaire, celle de posséder de bonnes connaissances métiers, plus particulièrement dans ce travail où l'un des enjeux majeurs est de comprendre le business d'une société de production, c'est-à-dire, la production et la vente de biens économiques sur un marché compétitif.

Pour se faire, il est nécessaire de comprendre quels éléments entre en ligne de compte pour simuler une entreprise. Les chapitres qui suivent tendent à les expliquer.

3.1.1. Plus-value

La plus-value qu'une société apporte à ses clients est ce qu'elle produit en transformant ou adaptant des éléments entrant dans sa chaîne de valeur. Un bon scénario se doit de définir clairement quels sont les éléments que produit la société, comment elle y parvient et avec quels moyens matériels, financiers et humains.

3.1.2. Opérations

Dans une entreprise de production classique, la chaîne de valeur globale est représentée comme une succession d'étapes, qui à tour de rôle, ajoutent de la valeur sur la matière première travaillée. Ces étapes sont quatre processus bien distincts :

- Prévision des ventes et planification de la production.
- Approvisionnement en matières premières ou en produits semi-ouvrés.
Pour disposer des ressources nécessaires à la production.
- Production s'organisant en atelier ou en ligne pour fabriquer les articles dédiés à la vente.

¹⁵ <http://www.formavox.com/serious-game-cles-de-succes-formation-entreprise>

- Vente des articles fabriqués pour générer des bénéfices.

Il est nécessaire d'analyser de manière globale chacun de ces processus. Ce qui est fait dans la partie pratique, chapitre 13 Analyse des processus opérationnels.

3.1.3. Parties prenantes

La diversité des relations entre l'entreprise et les systèmes qui l'entourent la confronte aux exigences et attentes les plus différentes des acteurs du système. L'une des approches d'analyse et de maîtrise de ces relations est le concept des parties prenantes.

Une partie prenante est par conséquent tout acteur individuel ou collectif ayant une exigence envers une entreprise parce qu'il est concerné par son action. En règle générale, on distingue les groupes d'intérêts selon qu'ils sont internes ou externes et on les classe en quatre grandes catégories :

- Les groupes d'intérêts internes (propriétaires, management, salariés)
- Les groupes d'intérêts économiques (clients, fournisseurs, bailleurs de fonds)
- Les groupes d'intérêts sociaux (Etats, médias, groupes de protection des consommateurs, Eglises)
- Les groupes de défense de l'écosystème (associations, groupes de protection de la nature et de l'environnement)

Les diverses parties prenantes entretiennent des relations bilatérales et sont même connectées en de multiples réseaux (Thommen, 2007).

Il s'agit là d'en simuler quelques-unes au travers du scénario afin d'augmenter le réalisme du jeu sérieux.

3.2. Réalité des environnements économiques

La Haute Ecole de Gestion Arc forme des étudiants dans trois domaines différents ; des économistes d'entreprise, des informaticiens de gestion et des spécialistes en droit des affaires.

OdooSIM doit servir de support à ces trois filières, il devient inévitable d'offrir un scénario qui convienne aux trois types de besoins pédagogiques.

Pour se faire, les interactions entre les sociétés et leur marché doit être le plus réaliste possible. L'un des challenges de ce travail est de pouvoir mettre en œuvre des algorithmes qui permettent de prendre en considération les éléments d'un marché.

3.2.1. Principe micro-économique

Pour pouvoir trouver les bons algorithmes et les mettre en œuvre, il est nécessaire de capitaliser un certain savoir sur les bases micro-économiques. L'élasticité prix-demande est utilisée.

3.2.1.1. L'élasticité

« L'élasticité mesure le rapport de cause à effet qui unit deux variables. Plus l'élasticité est forte en valeur absolue, plus l'intensité de la relation de cause à effet est élevée, c'est-à-dire qu'une faible variation de la cause a des effets très importants. Les élasticités n'ont pas d'unité. On peut distinguer l'élasticité prix et l'élasticité revenus »¹⁶.

3.2.1.2. L'élasticité prix

L'élasticité prix mesure l'effet qu'entraîne une évolution du prix sur la consommation d'un produit. Plus l'élasticité est forte, plus la variation du prix a un effet important sur la consommation. L'élasticité prix répond à la question : « Comment évolue la consommation d'un produit lorsque son prix augmente ou lorsque son prix baisse ? ». L'élasticité prix se calcule de la manière suivante :

$$\text{élasticité prix } (e) = \frac{\Delta \text{consommation en \%}}{\Delta \text{prix en \%}}$$

Exemple :

Si une augmentation du prix de l'essence de 5 % entraîne une baisse de la consommation de 10 %, alors l'élasticité prix de l'essence est de :

$$e = \frac{-10\%}{+5\%} = -2\%$$

Le résultat précédent peut se formuler ainsi :

« Lorsque le prix de l'essence augmente de 1 %, la baisse de la consommation est de 2 %. »

Référence :

Le contenu de ce chapitre est repris tel qu'il est à l'adresse ci-dessous :



http://ses.editions-bordas.fr/eleve/webfm_send/89 ou dans les annexes
TB.AT.Référence-Micro-Economie.pdf

Les éléments présentés sont vérifiés grâce à un ouvrage spécialisé « L'essentiel de la Micro-économie » (Gendron, 2012).

¹⁶ http://ses.editions-bordas.fr/eleve/webfm_send/89 ou dans les annexes
TB.AT.Référence-Micro-Economie.pdf

4. Progiciels de gestion intégrés

Ils font partie des éléments centraux du travail. Ces artefacts informatiques sont de plus en plus utilisés dans les entreprises. Le marché des éditeurs est très fourni et lorsqu'on cherche son PGI, cela devient vite très compliqué tant l'offre est dense.

Ce chapitre a pour vocation de présenter de manière concise les éléments à connaître lorsqu'on parle de PGI.

4.1. Principales caractéristiques

Pour qualifier un logiciel informatique de progiciel de gestion intégré, ce dernier doit respecter un certain nombre de caractéristiques que voici :

- Gestion effective de plusieurs domaines de l'entreprise par des modules intégrés susceptibles d'assurer une collaboration de processus.
- Existence d'un référentiel unique de données aussi appelé master data dans sa version anglophone. Ce terme est souvent employé dans la langue française.
- Adaptations rapides aux règles de fonctionnement de l'entreprise.
- Unicité d'administration du sous-système applicatif.
- Uniformisation des IHM.
- Existence d'outils de développement ou de personnalisation de compléments applicatifs.

Une solution ne répondant pas aux trois premiers critères ne peut pas être qualifiée de progiciel de gestion intégré (Galeuchet et al., 2014).

4.2. Motivations d'acquisition

Pourquoi les entreprises investissent dans ces solutions ? Tout d'abord, il est important de mentionner que le progiciel de gestion intégré permet de soutenir tous les processus d'une entreprise de façon standardisée. Ces standards sont émis par des spécialistes métiers.

Les sociétés sont donc rassurées dans leur mode de fonctionnement vu qu'elles peuvent appuyer leurs opérations sur des manières de procéder allant faire leurs preuves.

Le fait que toutes les données sont transverses à l'entreprise permet de soulager les budgets liés à l'intégration des systèmes hétérogènes. En effet, avec le référentiel de données unique tous les départements accèdent aux mêmes données. Cet avantage est l'un des plus importants. Pour le comprendre, voici à quoi ressemble un système d'informations informatisés, où chaque département travaille de manière autonome :

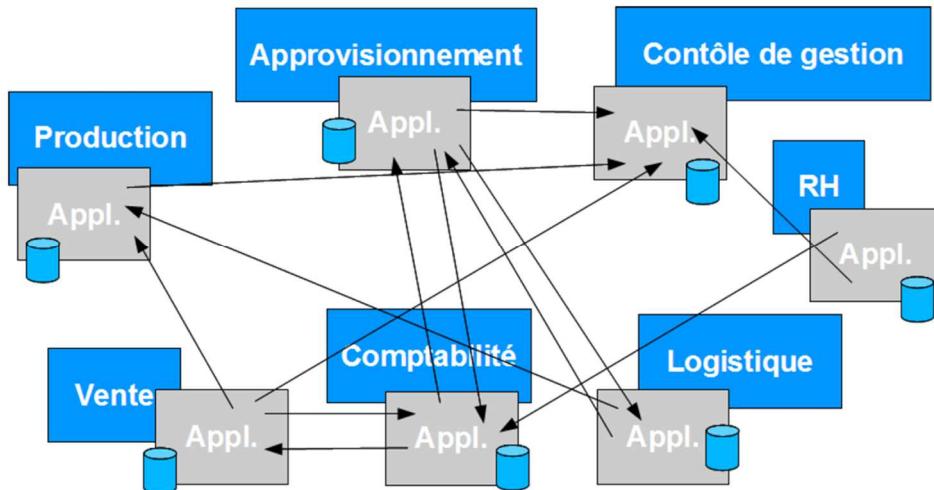


Figure 007 : SII organisé en silos fonctionnels (Galeuchet et al., 2014)

Complément d'informations :

Avec ce type de SII, chaque département gère sa base de données grâce à une application dédiée à leurs fonctions. On peut donc retrouver dans plusieurs bases de données le même client. Par conséquent, si un des départements modifie ce client, cette action impacte uniquement le client dans sa base de données. La problématique de ce genre d'opération est que pour garder un SII homogène, il faut mettre en place des mécanismes d'intégrations entre les applications pour que les modifications des uns se répercutent sur les autres.

i

En faisant l'acquisition d'un progiciel de gestion intégré, le système d'informations informatisé devient beaucoup plus simple et surtout beaucoup plus intégré évitant ainsi les silos fonctionnels :

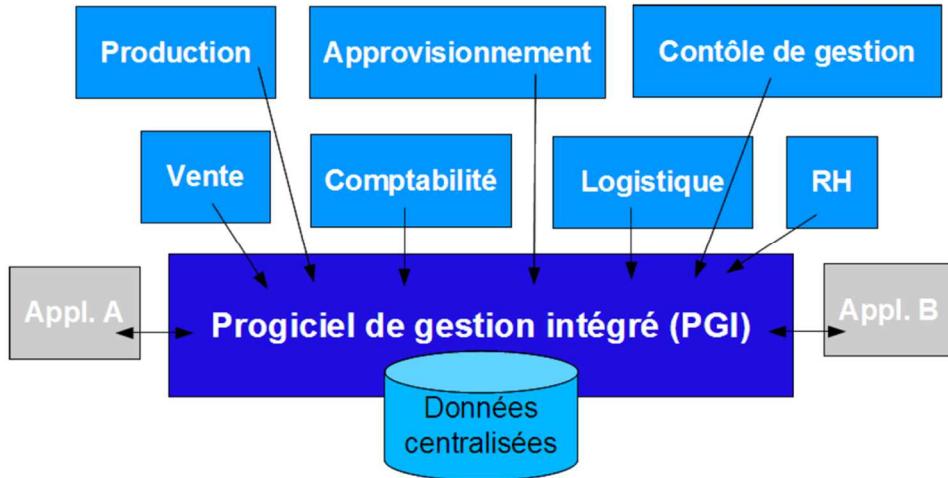


Figure 008 : SII mis en œuvre par un progiciel de gestion intégré (Galeuchet et al., 2014)

4.3. Processus standardisés

Les chapitres précédents l'ont déjà mentionné, un PGI met en œuvre « Des processus respectant les bonnes pratiques reconnues par les experts » (Galeuchet et al., 2014). De ce fait, les processus ressemblent généralement à l'exemple ci-dessous :

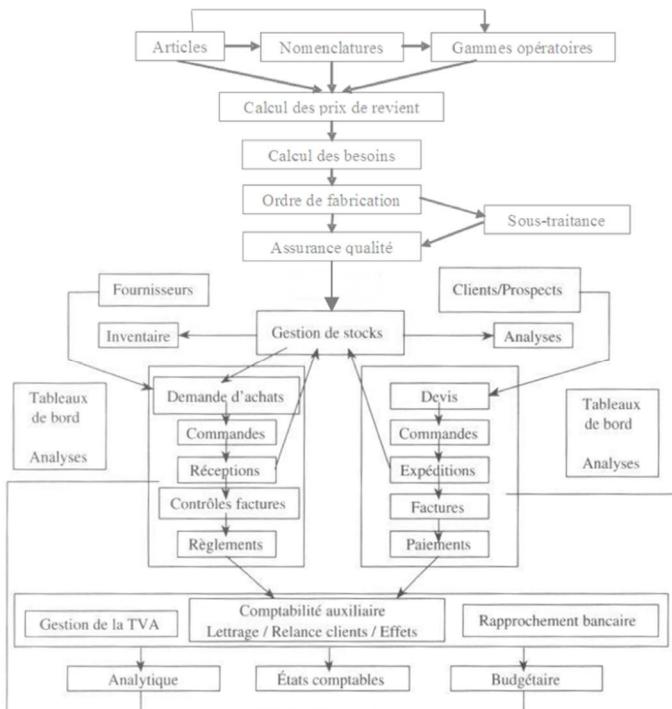


Figure 009 : Processus d'entreprise standardisés (Galeuchet et al., 2014)

4.4. Accès au service

Acquérir un progiciel de gestion intégré nécessite une décision. Comment est-ce qu'on accède au service ? Cela dépend de plusieurs facteurs. Celui qui semble être le plus souvent discriminant étant le facteur financier.

Les éditeurs proposent de donner accès à leur solution au travers plusieurs biais. Soit l'acquéreur prend la décision de s'approprier la solution et doit par conséquent disposer d'une infrastructure privée pour l'héberger. Soit elle n'a pas l'infrastructure nécessaire mais veut tout de même être propriétaire de sa solution. Ou finalement, l'acquéreur peut décider de louer la solution et d'y accéder par Internet sur la base d'un contrat de service.

4.4.1. On-premises

L'entreprise achète la licence et devient donc seul propriétaire du progiciel et d'un numéro de série bien distinct. Elle l'installe dans son propre environnement informatique et assure par ses propres moyens son exploitation et sa maintenance (Galeuchet et al., 2014).

4.4.2. Application Service Provider

L'entreprise est propriétaire du progiciel mais elle sous-traite son installation, son exploitation et sa maintenance à une entreprise de service tierce (Galeuchet et al., 2014).

4.4.3. Software as a Service

« L'entreprise n'est pas propriétaire du progiciel mais elle le loue, sur la base d'un contrat d'abonnement périodique, un service auprès de l'éditeur » (Galeuchet et al., 2014).

De cette manière, l'entreprise n'a pas besoin d'investir dans des infrastructures propriétaires et donc éviter les charges inhérentes à la maintenance d'une infrastructure privée. Pour l'éditeur, c'est aussi un gain en terme de productivité. La maintenance du socle applicatif est faite à un seul endroit sur les serveurs qui fournissent le service à l'ensemble des clients. Ce qui évite aussi le client de devoir maintenir la solution à jour.

5. Comparatif des PGI pour PME

Ce comparatif est récupéré d'une étude menée par Odoo SA (Odoo S.A, 2016). Il compare trois solutions axées petites et moyennes entreprises. Les solutions étudiées sont : Odoo Online, Microsoft Dynamics NAV et Sage 100 Entreprise i7.

Le comparatif commence par une introduction au positionnement des solutions sur deux axes. En ordonné, la couverture fonctionnelle et en abscisse la capacité d'intuitivité. Le résultat est qu'Odoo se positionne sur un marché où l'offre n'est pas conséquente :



Figure 010 : Analyse positionnement des PGI¹⁷.

Suite à cela, le détail des fonctionnalités est étudié.

5.1. Fonctionnalités

Dans les matrices ci-dessous, les solutions sont :

- Microsoft labélisée par PGI1
- Sage labélisée par PGI2
- Odoo labélisée par PGI3

Il est important de prendre en considération que toute l'analyse du comparatif n'est pas présentée car certains éléments ne concernent pas les besoins de ce travail.

¹⁷ www.odoo.com/fr_FR/

5.1.1. Vente

	PGI 1	PGI 2	PGI 3
Vente B2B			
Processus devis à commande	Oui	Oui	Oui
Ventes en magasin			
Point de vente	Non	Oui	Oui
Ventes en ligne			
eCommerce	Non	Non	Oui
Produits – Fonctions avancées			
Variantes Multi-Niveau	Oui	Non	Oui
Configuration des produits	Oui	Non	Oui
Kits	Oui	Oui	Oui

5.1.2. Stock

	PGI 1	PGI 2	PGI 3
Produits			
Traçabilité, lots et numéros de séries	Oui	Oui	Oui
Traçabilité des dates d'expirations	Non	Non	Oui
Unités de mesures multiples	Non	Oui	Oui
Rapports			
En temps réel	Non	Oui	Oui
Productivité			
Règles avancées de logistique, routes et stratégie push/pull	Oui	Non	Oui

5.1.3. Production

	PGI 1	PGI 2	PGI 3
Gestion			
MRP	Oui	Oui	Oui
Routings	Oui	Oui	Oui
Ordres de fabrication	Oui	Oui	Oui
Coûts	Oui	Non	Oui
Planification			
Planification de la production	Oui	Non	Oui

5.1.4. Approvisionnement

	PGI 1	PGI 2	PGI 3
Gestion			
Contrôle de réception	Oui	Oui	Oui
Routings	Oui	Oui	Oui
Ordres de fabrication	Oui	Oui	Oui
Coûts	Oui	Non	Oui
Exécution			
Make-to-Order	Oui	Oui	Oui
Make-to-Stock	Oui	Oui	Oui

6. Choix du PGI

Le choix du progiciel de gestion intégré à utiliser dans le jeu sérieux OdooSIM se porte sur Odoo.

En effet, la phase préliminaire à ce projet avait déjà permis d'explorer les possibilités offertes par Odoo pour manipuler son contenu à partir d'un code informatique externe. Le résultat avait été probant avec comme réponse la proposition d'une interface de programmation disponible sous forme de web services. Suite à cela, le fait qu'Odoo soit une solution Open Source va dans son sens.

Finalement, la prise en considération du comparatif ci-dessus permet de prendre la décision qui suit, Odoo est le candidat parfait pour les besoins de ce travail.

7. Odoo

Qu'est-ce qu'Odoo et d'où vient-il ? Odoo, « anciennement OpenERP et Tiny ERP, est initialement un progiciel open-source de gestion intégré comprenant de très nombreux modules permettant de simplifier la gestion d'entreprise dans son ensemble. Le logiciel est utilisé par plus de deux millions d'utilisateurs pour gérer leurs entreprises à travers le monde » (Wikipédia, 2016). Mais c'est aussi « un logiciel libre disponible en trois versions, deux sont locales tandis que la troisième est hébergée dans le Cloud. Plus de 7'300 applications sont disponibles grâce aux développements d'Odoo S.A et de sa communauté. ... Dans cette solution hautement modulaire, chaque opération de l'entreprise peut se faire via une application dédiée. Ceci permet aux entreprises en pleine croissance de commencer avec quelques apps et ensuite d'en adopter d'autres en fonction de ses besoins qui changent et évoluent. » (Odoo S.A, 2016).

Odoo® en quelques chiffres c'est aussi :



Figure 011 : Partenaires, utilisateurs, pays et applications (Odoo S.A, 2016).

7.1. Couverture fonctionnelle

Odoo est très riche en applications dont toutes ont une responsabilité bien précise comme en témoigne cette phrase « One need, one App. Expand as you grow »¹⁸.

La philosophie employée est de proposer une application par besoin fonctionnel. D'après les tests effectués dans l'analyse préliminaire, cela fonctionne plutôt bien. Pour donner un aperçu de l'offre, voici quelques possibilités d'applications :

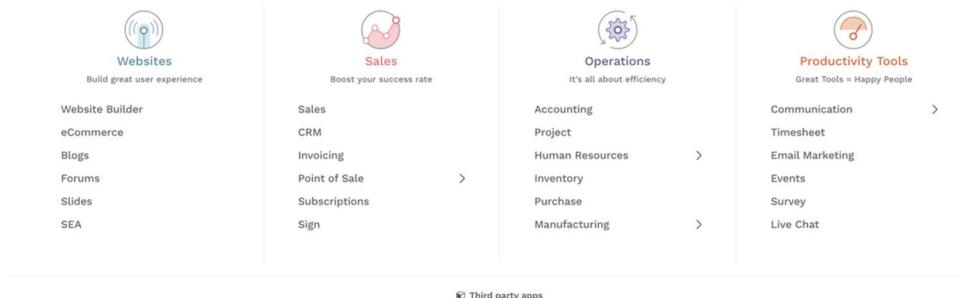


Figure 012 : Applications proposées par Odoo S.A¹⁹

¹⁸ https://www.odoo.com/fr_FR/

¹⁹ https://www.odoo.com/fr_FR/

7.2. Versions disponibles

Odoo S.A commercialise trois versions différentes²⁰ :

- Odoo Online
- Odoo Entreprise
- Odoo Community

Elles diffèrent toutes les unes des autres en termes d'accès au service, de prix et de fonctionnalités.

7.2.1. Online

Cette version est gratuite pour autant que l'on exploite une seule application. Elle est surtout proposée pour effectuer des tests. La version payante quant à elle est facturée à partir de 20.00 Euros par mois et par utilisateur. La facturation est progressive en fonction du nombre d'application installée. Le prix pour toutes nouvelles applications est de 20.00 Euros supplémentaires. Quant à l'accès au service, ce dernier s'effectue en mode SaaS sur les serveurs odoo.com.

Un des avantages très appréciable de cette version est sans aucun doute son programme éducatif²¹. Lorsque l'on s'inscrit à ce programme, on peut accéder au service Online avec toutes les applications, de manière illimitée et gratuite. Toutefois, certaines contraintes sont présentes.

7.2.1.1. Le programme éducatif

Le programme éducatif permet aux enseignants et aux étudiants du monde entier d'échanger sur une plateforme commune.

Teachers have the opportunity to:



Figure 013 : Opportunités offertes par la plateforme éducative Odoo²².

Ce programme donne accès à toutes sortes d'avantages et permet à n'importe quelle personne de venir partager ses connaissances ou venir apprendre à utiliser la solution.

Pour ce travail, ce programme a permis d'employer la solution la plus récente et à jour d'Odoo en mode SaaS sans contraintes autre que le nom des bases de données qui doivent être préfixée de **edu-**.

Un guide est disponible dans la partie des annexes (se référer au document **EN_Odoo 9 Guidelines Education Oct15.pdf**) afin de mettre en place un compte et créer des bases de données à vocation éducative.

²⁰ https://www.odoo.com/fr_FR/page/editions

²¹ https://www.odoo.com/fr_FR/page/education-program

²² https://www.odoo.com/fr_FR/page/education-program

7.2.2. Entreprise

Cette version est, selon Odoo, adaptée aux entreprises de plus de cinquante employés nécessitant un accès au PGI. Elle est accessible en mode **On-premises** ou en **Application Service Provider**. Son prix est de 20.00 Euros par mois et par utilisateur. Dans cette version, toutes les fonctionnalités sont débloquées.

7.2.3. Community

Cette version sous licence LGPLv3 est disponible en libre téléchargement sur le site [odoo.com](https://www.odoo.com)²³. Elle ne possède de loin pas toutes les fonctionnalités de ses sœurs. Cependant, tous les processus opérationnels standards sont couverts.

7.2.3.1. Licence *LGPLv3*

Cette licence limitée, ou amoindrie, c'est-à-dire qu'elle est moins restrictive que la licence GNU GPL dont elle découle, a été créée pour permettre à certains logiciels libres de pénétrer tout de même certains domaines où le choix d'une publication entièrement libre de toute l'offre était impossible.

Cette licence s'applique souvent aux bibliothèques. La signification originale du sigle LGPL est d'ailleurs Library General Public License. Le sens du L de LGPL a été modifié de Library en Lesser afin d'éviter les nombreuses inductions en erreurs. En effet, il arrivait souvent qu'un développeur, qui souhaitait publier sous licence libre une bibliothèque qu'il avait écrite, choisisse la LGPL en ignorant les limitations dans le copyleft (caractère héréditaire de la GPL, qui nécessite que tout programme utilisant une composante en GPL soit en GPL, notamment) par rapport à la GPL (Wikipédia, 2015)

²³ https://www.odoo.com/fr_FR/page/download

7.3. Pour les développeurs

Odoo S.A propose toute une ribambelle de supports aux développements de sa solution. Les spécifications de l'ORM et les tutoriaux à disposition des internautes expliquent comment effectuer toutes sortes de manipulations visant à étendre les fonctionnalités.

7.3.1. Tutoriaux

Comme cité précédemment, ils donnent accès à des marches à suivre pour effectuer diverses actions avec la solution. Ci-dessous, les principaux tutoriaux qui sont disponibles :

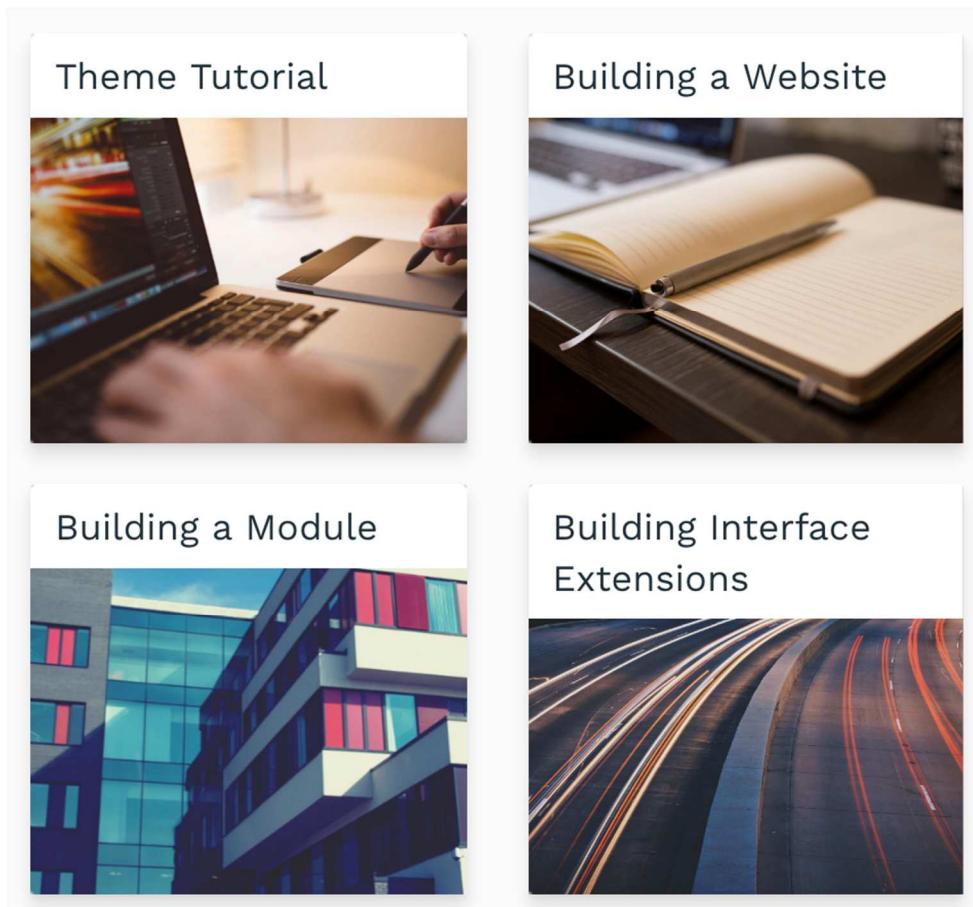


Figure 014 : Tutoriaux en libre accès sur le site [odoo.com²⁴](https://www.odoo.com/documentation/9.0/index.html).

²⁴ <https://www.odoo.com/documentation/9.0/index.html>

7.3.2. Configuration

Toute la partie installation et configuration du PGI est aussi présentée au travers des différents liens :

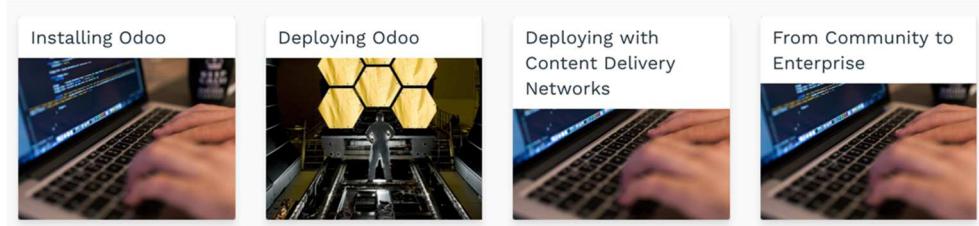


Figure 015 : Liens d'accès aux diverses parties d'installation et configuration²⁵.

7.3.3. Références

Plus technique et plus ciblé développeur, les références qui sont à employer lorsqu'il est nécessaire d'étendre les fonctionnalités du PGI. Les diverses rubriques sont aussi à disposition pour acquérir du savoir et comprendre la structure du PGI :

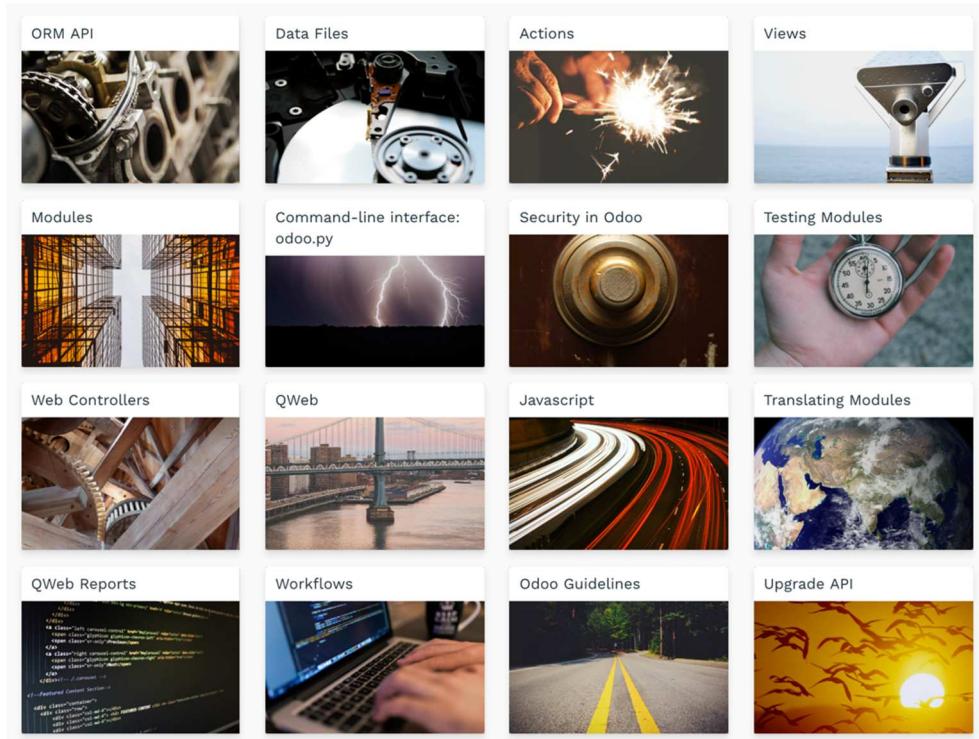


Figure 016 : Accès aux informations pour développeurs²⁶.

²⁵ <https://www.odoo.com/documentation/9.0/index.html>

²⁶ <https://www.odoo.com/documentation/9.0/index.html>

7.3.4. Interface de programmation

Le principal intérêt de ce travail est de pouvoir agir sur le PGI par programmation. Dans le but de pouvoir dérouler le scénario et appliquer automatiquement des changements dans les instances des équipes de participants.

Odoo donne accès à toute la documentation nécessaire pour prendre en main cette API²⁷.

Cette documentation permet aux développeurs de se connecter et de s'authentifier auprès de l'API pour effectuer les principales opérations de traitement des données.

7.3.4.1. Connexion et authentification

Cette étape est nécessaire et permet d'obtenir l'identifiant du compte utilisé pour requêter sur le modèle, lequel est obligatoire pour toutes les opérations où une authentification par mot de passe est nécessaire. Cette méthode s'exécute sur le point de terminaison **xmlrpc/2/common** et appelle la fonction RPC **execute** :

```

1  int uid = (int)client.execute(
2      common_config, "authenticate", asList(
3          db, username, password, emptyMap()));
4

```

Figure 017 : Code pour s'authentifier auprès d'une instance Odoo par l'API.

Cet appel retourne un **Integer** partant de 1 si l'authentification est validée et en cas d'erreur la valeur 0.

L'invocation des opérations qui suivent est effectuée à partir du point de terminaison appelé **xmlrpc/2/object** et de la fonction RPC nommée **execute_kw**. Chaque invocation nécessite les paramètres suivants :

- La base de données.
- L'identifiant de l'utilisateur, lequel est récupéré grâce à l'authentification.
- Le mot de passe de l'utilisateur employé.
- Le nom du modèle sur lequel l'opération doit s'exécutée.
- Le nom de la méthode. En principe une des opérations CRUD.
- Un **ArrayList** de paramètres passés par position.
- Optionnellement, un objet de type clé-valeur (**HashMap**).

²⁷ https://www.odoo.com/documentation/9.0/api_integration.html

7.3.4.2. *Création*

La création d'enregistrement s'effectue en appelant la méthode **create**, le code est le suivant :

```

1  final Integer id = (Integer)models.execute("execute_kw", asList(
2      db, uid, password,
3      "res.partner", "create",
4      asList(new HashMap() {{ put("name", "New Partner"); }}))
5  ));
6

```

Figure 018 : Code pour créer une enregistrement dans le modèle res.partner.

Cette méthode retourne l'identifiant au format **Integer** de l'enregistrement qu'elle vient d'insérer.

7.3.4.3. *Lecture*

Pour lire des enregistrements, Odoo propose plusieurs solutions²⁸. Dans le cadre de ce travail, la méthode **search_read** est régulièrement employée et c'est celle-ci qui est présentée ci-dessous :

```

1  // Requête avec critères, choix des champs retournés et limite de tuples
2  asList((Object[])models.execute("execute_kw", asList(
3      db, uid, password,
4      "res.partner", "search_read",
5      asList(asList(
6          asList("is_company", "=", true),
7          asList("customer", "=", true))),
8      new HashMap() {
9          put("fields", asList("name", "country_id", "comment"));
10         put("limit", 5);
11     })
12  )));
13  // Retourne
14  [[
15      {
16          "comment": false,
17          "country_id": [ 21, "Belgium" ],
18          "id": 7,
19          "name": "Agrolait"
20      },
21      {
22          "comment": false,
23          "country_id": [ 76, "France" ],
24          "id": 18,
25          "name": "Axelor"
26      },
27      {
28          "comment": false,
29          "country_id": [ 233, "United Kingdom" ],
30          "id": 12,
31          "name": "Bank Wealthy and sons"
32      },
33      {
34          "comment": false,
35          "country_id": [ 105, "India" ],
36          "id": 14,
37          "name": "Best Designers"
38      },
39      {
40          "comment": false,
41          "country_id": [ 76, "France" ],
42          "id": 17,
43          "name": "Camptocamp"
44      }
45  ]]

```

Figure 019 : Code pour lister un maximum de 5 enregistrements correspondants aux critères passés par l'ArrayList.

²⁸ https://www.odoo.com/documentation/9.0/api_integration.html

7.3.4.4. *Mise à jour*

Pour mettre à jour un enregistrement, la méthode **write** nécessite de connaître l’identifiant du ou des tuples ciblés. Le code ci-dessous permet de mettre à jour un élément distinct :

```

1  models.execute("execute_kw", asList(
2      db, uid, password,
3      "res.partner", "write",
4      asList(
5          asList(id),
6          new HashMap() {{ put("name", "Newer Partner"); }}
7      )
8  ));
9
10 // Récupération du nom après la mise à jour
11 asList((Object[])models.execute("execute_kw", asList(
12     db, uid, password,
13     "res.partner", "name_get",
14     asList(asList(id))
15 )));
16
17 [[78, "Newer partner"]]
18

```

Figure 020 : Code pour mettre à jour l’attribut name de l’enregistrement portant l’identifiant id plus code permettant de visualiser la mise à jour.

7.3.4.5. *Suppression*

Pour supprimer un enregistrement, la méthode **unlink** à besoin de l’identifiant (ou des identifiants) nécessaire à l’identification du tuple dans le modèle :

```

1  models.execute("execute_kw", asList(
2      db, uid, password,
3      "res.partner", "unlink",
4      asList(asList(id)));
5

```

Figure 021 : Code pour supprimer l’enregistrement portant l’identifiant id.

8. Business Process Modeling Notation (BPMN)

Business Process Model and Notation a été développée au départ par la Business Process Management Initiative (BPMI), et est maintenant maintenue par l'Object Management Group (OMG) depuis leur fusion en 2005. Depuis mars 2011, la version actuelle de BPMN est la 2.0. » (Briol, 2008).

Ce standard de notation permet une communication efficiente entre personnes aux métiers différents. C'est pourquoi, elle est utilisée pour représenter les processus proposés dans les différentes rubriques du document.

Son « but principal ... est de fournir une notation qui soit réellement compréhensible par tous les utilisateurs de l'entreprise, depuis les analystes métier qui créent les ébauches initiales des processus, jusqu'aux développeurs responsables de mettre en place la technologie qui va exécuter les processus applicatifs correspondants, et finalement, jusqu'aux utilisateurs de l'entreprise qui vont mettre en œuvre ces processus. Ainsi, BPMN crée un pont standardisé pour combler le vide entre la modélisation des processus métier et la mise en place des procédures. Actuellement, il y a des dizaines d'outils de modélisation et de méthodologies pour les procédures d'entreprise. » (Briol, 2008).

8.1. La palette d'éléments BPMN 2.0

BPMN et sa version 2.0 propose une palette assez large de composants servant à représenter la dynamique des organisations et leurs processus.

Ci-dessous l'intégralité des composants utilisables :

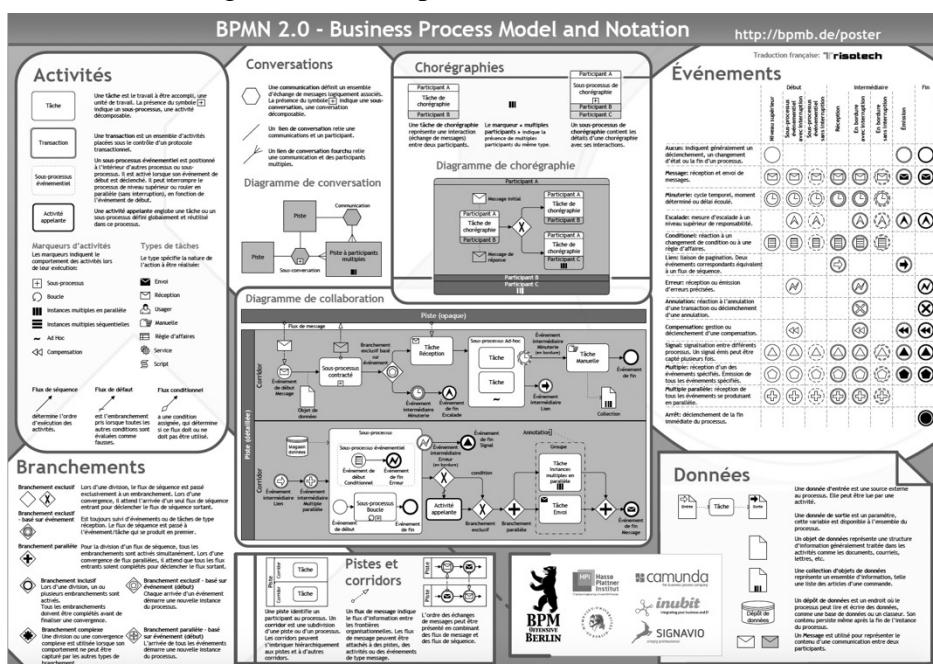


Figure 022 : Palette des éléments BPMN 2.0²⁹.

²⁹ http://www.bpmn.de/images/BPMN2_0_Poster_FR.pdf

9. Synthèse théorique

Tous les chapitres de cette partie servent à atteindre les résultats présentés dans la partie pratique.

Ce qui peut être affirmé est qu'OdooSIM est le composant simulateur de l'écosystème du jeu sérieux, c'est-à-dire, un artefact logiciel indépendant du PGI Odoo qui lui est la partie qui fait le lien entre les participants et le simulateur.

OdooSIM a pour mission de proposer un nouveau support de formation ludique qui prend la forme d'un jeu sérieux. Ce jeu sérieux se contextualise à l'aide d'un scénario qui doit être prenant et permettre de capter très fortement l'attention des participants.

Le scénario à développer doit correspondre au contexte d'apprentissage, c'est-à-dire la gestion d'entreprise. Ce scénario doit être joué par le simulateur, lequel doit être capable d'agir sur le PGI par programmation grâce au Web Service API qu'Odoo propose.

Partie Pratique

Table des matières

1.	PRÉAMBULE	1
2.	DÉMARCHE.....	2
3.	PROBLÉMATIQUES	3
3.1.1.	Scénario	3
3.1.2.	Complexité métier	3
3.1.3.	Apport pédagogique	4
3.1.4.	Accès au service	4
3.1.4.1.	Démarche	4
3.1.4.2.	Résultats	4
3.1.4.3.	Limites et recommandations	5
3.1.5.	Créer et configurer un accès pour les participants	6
3.1.5.1.	Démarche	6
3.1.5.2.	Résultats	6
3.1.5.3.	Limites et recommandations	6
3.1.6.	Dates virtuelles versus dates réelles	7
3.1.6.1.	Démarche	7
3.1.6.2.	Résultats	7
3.1.6.3.	Limites et recommandations	8
3.1.7.	Découverte des éléments métiers	9
3.1.8.	Sécurité	9
3.1.9.	Réalisme du marché.....	10
3.1.10.	Architecture du simulateur	10
4.	DÉVELOPPEMENT DU SCÉNARIO	11
4.1.	<i>Démarche et critères de développement</i>	11
4.1.1.	Définir un contexte accrocheur.....	12
4.1.2.	Définir la valeur ajoutée.....	12
4.1.3.	Définir le type d'affaire	13
4.1.4.	L'organisation logistique	13
4.1.5.	Définir les clients et leurs caractéristiques	13
4.1.6.	Définir les aspects financiers.....	14
4.1.7.	Définir des contraintes.....	14
4.2.	<i>Brewery & Co.....</i>	15
4.2.1.	Introduction	15
4.2.2.	Domaine brassicole.....	16
4.2.2.1.	Ingrédients.....	16
	Le malt d'orge	16
	Autres matières premières amylacées	17
	Le houblon.....	17
	L'eau.....	17
	La levure	18
4.2.2.2.	La fabrication de la bière	18
	Le brassage	18
	La fermentation.....	19
	La garde.....	19
	La filtration	19
	La mise en bouteille et le conditionnement.....	19
4.2.3.	Description de l'entreprise.....	20
4.2.3.1.	Les produits commercialisés.....	20
	Les nomenclatures	20
4.2.3.2.	Gestion des opérations de production	20
4.2.3.3.	La chaîne de valeur.....	21
4.2.3.4.	Support aux opérations	22
4.2.3.5.	Infrastructure	23

4.2.3.6.	Matériels de production	24
4.2.3.7.	Frais fixes mensuel	25
4.2.3.8.	Coûts de revient	25
Produit BW01	26	
Produit BW02	26	
Produit BW03	27	
Produit BW04	27	
4.2.3.9.	Exemple de rendement optimum.....	28
4.2.4.	L'environnement.....	28
4.2.4.1.	Le marché	28
4.2.4.2.	Les matières premières	29
4.2.4.3.	Les fournisseurs.....	29
4.2.4.4.	Les revendeurs	30
Les parts de marchés et les préférences	31	
Détaillant.....	31	
Supermarché.....	31	
Établissement public	32	
4.2.4.5.	Les bailleurs de fonds	32
4.2.4.6.	La fluctuation de la demande	33
4.2.5.	Les modalités de jeu.....	33
4.2.5.1.	Simulation du temps.....	33
4.2.5.2.	Évaluation de la performance	34
4.3.	<i>Limites et recommandations</i>	34
5.	PORTABILITÉ DES CONTRAINTES SCÉNARISTIQUES SUR ODOOSIM	35
5.1.	<i>Capacité de stockage local</i>	35
5.2.	<i>Marché de l'approvisionnement</i>	35
5.3.	<i>Délais</i>	36
5.4.	<i>Frais fixes</i>	37
5.5.	<i>Liquidités</i>	38
5.6.	<i>Récapitulation</i>	39
6.	DÉVELOPPEMENT DU PROOF OF CONCEPT.....	40
6.1.	<i>Problématiques</i>	40
6.2.	<i>Langage de programmation</i>	40
6.3.	<i>Fonctionnalités attendues</i>	41
6.4.	<i>Cycle opérationnel</i>	42
6.5.	<i>Exécution du simulateur</i>	43
6.5.1.	Fichier de configuration	44
6.5.2.	Phase de configuration	44
6.5.3.	Modèle objet.....	45
6.5.4.	Phase de simulation	48
6.5.4.1.	Gestion de la demande.....	48
6.5.4.2.	Gestion de l'offre.....	49
6.5.4.3.	Algorithme des ventes.....	49
6.5.4.4.	Mise en attente du Thread	49
6.5.5.	Phase de restitution	49
6.5.6.	Démonstration du résultat.....	49
6.6.	<i>Limites et améliorations</i>	50
6.6.1.	Pour la version finale	50
6.6.2.	Création des bases de données.....	50
6.6.3.	Gestion des droits d'accès utilisateurs.....	50
6.6.4.	Fichier de configuration	51
6.6.5.	Messages d'erreurs.....	51
6.6.6.	Game Loop	51
6.7.	<i>Réalité comptable</i>	52
6.7.1.	Constatation.....	52

6.7.2.	Démarche de résolution.....	52
6.7.3.	Solution.....	53
7.	<i>SPÉCIFICATIONS ODOOSIM</i>	54
7.1.	<i>Architecture</i>	54
7.2.	<i>Cycle opérationnel</i>	54
7.3.	<i>Pistes et recommandations pour les activités automatisées</i>	54
7.3.1.	Activité PL02	55
7.3.2.	Activités PR02, PR03 et PR04	55
7.3.3.	Activité MA02	59
7.3.4.	Limites et recommandations	60
8.	<i>ANALYSE DES PERFORMANCES</i>	61
8.1.	<i>Mesure temps des I/O</i>	61
8.1.1.	Alternative	62
8.2.	<i>Calcul théorique du temps d'exécution</i>	64
8.2.1.	Partie logistique	64
8.2.2.	Partie des ventes.....	65
8.2.3.	Vérification.....	66
8.3.	<i>Projection</i>	67
8.4.	<i>Limites et recommandations</i>	67
9.	<i>DÉVELOPPEMENT DE L'ALGORITHME DE CHOIX DE L'OFFRE</i>	68
9.1.	<i>Démarche</i>	68
9.2.	<i>L'algorithme de sélection de l'offre optimale</i>	68
9.3.	<i>Variation de la consommation</i>	69
9.4.	<i>Variation du prix</i>	70
9.5.	<i>Élasticité prix</i>	70
9.6.	<i>Calcul final de la demande adaptée au prix du marché</i>	71
9.7.	<i>Limites et recommandations</i>	71
10.	<i>ANALYSE DE L'ASPECT PÉDAGOGIQUE</i>	71
10.1.	<i>Démarche</i>	71
10.2.	<i>Critères de sélection</i>	72
10.3.	<i>Résultats</i>	72
10.4.	<i>Limites et recommandations</i>	75
11.	<i>ANALYSE DES OBJETS MANIPULÉS</i>	76
12.	<i>ANALYSE DE L'ASPECT SÉCURITÉ</i>	78
12.1.	<i>Démarche</i>	79
12.2.	<i>Application Contacts</i>	79
12.3.	<i>Application Inventory</i>	79
12.4.	<i>Application Purchases</i>	79
12.5.	<i>Application Manufacturing</i>	80
12.6.	<i>Application Sales</i>	80
12.7.	<i>Application Accounting</i>	80
12.8.	<i>Droit d'accès aux modèles et données</i>	81
12.9.	<i>Investigations</i>	82
12.10.	<i>Limites et recommandations</i>	84
13.	<i>ANALYSE DES PROCESSUS OPÉRATIONNELS</i>	85
13.1.	<i>Démarche</i>	85
13.2.	<i>Chaîne de valeur globale</i>	85
13.2.1.	Partie logistique.....	86
13.2.1.1.	La planification	86
13.2.1.2.	L'approvisionnement.....	86

13.2.1.3.	La production.....	86
13.2.2.	Partie des ventes	87
13.3.	<i>Limits et recommandations</i>	87
14.	PORTABILITÉ DU SCÉNARIO SUR ODOO.....	87
14.1.	<i>Démarche</i>	87
14.2.	<i>Configuration globale</i>	90
14.2.1.	Création de l'instance	90
14.2.2.	Installation des applications	91
14.2.3.	Master Data	92
14.3.	<i>Inventory Management</i>	94
14.3.1.	Configuration	94
14.3.1.1.	Valorisation des stocks	95
14.3.1.2.	Calcul des besoins.....	96
14.4.	<i>Purchase Management</i>	96
14.4.1.	Configuration	96
14.5.	<i>Manufacturing</i>	97
14.6.	<i>Sales Management</i>	97
14.6.1.	Configuration	97
14.7.	<i>Guide des opérations</i>	98
14.7.1.	Planification	99
14.7.1.1.	Effectuer les prévisions PL01	99
14.7.1.2.	Exécuter le calcul des besoins PL02.....	100
Autres indications.....		101
14.7.2.	Approvisionnement	101
14.7.2.1.	Confirmer les demandes d'achat PR01.....	101
14.7.2.2.	Réceptionner la marchandise PR02.....	103
14.7.2.3.	Créer la facture fournisseur PR03.....	107
14.7.2.4.	Comptabiliser le paiement fournisseur PR04	107
14.7.3.	Production	109
14.7.3.1.	Lancer des ordres de fabrication MA01	109
14.7.3.2.	Confirmer la production MA02.....	110
14.7.4.	Vente	111
14.7.4.1.	Changer les prix de vente SA05	111
14.7.4.2.	Créer une vente SA01	111
14.7.4.3.	Expédier la marchandise SA02	113
14.7.4.4.	Facturer le client SA03	115
14.7.4.5.	Comptabiliser le paiement client SA04.....	116
14.8.	<i>Limits et recommandations</i>	116
15.	CONCLUSION	117
15.1.	<i>Retour d'expérience</i>	117
15.2.	<i>Limits</i>	118
15.2.1.	Accès SaaS.....	118
15.2.2.	Web Service API.....	118
15.2.3.	Performances.....	118
15.3.	<i>Recommandations</i>	119
15.3.1.	Accès au service Odoo	119
15.3.2.	Performances.....	120
15.3.3.	Utilisation des workflows.....	121
15.3.4.	Gestion des dates virtuelles.....	124
15.4.	<i>Postface personnelle</i>	124
16.	BIBLIOGRAPHIE	126

1. Préambule

Cette partie du document présente les résultats obtenus au travers de différents chapitres.

Le premier chapitre parle d'une manière globale des problématiques rencontrées. Il redirige le lecteur dans les bonnes sections où elles sont détaillées et résolues.

Le chapitre suivant donne le détail du développement du scénario Brewery & Co. Il sensibilise le lecteur à la démarche et aux critères à respecter. Finalement, les limites et les recommandations sont données pour informer où le développement du scénario se situe.

Les contraintes créées au travers du scénario sont superposées au principaux processus que met en œuvre Brewery & Co. L'idée de ce chapitre est de donner une vue schématique de la situation de chacune des principales contraintes que doit implémenter le simulateur.

La quatrième partie est dédiée à la réalisation de la version limitée du simulateur, OdooSIM. Elle présente les problématiques soulevées, l'environnement choisi pour développer, le cycle opérationnel géré et finalement donne le concept global de l'exécution du simulateur. Comme pour chaque chapitre, des recommandations et les limites sont données. Une ultime partie est venue s'ajouter en fin de travail, celle de la problématique de réalité comptable. En effet, Deux petites semaines avant de rendre le travail, un problème a été constaté. Le processus de vente et de comptabilité n'était pas 100% intégrés. De ce fait, les écritures comptables censées représenter les transactions financières avec les clients n'étaient pas totalement comptabilisées et les comptes ne balançaient pas correctement. Toute la démarche de résolution est présentée car le problème a pu être résolu à temps.

En se basant sur l'expérience accumulée en développant la version proof of concept, les spécifications pour la version finale sont présentées.

Le développement du simulateur à soulever principalement une problématique, celle de la cohérence entre les dates virtuelles et celle simulées. Une solution est trouvée et consiste à récupérer chaque objet généré et mettre à jour les dates que les participants voient dans l'iHM. Le problème de cette solution, c'est qu'elle génère beaucoup de transactions avec l'API. Pour calculer le coût en terme de temps, le chapitre Analyse des performances est là pour cela.

Une des grandes valeurs apportées à la version proof of concept se situe au niveau de l'algorithme mis en œuvre pour la réalisation des ventes. En effet, dans cette version, les clients virtuels sont sensibles aux variations de prix. C'est ce qu'explique le chapitre Développement de l'algorithme de choix de l'offre.

Les quatre chapitres suivants sont dédiés à des analyses de diverses natures. L'aspect pédagogique est abordé, les modèles employés, l'aspect sécurité et l'analyses des processus opérationnels de Brewery & Co.

Le chapitre Portabilité du scénario sur Odoo est celui qui a permis de débuter le travail d'investigation sur le fonctionnement à reproduire dans le simulateur. Il est à la base de toute la démarche.

Pour conclure, la dernière partie éclaire le lecteur sur l'état du travail, les recommandations, les limites et la conclusion personnelle de l'auteur.

2. Démarche

La démarche employée pour atteindre les objectifs et les résultats présentés dans cette partie du document s'effectue de manière parallèle dans un premier temps, puis séquentiellement.

Dès le début, la problématique métier et technique sont très vite abordées en même temps. Ceci, afin de faire la plus grande avance possible dans des délais assez courts.

Le métier touche principalement à la prise de connaissance des processus d'une société de production et au développement d'un scénario de jeu, le plus réel possible. À ce moment-là, la partie technique correspond à la prise en main de l'API, lecture de la documentation et autoformation.

Dès que le scénario est sous toit, l'étape qui suit, est celle de configuration du PGI. Cette configuration doit être faite pour correspondre aux besoins de Brewery & Co.

Toutes les étapes sont référencées et analysées grâce à l'activation du mode développeur sur l'IHM. Grâce à cela, la structure technique d'Odoo est découverte et de bonnes pistes sont identifiées pour l'étape de conception du simulateur.

Une fois que cela est fait, une partie d'analyse pédagogique débute. Elle permet d'analyser chacune des activités que Brewery & Co doit réaliser pour atteindre son but, celui de vendre des produits. Cette analyse donne comme résultat les activités qui doivent exécuter par les participants et celles qui doivent être exécutées par le simulateur.

Une fois que le cadre général est posé et que l'on sait qu'est-ce que doit faire le simulateur, le développement de la version proof of concept débute.

3. Problématiques

Ce chapitre expose les principales problématiques auxquelles ce travail se confronte. Elles sont présentées d'une manière générale puis détaillées au travers des différents chapitres qui composent cette partie pratique.

3.1.1. Scénario

Cette problématique touche l'élément central du jeu sérieux OdooSIM. Son scénario doit être pensé et développé de sorte à ce qu'il puisse être le facteur premier dans l'engouement des participants pour la solution.

La difficulté est double. Elle réside dans la manière qui est utilisée de mettre en scène une société de production, ses activités internes et la façon qu'elle a de traiter avec son environnement.



Complément d'informations :

Les résultats obtenus sont détaillés dans le chapitre Développement du scénario.

3.1.2. Complexité métier

La modélisation des processus mis en œuvre par la société Brewery & Co n'est pas chose aisée. En effet, elle demande de la part de l'analyste des compétences métiers et dans une certaine mesure un bagage pratique.

Ce travail consistant à poser les fondations nécessaires au développement d'un jeu sérieux pour l'apprentissage de la gestion d'entreprise à l'aide d'un PGI, le réalisme doit être de mise pour qu'il soit utile. Par conséquent, des efforts ont été consentis dans la compréhension des processus internes aux entreprises de production.



Complément d'informations :

Les résultats obtenus sont détaillés dans le chapitre Analyse des processus opérationnels.

3.1.3. Apport pédagogique

Un des aspects majeurs de ce travail, se situe dans sa capacité à apporter de nouvelles compétences aux participants. De ce fait, il est primordial de bien choisir quelles opérations doivent être menées par les participants. La problématique est de définir des critères pertinents permettant d'effectuer ces choix. Une fois définis, il faut analyser toutes les activités de la chaîne de valeur et les confronter aux critères pour décider si elles apportent de la valeur à l'apprentissage.

Complément d'informations :



Les résultats obtenus sont détaillés dans le chapitre Analyse de l'aspect pédagogique.

3.1.4. Accès au service

Dès que le PGI et sa version sont choisis, il faut définir une méthode d'accès. Le chapitre Accès au service mentionne, plusieurs possibilités pour y parvenir. On-premises, ASP ou encore SaaS sont autant de solutions existantes et de pistes à analyser pour trouver la bonne solution dans le contexte de ce travail.

3.1.4.1. Démarche

Pour effectuer un choix, l'élément prépondérant est le temps à investir pour être opérationnel. Ce qui ne doit pas forcément être le cas pour le développement final. Le travail portant sur une dizaine de semaines, il faut pouvoir débuter l'analyse le plus rapidement possible. La décision se porte à la fois sur la théorie relative aux modes d'accès et sur la documentation qu'Odoo met à disposition sur son site web.

3.1.4.2. Résultats

Tout d'abord, Odoo propose différentes versions de son PGI. Entreprise, Community ou encore Online. Le résultat de ce premier choix se tourne vers la version en ligne. L'accès à cette version est étroitement lié à son nom, Online et par conséquent en mode SaaS.

Ce choix est fait pour éviter de consentir des efforts dans la mise en place du PGI en interne. En utilisant la version Online SaaS, le travail peut s'affranchir de toutes les étapes nécessaires à la mise en place d'un environnement d'hébergement, de l'installation et de la configuration de la solution.

De plus, le programme éducatif qu'Odoo propose avec la version Online correspond exactement aux besoins de ce travail.

3.1.4.3. *Limites et recommandations*

La version Entreprise n'a pas été investiguée. Cependant, elle est semblable à la version Online en termes de fonctionnalités, c'est-à-dire full-options. La différence réside dans le mode d'accès. Cette version est accessible uniquement en s'appropriant le logiciel. Ensuite, deux solutions existent pour sa mise en place, On-premises ou ASP. Entre autre, il semble important de mentionner qu'elle est payante et de ce fait, ne correspond pas à l'objectif de diminution des coûts qu'OdooSIM doit atteindre.



Rappel :

Un des objectifs d'OdooSIM est d'offrir un avantage financier par rapport à la solution ERPSim.

Pour la version Community, elle est utilisée pour quelques tests sur un serveur Synology privé. Ses fonctionnalités sont moins importantes mais peuvent largement faire l'affaire pour les besoins d'OdooSIM. Elle supporte les principaux processus utilisés par Brewery & Co. La grande différence se situe au niveau de l'interface graphique qui est bien moins soignée et séduisante que celle de la version Online.

Pour la suite à donner au projet OdooSIM, il est recommandé d'analyser plus en profondeur cette version Community. En effet, elle peut être une solution envisageable dans la mesure où, en l'utilisant, toute la partie de création et configuration des bases de données peut être faite par l'API.

L'API est aussi disponible dans cette version et permet de manipuler un nombre plus élevé d'éléments que ce qu'il est possible de faire sur les serveurs publics odoo.com. Le blocage des fonctionnalités de gestion des bases de données est une position prise par Odoo S.A. Elle est tout à fait compréhensible. Il est bien trop dangereux pour Odoo S.A. de donner accès à ces fonctions sur leurs serveurs publics où des environnements de production sont hébergés. Quiconque mal intentionné pourrait essayer de provoquer un déni de service en bombardant les serveurs odoo.com de requêtes gourmandes en ressources telles que celles de création de base de données.

3.1.5. Créer et configurer un accès pour les participants

Lorsqu'une nouvelle partie de jeu débute, une instance Odoo doit être mise à disposition des participants.

3.1.5.1. Démarche

Pour résoudre cette problématique, plusieurs pistes sont empruntées. Créer et configurer les bases de données manuellement. Dans le but d'identifier les principales étapes et les éléments impactés, puis par programmation en requêtant l'API.

3.1.5.2. Résultats

Premièrement plusieurs choix d'implémentation peuvent exister. Tout faire dans une seule base de données comme c'est le cas dans ERPSim ou de créer autant de base de données qu'il y a d'équipe. Dans la première solution, il faut mettre en œuvre un moyen de différenciation des objets entre les diverses équipes. Dans ERPSim, ce mécanisme se concrétise en préfixant les objets avec les initiales des groupes. Exemple :

Ordre d'achat de l'équipe A = AA Order d'achat.

Cette solution est difficile à mettre en œuvre d'un point de vue configuration. Dans Odoo, la possibilité de configurer le PGI en mode multi sociétés est une piste à envisager pour fonctionner ainsi.

Le choix fait dans le cadre de ce travail est d'utiliser une base de données par équipe. Elle a pour avantage de rendre la configuration aisée et le programme éducatif n'impose aucune restriction dans le nombre d'instance créée.

3.1.5.3. Limites et recommandations

Les échanges sont multipliés par le nombre d'équipe. Cela signifie que pour récupérer des demandes d'achats, il faut se connecter et exécuter la requête autant de fois qu'il y a d'équipe en jeu. Cette manière de procéder soulève une problématique secondaire, les performances.

Le cycle des opérations à mener par le simulateur est cadencé par le temps de simulation d'un jour. Il peut devenir très vite difficile de tout exécuter dans cette fenêtre de temps réduite (60 secondes dans le cadre de ce travail).

D'ailleurs, dans la version proof of concept du simulateur, cette fenêtre de temps ne suffit plus dès qu'il y a plus de trois sociétés en jeu. De plus, ce n'est que la partie des ventes qui est gérée.

3.1.6. Dates virtuelles versus dates réelles

En effectuant à la main les opérations du chapitre Guide des opérations, une problématique de taille s'est ajoutée au travail, celle de la cohérence des dates dans le scénario.

En effet, lorsqu'une partie est jouée, plusieurs jours se simulent en quelques minutes à peine (60 secondes = 1 jour). La conséquence est que toutes les transactions qui s'effectuent dans le PGI sont enregistrées en fonction de la date réelle, c'est-à-dire la date du jour à laquelle la session de jeu est dispensée.

La problématique se situe au niveau de l'interprétation faites par les participants en visualisant les données du PGI. Sans solution à ce problème, les participants voient tous les éléments créés dans Odoo avec la même date, cela n'est tout simplement pas envisageable. Les raisons sont simples, comment est-ce que les participants peuvent tirer des enseignements si aucune analyse pertinente ne peut être faite. La réponse est simplement d'aucune manière.

L'exemple qui représente le mieux ce problème est celui associé aux ventes. Si les dates ne sont pas modifiées, cela veut dire que durant une simulation où les participants changent de prix durant des jours simulés bien précis, il n'est simplement pas possible de visualiser l'impact que ces derniers ont provoqué. Effectivement, les ventes sont enregistrées toutes au même jour et par conséquent, les participants ne peuvent pas trouver d'éléments de réponse à des réflexions de ce type-là :

« Nous avons modifié le prix durant le jour 4, allons vérifier la différence du volume de vente générée entre le jour 4 et le jour 5 »

Ils trouvent uniquement le volume de vente agrégé sur la date du jour réelle. Cela n'a par conséquent aucun sens et aucune valeur dans une démarche pédagogique.

Il devient nécessaire de trouver comment modifier les dates et surtout sur quelle base temporelle s'appuyer pour que le scénario puisse apporter un effet réaliste. Le réalisme qui est l'un des critères qualitatifs les plus importants dans l'évaluation d'un scénario de jeux sérieux et dans la motivation qu'il génère envers les participants.

3.1.6.1. Démarche

Sur la base du référentiel créé (référence au chapitre Analyse des objets manipulés) reprendre toutes les dates qui apportent du sens au scénario et les modifier pour qu'elles soient concordantes avec le jour simulé.

3.1.6.2. Résultats

Après plusieurs tentatives infructueuses, une solution semble avoir été trouvée. Elle consiste à se baser sur une base temporelle fixe, c'est-à-dire qu'à chaque nouvelle simulation, le jeu débute au jour du 01 janvier de l'année réelle. Au fur et à mesure que les jours avancent, les dates se succèdent à partir de cette date référence.

En prenant en considération qu'une session de jeu est constituée de 4 rounds de 20 jours chacun, les éléments du PGI se créent tous entre les dates du 01 janvier au 20 janvier pour le round 1. Du 01 février au 20 février pour le round 2. Du 01 mars au 20 mars pour le round 3 et finalement du 01 au 20 avril pour le round 4.

La seule limite à cette solution consiste à ne pas définir des rounds de plus de 28 jours. Si cela se produit, alors arrivé au deuxième round, les données du PGI ne peuvent être modifiées car le 29, 30 ou encore 31 février n'existe pas dans les calendriers. L'opération visant à modifier l'élément n'est pas prise en compte par Odoo et les dates ne sont pas mises à jour. Si cela se produit alors le scénario et les données du PGI ne sont plus concordants. On retombe dans un problème de cohérence.

Cette solution a été implémentée dans la version proof of concept du simulateur et semble donner pleine satisfaction. De ce fait, toutes les recommandations et les hypothèses liées aux autres responsabilités du simulateur se base sur elle.

3.1.6.3. *Limites et recommandations*

Certains modèles n'autorisent pas la mise à jour de leurs attributs **datetime**. Odoo donne la possibilité de modifier ces restrictions en se rendant dans l'application **Settings / Technical / Database Structure / Models** et en ciblant l'objet pour lequel on veut retirer la restriction nommée **Read Only**.

Cependant, via l'interface d'administration graphique, dès que l'on retire la restriction et qu'on applique la modification, Odoo répond ceci :

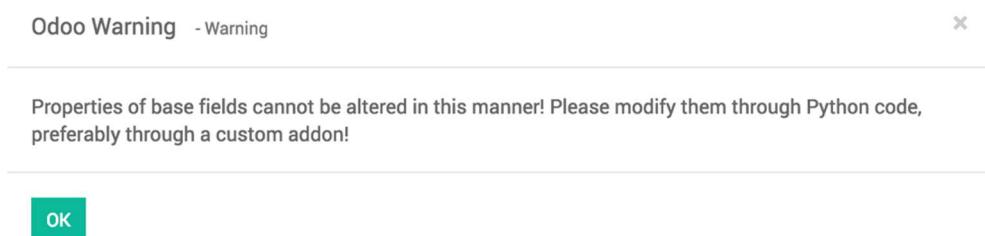


Figure 023 : Message d'avertissement.

Par conséquent, le travail à effectuer pour contourner ces restrictions nécessite de se rendre dans le code Python pour tous les champs bloqués de la sorte.

Autre limite constatée, celle du nombre de transactions supplémentaires qu'il est nécessaire de gérer pour mettre à jour les dates réelles en dates simulées. En optant pour cette solution, il faut être conscient qu'il est inévitable d'avoir à générer ce grand nombre de requêtes, lesquelles ralentissent fortement les performances du simulateur.

Une solution en lien avec la recommandation d'accès au service grâce à un serveur privé (se référer au chapitre Problématiques) peut permettre de contourner ce problème. Cette solution consiste à déployer sur le serveur qui héberge la solution Odoo Community, une routine Python qui lorsque le simulateur se lance, s'exécute parallèlement à la même cadence que les jours

sont simulés. Il est même envisageable d'appeler cette routine à partir du simulateur si elle expose une méthode invocable depuis l'extérieur. Son travail est de mettre à jour l'horloge du serveur Odoo en modifiant la date. Bien entendu ceci n'est qu'une piste hypothétique. Il s'agit de la vérifier.

3.1.7. Découverte des éléments métiers

Le scénario tel qu'il est défini dans le chapitre Brewery & Co. impose la mise en œuvre de processus bien distincts et par conséquent de savoir manipuler les éléments métiers associés. La problématique sous-jacente est d'identifier tous ces éléments et de constituer un référentiel qui doit être utilisé pour la partie développement du simulateur.

En effet, pour chaque activité à mettre en œuvre pour réaliser la chaîne de valeur Brewery & Co, il liste les objets manipulés et générés par Odoo. Le but est de pouvoir les mettre à jour pour qu'ils soient cohérents avec le temps virtuel.



Complément d'informations :

Les résultats obtenus sont détaillés dans le chapitre Analyse des objets manipulés.

3.1.8. Sécurité

Dans tous projets de développement, une partie est liée à l'aspect sécuritaire. OdooSIM n'y échappe pas. En effet, il est nécessaire de restreindre les possibilités fonctionnelles aux participants pour éviter qu'ils ne modifient les données du scénario ou qu'ils soient tentés de tricher.

Certains modèles sont accessibles à partir de plusieurs applications dans Odoo. Par exemple, cela est le cas pour les factures **account.invoice** qui peuvent être manipulées à partir de l'application **Accounting**, **Purchases** et **Sales**. Mais aussi les transferts de stock **stock.picking** qui peuvent être manipulés à partir de l'application **Inventory** mais aussi par le biais de **Purchases** ou **Sales**.

De ces constatations, deux éléments sont à sécuriser. L'accès global aux applications et à un niveau plus fin, les droits sur les modèles.



Complément d'informations :

Les résultats obtenus sont détaillés dans le chapitre Analyse de l'aspect sécurité.

3.1.9. Réalisme du marché

Le jeu donne aux participants la possibilité de s'entraîner à la gestion d'entreprise au travers de différents rôles managériaux clés. Dans le cadre de ce travail, le rôle de chef des ventes est simulé. En effet, la partie vente est choisie pour être mise en œuvre dans le proof of concept du simulateur.

Il convient de simuler un marché se rapprochant de la réalité. Par conséquent, le mécanisme d'élasticité prix-demande est implémenté. Effectivement, les tâches qu'incombent aux participants dans cette version du simulateur se limitent à la mise à jour des prix de vente. Cette manipulation a un impact sur les quantités demandées par les consommateurs.

Un algorithme de sélection de la meilleure offre sur le marché est mis en place et dès qu'il a identifié la meilleure offre, il adapte la quantité demandée par rapport à la variation du prix observé (variation qui se calcule avec un prix optimal contenu dans le fichier de configuration du simulateur).



Complément d'informations :

Les résultats obtenus sont détaillés dans le chapitre Développement de l'algorithme de choix de l'offre.

3.1.10. Architecture du simulateur

Le simulateur doit avoir une très bonne rapidité d'exécution. Il exécute une multitude d'opérations chaque jour qu'il simule. Ces opérations sont principalement de faire évoluer le contenu de tous les PGI en jeu.

En se référant au temps de 60 secondes nécessaires à la simulation d'un jour dans le jeu, cela implique que le simulateur puisse exécuter toutes ses opérations en 59 secondes. Cette responsabilité est problématique du fait que les échanges avec l'API sont assez coûteux en termes de temps d'exécution. De plus ce temps se multiplie autant de fois qu'il y a d'équipe en jeu. Par conséquent, il est nécessaire de trouver une solution architecturale pour limiter les échanges.



Complément d'informations :

Les résultats obtenus sont détaillés dans le chapitre Développement du proof of concept.

4. Développement du scénario

Le développement d'un jeu sérieux doit s'appuyer sur une histoire. Cette histoire met en relations des étapes pour former un scénario. La qualité de ce dernier influe sur le niveau d'intérêt des participants. L'immersion du participant étant directement impactée par cette composante qualitative.

Dans le cadre de ce travail, il faut tout d'abord proposer un nouveau scénario de jeu afin d'accompagner les participants dans leur progression.

Ce scénario est choisi en adéquation avec le thème que l'on veut faire assimiler aux participants. Pour se faire, il faut trouver des idées cohérentes avec la thématique afin de mettre en bonne situation les participants.

Complément d'informations :



La thématique de ce travail est de développer un nouveau scénario de jeu qui met en œuvre la gestion d'une entreprise de type PME à l'aide d'un progiciel de gestion intégré.

Tous les éléments du scénario doivent donner du sens pour atteindre l'objectif pédagogique recherché. De plus, les participants doivent toujours être tenus en haleine par l'attractivité du scénario.

Par exemple, il est malvenu de proposer un scénario qui met en œuvre une multinationale avec un montage financier hors du commun pour sensibiliser le participant à la gestion d'une petite ou moyenne entreprise.

La suite de ce chapitre se présente sous forme de deux parties. Premièrement, les critères de validité pour le scénario choisi. Deuxièmement, la présentation du scénario Brewery & Co développé dans le cadre de ce travail.

Complément d'informations :



Les critères utilisés sont propres à la thématique et ne doivent pas être repris tels quels dans le cadre d'un développement de jeu sérieux autre que celui abordant la thématique de la gestion d'entreprise

4.1. Démarche et critères de développement

Afin d'être dans un cadre réaliste, il se doit de comprendre et d'intégrer les différents éléments inhérents à la thématique. Ce chapitre présente la démarche de création d'un scénario et les éléments qui doivent être pris en considération dans le cadre du développement d'un scénario aligné à la gestion d'entreprise.

Pour obtenir les résultats espérés, il faut se baser sur les scénarios du modèle ERPSim. Principalement le scénario Muesli qui met en scène une société de production comme c'est le cas pour Brewery & Co. Parallèlement, la littérature évoquant la gestion d'entreprise est d'une grande aide. Pour les

besoins du travail l'ouvrage **Introduction à la gestion d'entreprise par Jean-Paul Thommen** fait office de référence.

Les chapitres suivants doivent être compris comme étant les critères minimaux à mettre en œuvre dans la réalisation d'un nouveau scénario.

4.1.1. Définir un contexte accrocheur

Pour garder à l'esprit que le scénario doit susciter l'intérêt des participants, il est nécessaire de bien choisir le contexte dans lequel ils doivent se plonger.

Pour se faire, il faut définir clairement le public ciblé par le jeu. Dans ce cas, ce sont principalement des étudiants d'une vingtaine d'années.

La question qui se pose ensuite est :

« Qu'est-ce qui peut motiver un étudiant d'une vingtaine d'année dans un cadre qui reste sérieux et dans lequel il peut tirer des leçons ? ». La réponse apportée se situe dans le domaine brassicole. L'hypothèse est qu'il puisse s'agir d'un contexte motivant pour les participants.

Une fois que le contexte général est mis en place, il faut imaginer des éléments. Par exemple, il est recommandé d'enrôler les participants dans des fonctions au sein de l'entreprise. Par exemple : Il endosse la responsabilité de manager des ventes pour une certaine région. Par conséquent, le participant se sent dans un rôle différent que celui qui est le sien au quotidien. Sa motivation résultante est augmentée.

Suite à cela, les opérations à réaliser doivent être organisées dans un environnement. Des marchés différents, des spécificités différentes, des intérêts différents ou encore du pouvoir d'achat différent. Cela rend le scénario plus réaliste car toutes ces composantes font que les participants ne sont pas seulement dans un marché unique.

4.1.2. Définir la valeur ajoutée

Toutes sociétés apportent une valeur ajoutée aux yeux des clients qu'elles ciblent. Le scénario doit définir clairement ce que la société prévoit de fournir aux consommateurs. Cette valeur ajoutée peut être de deux natures différentes, tangible ou intangible. Dans la catégorie des biens tangibles, les produits sont palpables. Dans l'autre, ils représentent des services.

Le choix du type de bien que la société réalise impacte de manière considérable la complexité de la chaîne de valeur. Alors qu'un bien de type service la rend plus concise, son contraire la rend plus longue.

Une fois que le choix est opéré, il faut contextualiser les biens. Exemple :

- Vendre un service de sauvegarde de données.
- Vendre un service d'entretien de véhicule.
- Vendre un produit comestible.
- Vendre un produit de type composant informatique.

4.1.3. Définir le type d'affaire

Quel genre d'opérations l'entreprise effectue ? C'est-à-dire, que fait-elle pour produire ses articles ? Elle les transforme ou elle fait uniquement de la distribution ?

Ces deux manières de travailler ont des répercussions sur la chaîne de valeur. Alors qu'une société qui achète des matières premières et les transforme en produits finis possède une chaîne de valeur :

Approvisionnement – Production – Vente

Celle qui fait de la distribution, s'abstient du processus de production.

Dans le cas où la société de production est gardée, il faut ensuite définir les produits et les matières premières.

Une fois que cela est mis en place, les nomenclatures doivent permettre de lier le tout.

4.1.4. L'organisation logistique

La mise en place d'une stratégie logistique doit être prévue. Soit la société travaille en flux poussé ou alors en flux tiré. Pour effectuer ce choix, il est nécessaire de prendre en considération le marché que nous voulons pénétrer. Si le marché à un volume conséquent de demandes et quelles peuvent être anticipées alors la stratégie en flux poussé est à privilégier. Elle a pour principale vocation « d'engager de l'argent (achat de matières premières, paiement des salariés, utilisation des moyens de production...) sans être réellement sûre à 100% qu'elle pourra vendre le fruit de sa production (et donc par conséquent être payé). »³⁰.

À l'opposé, si le produit à une durée de vie plutôt courte (au sens technologique par exemple) et que le marché est difficile à prévoir alors la société articule sa production en flux tiré. Ce qui a pour effet de ne pas avoir à gérer de stock ou très peu. « Dans cette organisation, c'est la demande d'un client (client externe : consommateur / client interne : un autre service de l'entreprise) ou une consommation qui sera l'élément déclencheur d'une mise en fabrication d'un produit. »³¹

4.1.5. Définir les clients et leurs caractéristiques

Afin de mettre en œuvre l'activité génératrice de revenu d'une société, il faut décider de quelle manière les biens sont distribués.

Par conséquent, il faut imaginer des clients et leur attribuer des caractéristiques. Les hypothèses sont que les caractéristiques doivent être au minimum les suivantes :

- Préférences pour certains produits.
- Pouvoir d'achat et sensibilité aux changements de prix.
- Besoins quantitatifs.

³⁰ <http://logistique-pour-tous.fr/flux-pousse-ou-flux-tire/>

³¹ <http://logistique-pour-tous.fr/flux-pousse-ou-flux-tire/>

- Conditions de paiement en termes de jours.

Grâce à ces caractéristiques le jeu peut sensibiliser les participants à la segmentation des marchés. Elles permettent aussi de définir des stratégies de pénétration de marché qui peuvent différées entre les équipes de participants. Ceci apporte une réelle valeur pédagogique au scénario.

4.1.6. Définir les aspects financiers

Une société effectue des transactions financières quotidiennement. Cette partie de la démarche doit permettre de définir la structure du bilan.

Entre outre, ses coûts d'achats, de production et de distribution sont à prendre en considération afin d'attribuer des coûts de revient aux produits.

La difficulté réside dans la cohérence des éléments entre eux. Des chiffres donnés aux hasards n'ont pas leurs places dans cette démarche.

4.1.7. Définir des contraintes

Pour ajouter de la complexité au scénario, il faut analyser l'environnement réel et essayer d'en identifier les contraintes qui y sont associées.

Ces contraintes ont l'effet de provoquer des erreurs auprès des participants. Par conséquent, ils doivent en tirer des leçons en analysant leurs erreurs. C'est grâce à cela que les bénéfices des jeux sérieux sont mesurables.

Pour se faire, il est recommandé de prendre la chaîne de valeur et d'essayer d'y superposer des contraintes, de volume, de temps ou encore financières.

Le but étant de provoquer des erreurs auprès des participants pour qu'ils puissent en tirer des apprentissages.

4.2. Brewery & Co.

Ce chapitre présente le scénario retenu pour ce nouveau jeu sérieux. Il détaille l'intégralité des éléments qui le constituent.

Cette partie donne une vision claire et précise de ce qu'il faut prendre en considération pour démarrer en tant que participant. Elle est complémentaire à l'annexe Votre Job chez Brewery & Co.

4.2.1. Introduction

En tant que jeune étudiant diplômé de la Haute Ecole de Gestion Arc à Neuchâtel, vous venez de postuler pour un poste de management dans la société Brewery & Co. à Berne en Suisse. Cette société est active dans la vente de plusieurs variétés de bières artisanales.

Le marché de la bière est très lucratif et les opportunités de vente sont très présentes. Pour appuyer ces dires, les chiffres de l'année 2015 parlent d'eux-mêmes : soit 3'438'047 hectolitres ont été produits et 4'623'798 hectolitres ont été bus par la population Suisse (Association suisse des brasseries, 2016).

La bière se compose principalement d'eau à plus de 90%. Ensuite, l'adjonction de malt d'orge ou d'autres matières semblables comme le froment, le seigle, le maïs, le riz ou l'avoine est nécessaire. Tous ces malts ont comme principale caractéristique de contenir de l'amidon. Un autre composant très important est le houblon. « Le houblon confère à la bière son agréable amertume et, selon le type de bière, un arôme houblonné. Il assure en outre son aptitude à la garde, ainsi que la tenue de la mousse. ». Finalement, la levure « Pour que le moût entre en fermentation, on lui adjoint de la levure, qui influe de manière déterminante sur le caractère et la saveur de la bière. Les levures sont des micro-organismes présents presque partout dans l'air ; ils se répartissent en différentes souches. Les brasseries utilisent des levures de culture pures, soit des souches tout à fait identiques de levures, pour pouvoir garantir la qualité régulière du produit final. On distingue généralement deux types principaux de levures : celles à fermentation basse, et celles à fermentation haute. » (Association suisse des brasseries, 2016).

Vous êtes engagé dans l'une des 12 brasseries du pays³² et vous intégrez une équipe qui s'occupe de la gestion des opérations. Le pouvoir qui vous est conféré, vous permet de prendre des décisions stratégiques afin d'amener la société à un certain niveau de prospérité. Principalement, vous êtes amené à décider quotidiennement les éléments tels que la gamme de produit que vous commercialisez, les marchés que vous pénétrez, la prévision des ventes, la planification de votre production et la politique de prix que vous pratiquez envers vos revendeurs. Tout cela en pratiquant régulièrement des analyses de vos performances et en effectuant des analyses de marché. Vous devez aussi veiller à respecter des règles d'hygiènes très strictes et des bases légales édictées par la Confédération.

³² Donnée fictive pour le scénario

4.2.2. Domaine brassicole

Le scénario retenu pour créer le jeu sérieux est celui relatif à l'industrie de la bière. Ce chapitre présente les principaux éléments du domaine à connaître afin de pouvoir s'en imprégner et en comprendre les principaux éléments.

4.2.2.1. Ingrédients

Les chapitres qui suivent présentent les matières premières utilisées dans la fabrication de la bière.

Le malt d'orge

Dans les cultures anciennes, le produit de base utilisé pour la fabrication de la bière était très probablement le pain trempé. Des fouilles archéologiques réalisées au Proche-Orient ont permis de découvrir des iconographies illustrant la préparation de la bière : le vannage des grains de blé, la conversion du blé en pain, le ramollissement dans l'eau et le processus de fermentation. De nos jours, le produit de base de la bière n'est plus la soupe de pain, mais essentiellement le malt d'orge.

Couramment, le malt d'orge est le plus souvent désigné tout simplement par « malt ». Fabriqué à partir d'une sorte particulière d'orge, dite orge de brasserie, il est un issu de céréales successivement trempées, germées, puis séchées. Le grain est constitué essentiellement d'amidon, ainsi que de protéines et de sels minéraux. Mais ces substances ne se laissent pas extraire sans autres, car l'amidon contenu dans le grain d'orge est emprisonné par des parois cellulaires non solubles dans l'eau.

Pour servir à la préparation de la bière, le malt d'orge doit subir des opérations en malterie : d'abord le trempage, puis la germination. Comme le germe se nourrit de l'amidon contenu dans le grain, des enzymes se forment pour dégrader les parois des cellules et rendre ainsi l'amidon soluble dans l'eau, ce qui permettra par la suite de libérer du grain les granules d'amidon et de le saccharifier.

Après environ six jours, le processus de germination est suffisamment avancé : les céréales en germination (malt vert) subissent un processus de séchage appelé « touraillage ». Le touraillage, plus ou moins prononcé selon les besoins, permet de conserver la matière première. Un séchage modéré donne du malt blond, tandis qu'un touraillage intensif (on parle aussi de torréfaction) donne un malt destiné à une bière brune. A ce stade, le malt de brasserie est débarrassé des radicelles, nettoyé et poli. Il est ensuite stocké en silo jusqu'à la livraison à la brasserie.

Ainsi, les bases de la couleur et du goût de la bière sont déjà jetées à la malterie, en faisant varier les paramètres que sont le temps, l'humidité, la température et la ventilation. Le malt peut avoir un goût qui va du sucré au corsé, et son spectre chromatique s'étend du beige clair au noir, en passant par le brun. Il n'y a ni culture d'orge de brasserie à grande échelle ni malterie en Suisse. Aussi les brasseurs suisses s'approvisionnent-ils en malt

principalement en France, en Allemagne et en République tchèque (Association suisse des brasseries, 2016c).

Autres matières premières amylacées

Outre le malt d'orge, d'autres matières premières contenant de l'amidon peuvent entrer dans la fabrication de la bière, certaines étant maltées, d'autres non.

Il s'agit principalement de céréales comme le froment, le seigle, le maïs, le riz, l'avoine, l'épeautre ou l'amidonnier, mais aussi d'ingrédients comme les châtaignes ou les dattes (Association suisse des brasseries, 2016c).

Le houblon

Comment l'heureuse idée d'associer le houblon au malt pour le brassage de la bière est-elle venue à l'esprit des Anciens ? La question n'est pas tranchée. Les premières preuves historiques de l'utilisation du houblon à des fins brassicoles datent du début du Moyen Age. Dans la pharmacopée populaire, le houblon a toujours joué un rôle important, et il entre encore aujourd'hui dans la fabrication de certains produits pharmaceutiques.

Le houblon est une plante grimpante qui s'agrippe à des fils de fer allant jusqu'à 8 m de haut. Le brasseur utilise les fleurs non fécondées de la plante femelle, appelées « cônes », qui contiennent de nobles substances aromatiques, résines amères et tannins. Le houblon confère à la bière son agréable amertume et, selon le type de bière, un arôme houblonné. Il assure en outre son aptitude à la garde, ainsi que la tenue de la mousse.

La récolte du houblon a lieu à la fin de l'été. Le houblon en cône doit être conservé au frais, car les substances amères et aromatiques se dégradent sous l'effet de la chaleur et de l'oxygène.

Afin de mieux préserver ses qualités et pour des raisons pratiques, le houblon en cône est aujourd'hui souvent pressé en petits granulés cylindriques, les « pellets », ou raffiné pour obtenir un extrait, les deux formes étant utilisées telles quelles pour la fabrication de la bière.

En Suisse, le houblon est cultivé dans la région de Stammheim (ZH), dans le Fricktal (AG) et dans l'Aaregäu (SO). Les importations proviennent essentiellement d'Allemagne, l'un des principaux producteurs de houblon. La plus grande culture d'un seul tenant au monde se trouve dans la région de l'Hallertau, en Bavière (Association suisse des brasseries, 2016c).

L'eau

Avec une part supérieure à 90 %, l'eau est la composante principale de la bière ; elle revêt une grande importance pour le brasseur. Les exigences de qualité des brasseries dépassent en règle générale celles applicables à l'eau du robinet. L'eau doit être fraîche, pauvre en calcaire et présenter des propriétés bactériologiques et physico-chimiques irréprochables.

Le degré de dureté de l'eau est déterminant. Pour une bière blonde, on se servira généralement d'une eau douce, alors qu'une eau plus dure pourra être

utilisée pour les bières brunes. Une eau trop calcaire peut, dans certains cas, assombrir la bière et provoquer un goût légèrement âcre, car la substance réagit avec des produits du malt. Autrefois, en l'absence de techniques de traitement, la qualité de l'eau était un aspect déterminant pour le choix du site d'implantation d'une brasserie (Association suisse des brasseries, 2016c).

La levure

Pour que le moût entre en fermentation, on lui adjoint de la levure, qui influe de manière déterminante sur le caractère et la saveur de la bière. Les levures sont des micro-organismes présents presque partout dans l'air ; ils se répartissent en différentes souches. Les brasseries utilisent des levures de culture pures, soit des souches tout à fait identiques de levures, pour pouvoir garantir la qualité régulière du produit final.

On distingue généralement deux types principaux de levures : celles à fermentation basse, et celles à fermentation haute.

Type	Caractéristiques	Comportement à la fermentation
Basée:	Température de fermentation: 5 – 15 °C, levures unicellulaires, arôme léger, typique des bières Lager	Les levures à fermentation basse se déposent au fond de la cuve à la fin de la fermentation.
Haute:	Température de fermentation: 15 – 25 °C, levures en chaînes, arômes plus fruités, typiques des bières de froment, p. ex.	Les levures à fermentation haute remontent pour former un chapeau à la surface du brassin sous l'action des bulles de gaz carbonique formées par la fermentation.

Figure 024 : Tableau des levures.

Tout en étant proches parents, les deux types de levures influent profondément sur le caractère de la bière et confèrent chacun au produit fini un bouquet et un goût caractéristiques. En Suisse, les principales sortes de bières sont obtenues à partir de levures à fermentation basse (Association suisse des brasseries, 2016c).

4.2.2.2. La fabrication de la bière

Ce chapitre passe en revue les principales étapes nécessaires à la production de la bière.

Le brassage

Le malt est d'abord broyé dans un concasseur, puis trempé dans la cuve matière. Le trempage est l'opération qui consiste à mélanger la farine de malt à de l'eau de brassage chaude. La « soupe » qui en résulte, appelée « maïsche », est chauffée à feu doux jusqu'à environ 76 °C, en respectant certains paliers de température. Pendant ce processus, une grande partie de l'amidon contenu dans le malt est transformé en sucre par des enzymes naturelles. Les minéraux, vitamines et une partie des protéines se dissolvent.

Passée dans la cuve de filtration, la maïsche est ensuite séparée des parties solides non solubles, comme l'enveloppe des céréales. Les restes solides, les « drêches », sont valorisées comme aliments pour animaux ou peuvent entrer dans la composition de certaines farines pour la fabrication de pain.

Le liquide ainsi obtenu, le moût, est ensuite additionné de houblon et cuit dans la chaudière à moût. Ce processus permet de libérer les arômes et les tanins. Peu à peu, le moût subit une concentration pour atteindre la teneur en essence souhaitée (« moût d'origine »). Par « moût d'origine », on entend l'extrait dissout dans le moût avant fermentation (maltose, protéines, vitamines et substances aromatiques).

Le moût encore bouillant est ensuite clarifié dans le whirlpool puis rapidement refroidi à la température requise pour la fermentation. En résumé, le brassage consiste à faire passer certaines substances solides du malt dans une solution liquide – le moût. Le processus dure environ sept heures (Association suisse des brasseries, 2016)

La fermentation

Après refroidissement du moût, il estensemencé avec de la levure. Au cours de la fermentation, celle-ci transforme une grande partie du maltose en alcool, gaz carbonique et substances aromatiques.

Après quelques heures, une mousse abondante se forme à la surface de la cuve de fermentation (les « kräusen » dans le jargon). La fermentation principale se termine environ une semaine plus tard : le moût s'est transformé en « bière jeune » (Association suisse des brasseries, 2016).

La garde

La bière jeune est ensuite logée dans des cuves de garde pour la seconde fermentation, la maturation et la clarification. Pendant la garde, elle est alors refroidie à des températures proches de 0 °C.

La pression qui en résulte sature naturellement la bière de son propre gaz carbonique, ce qui est important pour la formation de la mousse, la saveur et la digestibilité (Association suisse des brasseries, 2016).

La filtration

Après quelques semaines, la bière arrive à maturité pour être consommée. Elle est alors prête à être mise en bouteille après une dernière filtration fine. Cette opération n'est pas appliquée aux bières dites non filtrées, afin de préserver les levures (Association suisse des brasseries, 2016).

La mise en bouteille et le conditionnement

Afin d'éviter les pertes de gaz carbonique et la formation de mousse, la bière est soutirée et transvasée, sous contre-pression, en fûts (keg, container), en bouteilles, en canettes ou dans des citerne du système « beer drive ».

Les fûts usagés qui restent dans un état irréprochable et les bouteilles consignées sont soumis à un lavage minutieux avant réutilisation. Tous les contenants doivent être étiquetés conformément aux prescriptions légales, avec notamment la mention du type de bière, de la teneur en alcool, des ingrédients et du fabricant. Après le conditionnement pour le transport, la bière est prête à être livrée (Association suisse des brasseries, 2016).

4.2.3. Description de l'entreprise

L'idée ici est de vous familiariser avec votre nouvel environnement de travail. Les produits que vous allez devoir vendre sont présentés, le processus de production tel qu'il est mis en place actuellement, la chaîne de valeur global, l'outil informatique que la société exploite, l'infrastructure, les machines de production et le procédé global de fabrication.

4.2.3.1. *Les produits commercialisés*

Brewery & Co commercialise quatre sortes de bière. Ces bières divergent les unes des autres dans leur composition :

Illustration	Code	Nom	Composition par litre
	BW01 (A/B)	Lager Beer	Eau 10L Malt d'orge 130gr. Houblon 2gr. Levure 6gr.
	BW02 (A/B)	Honing Beer	Eau 10L Malt de froment 130gr. Houblon 2gr. Levure 6gr. Miel de fleur 25gr.
	BW03 (A/B)	Bio Beer	Eau 10L Malt d'orge 130gr. Houblon 2gr. Levure 6gr.
	BW04 (A/B)	Spice Beer	Eau 10L Malt d'orge 130gr. Houblon 2gr. Levure 6gr. Epices 50gr.

Les nomenclatures

Elles constituent le lien entre un article dit le composé et ses composants, c'est-à-dire, les matières premières. Elles listent donc les composants et leur quantité pour un composé. Vous trouvez le détail de chacune d'elle dans la matrice ci-dessus.

4.2.3.2. *Gestion des opérations de production*

Etant donné la nature des biens produits, la stratégie adoptée par vos prédecesseurs est celle dite de la production en flux poussé.

Cette manière de produire est la plus appropriée pour des produits dont la demande est forte. Ainsi, « Le flux poussé consiste à produire un bien avant qu'un besoin particulier n'ai été formulé par un client. Cela signifie donc que

l'entreprise va engager de l'argent (achat de matières premières, paiement des salariés, utilisation des moyens de production...) sans être réellement sûre à 100% qu'elle pourra vendre le fruit de sa production (et donc par conséquent être payé). Il faut noter également qu'une fois le processus de production terminé, le produit sera stocké en attendant qu'un client ne l'achète. Il en résultera donc des frais de stockage sur des produits finis !» (Logistique pour tous.fr, 2016).

4.2.3.3. La chaîne de valeur

Brewery & Co articule ses opérations autour d'une chaîne de valeur bien huilée ayant déjà fait ses preuves. Elle se compose de quatre processus opérationnels, plus un processus de support. Ce dernier, la comptabilité, a pour mission d'enregistrer tous les flux monétaires de la société avec son environnement extérieur.

Ci-dessous, les principales opérations de ces divers processus opérationnel :

Processus de planification	
Prévision	Les opérations Brewery & Co s'effectuant en flux poussé. Le manager des ventes a pour mission de prévoir la demande afin de créer des besoins indépendants. Ce sont ces besoins qui permettent au MRP de calculer les quantités à produire et les approvisionnements nécessaires en matières premières.
Calcul des besoins	Le MRP doit être lancé pour créer les ordres de fabrication nécessaires à l'atteinte des objectifs de vente formulés par les prévisions faites en amont.
Processus d'approvisionnement	
Demande d'achat	Le manager des achats s'occupe d'acheter les matières premières nécessaires à la production des besoins indépendants. Pour se faire, il valide les demandes d'achats générées par le calcul des besoins ou alors les créées manuellement.
Réception marchandises	Les employés du département réception marchandise, s'occupent de réceptionner la livraison et de vérifier les quantités livrées.
Réception et enregistrement de la facture	La facture est envoyée au département comptabilité qui s'occupe de l'enregistrer dans le PGI.
Paiement fournisseur	En respectant les conditions de paiement du fournisseur, la comptabilité se charge de solder le montant de la facture.
Processus de production	

Lancement des OF	Lorsque les matières premières sont disponibles, il est nécessaire de lancer les ordres de fabrication pour débuter la production qui s'étend sur une durée de 5 jours.
Confirmer la production	Dès que le processus de fabrication arrive à terme, il est nécessaire de confirmer la fabrication de l'OF dans le PGI.
Processus de vente	
Effectuer une offre	Lorsqu'un prospect est intéressé par un article, il est nécessaire de lui faire parvenir une offre correspondante à sa demande.
Créer une vente	Une fois que les parties se sont mises d'accord sur les conditions, une vente ferme est créée dans le PGI.
Livraison	La quantité demandée de l'article est prélevée en stock et envoyée à l'adresse de livraison du client.
Facturation	Les articles sont facturés au client sur la base des conditions négociées.
Comptabilisation des débiteurs	Le relevé bancaire indique que le client a payé le montant de sa facture. Il s'agit de comptabiliser cette écriture dans le PGI.

4.2.3.4. *Support aux opérations*

Pour vous aider dans votre mission, **Brewery & Co** utilise un progiciel de gestion intégré du nom d'**Odoo®**. Ce PGI est largement utilisé dans les entreprises telles que la vôtre. Plusieurs millions d'utilisateurs à travers le monde ont aujourd'hui à faire à cet outil informatique.

L'une des premières tâches qui vous incombe est de vous familiariser avec cet outil afin de piloter vos opérations quotidiennes de la manière la plus efficiente possible. Vous verrez, Odoo vous donne accès à une vision unifiée de tout ce qu'il se passe dans votre entreprise.

4.2.3.5. Infrastructure

Pour produire vos bières, vous possédez vos propres installations. Votre entreprise se situe en périphérie de la capitale. Elle dispose d'un pont de livraison, d'un pont de chargement pour les livraisons, d'une grande halle de production ainsi que d'une partie dédiée aux départements administratifs. Voici le plan de votre infrastructure :

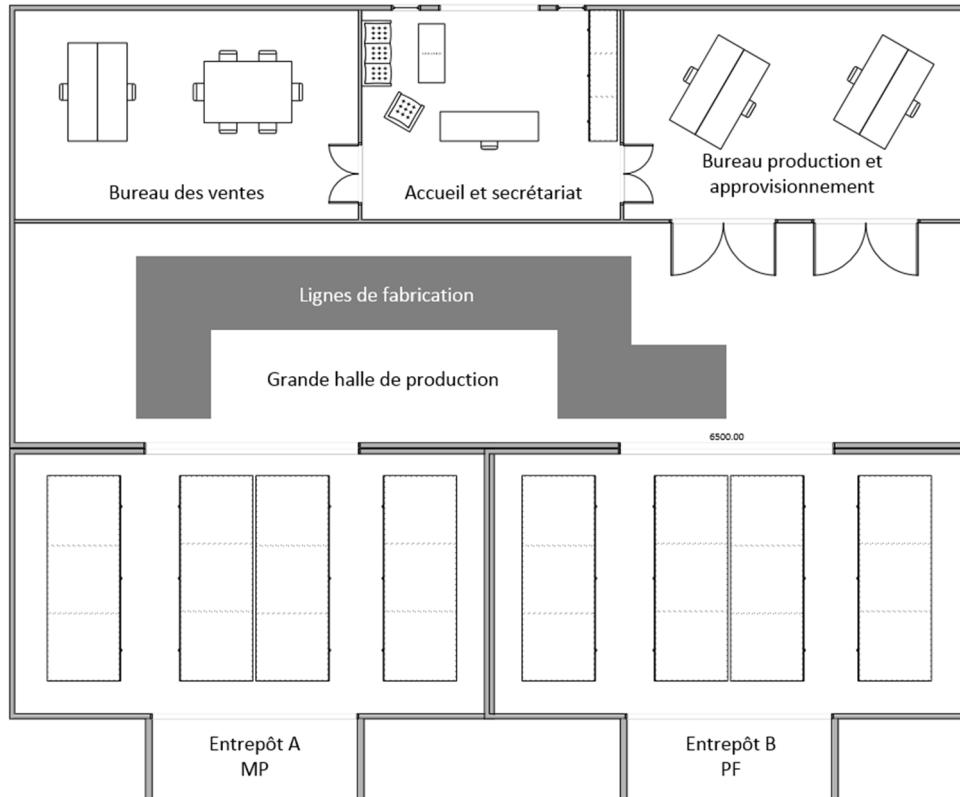


Figure 025 : Plan d'aménagement du site de production Brewery & Co.

Ceci est le résultat d'un investissement d'un million de CHF entre le terrain et le bâtiment.

L'entreprise dispose de deux endroits de stockage. L'entrepôt A permet de stocker les matières premières. L'entrepôt B est l'endroit où les produits finis vont atterrir à la fin du processus de fabrication. Les bouteilles et canettes vides utilisées pour le conditionnement se situent dans l'entrepôt A.

Ces entrepôts ont des limitations en termes de volumétrie à disposition. Ainsi, il est nécessaire de surveiller à ce que les limites ne soient pas dépassées.

L'entrepôt de matière première permet de stocker :

- 120 m³ d'eau.
- 4 cuves de 350 kg destinées au stockage des différents malts.
- 4 cuves de 6 kg destinées au stockage du houblon.
- 1 cuve de 70 kg destinée au stockage de la levure.
- 1 cuve de 150 kg destinée au stockage des épices.
- 1 étagère pouvant accueillir 75 pots d'un kilogramme de miel.

- 1 étagère pouvant accueillir 25'000 unités de conditionnements, tous types confondus.

Ces quantités permettent d'approvisionner les lignes de production pendant 1 mois sans avoir à commander de la matière première auprès des fournisseurs. L'entrepôt B est celui réservé aux produits finis. Il est organisé en ligne et est réfrigéré. Il permet d'accueillir :

- 50'000 unités (grand ou petit format).

Cette limite permet de stocker le résultat d'un mois de production.

Il est tout à fait possible d'étendre ces capacités de stockage auprès d'un tiers. Le partenaire Warouhousing Swiss AG loue ses services à la journée.

- 100.- CHF par récipient de 10 kilogrammes de matières premières.
- 200.- CHF par lot de 12'000 unités canettes ou bouteilles, vides ou pleines.

La location d'espace de stockage supplémentaire peut faire partie d'une stratégie. Attention toutefois à ne pas vous retrouver en sous-liquidité.

4.2.3.6. Matériels de production

La fabrication de bière s'effectue selon un ordonnancement d'opérations chronologiques. Pour se faire, vous disposez de trois lignes de production indépendantes d'une valeur de 150'000.- chacune. Elles peuvent produire mensuellement 4'000 litres de bières. Ceci signifie qu'avec un temps de production de 5 jours, l'entreprise produit 3'000 litres de bière toutes les semaines.

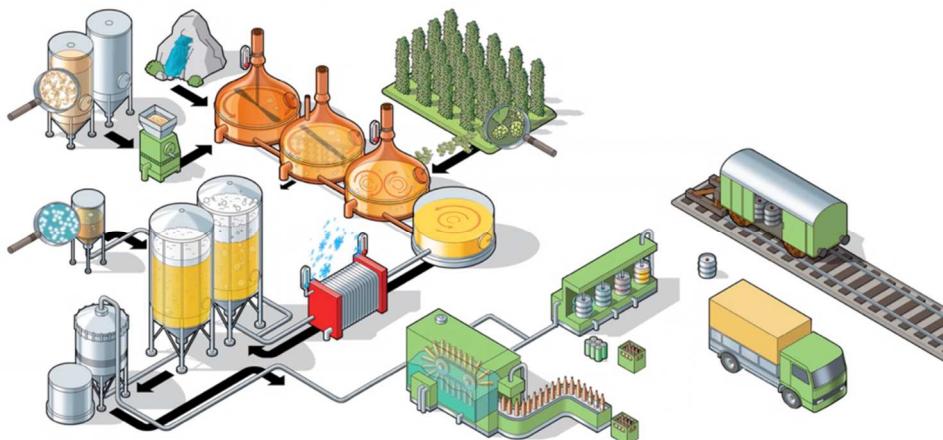


Figure 026 : Zoom sur une ligne de production³³

Votre gamme de produit est composée de quatre variétés, cependant, vous ne pouvez produire que trois sortes de bière simultanément. Si vous décidez de changer la bière produite par une ligne, cela impose un nettoyage et une

³³ <http://bier.ch/fr/autour-de-la-biere/le-brassage/infographie/>

désinfectassions des cuves. Cette opération dure 1 jour. Bien évidemment, votre production s'arrête durant ce laps de temps.

4.2.3.7. *Frais fixes mensuel*

Votre société doit prendre en compte diverses charges d'exploitation qui pèsent sur elle. Vous devez payer la main d'œuvre, des frais de bureau, des frais de licences informatiques et toutes les autres charges inhérentes à une entreprise.

Les coûts mensuelles se monte à :

- 5'000.- pour la main d'œuvre.
- 30'000.- pour les frais généraux et l'administration.
- 8'500.- pour les frais relatifs à votre emprunt bancaire.
- 3'750.- pour les coûts liés à l'amortissement de vos installations de production.

4.2.3.8. *Coûts de revient*

Le coût de revient de vos produits se compose des matières premières, des réceptacles, de la main d'œuvre directe et indirecte plus l'amortissement des installations de production et le remboursement de votre dette bancaire.

Par exemple, en prenant le coût de revient pour le produit BW01 (détails ci-dessous) conditionné au format 0.25 cl, les résultats obtenus sont les suivants :

Part	Nature des coûts	Considérations	Montant
32%	Matières premières	100% de la capacité productive employé (48'000 unités)	22'080.-
50%	Frais fixe d'exploitation	Salaire MOI/MOD et autres frais généraux	35'000.-
12%	Emprunt bancaire	Sur la base d'un prêt de 1'200'000.- CHF bloqué à 4.50% sur 25 ans	8'500.-
6%	Amortissement	450'000.- de machine industrielle à amortir sur 10 ans	3'750.-
		<i>Prix de revient :</i>	69'330.-
		Prix de revient unitaire :	1.444

Les chapitres qui suivent présentent les coûts de reviens bruts pour 4'000 litre de bière (production mensuelle par ligne). Pour connaître le prix de vente de vos produits, il vous suffit de sommer les coûts d'exploitation, le coût des matières premières basé sur les quantités présentées dans le chapitre Les matières premières et celui du conditionnement.

Produit BW01

Matières premières	Quantité	Prix de revient
Eau	4m ³	6.-
Malt d'orge	520kg	312.-
Houblon	8kg	144.-
Levure à bière	24kg	6'000.-
Sous-total		6'462.- CHF
Conditionnement 0.25 cl	16'000 unités	800.-
Total		7'262.- CHF
Conditionnement 0.50 cl	8'000 unités	80.-
Total		6'542.- CHF

Produit BW02

Matières premières	Quantité	Prix de revient
Eau	4m ³	6.-
Malt de fromant	520kg	228.80
Houblon	8kg	144.-
Levure à bière	24kg	6'000.-
Miel de fleur	100kg	1'000.-
Sous-total		7'378.80 CHF
Conditionnement 0.25 cl	16'000 unités	800.-
Total		8'178.80 CHF
Conditionnement 0.50 cl	8'000 unités	80.-
Total		7'458.80 CHF

Produit BW03

Matières premières	Quantité	Prix de revient
Eau	4m ³	6.-
Malt d'orge labélisé Bio	520kg	381.34
Houblon labélisé Bio	8kg	216.-
Levure à bière	24kg	6'000.-
Sous-total		6'603.34 CHF
Conditionnement 0.25 cl	16'000 unités	800.-
Total		7'403.34 CHF
Conditionnement 0.50 cl	8'000 unités	80.-
Total		6'683.34 CHF

Produit BW04

Matières premières	Quantité	Prix de revient
Eau	4m ³	6.-
Malt d'orge	520kg	312.-
Houblon	8kg	144.-
Levure à bière	24kg	6'000.-
Epices	2'000kg	1'600.-
Sous-total		8'062.- CHF
Conditionnement 0.25 cl	16'000 unités	800.-
Total		8'862.- CHF
Conditionnement 0.50 cl	8'000 unités	80.-
Total		8'142.- CHF

4.2.3.9. Exemple de rendement optimum

L’analyse qui suit part du principe que les processus de l’entreprise sont à 100% efficents. Elle se base sur le coût de revient du produit BW01.

En prenant en considération votre capacité de production de 12'000 litres, une activité sans interruption, une marge sur le produit de 35% (1.95 la bouteille) et l’ensemble de vos stocks vendus sur la période, vous pouvez atteindre un chiffre d’affaires de 93'600.- de CHF.

Cela signifie que vous pouvez obtenir des bénéfices à hauteur de 24'270.- CHF mensuellement.

4.2.4. L’environnement

Ce chapitre présente les éléments avec lesquels vous devez interagir durant votre mandat. Plus particulièrement, les caractéristiques principales des marchés, les matières premières, vos partenaires sur le marché de l’approvisionnement, vos revendeurs, la provenance de vos fonds, les tendances de fluctuations saisonnières et les principales contraintes.

4.2.4.1. Le marché

Vous évoluez dans un marché concurrentiel « Un marché concurrentiel est un marché sur lequel il y a de nombreux acheteurs et de nombreux vendeurs d’un produit identique, de sorte que chacun a un impact négligeable sur le prix de marché. » (Ionescu, 2014).

Cependant, vous n’êtes pas tout à fait sur un marché parfaitement concurrentiel. En effet, il est préférable de parler d’un marché dit oligopole. Dans ce type de marché, « on trouve des vendeurs peu nombreux qui ne se concurrencent pas toujours de manière agressive. » (Ionescu, 2014).

4.2.4.2. *Les matières premières*

Dans le but de créer vos produits, vous êtes dépendant du marché de l'approvisionnement. Pour faciliter le management des ressources utiles à la fabrication de vos bières, la décision de définir ce marché comme étant illimité est prise. Ce qui a pour effet de supprimer le risque de ne pas pouvoir s'approvisionner.

Ci-dessous, les prix³⁴ des matières premières utilisées dans vos recettes :

Matières premières	Prix au kilo
Eau (m ³)	1.50
Malt d'orge	0.60
Malt d'orge Bio	0.7333
Houblon	18.-
Houblon Bio	27.-
Levure à bière	250.-
Levure à bière Bio	275.-
Miel de fleur	10.-
Bouquet d'épices (sachet de 25 grammes)	0.20
Réceptacle en verre de contenance 0.25cl	0.05
Réceptacle en aluminium de contenance 0.50 cl	0.01

4.2.4.3. *Les fournisseurs*

Vous traitez principalement avec trois fournisseurs. Vous n'avez pas la possibilité de changer ces partenaires durant votre mandat. En effet, ces partenariats sont le fruit d'une longue collaboration fructueuse mise en place par vos prédécesseurs.

Les matières premières telles que le houblon, les différents malts, la levure et ainsi que les récipients s'obtiennent chez Maltlonne SA.

L'eau est obtenue chez la société cantonale Ourseau SA.

Pour les aliments plus artisanaux comme le miel et les épices, ceux-ci, s'obtiennent chez l'artisan Monsieur Ture.

³⁴ Les prix sont influencés par Viteos SA pour l'eau, www.brouwland.com pour les différents types de malt, <https://www.bieresdumonde.fr/156-levures-de-brasserie> pour la levure et de manière subjective pour le miel et le bouquet d'épices.

Tous ces partenaires se situent en Suisse. Ce qui facilite les échanges et évite de devoir travailler dans des devises différentes. L'une des informations très importantes, est celle des délais de livraison et des conditions de paiement que ces différents fournisseurs vous offrent :

- Maltlonne SA vous octroie des conditions de paiement à 30 jours et livre la marchandise dans les 15 jours.
- Ourseau SA vous octroie des conditions de paiement à 15 jours et livre par camion-citerne tous les 15 jours.
- Monsieur Ture, petit commerçant ne peut pas se permettre de vous offrir des conditions de paiement. Il a besoin rapidement de liquidité et par conséquent vous demande de régler à la réception de la marchandise qu'il livre au minimum 10 jours après commande.

4.2.4.4. *Les revendeurs*

Vous ne travaillez pas directement avec le consommateur final. Vous passez par 220 revendeurs agréés afin de distribuer vos produits.

Ces revendeurs sont atteignables par divers canaux de distribution. Chacun d'eux a ses spécificités et ses clients avec des profils variés et des préférences différentes. Il s'agit là de les prendre en considération afin d'augmenter vos ventes.

De plus, ces revendeurs se situent dans trois zones géographiques différentes. Voici leur répartition :

	Détaillant	Supermarché	Autres ³⁵	Totaux
Ouest	23	11	49	37.72%
Centrale	12	17	58	39.55%
Est	8	15	27	22.73%
Totaux	19.55%	19.55%	60.90%	220

³⁵ Restaurants, discothèques, bars et cafés.

Les parts de marchés et les préférences

Vos canaux de distributions ne possèdent pas tous les mêmes parts de marchés. Il est important de prendre en considération les informations ci-dessous pour vos prévoir de la meilleure manière possible vos futures ventes :

43 Détailleur

- 20% de la demande totale
- Attrait pour BW02, 03 et 04
- Format 25 cl
- 15% plus cher que les supermarchés
- Paie entre 5 et 10 jours

43 Supermarchés

- 55% de la demande totale
- Attrait pour tous les produits
- Format 50 cl
- Prix le plus bas possible
- Paie à 15 jours

134 Autres*

- 25% de la demande totale
- Uniquement la Lager Beer
- Format 25 cl
- Jusqu'à 50% plus cher
- Paie en une dizaine de jours

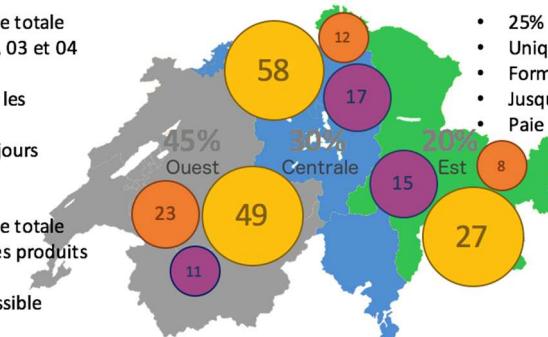


Figure 027 : Composition du marché Suisse.

Les trois régions Ouest, Centrale et Est ont des parts de marchés différentes. La Suisse Ouest possède 45% de la demande, la Suisse Centrale 35% et la Suisse de l'Est 20%. Ensuite, ces parts sont encore repartagés par type de revendeur. Ces parts de marchés sont présentées dans l'illustration ci-dessus.

Détaillant

Les détaillants sont le plus faible des canaux de distribution. Avec un total de vente de 20%, ils se placent derrière les deux autres.

Ces revendeurs apprécient de pouvoir proposer à leurs clients, une variété restreinte en terme de quantité mais axe son offre sur des produits premium tel que la bière bio, au miel ou encore la spécialité aux épices.

Ils vendent uniquement dans le format en bouteille de 25 cl et ne sont donc catégoriquement pas intéressés par les autres types de conditionnement.

Leurs clients sont peu sensibles aux prix pratiqués. Les produits se vendent généralement 15% plus cher que dans les supermarchés.

Les détaillants paient généralement les factures ouvertes entre 5 et 10 jours. Cependant, gardez en tête que vous leur accordez un délai de paiement à 15 jours.

Supermarché

En terme de quantité vendue, ce canal de distribution représente le plus important avec des parts de vente grimpant à 55%.

L'éventail de l'offre qu'ils proposent est large. Ils vendent toutes vos bières dans le format en canette de 50cl.

Les clients de ces établissements sont très sensibles aux prix. Ils veulent acheter au prix le plus bas possible.

Les supermarchés attendent la fin de leur droit en termes de conditions de paiement. De ce fait, vous recevez leurs paiements après 15 jours.

Établissement public

Les établissements publics où les gens se rencontrent pour échanger et boire un verre sont aussi très rentables pour Brewery & Co. Ils représentent 25% de la demande totale.

Les produits proposés sont uniquement la bière BW01 et la BW04 d'octobre à mars. De plus, ils commandent dans un seul format, celui en bouteille de verre d'une contenance de 25cl.

Les clients ne sont pas regardant quant aux prix pratiqués. Il n'est pas rare de voir des ventes avec des marges d'environ 50% par rapport aux ventes effectuées dans les supermarchés.

Ces revendeurs paient généralement dans les 10 jours vos factures, ce qui représente un avantage. Vous obtenez rapidement des liquidités en retour.

4.2.4.5. Les bailleurs de fonds

L'acquisition de votre entreprise et de ses installations vous ont contraint à trouver des investisseurs et un établissement bancaire vous octroyant un crédit.

Votre structure financière se compose de la manière suivante :

ACTIF		PASSIF	
Circulants			A court terme
<i>Compte VUBS SA</i>	150'000	0	<i>Dettes fournisseurs</i>
Immobilisés			A long terme
<i>Machines</i>	450'000	1'200'000	<i>Dette bancaire</i>
<i>Bâtiments</i>	725'000	400'000	<i>Capital</i>
<i>Terrain</i>	275'000		
Total :	1'600'000	1'600'000	: Total

L'une de vos missions est de rembourser petit à petit votre emprunt et de rapporter de l'argent à vos actionnaires en dégageant des bénéfices assez importants pour d'une part constituer une réserve et de l'autre, distribuer des dividendes.

Vous débutez votre mission avec un solde de liquidité de 150'000.- CHF pour effectuer vos approvisionnements et payer les charges inhérentes à l'exploitation de votre business.

4.2.4.6. La fluctuation de la demande

Durant l'année, les ventes fluctuent à diverses périodes. La demande pour vos produits diffère entre les saisons chaudes et les saisons froides.

Par exemple, il n'est pas rare de voir les ventes de bière au miel et aux épices s'envoler de novembre à fin janvier. Tandis que la bière standard se boit de manière plus prononcée durant les périodes ensoleillées. Pour la bière labélisée Bio, sa consommation est stable tout du long de l'année. Voici un tableau des valeurs ainsi qu'un graphique illustrant ces propos :

Article	Round 1 (hiver)	Round 2 (printemps)	Round 3 (été)	Round 4 (automne)
BW01	+0%	+5%	+20%	-5%
BW02	+8%	-3%	-3%	+5%
BW03	+0%	+0%	+0%	+0%
BW04	+15%	-20%	-30%	+45%

C'est donc de votre ressort de prendre en considération ces éléments et de planifier la demande et vos lots de production en connaissance de cause.

4.2.5. Les modalités de jeu

Ce chapitre présente les modalités du jeu. Principalement, les temps à disposition et le détail de l'évaluation de votre performance.

4.2.5.1. Simulation du temps

Le temps imparti pour jouer une partie Brewery & Co se compose de quatre rounds de 10 minutes. Un round simule 10 semaines, c'est-à-dire 1 semaine réelle est exécutée dans le jeu en à peine une minute. Vous débutez le jeu à la semaine numéro dix jusqu'à la numéro 20, ce qui vous positionne en saison de printemps. Ce choix est fait pour éviter de commencer avec la saison hivernale où la demande est très forte. De cette manière, vous pouvez débuter en douceur.

Ci-dessous, le détail des rounds et des temps simulés :

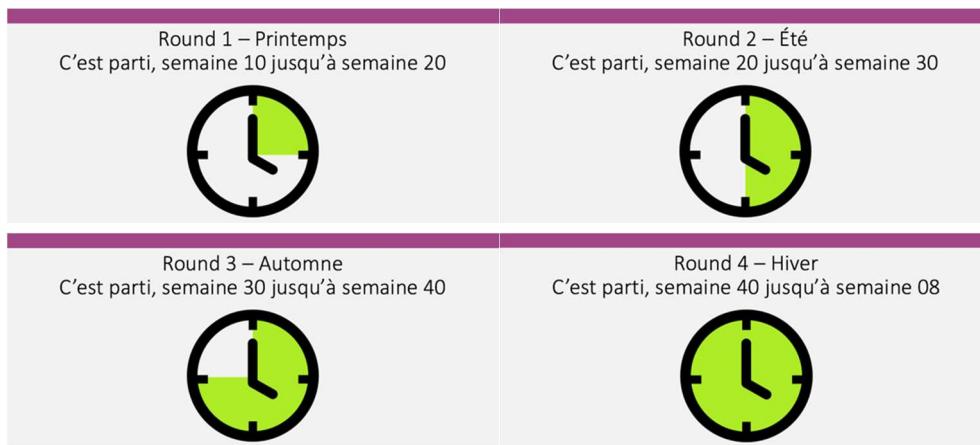


Figure 028 : Détail des rounds, semaines et saisons dans le jeu OdooSIM.

4.2.5.2. Évaluation de la performance

Votre évaluation est faite sur le bénéfice que vous générez. Il est par conséquent nécessaire de ne pas confondre chiffre d'affaires et bénéfices. En effet, vous devez payer beaucoup de charges et par conséquent, le bénéfice résultant est bien plus maigre que le chiffre d'affaires.

4.3. Limites et recommandations

Le scénario et son détail est développé et présenté. Il est toutefois important de mentionner que son développement a été effectué par un informaticien de gestion. Même si tout a été mis en œuvre pour qu'il soit le plus réaliste possible, des écarts avec la réalité peuvent éventuellement exister.

Il est recommandé d'analyser le résultat fourni et d'y apporter une vision gestionnaire d'entreprise. Principalement, les capacités des stocks et les frais fixes sont basés sur des hypothèses et doivent être revus et validés.

Le paiement bancaire mensuel prend en compte le remboursement de la dette à hauteur de 4'000.- CHF en plus des intérêts. Dans la réalité, ce n'est pas forcément le cas. La majeure partie du temps, seul les intérêts sont payés mensuellement. Quant au remboursement, il intervient après le délai sur lequel les parties se sont entendues. Ce principe de prêt est principalement négocié lorsque les entreprises ont un besoin pressant en liquidité pour investir. En amont elles ont déjà effectuées des analyses de rentabilité et savent que l'investissement rapporte plus que le prêt. Ainsi, au moment de l'échéance, elles peuvent rembourser l'établissement préteur. De plus, les intérêts doivent être dégressifs au fur et à mesure que le remboursement s'effectue, ce qui n'est pas le cas dans les calculs actuels.

La partie simulation du temps est à approfondir. En effet, ce n'est qu'une alternative qui peut sûrement être discutée. En effet, la simulation d'une année telle qu'elle est préconisée peut-être longue dans le cadre d'une ou deux leçons de 45 minutes, à voir les besoins exacts avec le corps professoral. Il faut simplement garder à l'esprit l'influence des saisons et essayer de le représenter au mieux dans la disposition des rounds que l'on choisit.

5. Portabilité des contraintes scénaristiques sur OdooSIM

Ce chapitre explique, où est-ce qu'au niveau de la chaîne de valeur les contraintes exercent des pressions. La chaîne de valeur globale peut être consultée au chapitre Cycle opérationnel.

Il donne les recommandations nécessaires à la mise en œuvre dans le simulateur. En effet, c'est ce dernier qui s'occupe de la gestion des contraintes. Il le fait à chaque jour de simulation, c'est-à-dire à chaque fois qu'il exécute sa phase numéro deux (se référer au chapitre Phase de simulation).

5.1. Capacité de stockage local

Les stocks de la société ont des limites en termes de volumétrie. En se référant au chapitre Infrastructure qui présente le scénario, le constat est que l'entreprise dispose de deux entrepôts. Le premier pour stocker ses matières premières et le second pour les produits finis. Cette contrainte de volumétrie pèse sur les processus de planification, d'approvisionnement et de comptabilité.

La planification est impactée du fait qu'il faut prévoir de calculer la somme des quantités maximales des règles d'approvisionnement pour chacun des produits. Cette somme ne doit pas être plus élevée que la capacité totale de l'entrepôt B si la stratégie de l'entreprise est de ne pas vouloir acquérir de l'espace de stockage chez un tiers.

L'approvisionnement doit tenir compte de la limite de stockage dans l'entrepôt A lorsqu'elle commande.

La comptabilité doit facturer les montants adéquats par tranche de volume dépassée. Pour le faire, le simulateur doit calculer la somme des quantités entreposées chaque jour simulé. La quantité des matières premières ne doit pas excéder la capacité totale de l'entrepôt A. La quantité de produits finis ne doit pas dépasser la quantité maximale que l'entrepôt B peut héberger. Si c'est le cas, l'écriture comptable suivante doit être passée :

4400 prestations de travail auprès d'un tiers à 10201 Bank pour X CHF

5.2. Marché de l'approvisionnement

Dans la version actuelle du scénario, le marché de l'approvisionnement est illimité. Il est envisageable de définir une contrainte quantitative au niveau des disponibilités de ce marché.

Si tel est le cas, cette contrainte pèse sur les processus d'approvisionnement et par conséquent sur celui de la production.

La limite doit être gérée au niveau du simulateur. Lorsqu'une société effectue un approvisionnement, le solde restant sur le marché doit être comparé à la

quantité demandée. Si la quantité demandée n'excède pas le solde alors l'achat peut être conclu. Dès que l'achat s'effectue, le solde doit être diminué de la quantité qui vient d'être achetée.

5.3. Délais

Plusieurs contraintes de ce type-là sont présentes dans le scénario Brewery & Co. Les délais sont ceux :

- Du temps de procuration des matières premières chez les différents fournisseurs.
- Du temps accordé par les fournisseurs pour le paiement des factures.
- Du temps que l'entreprise accorde aux différents types de revendeur pour payer leurs dûs.
- Du temps nécessaire à la fabrication des bières

Le temps de procuration des matières premières se comptabilise en nombre de jour. Au niveau du processus d'approvisionnement cette contrainte agit à cet endroit :

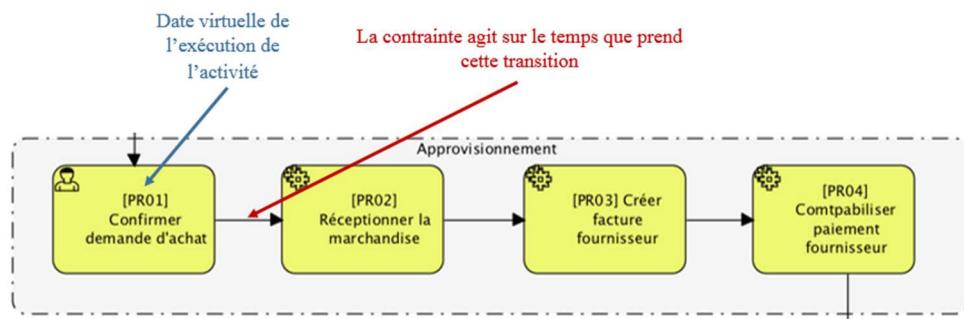


Figure 029 : Contraintes sur le processus d'approvisionnement.

Pour gérer ce délai, le simulateur doit comparer la date de simulation actuelle à la date où la commande a été confirmée.

Le temps qu'accordent les fournisseurs au paiement des factures permet de garder des liquidités en sa possession le plus longtemps possible. Cette contrainte agit à cet endroit :

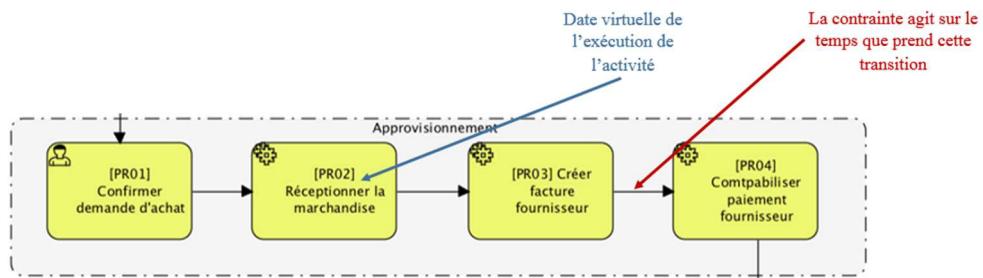


Figure 030 : Contraintes sur le processus d'approvisionnement.

La gestion de ce délai incombe au simulateur de lire la date virtuelle de l'exécution de l'activité **PR02** et de la comparer à la date virtuelle du jour où la vérification s'effectue. Si l'échéance est atteinte, le simulateur doit effectuer le paiement **PR04**.

Le temps qu'accorde l'entreprise à ses revendeurs pour le paiement de leurs factures doit être pris en considération pour évaluer les liquidités futures. Ce délai qui est une contrainte sur le processus des ventes, se met en place à cet endroit :

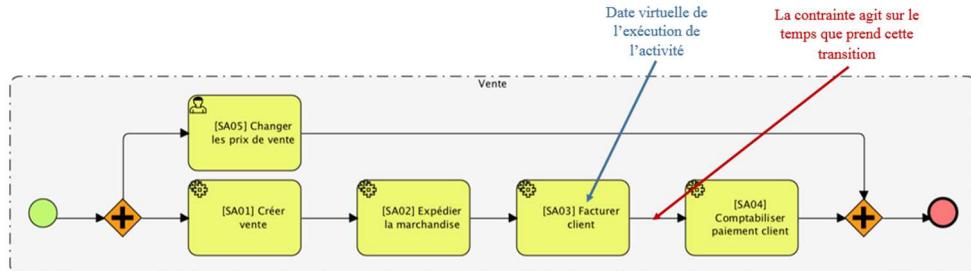


Figure 031 : Contraintes sur le processus de vente.

Pour gérer cette contrainte, le simulateur doit lire la date virtuelle où l'activité **SA03** a lieu et la comparer à la date de simulation courante. Si le délai est arrivé à échéance, il doit comptabiliser le montant de la facture en exécutant l'activité **SA04**.

La contrainte de temps nécessaire à la fabrication d'un lot de produit exerce une pression sur le processus de production. Il agit à cet endroit :

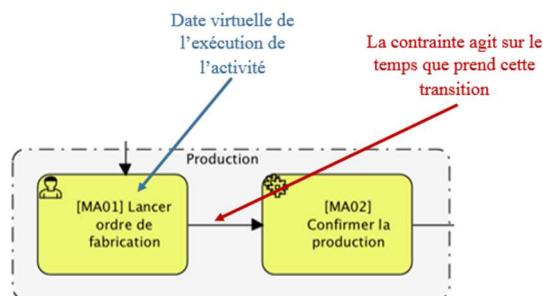


Figure 032 : Contraintes sur le processus de production.

Pour la gérer, le simulateur doit lire la date virtuelle où l'activité **MA01** est exécutée par les participants et la comparer à la date courante de simulation. Si le délai est arrivé à échéance, le simulateur doit exécuter l'activité automatisée **MA02**.

5.4. Frais fixes

Les frais fixes sont gérés dans le processus de comptabilité. Ils doivent être comptabilisés à la fin de chaque mois virtuel. Le choix de les comptabiliser à ce moment-là est pris pour éviter d'effectuer des transactions avec l'API chaque jour simulé. Ceci a pour effet, d'alléger la phase de simulation dans le simulateur. Les écritures comptables à passer sont :

5000 Salaires à 10201 Bank pour 35'000.- CHF

2100 Dettes bancaires à 10201 Bank pour 4'000.- CHF

6900 Charges financières à 10201 Bank pour 4'500.- CHF

6800 Amortissements à 1500 Machines et appareils pour 3'750.- CHF

5.5. Liquidités

Les liquidités peuvent devenir une contrainte, dès le moment où elles ne sont plus disponibles. En effet, dans le jeu, une société fait faillite à partir du moment où elle n'a plus assez de liquidité pour payer ses factures et/ou ses charges (dettes bancaires, salaires, etc...).

Pour gérer cela, le simulateur doit tous les jours vérifier qu'après avoir comptabilisé les factures ouvertes arrivées à échéances, l'entreprise n'évolue pas en sous liquidités grâce au bilan :

Current Assets		
Bank and Cash Accounts		
10201 Bank	150,000.00 CHF	
Total Bank and Cash Accounts	150,000.00 CHF	

Figure 033 : Compte 10201 Bank lequel contient les liquidités de la société.

Si c'est le cas, le simulateur donne droit à 30 jours pour rétablir un niveau de liquidité lui permettant de couvrir ses charges. Si arrivé à ces 30 jours l'entreprise n'est pas revenue dans les chiffres noirs, elle est éjectée de la partie sous prétexte d'une faillite.

La recommandation fait état qu'il faut stocker dans le graphe d'objet et plus précisément dans une instance représentant la société, la date virtuelle à laquelle l'entreprise entre en sous liquidités. Suite à cela, le simulateur vérifie chaque jour, la santé financière de la société. Si elle ne parvient pas à rétablir sa solvabilité dans le délai imparti, le simulateur doit supprimer l'instance de l'objet qui référence la société.

Complément d'informations :



Chaque passage dans la zone rouge doit être comptabilisé par le simulateur et en fin de partie une pénalité doit être infligée à la société. Cette pénalité doit être comptabilisée par l'écriture suivante :

6700 Autres charges d'exploitation à 10201 Bank pour X CHF

5.6. Récapitulation

L'image ci-dessous exprime les contraintes de chaque activité :

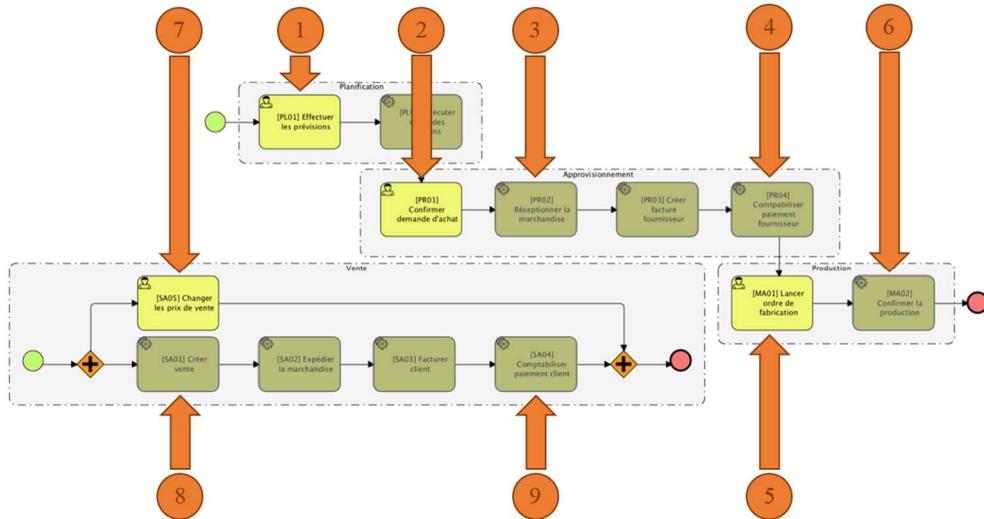


Figure 034 : Contraintes pesant sur la chaîne de valeur Brewery & Co.

N°	Gérée par	Explication
1	Participants	Si les participants ne souhaitent pas acheter du stockage supplémentaire, il est nécessaire que la somme des seuils maximaux de toutes les règles d'approvisionnement automatique n'excèdent pas la capacité de stockage des produits finis.
2	Participants	Les ordres d'achats doivent être confirmés uniquement si les liquidités sont suffisantes pour s'acquitter de la facture fournisseur qui s'en suit.
3	Simulateur	Cette activité ne peut être performée uniquement lorsque le délai d'attente des matières premières est arrivé à échéance.
4	Simulateur	Cette activité a lieu à l'échéance du délai qu'accorde le fournisseur à la société.
5	Participants	Avant de lancer un ordre de fabrication, les participants doivent s'assurer qu'une ligne de production soit libre.
6	Simulateur	Cette activité s'effectue dès que le temps nécessaire à la fabrication d'un lot est atteint.
7	Participants	Les consommateurs réagissent aux variations de prix. Un prix trop bas augmente la demande. Un prix trop haut diminue la demande.
8	Simulateur	L'article doit être disponible.
9	Simulateur	Le fichier settings.xml défini pour chacun des types de revendeur un nombre de jours minimaux

		et un nombre de jours maximaux. Ces jours définissent la borne dans laquelle le revendeur paie ses factures. Le simulateur doit effectuer un random entre ces valeurs et une fois que le temps a atteint cette valeur, il comptabilise le montant.
--	--	--

6. Développement du proof of concept

Les éléments qui ont été mis en œuvres durant le développement de la version proof of concept d'OdooSIM sont expliqués dans les chapitres qui suivent.

6.1. Problématiques

Le simulateur doit configurer les instances Odoo nécessaires à la réalisation d'une nouvelle partie. Cette phase doit permettre d'obtenir une instance répondante aux besoins du scénario (master data et configuration générale) pour chacune des équipes.

Le simulateur a comme contrainte d'avoir à effectuer dans un temps court beaucoup d'opérations avec toutes les instances qui sont en concurrence dans une partie de jeu nouvellement créée. Ces opérations sont toutes celles nécessaires à faire évoluer le scénario. Elles s'effectuent grâce l'API. Malheureusement, le fait d'invoquer un service par Internet sur un élément distant provoque des temps d'attente trop long. Par conséquent, il s'agit de trouver un moyen pour limiter les interactions durant la phase de simulation.

L'autre grande problématique à laquelle le développement se confronte est de dater les opérations avec les dates virtuelles et non pas les dates réelles. En effet, une partie se joue en moins d'une heure et par conséquent, toutes les opérations générées sont datées avec une date unique, celle du jour de la simulation. Pour que le scénario soit cohérent et que les participants puissent tirer des analyses pertinentes, il faut trouver une solution qui permette de dater les opérations avec les dates virtuelles.

6.2. Langage de programmation

Pour développer cette version du simulateur, le langage Java est choisi. Les motivations de ce choix sont principalement subjectives. En effet, le travail s'étalant sur un délai court de quatorze semaines, il n'est pas envisageable de prendre en main un nouveau langage de programmation. De part ce fait, le choix s'est porté sur ce langage qui est appris durant la formation d'informaticien de gestion.

D'une manière objective, le langage Python semble être une meilleure solution. En effet, Odoo est écrit grâce à ce langage. Python permet d'étendre certaines fonctionnalités directement dans le code source de la solution. Ce qui n'est pas réalisable avec Java. Même si ce besoin n'est pas existant dans cette version du simulateur, il peut le devenir dans la réalisation finale.

6.3. Fonctionnalités attendues

Dans le cahier des charges, une liste des fonctionnalités à couvrir est établie et permet de piloter le projet de développement proof of concept avec des objectifs mesurables.

Cette liste est formulée sous forme d'user stories. Voici son contenu :

- « En tant que participant (faisant parti d'un groupe), je veux qu'il me soit possible de m'authentifier sur une instance Odoo SaaS qui m'est dédiée afin de pouvoir rejoindre une partie ».
- « En tant que responsable des ventes, je veux qu'il me soit possible de gérer mes conditions de vente sur une instance Odoo Saas afin d'impacter le volume des ventes ».
- « En tant que responsable des ventes, je veux qu'il me soit possible d'investir dans une campagne publicitaire ciblée sur une instance Odoo Saas afin d'augmenter l'attrait des consommateurs pour un produit distinct ».
- « En tant que responsable des ventes, je veux qu'il me soit possible de visualiser ma progression dans le PGI afin de m'informer et de mesurer ma performance ».
- « En tant que professeur, je veux qu'il me soit possible de paramétrier une nouvelle partie afin de démarrer une nouvelle simulation ».
- « En tant que professeur, je veux qu'il me soit possible de démarrer une nouvelle partie afin de dispenser une leçon ».
- « En tant que professeur, je veux qu'il me soit possible d'avoir un retour sur la partie afin de visualiser les opérations survenues durant la partie ».

Ces sept user stories doivent être réalisées dans cette version du simulateur.

6.4. Cycle opérationnel

Cette version du simulateur doit couvrir l'entier du processus des ventes. Ce processus est détaillé par l'illustration ci-dessous :

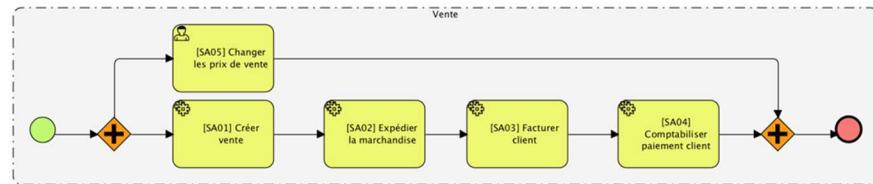


Figure 035 : Processus des ventes dans Odoo.

Cependant, ce n'est pas l'ultime point qui doit être géré. Pour qu'une vente puisse s'effectuer, plusieurs étapes précédentes sont nécessaires. Dans l'ordre chronologique, le simulateur doit tout d'abord remplir les stocks des sociétés en jeu. Il fait cela car pour vendre, il faut de la disponibilité. Cette version ne gérant pas la partie production, cette création de disponibilité s'effectue ainsi. Ensuite, il doit générer de la demande virtuelle pour chacun des produits du scénario et à partir des instances Odoo créer l'offre. Dès que les consommateurs et les fournisseurs sont générés, il faut parcourir toutes les demandes et les confronter aux offres pour sélectionner l'offre optimale d'un point de vue consommateur, c'est-à-dire l'offre la moins cher. Une fois que l'offre est ciblée, une vente s'effectue. Au niveau du processus présenté ci-dessus, cela se concrétise en amorçant par l'activité **SA01** et en effectuant toutes les opérations en aval.

6.5. Exécution du simulateur

L'exécution du simulateur se divise en trois phases distinctes. Ces phases ont toutes des responsabilités différentes.

La première phase effectue la configuration globale des instances. Cela comprend la configuration d'une partie des applications nécessaires au support des processus, la mise à disposition d'un accès aux différentes équipes et le master data rempli en fonction du scénario. Tous les éléments nécessaires à cette configuration sont présents dans un fichier de configuration au formalisme XML nommé **settings.xml**.

La seconde phase s'occupe de la simulation du temps virtuel et joue le cycle opérationnel de manière itérative. Elle l'exécute autant de fois qu'il y a de jours paramétrés (**nombre de round * nombre de jours par round**). Sur cette phase-là, les contraintes sont le temps à disposition et la cohérence des dates.

La troisième phase s'occupe de calculer le résultat des ventes de chaque équipe et affiche le classement des équipes. La vision globale du schéma d'exécution du simulateur peut être modélisée de la manière suivante :

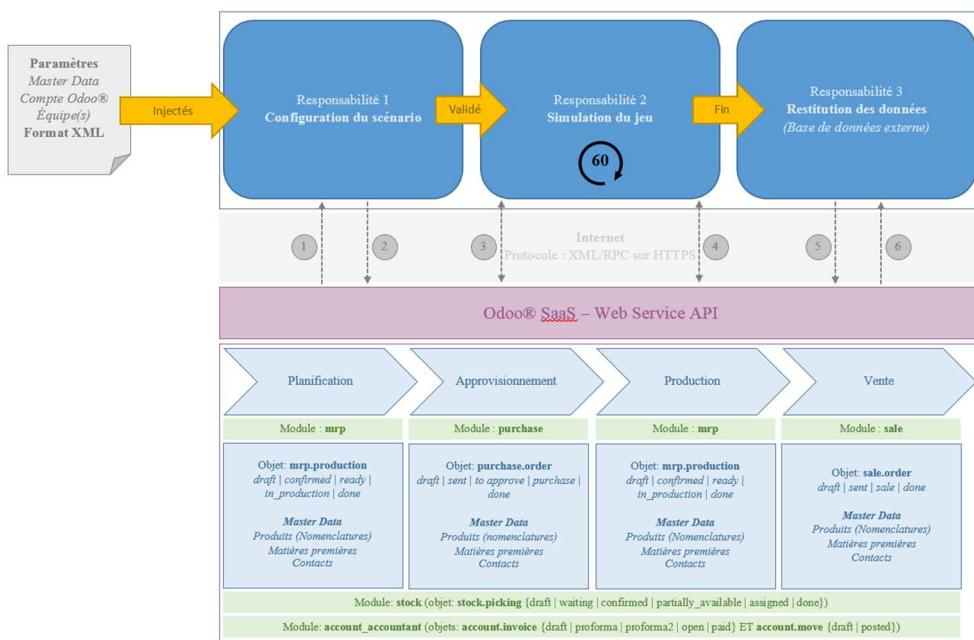


Figure 036 : OdooSIM.

Le fichier de configuration est lu par le simulateur durant la phase 1 (responsabilité 1). Dès que les éléments de la première phase sont tous terminés, la phase 2 démarre et est exécutée autant de fois qu'il y a de jours à jouer. Finalement, la phase 3 s'occupe du classement des équipes par rapport aux chiffres d'affaires.

Ce qui est visualisable sur le schéma ci-dessus, est que chaque phase communique avec Odoo par le biais de l'API. Puis, Odoo gère les processus grâce aux applications installées, aux modèles et aux données du master data.

6.5.1. Fichier de configuration

Le formalisme XML est choisi car il est assez verbeux pour être compris par la plupart des personnes. Ce fichier représente plusieurs éléments importants pour l'exécution du simulateur. Il abrite les données d'accès à l'API et du scénario Brewery & Co. Ainsi, grâce à ce fichier, le simulateur peut se connecter aux instances Odoo et insérer les données du scénario dans le master data de ces dernières. Tout ceci s'effectue bien entendu par l'API.

6.5.2. Phase de configuration

Pour ne pas empiéter sur le temps nécessaire à l'exécution d'un jour simulé, beaucoup d'opérations ne devant être exécutées qu'une seule fois sont faites dans cette phase-là.

Ce choix est motivé par le fait que cette phase peut être amorcée à n'importe quel moment et la durée n'est pas un critère primordial de qualité à ce niveau. En effet, il est tout à fait envisageable de lancer la phase de configuration avant de dispenser une leçon et une fois que les explications sont données et que les équipes sont prêtes, là, lancer la phase 2.

Par conséquent, la phase de configuration exécute les opérations suivantes :

- 1) Elle récupère les informations de simulation à partir du fichier de configuration. Ces informations sont le volume de demande quotidienne, le nombre de rounds à jouer, le nombre de jours par round, le temps d'un jour en seconde et la quantité de produit renouvelé chaque jour.
- 2) Elle récupère les informations de connexions à l'API, à partir du fichier de configuration.
- 3) Elle génère dans son graphe objet les parties prenantes, les sociétés, les produits et le marché.
- 4) Elle persiste les éléments du graphe objet dans les instances Odoo. Puis, elle récupère l'identifiant généré par le PGI et l'ajoute dans les informations en mémoire dans le graphe objet.

Cette technique de créer les éléments à double est la solution trouvée pour diminuer les échanges avec l'API dans la phase de simulation.

Par exemple, pour effectuer une vente dans Odoo, il faut créer un enregistrement dans la table des ventes laquelle possède plusieurs clés étrangères. Si les informations de ces clés étrangères ne sont pas dans le graphe objet, il faut les récupérer avec des requêtes en direction de l'API.

En reprenant l'exemple d'une vente, pour l'insérer dans Odoo, il faut les références vers les identifiants ; du produit vendu, du client qui achète, du compte qui comptabilise la transaction, du vendeur et de son équipe. Sans avoir les informations en mémoire, cela implique d'effectuer 4 requêtes supplémentaires vers l'API pour chaque vente insérée (le vendeur étant celui qui invoque la création, ce qui équivaut à toujours affecter le compte administrateur aux ventes générées).

6.5.3. Modèle objet

Afin de pouvoir enregistrer les informations en mémoire durant l'exécution d'une partie, le modèle objet qui suit est réalisé. Il permet de stocker toutes les informations nécessaires afin d'éviter d'effectuer des requêtes vers l'API trop régulièrement.

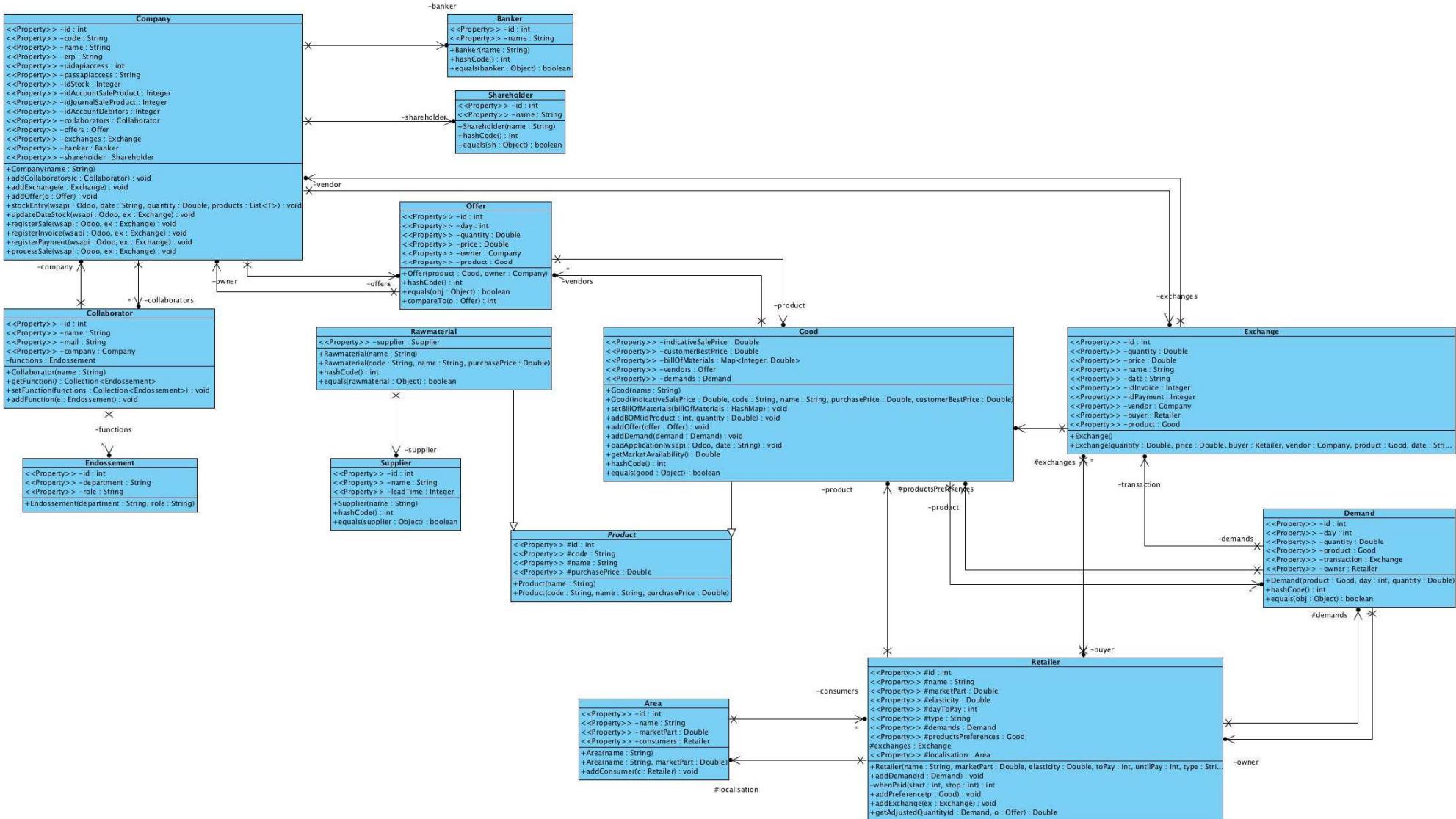


Figure 037 : Modèle du domaine utilisé dans la version proof of concept.

Le concept général de ce modèle est de stocker les informations sur les équipes en jeu dans la classe **Company**. Cette classe possède les informations nécessaires à l'échange avec l'API. Les attributs **erp**, **uidapiaccess** et **passapiaccess** permettent de s'authentifier auprès de l'API. Au niveau des méthodes qu'elle possède, **stockEntry(...)** permet d'effectuer l'autoreconstitution du stock qui survient chaque jour de simulation. La méthode **processSale(...)** permet d'enregistrer une nouvelle vente. Elle appelle dans le bon ordre, les méthodes nécessaires à l'exécution de tout le processus de vente dans Odoo. En terme d'associations, cette classe possède des collaborateurs **Collaborator** qui eux, possèdent une fonction particulière. Elle référence aussi une instance de l'établissement bancaire **Banker** avec qui elle traite et aussi vers son actionnaire **Shareholder**. Afin d'avoir une vision sur les échanges qu'elle effectue sur le marché, une liste de référence vers la classe **Exchange** fait partie de ses attributs.

Ces échanges sont en réalité des ventes. Elle possède la date virtuelle à laquelle elle est réalisée au format **String**, le prix convenu, la référence vers l'identifiant généré par Odoo lors de l'insertion dans l'attribut **idInvoice**. La référence vers le paiement effectué dans Odoo avec l'attribut **idPayment**. Les informations sur le vendeur **Company**, l'acheteur **Retailer** et le produit **Good** sont aussi contenues.

Les revendeurs sont contenus dans la classe **Retailer**. Cette classe possède plusieurs attributs intéressants. Le premier, **name** qui affecte un nom en rapport avec l'information du type de revendeur, lequel est renseigné dans le fichier de configuration XML. **marketPart** qui définit sa part de marché dans sa région **localisation** référence à la classe **Area**. **elasticity** qui permet de stocker l'information relative à sa sensibilité face aux variations de prix. Cet attribut permet de calculer la quantité qu'il achète par rapport à un prix donné qui n'est jamais égal au prix optimal. La méthode qui permet d'ajuster le quantité par rapport aux prix des diverses offres se nomme **getAdjustedQuantity(...)**. Pour revenir sur les attributs, **productPreferences** permet de stocker les produits pour lesquels le revendeur à attrait. Cet attribut permet de faire le lien avec le scénario et les produits commercialisés par les différents types de revendeurs.

Une petite parenthèse sur une subtilité du modèle, la généralisation/spécialisation entre la classe générale **Product** et ses spécialisations **Rawmaterial** et **Good**. Elle est mise en place car au final, qu'un produit soit une matière première ou un article commercialisé, beaucoup d'informations sont communes.

À la base, les deux types d'objet étaient contenus dans une seule liste de **Product** grâce au polymorphisme. Par la suite, cette liste était uniquement servie pour itérer dans les articles commercialisés alors elle a évolué vers une liste de **Good** pour éviter de devoir itérer sur une liste de dizaine d'objets. Cependant, cette subtilité peut ouvrir des horizons grâce à la puissance du monde OO elle est donc maintenue.

6.5.4. Phase de simulation

L'une des contraintes liées à cette phase et que doit gérer le simulateur est de pouvoir réaliser toutes les opérations dites automatiques de toutes les équipes dans une fenêtre de temps cadencée par le paramètre **temps de simulation d'un cycle opérationnel** du scénario.

Cela implique qu'en un maximum inférieur au temps défini pour un cycle opérationnel, toutes les opérations opérationnelles automatisées **SA01**, **SA02**, **SA03** et **SA04** doivent pouvoir être réalisées plus celles visant à générer la demande, l'offre, l'auto approvisionnement des stocks et la mise en œuvre de l'algorithme de sélection de l'offre. Visuellement, cette phase doit gérer ce qui suit :

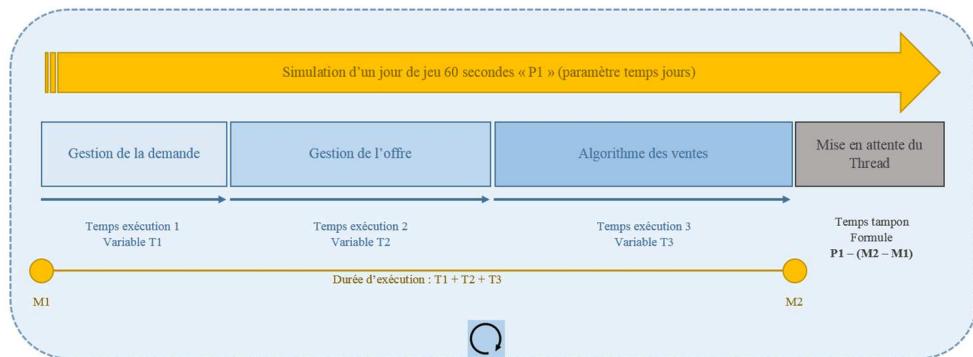


Figure 038 : Zoom sur la phase 2.

Ce schéma présente les étapes que doit pouvoir exécuter le simulateur dans le temps imparti. Idéalement, une pause doit être marquée à la fin des opérations pour remplir le temps nécessaire à la simulation d'un jour. L'étape **Algorithme des ventes** couvre aussi le cycle opérationnel du processus des ventes.

6.5.4.1. Gestion de la demande

La gestion de la demande consiste à la créer. Pour se faire, le simulateur lit dans le fichier de configuration **settings.xml** les données relatives à la constitution du marché.

Ensuite, il itère sur tous les consommateurs précédemment créés dans la phase de configuration et selon leur appartenance à une région et à un type de revendeur il leur attribue une demande par produit qu'ils désirent obtenir. La quantité demandée est calculée à partir des données du fichier de configuration (se référer au chapitre L'algorithme de sélection de l'offre optimale).

6.5.4.2. Gestion de l'offre

L'offre se crée en itérant sur la liste des produits disponibles, c'est-à-dire huit dans le scénario Brewery & Co. Puis pour chaque objet de type société (une instance par société en jeu) une requête vers l'API est envoyée pour récupérer le prix de vente appliqué et la quantité en stock disponible.

Attention, si le prix de vente qu'applique la société est au-dessus du prix plafond (dans le fichier de configuration / nœud **product_sellable** / attribut **maxPrice**) alors l'offre n'est pas prise en considération.

6.5.4.3. Algorithme des ventes

Cette partie débute par ajouter une quantité en stock dans chacune des instances pour chacun des produits.

Ensuite, l'algorithme des ventes effectue une sélection de la meilleure offre pour une demande donnée. Ce qui implique d'itérer dans les demandes et pour chacune d'elle, itérer dans les offres disponibles. Dès que l'offre est identifiée comme étant la plus optimale, le processus de vente se réalise dans l'instance Odoo de l'équipe qui a conclu la vente.

6.5.4.4. Mise en attente du Thread

Le thread principal de l'application est mis en attente le temps d'atteindre le temps imparti pour un jour simulé.

Pour se faire, le temps est enregistré au début de la phase 2 et à la fin lorsque toutes les opérations sont faites. En obtenant la différence entre le délai et la soustraction de la fin avec le début enregistré, le temps d'attente est obtenu. Le thread est mis en pause durant ce laps de temps puis reprend une itération supplémentaire dans le cas où il reste des jours à simuler.

6.5.5. Phase de restitution

Cette phase s'occupe de lire dans la comptabilité des instances en jeu, le chiffre d'affaires générés et affiche une sortie ordonnée sous forme d'un classement des équipes par chiffres d'affaires décroissants.

6.5.6. Démonstration du résultat

Une vidéo démontre le bon fonctionnement du simulateur. Les paramètres de jeu utilisés sont les suivants :

- Nombre d'équipe : **2**.
- Nombre de produit vendu : **8**.
- Prix optimal du produit vendu : **3.80 CHF**.
- Nombre de rounds : **2**.
- Nombre de jour par round : **10**.
- Temps en seconde d'un jour virtuel : **10 secondes**.

Cette vidéo présente le code exécuté dans la phase 1. Puis, le détail d'une itération de la phase 2. Finalement, elle présente les résultats des ventes en

effectuant des variations de prix sur le produit dans les deux instances utilisées.

Visualisation :



La vidéo peut être lue sur YouTube à partir du lien suivant :

<https://www.youtube.com/watch?v=tUvzrxGYt44&feature=youtu.be>

6.6. Limites et améliorations

La lumière est mise sur les limites de cet artefact estampillé « Proof of concept ». Etant donné la durée relativement courte de ce travail, il est difficile d'affirmer que cet artefact logiciel respecte toutes les bonnes pratiques assimilées durant la formation d'informaticien de gestion. En effet, certains choix sont pris pour accélérer le développement, ce qui a pour conséquence d'abandonner certaines parties des tests et personnalisation des messages d'erreurs qui, normalement doivent figurer dans un projet de développement logiciel.

6.6.1. Pour la version finale

Tout le processus des ventes est opérationnel. Cependant, le temps d'attente entre la livraison et le paiement du client n'est pas fait. Il est nécessaire de la mettre en place pour reprendre ce qui est fait actuellement dans la version finale.

6.6.2. Crédit des bases de données

Cette version ne donne pas la possibilité de créer les bases de données automatiquement. Cela est dû au fait que la version Online en mode SaaS ne le permet pas. Par conséquent, avant chaque lancement d'une nouvelle partie sur le simulateur proof of concept, il est nécessaire de créer les instances à partir du portail odoo.com.

6.6.3. Gestion des droits d'accès utilisateurs

Les droits tels qu'ils sont présentés dans le chapitre Droit d'accès aux modèles et données ne sont pas gérés dans cette version. Cette fonctionnalité manquante offre aux utilisateurs de la version des droits d'administration et par conséquent, la possibilité d'effectuer des opérations sur le master data qui mettent en péril la bonne exécution du simulateur. Lorsque les testeurs de cette version l'utilisent, il est important d'effectuer uniquement les manipulations expliquées dans le document **TB.AT.JobAids Brewery and Co.pdf**.

6.6.4. Fichier de configuration

Le fichier **settings.xml** est l'élément sine qua non à la bonne exécution du simulateur.

Dans l'état actuel, ce fichier doit être généré à la main sur la base du modèle fourni. Cependant, ce fichier XML ne possède pas de schéma, ce qui rend sa validation sémantique impossible.

Lors d'une prochaine itération du travail, il est fortement recommandé de créer un schéma qui puisse être garant de la bonne conformité du fichier.

Pour la création de ce fichier, l'idée finale est de disposer d'un IHM dont l'objectif est de générer ce fichier de manière automatique. Par exemple, un portail web avec un formulaire permettant l'édition des produits, des équipes, des marchés, etc. serait une solution viable.

Un autre problème a été soulevé lors des tests réalisés. Les requêtes xPath ne gère pas les apostrophes. Lorsque la valeur d'une balise ou d'un attribut possède une apostrophe, la requête échoue. Pour éviter cela, il est nécessaire de mettre en place un mécanisme qui nettoie les chaînes de caractères.

6.6.5. Messages d'erreurs

Les messages d'erreur ne sont pas logués. Il est recommandé de les traiter à l'aide de la classe `Logger`³⁶.

6.6.6. Game Loop

La Game Loop utilisée pour itérer dans les jours virtuels possède une faiblesse majeure, elle se déroule d'une traite sans pouvoir interagir avec. Les actions comme mettre en pause, reprendre, débuter ou encore stopper la boucle de jeu doivent être implémentées afin de pouvoir agir sur la phase de simulation.

Actuellement, si des questions sont soulevées par les participants, le simulateur ne peut pas être stoppé pour y répondre.

Afin de mettre en place ce mécanisme, il est recommandé de mettre en œuvre un Thread³⁷ spécifique à la boucle de jeu. Ainsi, à partir du Thread principal, les actions d'interactions peuvent être exécutées sur le Thread secondaire.

³⁶ <https://docs.oracle.com/javase/7/docs/api/java/util/logging/Logger.html>

³⁷ <https://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html>

6.7. Réalité comptable

Suite à la première version du proof of concept, une carence grave en comptabilité a été constatée. En effet, les ventes ne se comptabilisent pas correctement.

6.7.1. Constatation

Lorsqu'une vente est créée, la méthode **processSale(...)** de la classe **Company** est appelée. Elle effectue les insertions dans Odoo de manière indépendante les unes des autres alors que lorsqu'on effectue le processus de vente entièrement via l'interface cliente Web, la plupart des éléments, se créent en quelque clics. Le transfert de stock, la facturation (avec toutes les écritures comptables nécessaires) et la conclusion du processus de vente.

Dans l'état actuel de fonctionnement du simulateur, la méthode précitée effectue dans l'ordre chronologique les éléments suivants :

- Création de la vente **sale.order**.
- Création du transfert **stock.picking**, **stock.immediate.transfert**, **stock.pack.operation** et **stock.move**.
- Création d'une facture **account.invoice**, qu'elle rattache par un champ textuelle **origin** à la vente.
- Création d'un paiement **account.payment**.

Ces éléments sont créés indépendamment les uns des autres et par conséquent, le processus de vente et son processus de support, la comptabilité, ne sont pas intégrés totalement. De ce fait, une vente n'est pas correctement comptabilisée et les comptes ne balancent pas correctement.

6.7.2. Démarche de résolution

Pour rétablir le tir, une solution est mise en œuvre. Reprendre le flux de réalisation d'une vente et analyser dans le détail, les interactions du navigateur avec Odoo grâce aux outils de développement.

6.7.3. Solution

Suite à cette démarche, le processus de vente est totalement intégré aux processus de support inventaire et comptabilité. Chaque étape identifiée grâce à la démarche de résolution est réalisée par programmation via l'API dans la nouvelle méthode de la classe **Company** nommée **processQuickSale2()**.

Au niveau du bilan, le compte courant de la société 10201 Bank est aussi débité de bonne manière. Le compte des débiteurs 1100 Débiteurs est lui à zéro vu que toutes les factures sont payées directement.

Complément d'informations :

La nouvelle méthode **processQuickSale2()** ne prend pas en charge les dates simulées au niveau des mouvements de stock **stock.move**. Toutefois, les dates des ventes **sale.order**, des factures **account.invoice** et des paiements **account.payment** sont à jour.



Cette déconvenue est due au fait que cette problématique s'est révélée très tardivement et par conséquent, il n'était plus possible de tout refaire. De plus, les objets liés à un **stock.move** sont trop nombreux.

7. Spécifications OdooSIM

Sur la base du savoir capitalisé durant le développement de la version proof of concept, ce chapitre présente les recommandations hypothétiques pour le développement de la version intégrale.

7.1. Architecture

Pour l'architecture, il est important d'utiliser les trois phases telles qu'utilisées pour la version proof of concept. Cependant, dans la dernière phase, il est nécessaire de persister toutes les données générées (variation des stocks, ventes, achats, etc...) dans une base de données indépendantes.

Le but de cette action de persistance est de pouvoir constituer un entrepôt de données historiques sur les parties jouées. Ainsi, des analyses peuvent être tirées et des leçons apprises sur les événements survenus précédemment.

7.2. Cycle opérationnel

La chaîne de valeur complète de la société Brewery & Co se présente ainsi :

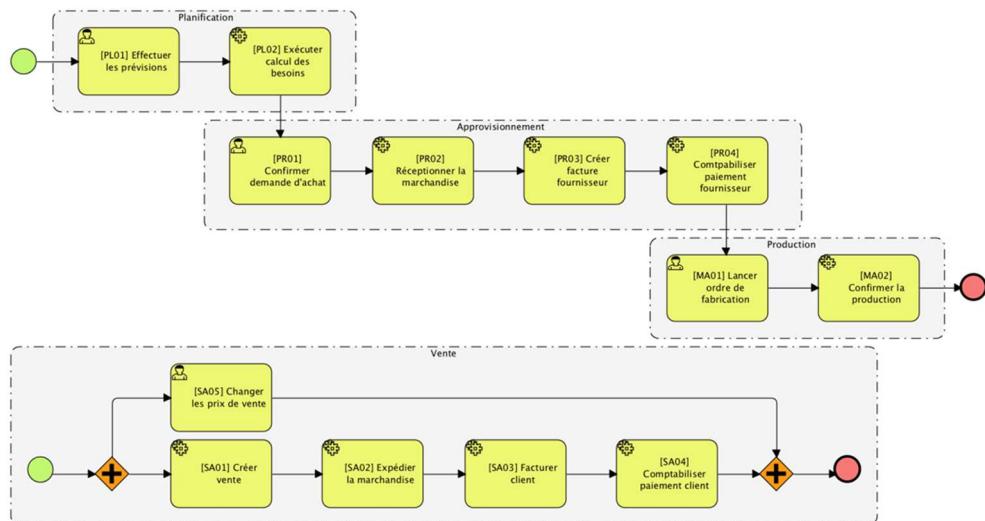


Figure 039 : Chaîne de valeur globale de Brewery & Co. telle qu'elle est mise en œuvre dans Odoo.

7.3. Pistes et recommandations pour les activités automatisées

Comme présenté au chapitre Analyse de l'aspect pédagogique, toutes les activités de la chaîne de valeur Brewery & Co ne sont pas jouées par les participants. D'une part, car elles n'ont pas toutes de la valeur pédagogique et de l'autre, car les participants n'auraient pas l'impression d'être dans un environnement attractif et automatisé.

Ce chapitre passe en revue toutes les activités définies comme étant automatisables par le simulateur.

7.3.1. Activité PL02

Lorsque manuellement le MRP est exécuté, cela crée les appels suivants :

```
▼ Request Payload view source
  ▼ {jsonrpc: "2.0", method: "call",...}
    id: 383651606
    jsonrpc: "2.0"
    method: "call"
    ▶ params: {model: "procurement.orderpoint.compute", method: "create", args: [{}],...}
      ▶ args: [{}]
        ▶ 0: {}
      ▶ kwargs: {context: {lang: "en_US", tz: false, uid: 1, params: {action: 260}}}
        ▶ context: {lang: "en_US", tz: false, uid: 1, params: {action: 260}}
          lang: "en_US"
          ▶ params: {action: 260}
            action: 260
            tz: false
            uid: 1
          method: "create"
          model: "procurement.orderpoint.compute"
```

Figure 040 : Outil de développement du navigateur Google Chrome.

Tout d'abord, la méthode **create** est appelée sur le modèle **procurement.orderpoint.compute**. Puis, la méthode **procure_calculation** est appelée sur l'objet précédemment créé en passant en paramètre son id.

```
▼ Request Payload view source
  ▼ {jsonrpc: "2.0", method: "call",...}
    id: 401631226
    jsonrpc: "2.0"
    method: "call"
    ▶ params: {model: "procurement.orderpoint.compute", method: "procure_calculation", domain_id: null,...}
      ▶ args: [[1], {lang: "en_US", tz: false, uid: 1, params: {action: 260}}]
        ▶ 0: [1]
          ▶ 0: 1
        ▶ 1: {lang: "en_US", tz: false, uid: 1, params: {action: 260}}
          lang: "en_US"
          ▶ params: {action: 260}
            action: 260
            tz: false
            uid: 1
          context_id: 1
          domain_id: null
          method: "procure_calculation"
          model: "procurement.orderpoint.compute"
```

Figure 041 : Outil de développement du navigateur Google Chrome.

7.3.2. Activités PR02, PR03 et PR04

Une fois que les participants ont confirmé les demandes d'achat générées, le simulateur doit réceptionner la marchandise.

Il faut récupérer les ordres d'achat dans le statut **purchase** :

```
▼ Request Payload view source
  ▼ {jsonrpc: "2.0", method: "call", params: {model: "purchase.order",...}, id: 53897030}
    id: 53897030
    jsonrpc: "2.0"
    method: "call"
    ▶ params: {model: "purchase.order",...}
      ▶ context: {lang: "en_US", tz: false, uid: 1, search_default_todo: 1, show_purchase: true, params: {action: 365},...}
        ▶ domain: [{"state": "not in", ["draft", "sent", "bid", "confirmed"]}]
          ▶ 0: {"state": "not in", ["draft", "sent", "bid", "confirmed"]}
        ▶ fields: ["message_unread", "name", "date_order", "partner_id", "company_id", "date_planned", "origin",...]
          limit: 80
          model: "purchase.order"
          offset: 0
          sort: ""
```

Figure 042 : Outil de développement du navigateur Google Chrome.

Ce qui est réalisable grâce aux conditions dans **domain**. La validation de l'entrée en stock génère ceci :

```
▼ Request Payload view source
  ▼ {jsonrpc: "2.0", method: "call",...}
    id: 318751197
    jsonrpc: "2.0"
    method: "call"
    ▶ params: {model: "stock.picking", method: "do_new_transfer", domain_id: null, context_id: 1, args: [[2],...]}
      ▶ args: [[2],...]
        ▶ 0: [2]
        ▶ 1: {lang: "en_US", tz: false, uid: 1, params: {action: 365}, active_model: "purchase.order", active_id: 1,...}
          context_id: 1
          domain_id: null
          method: "do_new_transfer"
          model: "stock.picking"
```

Figure 043 : Outil de développement du navigateur Google Chrome.

Un objet **stock.picking** est généré et la méthode **do_new_transfert** est appelée. Dans l'IHM, de manière manuelle, Odoo demande confirmation que

l'entrée en stock concerne toute la quantité. Les appels pour effectuer ceci sont :

```
▼ Request Payload view source
  ▷ {jsonrpc: "2.0", method: "call",...}
    id: 848835694
    jsonrpc: "2.0"
    method: "call"
    ▷ params: {model: "stock.immediate.transfer", method: "process", domain_id: null, context_id: 1,...}
      ▷ args: [[2], {lang: "en_US", tz: false, uid: 1, params: {action: 365}, active_model: "purchase.order",...}]
        ▷ 0: [2]
        ▷ 1: {lang: "en_US", tz: false, uid: 1, params: {action: 365}, active_model: "purchase.order",...}
          active_id: 1
          ▷ active_ids: [1]
          active_model: "purchase.order"
          lang: "en_US"
          ▷ params: {action: 365}
          search_disable_custom_filters: true
          tz: false
          uid: 1
        context_id: 1
        domain_id: null
        method: "process"
        model: "stock.immediate.transfer"
```

Figure 044 : Outil de développement du navigateur Google Chrome.

L'appel à la méthode **process** sur le modèle **stock.immediate.transfer**. Ensuite, pour générer la facture, il est nécessaire d'appeler la méthode **default_get** du modèle **account.invoice** en passant en paramètre les éléments relatif à la demande d'achat :

```
▼ Request Payload view source
  ▷ {jsonrpc: "2.0", method: "call", params: {model: "account.invoice", method: "default_get",...}}
    id: 737397202
    jsonrpc: "2.0"
    method: "call"
    ▷ params: {model: "account.invoice", method: "default_get",...}
      ▷ args: [{"comment": "", "message_follower_ids": "", "date_due": "", "partner_bank_id": "", "number": "", "company_id": ""}]
      ▷ kwargs: {,...}
        ▷ context: {lang: "en_US", tz: false, uid: 1, params: {action: 365}, active_model: "purchase.order", active_id: 1,...}
          active_id: 1
          ▷ active_ids: [1]
          active_model: "purchase.order"
          default_purchase_id: 1
          lang: "en_US"
          ▷ params: {action: 365}
          search_disable_custom_filters: true
          type: "in_invoice"
          tz: false
          uid: 1
        method: "default_get"
        model: "account.invoice"
```

Figure 045 : Outil de développement du navigateur Google Chrome.

C'est exactement ce qui est fait, **active_id** renseigne l'id de l'**active_model purchase.order**. Une fois que cela est fait, il est nécessaire de créer la facture avec la méthode **create**. En effet actuellement avec **default_get** la facture est simplement générée mais pas persistée, pour le faire :

```
method: "call"
params: {model: "account.invoice", method: "create",...}
args: [{partner_id: 10, reference: false, purchase_id: false, origin: "P000001", date_invoice: false,...}]
  ▷ 0: {partner_id: 10, reference: false, purchase_id: false, origin: "P000001", date_invoice: false,...}
    account_id: 49
    comment: false
    company_id: 1
    currency_id: 6
    date: false
    date_due: false
    date_invoice: false
    fiscal_position_id: false
    ▷ invoice_line_ids: [[0, false, {purchase_id: 1, name: "P000001: Eau traitée", sequence: false, price_unit: 0.15,...}]]
      ▷ 0: [0, false, {purchase_id: 1, name: "P000001: Eau traitée", sequence: false, price_unit: 0.15,...}]
        journal_id: 2
        message_follower_ids: false
        message_ids: false
        name: false
        origin: "P000001"
        partner_bank_id: false
        partner_id: 10
        payment_term_id: false
        purchase_id: false
        reference: false
        ▷ tax_line_ids: [[0, false,...]]
        user_id: 1
      ▷ kwargs: {,...}
      method: "create"
      model: "account.invoice"
```

Figure 046 : Outil de développement du navigateur Google Chrome.

Dès qu'elle est créée, il est nécessaire d'invoquer le workflow du modèle **account.invoice** afin de passer la facture du statut **Draft** à **Open** :

```
▼ Request Payload view source
  {jsonrpc: "2.0", method: "call", params: {model: "account.invoice", id: 1, signal: "invoice_open"},...}
    id: 644492120
    jsonrpc: "2.0"
    method: "call"
    params: {model: "account.invoice", id: 1, signal: "invoice_open"}
      id: 1
      model: "account.invoice"
      signal: "invoice_open"
```

Figure 047 : Outil de développement du navigateur Google Chrome.

L'appel de la fonction RPC **exec_workflow**³⁸ avec comme signal **invoice_open** permet de changer l'état de la facture qui porte l'id numéro 1. Pour enregistrer un paiement, il faut charger une nouvelle instance du modèle **account.payment** en appelant la méthode **default_get** en passant les paramètres qui font le lien avec la bonne instance d'**account.invoice** :

```
▼ Request Payload view source
  {jsonrpc: "2.0", method: "call", params: {model: "account.payment", method: "default_get",...}}
    id: 591204046
    jsonrpc: "2.0"
    method: "call"
    params: {model: "account.payment", method: "default_get",...}
      args: [{"communication", "currency_id", "invoice_ids", "payment_difference", "partner_id",...}]
      kwargs: {,...}
        context: {lang: "en_US", tz: false, uid: 1, params: {action: 365}, active_model: "account.invoice",...}
          active_id: 1
          active_ids: [1]
          active_model: "account.invoice"
          default_invoice_ids: [14, 1, null]
          lang: "en_US"
          params: {action: 365}
          search_disable_custom_filters: true
          type: "in_invoice"
          tz: false
          uid: 1
        method: "default_get"
        model: "account.payment"
```

Figure 048 : Outil de développement du navigateur Google Chrome.

La bonne instance est récupérée en utilisant **active_id**, **active_ids** et **active_model**. Une fois que le paiement est généré, tout comme pour la facture, il faut le comptabiliser. Pour se faire :

```
▼ Request Payload view source
  {jsonrpc: "2.0", method: "call", params: {model: "account.payment", method: "create",...}, id: 878934383}
    id: 878934383
    jsonrpc: "2.0"
    method: "call"
    params: {model: "account.payment", method: "create",...}
    args: [{payment_type: "outbound", partner_type: "supplier", partner_id: 10, journal_id: 6,...}]
      0: {payment_type: "outbound", partner_type: "supplier", partner_id: 10, journal_id: 6,...}
        amount: 187.5
        communication: "BILL/2016/0001"
        currency_id: 6
        journal_id: 6
        partner_id: 10
        partner_type: "supplier"
        payment_date: "2016-07-04"
        payment_difference_handling: "open"
        payment_method_id: 2
        payment_type: "outbound"
        writeoff_account_id: false
      kwargs: {,...}
      method: "create"
      model: "account.payment"
```

Figure 049 : Outil de développement du navigateur Google Chrome.

³⁸ http://www.odoo.com/documentation/9.0/api_integration.html section Workflow manipulations

La méthode **create** est invoquée sur le modèle **account.payment**. Petite remarque supplémentaire, à ce moment-là, l'attribut **payment_date** reçoit la date serveur. Il est important d'effectuer une requête de modification **write** sur cet enregistrement pour mettre à jour la date avec la date simulée. Finalement, le paiement est directement comptabilisé et son statut passe à **Paid** grâce l'appel à la méthode **post** :

```
▼ Request Payload view source
  {jsonrpc: "2.0", method: "call",...}
    id: 114406892
    jsonrpc: "2.0"
    method: "call"
  ▶ params: {model: "account.payment", method: "post", domain_id: null, context_id: 1,...}
    args: [[1], {lang: "en_US", tz: false, uid: 1, params: {action: 365}, active_model: "account.invoice",...}]
      context_id: 1
      domain_id: null
      method: "post"
      model: "account.payment"
```

Figure 050 : Outil de développement du navigateur Google Chrome.

Une fois que la réception marchandise est faite et que la comptabilisation de la transaction financière également, la commande d'achat peut être terminée grâce à l'appel de la méthode **button_done** sur le modèle **purchase.order** :

```
▼ Request Payload view source
  {jsonrpc: "2.0", method: "call",...}
    id: 343248148
    jsonrpc: "2.0"
    method: "call"
  ▶ params: {model: "purchase.order", method: "button_done", domain_id: null, context_id: 1, args: [[1],...]}
    args: [[1],...]
    context_id: 1
    domain_id: null
    method: "button_done"
    model: "purchase.order"
```

Figure 051 : Outil de développement du navigateur Google Chrome.

Dans **args** il faut prendre soin de passer le bon id qui référence la commande d'achat pour laquelle la fin du processus doit être exécutée.

7.3.3. Activité MA02

En résumé, cette activité doit passer les ordres de fabrication ayant atteint le temps nécessaire à la fabrication des bières du statut en cours de production à terminer. Elle doit aussi s'assurer que les participants n'ont pas triché en lançant plus de trois ordres de fabrication simultanément. En effet, ce nombre étant une des contraintes du scénario.

Pour se faire, il faut tout d'abord obtenir la liste des ordres de fabrication dans le statut **in_production** grâce à la méthode **search_read** sur le modèle **mrp.production**. Une fois que cela est réalisé, il est nécessaire de tester si la production a atteint son délai. Si c'est le cas, l'ordre de fabrication doit être terminé. L'implication est que l'OF passe du statut **in_production** à **done**. Pour se faire, il est nécessaire d'appeler la méthode **default_get** sur le modèle **mrp.production.produce** en spécifiant le lien avec l'OF grâce à **active_id**, **active_ids** et **active_model** :

```
▼ Request Payload view source
  ▷ {jsonrpc: "2.0", method: "call", params: {model: "mrp.product.produce", method: "default_get",...},...}
    id: 50616
    jsonrpc: "2.0"
    method: "call"
  ▷ params: {model: "mrp.product.produce", method: "default_get",...}
    ▷ args: [{"lot_id": "tracking", "product_id": "mode", "product_qty": "consume_lines"}]
      ▷ 0: {"lot_id": "tracking", "product_id": "mode", "product_qty": "consume_lines"}
    ▷ kwargs: {}
      ▷ context: {lang: "en_US", tz: false, uid: 1, params: {action: 325}, active_model: "mrp.production", active_id: 1,...}
        active_id: 1
      ▷ active_ids: [1]
        active_model: "mrp.production"
        lang: "en_US"
      ▷ params: {action: 325}
        search_disable_custom_filters: true
        tz: false
        uid: 1
      method: "default_get"
      model: "mrp.product.produce"
```

Figure 052 : Outil de développement du navigateur Google Chrome.

Puis de confirmer le tout, c'est-à-dire tous les mouvements de stock, la sortie des matières premières consommées et l'entrée des produits finis. Pour réaliser cela :

```
▼ Request Payload view source
  ▷ {jsonrpc: "2.0", method: "call", params: {model: "mrp.product.produce", method: "create",...},...}
    id: 263889234
    jsonrpc: "2.0"
    method: "call"
  ▷ params: {model: "mrp.product.produce", method: "create",...}
    ▷ args: [{mode: "consume_produce", product_qty: 5000, product_id: 13, tracking: "none", lot_id: false,...}]
      ▷ 0: {mode: "consume_produce", product_qty: 5000, product_id: 13, tracking: "none", lot_id: false,...}
        ▷ consume_lines: [[0, false, {lot_id: false, product_id: 1, product_qty: 1250}],...]
          ▷ 0: [0, false, {lot_id: false, product_id: 1, product_qty: 1250}]
            0: 0
            1: false
          ▷ 2: {lot_id: false, product_id: 1, product_qty: 1250}
            lot_id: false
            product_id: 1
            product_qty: 1250
          ▷ 1: [0, false, {lot_id: false, product_id: 2, product_qty: 5}]
          ▷ 2: [0, false, {lot_id: false, product_id: 4, product_qty: 165}]
          ▷ 3: [0, false, {lot_id: false, product_id: 7, product_qty: 10}]
          ▷ 4: [0, false, {lot_id: false, product_id: 12, product_qty: 5000}]
            lot_id: false
            mode: "consume_produce"
            product_id: 13
            product_qty: 5000
            tracking: "none"
        ▷ kwargs: {}
        method: "create"
      model: "mrp.product.produce"
```

Figure 053 : Outil de développement du navigateur Google Chrome.

Il est nécessaire d'appeler la méthode **create** sur le modèle **mrp.product.produce** préalablement obtenu avec **default_get**. En regardant de plus près les paramètres, le constat est que **consume_lines** informe sur les matières premières employées et **product_id** et **product_qty** sur le produit fini et son nombre fabriqué.

Finalement, terminer le processus grâce à la méthode **do_produce** sur le modèle **mrp.product.produce** en passant en paramètre l'id de l'instance préalablement créée grâce à **create** :

```
▼ Request Payload view source
  {jsonrpc: "2.0", method: "call",...}
    id: 284475031
    jsonrpc: "2.0"
    method: "call"
    ▶ params: {model: "mrp.product.produce", method: "do_produce", domain_id: null, context_id: 1, args: [[1],...]}
      ▶ args: [[1],...]
        ▷ 0: [1]
        ▷ 1: {lang: "en_US", tz: false, uid: 1, params: {action: 325}, active_model: "mrp.production", active_id: 1,...}
          active_id: 1
          active_ids: [1]
          active_model: "mrp.production"
          lang: "en_US"
          params: {action: 325}
          search_disable_custom_filters: true
          tz: false
          uid: 1
        context_id: 1
        domain_id: null
        method: "do_produce"
        model: "mrp.product.produce"
```

Figure 054 : Outil de développement du navigateur Google Chrome.

À ce moment-là, l'OF **mrp.production** est dans l'état **done**.

7.3.4. Limites et recommandations

Afin de donner de la pertinence aux opérations, il est nécessaire de comparer toutes ces étapes à celles du chapitre Guide des opérations. En effectuant cela, d'une part les appels à l'API pour chacune des étapes sont connus et de l'autre, les objets créés sont aussi identifiés. Il ne reste qu'à effectuer la mise à jour des dates pertinentes pour que les analyses côté participants puissent être fiables et parlantes pour les joueurs.

8. Analyse des performances

Un des points critiques se situe au niveau du temps à disposition pour effectuer les opérations lors d'un jour virtuel. Ce chapitre donne des pistes sur les temps nécessaires à l'exécution d'un jour virtuel dans le jeu OdooSIM.

8.1. Mesure temps des I/O

Cette analyse calcule les temps en millisecondes nécessaires au simulateur pour effectuer des opérations sur les données d'**Odoo Online 9.0 SaaS** via l'API. Elle porte sur les opérations basiques CRUD. Le modèle utilisé est **res.partner**. Chaque opération est mesurée unitairement et une moyenne est calculée sur la base de dix répétitions successives. Cette batterie de test est effectuée cinq fois dans trois conditions différentes :

- **C1** – Au domicile de Monsieur Anthony Tomat (wifi)
 - **C2** – Mêmes conditions que C1 en passant par le VPN de l'école
 - **C3** – En 4G (100 Mbit/s) grâce à l'opérateur mobile Suisse Sunrise

Voici les résultats obtenus :

Ce qu'il faut retenir c'est qu'une transaction, qu'importe son type, prend en moyenne un tiers de secondes. Cela signifie que la recommandation de diminuer les échanges est justifiée et doit être prise au sérieux afin de rester dans des temps d'exécution abordables.

8.1.1. Alternative

Cette alternative repose sur les mêmes tests. Cependant, ils sont effectués sur une version **Odoo Community 8.0.0-0109** installée sur un serveur NAS Synology connecté au réseau local par une connexion de 100Mbit/s.



Spécifications techniques du serveur :

Modèle DS213j avec processeur mono cœur 1.2 Ghz et 512 Mo de mémoire vive.

Les résultats obtenus sont les suivants :

Opération <i>Méthode API</i>	Moyenne des 10 opérations successives en millisecondes						Moyenne
	Type	T1	T2	T3	T4	T5	
Insertion <i>create</i>	Filaire	971	986	974	988	983	980.40
	Wifi	1037	1038	1057	1058	1038	1045.60
Mise à jour <i>write</i>	Filaire	142	139	154	140	140	143.00
	Wifi	148	151	150	150	149	149.60
Lecture <i>search_read</i>	Filaire	517	501	503	503	509	506.60
	Wifi	609	563	545	587	608	582.40
Suppression <i>unlink</i>	Filaire	174	168	158	155	158	162.60
	Wifi	169	170	168	164	166	167.40
Moyenne d'une transaction tous types confondus :						467.20	

Cette version montre que les transactions avec l'API ne sont pas plus performantes sur un réseau local qu'avec la version du SaaS. Cependant, il est important de spécifier que la configuration du serveur qui exécute Odoo **Community** possède des capacités de traitements réduites.

Ci-dessous, le résultat de la console de monitoring des ressources du serveur Synology lorsque les tests sont effectués :



Figure 055 : Moniteur des ressources DSM 6.0-7321 Update 2.

L'impact des opérations est visualisable facilement. Les flèches bleues foncées mettent en évidence le moment où les échanges sont opérés. Le processeur est très impacté puisqu'il tourne à plus de 90% de ses capacités. Les accès au disque sont eux aussi augmentés. La mémoire n'est quant à elle que peu impactée par les opérations.

Il faut porter les mêmes tests sur un environnement serveur réel afin de pouvoir comparer la version **Online** avec la **Community**. En effet, lorsque les requêtes sont passées au travers du SaaS, les serveurs qui répondent sont des infrastructures réparties avec de gros potentiels de traitement ce qui n'est malheureusement pas le cas du serveur employé pour ce comparatif.

8.2. Calcul théorique du temps d'exécution

Cette analyse donne un ordre de grandeur en temps d'exécution que le simulateur doit prévoir pour effectuer les activités de la chaîne de valeur. Pour le calcul, les données obtenues dans l'analyse faite au chapitre Mesure temps des I/O sont utilisées.

Les deux parties de la chaîne de valeur que sont la logistique et la vente sont analysées séparément. En effet, le nombre de transactions dans la partie des ventes est sensiblement affecté par les paramètres de volumétrie passés au simulateur.



Complément d'informations :

Certaines méthodes sont spécifiques et pas testées unitairement. Pour les comptabiliser, la moyenne générale est utilisée.

8.2.1. Partie logistique

Il faut compter 1 transaction de type insertion et 1 de type autre pour l'activité PL02. Pour le processus des achats, il faut compter 1 transaction de lecture principale plus 10 transactions (2 insertions / 1 māj / 7 autres) par demande d'achat pour lesquelles il faut terminer le processus. L'activité MA02 génère 1 première transaction de lecture pour récupérer les bons ordres de fabrication plus 3 (1 insertion / 2 autres) autres transactions destinées à terminer le processus de fabrication.



Hypothèse :

5 sociétés en compétition ayant pour un jour donné 5 demandes d'achat dont le processus doit se terminer et 3 ordres de fabrication pour lesquels le processus de fabrication est arrivé à terme.

En se basant sur l'hypothèse précédente, chaque société génère 63 transactions avec l'API. En détail, cela donne 14 insertions, 2 lectures, 5 mises à jour et 42 transactions d'un type spécial.

En se référant au temps moyen par type de transaction, les temps obtenus sont :

- 14 insertion de 399.94 millisecondes = 5 secondes.
- 2 lectures de 341.00 millisecondes = 1 secondes.
- 5 mises à jour de 157.20 millisecondes = 1 secondes.
- 42 autres de 269.24 millisecondes = 11 secondes.

Le temps nécessaire à chaque société est d'environ 18 secondes pour effectuer les transactions nécessaires à l'exécution de la partie logistique.

8.2.2. Partie des ventes

En se basant sur les résultats obtenus dans le proof of concept, le nombre de transactions nécessaires à l'exécution du processus de vente entier (sans l'étape manuelle SA05) est de 13 transactions. En détail, cela se partage en 3 insertions, 1 lectures, 1 mise à jour et 8 d'un type spécial.



Hypothèse :

5 sociétés en compétition qui réalisent 10 ventes quotidiennes.

Chaque société génère 30 insertions, 10 lectures, 10 mises à jour et 80 autres, pour un total de 130 transactions. Le temps nécessaire à l'exécution de ces transactions prend :

- 30 insertions de 399.94 millisecondes, soit 12 secondes.
- 10 lectures de 341.00 millisecondes, soit 3 secondes.
- 10 mises à jour de 157.20 millisecondes, soit 1 seconde.
- 80 autres de 269.24 millisecondes = 22 secondes.

Le résultat final pour la partie des ventes selon l'hypothèse préalable est que le temps nécessaire à l'exécution de ces processus de vente est de 38 secondes.

8.2.3. Vérification

Le chapitre Partie des ventes démontre que le temps théorique que met le simulateur à réaliser une exécution du processus de vente est de 4 secondes.

Il est important de vérifier cette affirmation par la pratique en effectuant une itération du processus de vente dans la version proof of concept du simulateur. Cette itération est effectuée 13 fois afin de calculer une moyenne permettant de mesurer l'écart entre la mesure réelle et le temps théorique.

N° d'exécution	Temps mesuré	Moyenne
1	5367	
2	3839	
3	4302	
4	4765	
5	3060	
6	4025	
7	3966	4'121 millisecondes
8	3900	
9	4895	
10	4303	
11	3312	
12	3968	
13	3872	

Le résultat théorique coïncide de fait avec les observations réelles. Il est envisageable de calculer l'exécution théorique d'une partie de jeu sur la base de ces informations.



Complément d'informations :

L'analyse est réalisée dans les conditions de **C1** (référence au chapitre Mesure temps des I/O).

8.3. Projection

En prenant les temps théoriques calculés dans les deux précédents chapitres, une projection dans l'exécution d'un jour virtuel peut être faite. Ceci, afin de calculer le temps que prend la réalisation de toutes les étapes automatisées de la chaîne de valeur Brewery & Co lors de la simulation d'un jour virtuel. Les variables se basent sur les mêmes hypothèses que pour les calculs précédents :

- Partie logistique : 5 sociétés, 5 OA et 3 OF.
- Partie vente : 5 sociétés qui réalisent 10 ventes.

Le détail du temps nécessaire au simulateur pour performer les opérations est le suivant :

- Environ 1 minute 30 secondes ($5 \times 18''$).
- 50 ventes qui durent unitairement 4 secondes, le temps nécessaire est de 3 minutes et 30 secondes.

Théoriquement, un jour virtuel peut être effectué en 5 minutes. Tout en prenant en considération que cinq équipes sont présentent sur le marché.

8.4. Limites et recommandations

Il est nécessaire de trouver une solution à la diminution du nombre d'échange. En effet, dans ces conditions, beaucoup de transactions sont déjà effectuées sans même mettre à jour les champs dates qui font du sens pour les participants. C'est donc bien plus de temps qu'il faut employer pour jouer un jour virtuel.

9. Développement de l'algorithme de choix de l'offre

Il a déjà été présenté mais la version proof of concept met en œuvre un algorithme qui rend la sélection de l'offre plus intelligente et sensible à la variation des prix que les sociétés en jeu effectuent.

9.1. Démarche

Pour que cela soit réalisable, il faut tout d'abord comprendre ce mécanisme d'élasticité. Une fois assimilé, il faut prendre connaissance de la formule mathématique permettant d'obtenir le résultat. Suite à cela, il est nécessaire de l'appliquer au niveau de l'algorithme de sélection de l'offre adéquate pour un consommateur virtuel donné.

9.2. L'algorithme de sélection de l'offre optimale

Cet algorithme sélectionne parmi toutes les offres disponibles sur le marché, la meilleure d'un point de vue consommateur.

Il effectue cette sélection de la manière suivante :

- 1) Il mélange toutes les offres et les tris par prix ascendants. En cas de prix égaux, il trie sur la quantité disponible en stock.
- 2) Il choisit celle qui se situe en tête de liste et ajuste la quantité demandée grâce au mécanisme de l'élasticité prix-demande.
- 3) Il effectue la transaction entre l'offreur et le demandeur.
- 4) Si la demande n'est pas satisfaite quantitativement, l'algorithme passe à la deuxième offre et ainsi de suite jusqu'à ce que la demande soit complètement satisfaite.

La formule mathématique qui est employée l'élasticité prix est la suivante :

$$\text{élasticité prix } (e) = \frac{\text{variation de la consommation}}{\text{variation du prix}}$$

Cette formule utilise des variations. Pour les calculer, il est nécessaire d'avoir deux mesures. La variation se calcule de cette manière :

$$\text{variation } (v) = \left(\frac{\text{prix}A - \text{prix}B}{\text{prix}A} \right) * 100$$

9.3. Variation de la consommation

La première mesure est celle qui est définie au niveau de chaque consommateur que le simulateur créé virtuellement. Cette mesure qui se traduit par une demande quantitative, est calculée en fonction des appartenances que possède un consommateur virtuel à une région et un type de revendeur (lesquels sont définis dans le scénario Brewery & Co). Cette quantité se calcule par le simulateur dans la phase de configuration lorsqu'il génère les consommateurs. Le calcul se base sur des informations qui se trouvent dans le fichier de configuration du scénario, **settings.xml** :

- Le volume total de demandes quotidiennes toutes confondues qui se situe dans la balise **markets** et dans l'attribut **baseVolumDemandDay**. Ce volume correspond à un produit.
- La part de marché de la région où se situe le consommateur. Cette valeur se retrouve dans la balise **market** et son attribut **part**.
- La part de marché du type de revendeur qui catégorise le consommateur. Cette valeur se trouve dans la balise **retailer** et son attribut **part**.
- Le nombre de consommateur ayant les mêmes caractéristiques. Cette valeur se trouve dans la balise **retailer** et son attribut **number**.
- Le nombre de produit (la variable est nommée **nbPref**) pour lesquels le consommateur possède des préférences. Cette information est donnée par le nombre de balise **product** enfant de la balise **preferences** dont le père est le consommateur.

Afin d'illustrer ces explications, voici la structure du fichier XML et les données qui viennent d'être présentées :

```

308   <markets baseVolumDemandDay="100000">
309     <market part="30.00">
310       <name>Suisse Est</name>
311       <retailers>
312         <retailer type="groceries" number="8" payMin="5" payMax="10" part="20.00" elasticity="-1.00">
313           <name>Détailleur N°</name>
314           <preferences>
315             <product>
316               <code>BW01A</code>
317             </product>
318             <product>
319               <code>BW03A</code>
320             </product>
321             <product>
322               <code>BW04A</code>
323             </product>
324           </preferences>
325         </retailer>
326         <retailer type="hypermarket" number="15" payMin="15" payMax="15" part="55.00" elasticity="-0.20">
327           <name>Supermarché N°</name>
328           <preferences>
329             <product>
330               <code>BW01B</code>
331             </product>
332             <product>
333               <code>BW02B</code>
334             </product>
335             <product>
336               <code>BW03B</code>
337             </product>
338             <product>
339               <code>BW04B</code>
340             </product>
341           </preferences>
342         </retailer>
343         <retailer type="other" number="27" payMin="8" payMax="12" part="25.00" elasticity="-3.5">
344           <name>Etablissement public N°</name>
345           <preferences>
346             <product>
347               <code>BW01A</code>
348             </product>
349             </preferences>
350           </retailer>
351       </retailers>
352     </market>

```

Figure 056 : Définition du marché Suisse Est dans le fichier de configuration settings.xml.

À la génération des consommateurs, le simulateur calcule la demande quantitative du consommateur en effectuant le calcul suivant :

$$\text{Demande consommateur} = \frac{\left(\frac{\text{baseVolumDemandDay}}{100} * \text{marketPart} \right)}{\left(\frac{\text{retailerNumber}}{100} * \text{retailPart} \right) / \text{nbPref}}$$

La deuxième mesure est celle que doit calculée le simulateur lorsqu'il met en œuvre l'algorithme de sélection de l'offre. Elle dépend des différents prix auxquels les marchandises se négocient sur le marché à un instant T de la simulation.

9.4. Variation du prix

La première mesure permettant de calculer la variation du prix est obtenue à partir du fichier **settings.xml**. Elle est définie comme étant le prix idéal d'un point de vue consommateur. Cette information se trouve sur chacun des produits, noeud **product_sellable** attribut **bestPrice**, que les sociétés commercialisent.

```
198 <product_sellable type="product" default_code="BW02B" list_price="2.00" standard_price="2.00" bestPrice="3.50">
```

Figure 057 : Définition du produit BW02B.

Complément d'informations :

Cette information du noeud **product_sellable** attribut **bestPrice** est celle que doivent trouver les participants ou plutôt vers laquelle, ils doivent se rapprocher pour maximiser leurs ventes.



L'attribut **maxPrice** du noeud **product_sellable** permet de donner un plafond au prix de vente d'un certain produit. En effet, sans plafond et en admettant que la demande est très élevée quantitativement par rapport à l'offre, même en appliquant la formule d'élasticité prix-demande, certains vendeurs peuvent vendre à des prix très hauts. Le but de cette valeur maximale est de limiter cet effet-là.

La deuxième mesure s'obtient en parcourant l'offre. C'est-à-dire en allant requérer les instances Odoo en jeu et en récupérant les prix appliqués sur les produits.

9.5. Élasticité prix

L'élasticité prix est donnée pour chaque type de consommateur, de revendeur. Elle se trouve aussi dans le fichier de configuration **settings.xml** du scénario Brewery & Co. Pour la récupérer, il est nécessaire d'atteindre le noeud **retailer** et l'attribut **elasticity**.

9.6. Calcul final de la demande adaptée au prix du marché

Il est nécessaire de modifier la formule de base afin de prendre en compte les inconnues du simulateur. La variation du prix est calculable, l'élasticité est connue, il reste à trouver une des deux mesures de la variation de la demande, celle de la nouvelle quantité que demande le consommateur par rapport aux prix du marché :

$$\left(\frac{(\Delta \text{Prix} * 100) * \text{élasticité}}{\text{demande}} \right) * \text{demande} - \text{demande} * -1$$

9.7. Limites et recommandations

La vidéo de présentation des résultats du proof of concept donne de bonnes indications sur les limites et les recommandations à prendre en considération sur le fonctionnement de cet algorithme.



Complément d'informations :

Disponible sur YouTube :

<https://www.youtube.com/watch?v=tUvzrxGYt44&feature=youtu.be>

10. Analyse de l'aspect pédagogique

Pour les besoins du jeu sérieux, les activités présentées dans les chapitres précédents sont catégorisées de deux manières différentes :

- **Automatique** : C'est-à-dire jouée par le simulateur.
- **Manuelle** : C'est-à-dire exécutées par les participants.

Cela est indispensable pour donner l'impression aux participants que le scénario évolue de manière automne. De plus, le jeu serait perçu comme étant ennuyant si les participants avaient à effectuer eux-mêmes la réalisation de tous les processus et leurs activités. Le jeu serait alors uniquement conçu pour créer des ventes et marquer des temps d'arrêt entre les opérations avec délai. Comme cela est le cas pour l'attente des matières premières ou encore l'attente sur la production.

10.1. Démarche

La démarche consiste à récupérer la liste de toutes les activités nécessaires au pilotage de Brewery & Co. et des analysées avec les critères qui doivent être préalablement établis.

10.2. Critères de sélection

Ce choix doit être fait sur la base de critères compréhensibles et reproductibles par d'autres personnes. De ce fait, le choix s'est porté sur deux critères simples et subjectifs.

Considérations :



L'auteur ne possède aucune compétence dans le domaine de la pédagogie. C'est pourquoi, les critères sont qualifiés comme subjectifs. Ils ont été déterminés par pure logique et ressenti. Cependant, ils semblent être pertinents.

Finalement, les activités sont catégorisées grâce aux critères qui suivent :

- Si l'activité est simple et répétitive, sans besoin d'analyse pour son exécution alors l'hypothèse est qu'elle n'apporte pas de plus-value pédagogique. Elle est donc automatisée par le simulateur.
- Si l'activité nécessite une analyse ou une réflexion alors l'hypothèse est qu'elle a du sens dans l'apprentissage et est de ce fait catégorisée comme étant manuelle. Elle se joue sur la base d'une prise de décision que doivent faire les participants.
- C'est grâce à ces activités que les participants font des erreurs. Par conséquent, elles représentent la plus-value de la solution.

Complément d'informations :



Les décisions que font les participants ne sont pas toujours justifiées. De ce fait, ils sont confrontés à leurs erreurs et peuvent en tirer des leçons. C'est grâce à ce mécanisme que les jeux sérieux sont très bénéfiques pour l'apprentissage.

Toutefois, les recommandations font état qu'une validation de ces choix par des personnes qualifiées en pédagogie serait une réelle plus-value pour OdooSIM.

10.3. Résultats

Présenter toutes les étapes et montrer les résultats de l'analyse selon les critères.

Activité	Responsable	Explications
PL01 Effectuer les prévisions	Participants	Prévoir les ventes et donc le besoin de production semble être un exercice important dans l'apprentissage de la gestion d'entreprise. Si les données de ventes ou l'étude de marché sont

		bien faites, les prévisions sont un des facteurs clés de succès. Les prévisions peuvent s'appuyer sur les données de jeu historique ou sur les supports liés au scénario.
PL02 Exécuter le calcul des besoins	Simulateur	Cette activité ne semble pas apporter de plus-value pédagogique. De plus, une problématique technique s'est posée quant aux éléments métiers (OF et OA) qui sont créés. En effet, dès que le MRP crée les besoins, il les date au jour réel alors que le simulateur joue des jours fictifs. Ceci dit, un problème de cohérence survient. En rendant cette tâche automatique, le simulateur exécute le calcul et récupère directement les besoins générés qu'il date au jour fictif. Ainsi, dès que les participants visualisent les OA et OF, ils les identifient comme étant cohérent avec les dates des jours simulés.
PR01 Confirmer les demandes d'achat	Participants	Cette opération doit être effectuée par les participants. En effet, elle suscite une interrogation. « Avons-nous les liquidités pour effectuer l'achat ? » ou encore comment se projeter dans un court terme afin d'évaluer la trésorerie future.
PR02 Réceptionner la marchandise	Simulateur	Cette tâche est simple et répétitive. En effet, le scénario ne prévoit pas de prendre en compte les risques de livraison et les erreurs qui peuvent survenir dans les quantités. Par conséquent, cette tâche est amenée à être exécutée à chaque fois de la même manière.
PR03 Créer la facture fournisseur	Simulateur	Une nouvelle fois, cette tâche est relativement simple. Dans la réalité, cela s'effectue par une simple saisie. De cette observation, l'hypothèse fait office que les participants ne tirent pas de leçon à l'exécuter eux-mêmes.
PR04	Simulateur	A la fin du mois, le département comptabilité paie ses créanciers. Cela

Comptabiliser le paiement fournisseur		n'étant pas une tâche qui nécessite analyse ou réflexion, elle est la candidate parfaite à l'automatisation.
MA01 Lancer des ordres de fabrication	Participants	Le lancement des ordres de fabrication doit s'effectuer en analysant ce qui se passe actuellement sur les lignes. « Avons-nous déjà atteint notre nombre simultané d'OF en cours ? » ou encore « Comment priorisons-nous les OF ? ». Toutes ces questions déclenchent réflexions et éléments stratégiques où les participants doivent faire des choix, prendre des décisions.
MA02 Confirmer la production	Simulateur	La confirmation de la production prend en compte le délai que le scénario impose en tant que contrainte. Pour ne pas donner la possibilité aux participants d'effectuer des tricheries, la décision est de rendre cette étape automatique. De plus, le scénario ne prévoit pas d'ajouter des contraintes comme des problèmes de fabrication ou des retards.

Complément d'informations :

Ce point a été présenté précédemment dans ce document. Cependant, au moment de passer en revue les résultats liés au processus de vente, il semble important de rappeler qu'un jeu sérieux se doit de donner l'impression aux participants, qu'il évolue et qu'il réagit aux décisions et aux actions qu'ils prennent. De ce constat, le processus de vente est le candidat parfait pour rendre le jeu fluide. Comme ce processus vient en bout de chaîne, il est le générateur de revenu pour la société. Toutes les décisions prisent en amont affectent les ventes. C'est une caractéristique propre au domaine étudié. De plus, l'objectif d'une partie OdooSIM est d'atteindre le plus haut bénéfice possible. C'est pour cela, que ce processus est majoritairement automatisé.

SA01 Créer une vente	Simulateur	La concrétisation d'une vente par une équipe est le résultat de plusieurs décisions. C'est donc tout naturellement que cette tâche est automatisée car elle est le résultat de plusieurs critères que doit remplir l'offre que créée les sociétés.
-------------------------	------------	--

SA02 Expédier la marchandise	Simulateur	Tout comme pour la réception d'une commande fournisseur, cette étape n'apporte aucune plus-value pédagogique.
SA03 Facturer le client	Simulateur	L'employé qui facture le client ne fait que de se référer à la vente réalisée et aux conditions négociées. C'est une simple tâche de saisie.
SA04 Comptabiliser le paiement	Simulateur	A nouveau, l'employé qui effectue cette tâche n'a qu'à se référer aux relevés des comptes et changer le statut d'une facture ouverte à payer.
SA05 Changer les prix de vente	Participants	Lors de la présentation de la tâche SA01, il a été fait mention que plusieurs critères devaient être présents pour qu'une société effectue une vente. Les prix des produits en est un. C'est donc une analyse de justesse qu'il faut effectuer pour statuer des prix que l'on désire pratiquer sur le marché. Cette décision affecte directement le volume de vente réalisé. C'est pour cela que cette tâche est à charge des participants.

Le constat de ces résultats est que la majeure partie des opérations s'effectuent par le simulateur. Seules les opérations où un impact sur les connaissances des participants semble être mesurable sont rendus manuelles.

Complément d'informations :

i Les activités sont identifiables par leur code. Pour visualiser comment elles s'effectuent dans Odoo et les impacts qu'elles ont sur les éléments du PGI, il est nécessaire de se référer au chapitre **Guides des opérations**.

10.4. Limites et recommandations

Les résultats obtenus sont faits à partir de critères purement subjectifs. Il est nécessaire de faire valider ces choix par des personnes qualifiées.

11. Analyse des objets manipulés

Limites de l'analyse :



Ce chapitre a été rédigé bien avant de trouver la démarche exposée au chapitre Démarche de résolution et qui permet d'avoir une vue totale des modèles employés à chaque étape. Par conséquent, l'analyse qui suit constitue un bon complément.

Toutes les opérations effectuées au chapitre Guide des opérations donnent lieu à la constitution d'un référentiel d'éléments Odoo manipulés au travers des différentes activités. Ce référentiel constitue une base solide sur laquelle s'appuyer pour développer le simulateur.

La matrice ci-dessous présente pour chaque activité de la chaîne de valeur Brewery & Co., le ou les modèles créés :

Activité	Modèles créés Odoo
PL01 : Effectuer prévisions	<ul style="list-style-type: none"> • stock.warehouse.orderpoint
PL02 Calculer besoins	<ul style="list-style-type: none"> • procurement.orderpoint.compute • procurement.order • purchase.order • purchase.order.line • mrp.production • stock.move
PR01 Confirmer OA	<ul style="list-style-type: none"> • stock.picking • stock.pack.operation
PR02 Réceptionner marchandise	<ul style="list-style-type: none"> • stock.immediate.transfer • stock.quant • account.move • account.move.line
PR03 Créer facture fournisseur	<ul style="list-style-type: none"> • account.invoice • account.invoice.line • account.move • account.move.line
PR04 Comptabiliser paiement fournisseur	<ul style="list-style-type: none"> • account.move • account.move.line • account.payment
MA01 Lancer OF	<ul style="list-style-type: none"> • Aucun

MA02 Confirmer OF	<ul style="list-style-type: none"> • mrp.product.produce • mrp.product.produce.line
SA01 Créer vente	<ul style="list-style-type: none"> • sale.order • sale.order.line • stock.picking • stock.pack.operation • stock.move
SA02 Expédier marchandise	<ul style="list-style-type: none"> • stock.immediate.transfer • stock.quant • account.move • account.move.line
SA03 Facturer client	<ul style="list-style-type: none"> • account.invoice • account.invoice.line • account.move • account.move.line
SA04 Comptabiliser paiement	<ul style="list-style-type: none"> • account.move • account.move.line • account.payment
SA05 Changer prix de vente	<ul style="list-style-type: none"> • aucun

Considération :

Cette matrice a été générée aux mieux. Cependant, il serait prétentieux d'affirmer qu'elle soit exhaustive. D'autres éléments peuvent être créés dans les méandres d'Odoo.

Pour davantage de détail, les paramètres techniques accessibles via l'application **Settings / Technical / Database Structure / Models** peuvent être d'une enrichissante aide. Par exemple, pour obtenir des informations en détail sur le modèle **stock.immediate.transfer**, taper le nom du modèle dans la barre de recherche, ouvrir son détail en cliquant dessus et utiliser le menu **Print / Model Overview**. Le détail du modèle se télécharge au format PDF :



Object: stock.immediate.transfer Name: Immediate Transfer					Type: Base Object Transient: True Apps: stock			
Security								
Group	R	W	C	U	Name			
Fields								
Name	Label	Type	Attribute		Rq	Ro	Tr	Idx
__last_update	Last Modified on	datetime			X			stock
create_date	Created on	datetime						stock
create_uid	Created by	many2one	- relation = res.users					stock
display_name	Display Name	char			X			stock
id	ID	integer			X			stock
pick_id	Pick id	many2one	- relation = stock.picking					stock
write_date	Last Updated on	datetime						stock
write_uid	Last Updated by	many2one	- relation = res.users					stock
Views								
Seq	Type	Name	XML ID		Inherited			
16	Form	view_immediate_transfer	stock.view_immediate_transfer					

12. Analyse de l'aspect sécurité

Ce chapitre présente les éléments à mettre en œuvre pour restreindre le champ d'action des participants dans le PGI. Tout d'abord, chaque application est listée et pour chacune d'elle, des décisions sont prises par rapport à : Est-ce que les participants peuvent y accéder ou pas et si oui, une présentation des éléments à modifier est proposée.

Les recommandations sur les droits à attribuer sur les différents modèles et données sont présentées. Les modèles analysés sont ceux listés au chapitre Analyse des objets manipulés et les données, celles du master data relatives au scénario.

Complément d'informations :



Les droits utilisés sont ceux qu'Odoo emploie. Ils sont codifiés de la manière suivante : Read Access (**RAS**), Write Access (**WAS**), Create Access (**CAS**) et Delete Access (**DAS**).

Finalement, les investigations effectuées durant ce travail sont présentées dans la dernière partie de ce chapitre.

12.1. Démarche

En se basant sur le référentiel obtenu au chapitre Analyse des objets manipulés et aux responsabilités qu'incombe la réalisation des diverses activités de la chaîne globale, définir les droits pour chacun des éléments métiers et les accès aux applications installées.

12.2. Application Contacts

L'accès à cette application durant la partie ne doit pas être donné. En effet, les participants n'ont pas à modifier les données des fournisseurs et des clients.

12.3. Application Inventory

L'activité de création des prévisions nécessite un accès à cette application. Cependant, les droits et les vues doivent être modifiés afin de donner accès uniquement à certaines parties. Tout d'abord, le menu doit être changé. Il est recommandé de passer du menu standard :



Figure 058 : Menu d'origine de l'application Inventory.

À quelque chose qui s'approche de ceci :



Figure 059 : Menu modifié de l'application Inventory.

De cette manière, seul l'accès aux règles d'approvisionnements est maintenu par le menu **Inventory Control / Reordering Rules** et l'accès aux rapports sur l'état des stocks **Reports /**

12.4. Application Purchases

L'activité de confirmation des commandes d'achats nécessite l'accès à cette application. Cependant, beaucoup des éléments qu'elle permet de manipuler n'ont pas à être disponibles pour les participants. Au niveau du menu d'origine :



Figure 060 : Menu d'origine de l'application Purchases.

Quelques modifications doivent être faites pour présenter quelque chose de similaire à cela :



Figure 061 : Menu modifié de l'application Purchases.

Seul l'accès au menu de configuration doit être retiré. Pour le reste, les droits sur les modèles doivent palier aux problèmes de sécurité.

12.5. Application Manufacturing

L'activité de lancement des ordres de fabrication nécessite un accès à cette application. Cependant, quelques fonctions doivent être bannies des possibilités offertes aux participants. Par ailleurs, le menu d'origine :

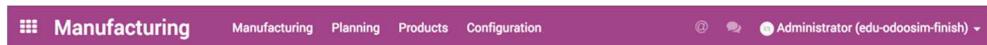


Figure 062 : Menu d'origine de l'application Manufacturing.

Doit donner accès à moins de fonctions. Il est recommandé de le faire évoluer vers ce résultat :



Figure 063 : Menu modifié de l'application Manufacturing.

Les produits, les nomenclatures et les paramètres de configuration de la production ne devant pas être modifiés par les participants.

12.6. Application Sales

Bien qu'il soit possible d'accéder à la modification des prix de vente par l'application **Purchases**, laquelle doit être accessible, il est recommandé de la laisser ouverte, pour deux raisons majeures. Premièrement, pour donner du sens au participant qui endosse la responsabilité de Sales Manager. Devoir accéder à ses produits à partir d'une application qui s'appelle **Purchases** ne fait pas tellement de sens. Deuxièmement, l'analyse des ventes se fait dans l'application **Sales**. Cependant, il est recommandé d'effectuer une modification de la barre de menu original :



Figure 064 : Menu d'origine de l'application Sales.

Pour qu'elle ressemble à ce résultat :



Figure 065 : Menu modifié de l'application Sales.

Ainsi, on coupe l'accès aux participants pour les fonctions de facturation et de configuration des paramètres de vente.

12.7. Application Accounting

Etant donné que les participants sont évalués à la performance financière, il est impensable de ne pas donner accès aux chiffres de la société. Cependant, pour les participants, toutes les opérations comptables doivent se limiter à de la consultation. Le menu original :



Figure 066 : Menu d'origine de l'application Accounting.

Doit être repensé et réélaboré pour ressembler à ceci :



Figure 067 : Menu modifié de l'application Accounting.

Tout ce qui touche au processus de vente et d'approvisionnement peut se faire à partir de leur application respective ainsi seules les fonctions de visualisation sont maintenues.

12.8. Droit d'accès aux modèles et données

En se basant sur le référentiel du chapitre Analyse des objets manipulés, tous les objets sont analysés et les recommandations quant aux droits d'accès sont données. La matrice se partage en deux parties. La première partie présente les modèles manipulés durant l'exécution de la chaîne de valeur. La seconde est mise en évidence et présente les modèles manipulés lors de la première phase (référence au chapitre Phase de configuration) :

Modèle Odoo	RAS	CAS	WAS	DAS
account.invoice	Y	N	N	N
account.invoice.line	Y	N	N	N
account.move	Y	N	N	N
account.move.line	Y	N	N	N
account.payment	Y	N	N	N
procurement.order	Y	Y	Y*	Y*
procurement.orderpoint.compute	N	N	N	N
purchase.order	Y	Y*	Y*	Y*
purchase.order.line	Y	Y*	Y*	Y*
mrp.product.produce	N	N	N	N
mrp.product.produce.line	N	N	N	N
mrp.production	Y	Y*	Y*	Y*
sale.order	Y	N	N	N
sale.order.line	Y	N	N	N
stock.immediate.transfer	N	N	N	N
stock.picking	Y	N	N	N
stock.pack.operation	Y	N	N	N
stock.quant	Y	N	N	N
stock.warehouse.orderpoint	Y	Y	Y	Y
mrp.bom	Y	N	N	N
product.product	Y	N	Y**	N

<i>product.template</i>	<i>Y</i>	<i>N</i>	<i>Y**</i>	<i>N</i>
<i>res.partner</i>	<i>Y</i>	<i>N</i>	<i>N</i>	<i>N</i>
<i>res.users</i>	<i>N</i>	<i>N</i>	<i>N</i>	<i>N</i>
<i>stock.location.route</i>	<i>N</i>	<i>N</i>	<i>N</i>	<i>N</i>

* Uniquement les enregistrements qu'ils créent eux-mêmes.

** Uniquement la modification de l'attribut prix de vente

12.9. Investigations

Cette analyse se base sur une investigation menée afin de répondre à la question « est-ce possible de gérer les droits finement sur Odoo ? ». La réponse est affirmative.

Suite à cela, certains essais réalisés dans le proof of concept permettent de s'en rendre compte.

Complément d'informations :



Lors de la phase de configuration, les droits étaient attribués aux participants via l'API. Cependant, par manque de temps d'une part et puis parce que ce point-là ne faisait pas parti du cahier des charges, cette phase de configuration n'a pas été maintenue.

Toutefois, un premier audit des groupes et des règles d'accès est réalisé à partir de l'application **Settings / Users / Groups**. Il constitue une base sur laquelle s'appuyer pour continuer le développement du simulateur.

Chaque utilisateur du système se voit attribuer un rôle au sein des applications installées. Par défaut, les utilisateurs reçoivent les rôles suivants :

An invitation email containing the following subscription link has been sent:
https://edu-odoosim-finish.odoo.com/web/reset_password?token=3sC7mwctpbFvTSpz7J&db=edu-odoosim-finish

A. Tomat

anthony.tomat@gmail.com

Related Partner **A. Tomat**

ACCESS RIGHTS **OAUTH** **PREFERENCES**

Application	
Sales	Manager
Inventory	Manager
Manufacturing	Manager
Accounting & Finance	Adviser
Purchases	Manager
Human Resources	Employee
Administration	Settings

Technical Settings	
Addressess in Sales Orders	<input type="checkbox"/>
Analytic Accounting	<input type="checkbox"/>
Analytic Accounting for Sales	<input type="checkbox"/>
Display incoterms on Sales Order and related invoices	<input type="checkbox"/>
Manage Different Stock Owners	<input type="checkbox"/>
Manage Lots / Serial Numbers	<input type="checkbox"/>
Manage Packages	<input type="checkbox"/>
Manage Product Packaging	<input type="checkbox"/>
Manage Properties of Product	<input type="checkbox"/>
Manage Secondary Unit of Measure	<input type="checkbox"/>
Manage Work Order Operations	<input type="checkbox"/>
Manage multiple warehouses	<input type="checkbox"/>
Pricelists On Product	<input type="checkbox"/>
Properties on lines	<input type="checkbox"/>
Multi Companies	<input type="checkbox"/>
Multi Currencies	<input type="checkbox"/>
Contact Creation	<input checked="" type="checkbox"/>
Portal	<input type="checkbox"/>
Public	<input type="checkbox"/>

Figure 068 : Paramètres de sécurité sur un profil utilisateur (mode développeur actif).

Les rôles créés par le système pour les applications sont les suivants :

- Sales : Aucun, See Own Leads, See all Leads et Manager.
- Inventory : Aucun, User ou Manager.
- Manufacturing : Aucun, User et Manager.
- Accounting & Finance : Aucun, Billing, Accountant et Adviser.
- Purchases : Aucun, Manager et User.
- Human Resources : Aucun et Employee.
- Administration : Aucun, Settings et Access Rights.

Ces rôles sont en faites des groupes de sécurité récupérables à partir du menu **Settings / Users / Groups**.

Mise en garde :

! Pour avoir accès aux groupes, il est nécessaire d'activer le mode développeur.

À l'aide de ces droits de bases, il est déjà tout à fait possible de limiter les fonctionnalités attribuées aux participants. Bien entendu, il est possible de créer ses propres groupes pour affiner les restrictions.

Il est recommandé de récupérer l'un des groupes d'accès à l'application que l'on veut sécuriser et aller dans son formulaire d'édition :

Figure 069 : Formulaire d'édition du groupe de sécurité Sales / Manager.

Puis à l'aide du menu **Action / Duplicate** dupliquer le groupe. En effectuant cette manipulation, le nouveau groupe est une copie du précédent. Il ne reste qu'à effectuer des modifications par tâtonnement pour atteindre l'objectif ciblé.

Les possibilités qu'offrent les groupes de sécurités sont disponibles sous forme d'onglets. Ces onglets sont **Users** qui offre la possibilité d'ajouter des utilisateurs au groupe. **Inherited** qui donne implicitement l'accès des utilisateurs qu'il abrite aux groupes de la liste. **Menus** qui permet de choisir l'agencement de la barre de menu. **Views** donne accès à des vues spécifiques. **Access Rights** qui liste les droits sur les objets du master data. **Notes** qui permet d'ajouter des notes explicatives.

Complément d'informations :



Les éléments de l'IHM peuvent être créés ou modifiés à partir de l'application **Settings / Technical / User Interface**.

L'utilité de l'onglet **Rules** n'est pas connue. Cette partie de l'analyse doit être revue au besoin.

12.10. Limites et recommandations

Toutes ces recommandations se basent sur des hypothèses qu'il s'agit de vérifier une à une. En effet, dans le cadre de ce travail, seul des essais partiels, certes concluant, ont été menés et il n'est pas possible d'affirmer avec certitude que toutes les recommandations sont exhaustives et fonctionnelles.

13. Analyse des processus opérationnels

Ce chapitre présente chacun des processus nécessaires à la fabrication des bières de Brewery & Co. Il est important dans la mesure où il permet de comprendre les processus qu'implémente la société choisie dans le scénario.

13.1. Démarche

Pour obtenir les résultats qui suivent, ERPSim et son scénario Muesli sont utilisés en tant que modèle. Pour compléter les besoins métiers, l'ouvrage de **Jean-Paul Thommen, Introduction à la gestion d'entreprise par Jean-Paul Thommen** fait office de seconde référence.

13.2. Chaîne de valeur globale

La chaîne de valeur est divisée en deux parties. Cette division fait apparaître distinctement la logistique et la vente :

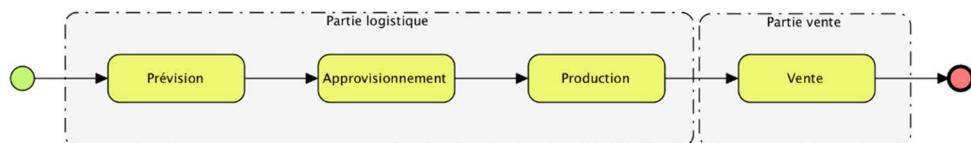


Figure 070 : Chaîne de valeur pour la société Brewery & Co.

Ce choix est fait car les observations démontrent que toutes les activités commerciales quelles qu'elles soient mettent en œuvre la partie des ventes. Pour la partie logistique, cela dépend du type de société. Une manufacture en aura une, tandis que la société de service pas.

Ce découpage apporte une certaine liberté pour la création du proof of concept.

Rappel :

- i** Un des objectifs majeurs du travail est de fournir une preuve de faisabilité sous forme d'un artefact logiciel. Ce proof of concept doit permettre de jouer à une version limitée du jeu OdooSIM.

13.2.1. Partie logistique

Le but de la logistique est de mettre à disposition des forces de ventes les produits destinés à la vente. Ces principaux objectifs sont de le faire à temps et en quantité adéquate.

13.2.1.1. La planification

La planification se situe au début de la chaîne des opérations. Elle influence toute la suite des opérations. En effet, elle agit comme le régulateur de ce qui va s'exécuter dans les processus en aval.



Figure 071 : Processus de planification en lien avec la stratégie flux-poussé.

La création des prévisions se concrétise en créant des règles d'approvisionnement pour chacun des produits que la société commercialise. Ensuite, le calcul des besoins crée des demandes d'approvisionnements, des ordres de fabrications et des demandes d'achats.



Complément d'informations :

L'activité de création des prévisions créé en sortie des requis indépendants. Ce terme propre au domaine de la logistique désigne les produits finis.

13.2.1.2. L'approvisionnement

L'approvisionnement des matières premières se fait en rapport avec les besoins du processus de production :



Figure 072 : Processus d'approvisionnement standardisé.

Lorsqu'on se trouve dans une stratégie logistique en flux poussé, les demandes d'achat sont dans la majeure partie du temps générées par le calcul des besoins. En effet, si l'on se réfère à l'exemple suivant :

Production de 100 vélos dont la nomenclature est composée de 2 roues, 1 cadre, 2 pédales, 1 dérailleur et 1 chaîne. Alors l'ordre de fabrication se constitue 100 vélos et avenant le cas où les composants ne sont pas disponibles l'ordre d'achat résultant est de 200 roues, 100 cadres, 200 pédales, 100 dérailleurs et 100 chaînes.

13.2.1.3. La production

Le processus de production est l'élément clé dans la génération de valeur. Il reçoit en entrée des matières premières brutes et fabrique les produits finaux que les consommateurs convoitent.

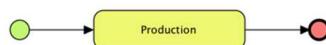


Figure 073 : Processus de production.

13.2.2. Partie des ventes

Cette partie représente la génératrice de revenue. Elle doit pouvoir compter sur les résultats de la chaîne logistique pour disposer de disponibilité pour les produits qu'elle compte proposer à ses clients.



Figure 074 : Processus de vente standardisé.

13.3. Limites et recommandations

Il est important de prendre en considération que ces modèles sont beaucoup influencés par leur mise en place dans la solution choisie. En effet, les éléments théoriques ont été mis en pratique dans Odoo et à partir de ces essais, les modèles ont été créés.

Cependant, les processus intégrés dans le PGI doivent être le plus standardisé possible. C'est ce qui fait la force d'un outil de ce type-là. De ce fait, les résultats présentés au travers des diagrammes doivent être pertinents.

14. Portabilité du scénario sur Odoo

Ce chapitre démontre comment intégrer Brewery & Co. et ses processus dans le PGI. Il présente la configuration de l'accès à une nouvelle instance, la configuration globale de la base de données, le master data à créer et passe en revue tous les processus que la société exploite pour produire ses bières.

Complément d'informations :



Odoo S.A propose un programme éducatif à sa communauté. Il donne accès à toutes les fonctionnalités du SaaS en version illimitée.

Pour la réalisation de ce travail, un compte affilié à ce programme est utilisé.

14.1. Démarche

La simulation manuelle permet d'obtenir un référentiel des objets et des workflows utilisés pour chaque activité. Sur cette base, il est possible d'utiliser divers outils pour obtenir des informations supplémentaires comme les relations qui existent entre les différents objets le type (many2many/one2many/many2one). De plus, il est également envisageable d'accéder aux contraintes que peuvent contenir certains modèles.

Afin d'avoir accès à ces informations, les outils suivants sont proposés par l'éditeur :

- La barre **Developer Tools** pour un audit simple. Elle est utile grâce aux menus **Action** pour tous ce qui touche à la visualisation des

attributs et des workflows (s'il y en a) et **View** pour obtenir des informations sur les métadonnées.

- Plus poussés, les paramètres techniques. Pour s'y rendre **Menu principal / Paramètres / Technique / Structure de base de données** et à partir de là, les modèles, les attributs, les contraintes et les relations many2many sont accessibles. Principalement utilisé, le Modèle qui donne accès à la liste exhaustive de tous les objets (tables) Odoo®. Il est aussi possible d'extraire un fichier PDF grâce au menu **Print / Model Overview** accessible directement dans le détail d'un modèle.
- De la même manière, les workflows peuvent être exportés sous forme PDF. Cependant, dans la version actuelle d'Odoo® les workflows sont gentiment abandonnés. De ce fait, la plupart des opérations se font uniquement grâce à des états sur les objets.

i Complément d'informations :

Ce n'est qu'après plusieurs semaines que ces outils proposés par l'éditeur ont été trouvés. Avant cela, la mise en place d'un bout de code permettait d'interroger la structure d'un modèle particulier par le biais de l'API. Voici le code de base (dans cet exemple c'est le modèle res.partner qui est inspecté) :

```
(Map<String, Map<String, Object>>)models.execute("execute_kw",.asList(
    db, uid, password,
    "res.partner", "fields_get",
    emptyList(),
    new HashMap() {{
        put("attributes", asList("string", "help", "type"));
    }});
));
```

Afin d'obtenir ces résultats, le mode développeur actif. Ce mode permet d'avoir accès à des informations techniques très intéressantes dans le cadre de ce travail.

i Complément d'informations :

- Le mode développeur s'active directement sur l'IHM Web. Pour se faire aller dans le menu supérieur droit / About :

[Activate the developer mode](#)

Par exemple, pour récupérer les informations relatives à l'objet manipulé lors de l'ajout d'une nouvelle vente, il suffit de créer la vente et de survoler les champs intéressants qu'il est nécessaire de manipuler pour les besoins du scénario.

Voici comment récupérer le nom de l'attribut nécessaire à renseigner le client dans une vente :

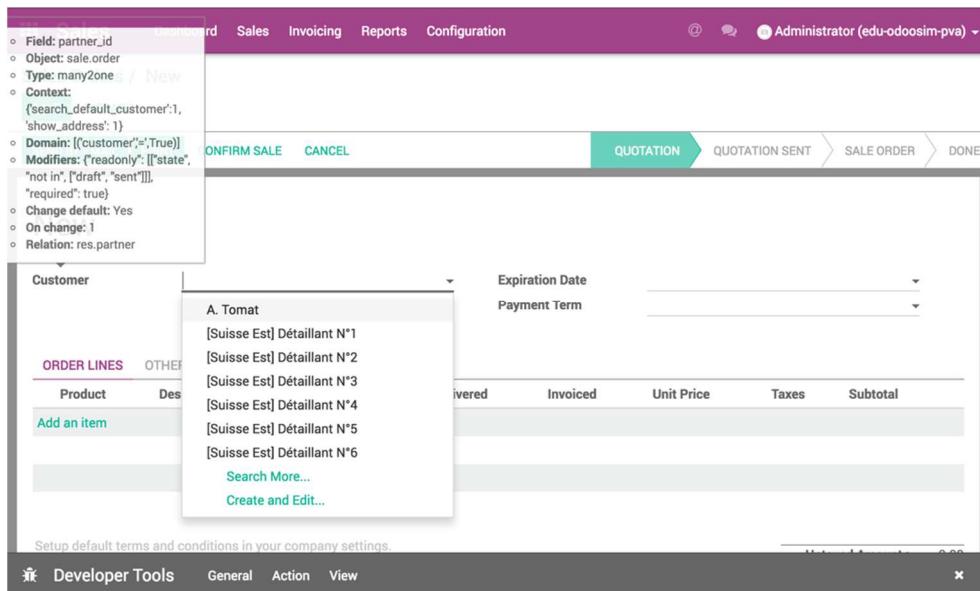


Figure 075 : Détail d'un survol à l'aide du mode développeur actif.

Le résultat obtenu est l'objet qu'exploite Odoo® pour effectuer l'opération. Pour une vente, l'objet se nomme **sale.order** et l'attribut **Field** permettant de renseigner le client se nomme **partner_id** du type **many2one**.

Une autre manière d'accéder aux informations, les outils de développement disponibles dans le navigateur Google Chrome.

Complément d'informations :



Certains modèles sont des **Transient Model** et n'ont donc pas d'existence réelle dans la base de données. Par exemple, lors de la confirmation d'un ordre de fabrication, l'objet **mrp.production.produce** est l'un de ces objets transients. Pour pouvoir créer cette opération de manière automatique dans le simulateur, il est primordial de créer cet objet transients le temps d'effectuer la confirmation. Pour comprendre comment cela s'effectue, les outils de développement sont d'une grande utilité.

Lorsque des interactions entre l'utilisateur et Odoo® s'exécutent, les outils de développement peuvent servir à donner des bonnes informations sur ce qui est envoyé à Odoo®. Ainsi, il est toute à fait possible de reproduire le même type d'appel dans le simulateur.

Voici un exemple de confirmation de la production et les communications qui s'exécutent :

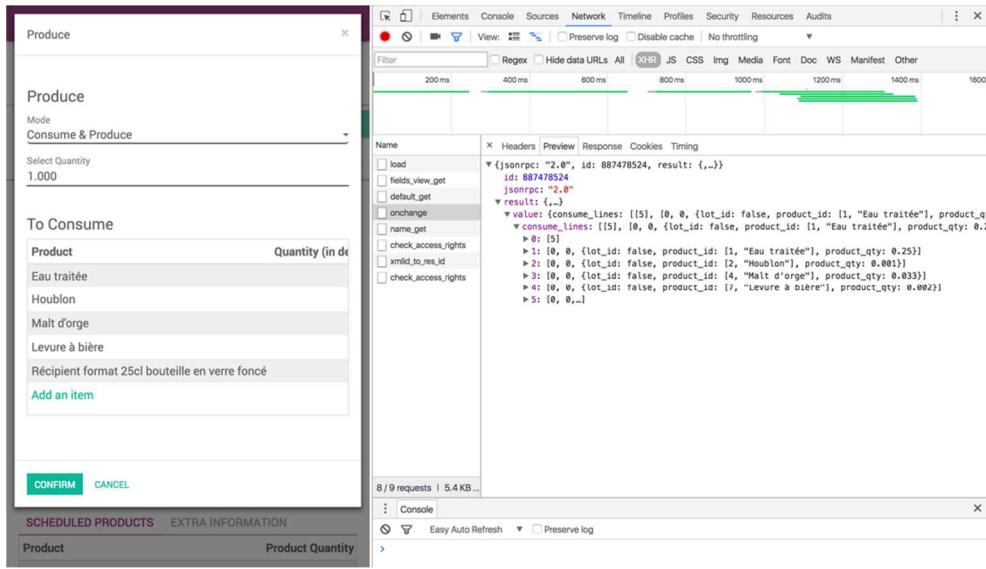


Figure 076 : Outil de développement du navigateur Google Chrome.

14.2. Configuration globale

Ce chapitre parcourt la mise en place d'un nouvel environnement pour la société Brewery & Co. Il détaille trois éléments, la création de l'instance, l'installation des applications et la configuration du master data.

Ensuite, il passe en revue chacune des applications nécessaires à la réalisation des processus. Pour chacune d'entre elles, il donne les spécifications la configuration à mettre en place pour répondre aux besoins Brewery & Co.

14.2.1. Crédation de l'instance

Afin de pouvoir accéder à une instance Odoo online, il faut tout d'abord la créer. Comme mentionné ci-dessus, l'utilisation d'un compte éducatif permet de s'affranchir des coûts nécessaires à l'activation de toutes les fonctionnalités.

Pour obtenir une instance, il est nécessaire de se rendre sur odoo.com, de s’authentifier et d'accéder au management des bases de données par le menu **Nom du compte / My Databases** puis **Manage your databases / Create a new database** et au moment de sélectionner une première application choisir **Accounting**. L'étape qui suit demande quelques informations globales permettant au moteur de création de base de données de configurer la nouvelle instance selon la législation et les normes du pays renseigné. Aussi, les paramètres d'accès sont demandés :

Name	Brewery & Co.		
Email	anthony.tomat@he-arc.ch		
Company Name	edu-odoosim-brewery	Phone Number	+41XXXXXXX
Country	Switzerland	Language	English
Company size	< 5 employees	Primary Interest	I am a teacher or a student
START NOW			

Figure 077 : Formulaire de création d'une nouvelle instance.



Complément d'informations :

La base de données est créée et un e-mail de confirmation est envoyé à l'adresse renseignée.

14.2.2. Installation des applications

Pour couvrir l'ensemble des besoins de la société Brewery & Co plusieurs applications doivent être installées.



Complément d'informations :

La version Odoo online 9 propose nativement les applications **Discuss**, **Contacts**, **Apps** et **Settings**. De plus, l'application sélectionnée lors de la phase de création est aussi disponible. Dans ce cas de figure **Accounting**.

Tout d'abord, il est nécessaire de se rendre dans l'application **Apps** et de choisir d'installer ce qui suit :

- Inventory Management (nom technique : stock)
- MRP (mrp)
- Sales Management (sale)
- Purchase management (purchase)

Les applications natives plus la comptabilité et ces dernières suffisent largement à couvrir les besoins de la société Brewery & Co.

14.2.3. Master Data

À présent, il est nécessaire d'introduire les données métiers que Brewery & Co. va manipuler durant son exploitation. L'outil propose deux guides d'implémentations pour garantir un paramétrage correct.

Le premier, **Accounting**, donne une suite d'opérations à effectuer qui ressemble à cela :

- Définition des données générales de la société (nom légal, adresse, numéro TVA, devise employée, conditions de paiement en nombre de jours et le logo). Elles ne sont pas employées dans le cadre de ce travail.
- Définition d'un journal pour enregistrer les transactions bancaires. La valeur par défaut **Bank (CHF)** correspond bien au besoin du scénario.
- Définition d'un plan comptable. Etant donné qu'il a fallu renseigner le pays lors de la création de l'instance, le plan proposé est normé et convient amplement.
- Définition des différentes taxes. Rien à modifier.
- Création de la base de données clients, produits et des soldes des comptes.

Ce qu'il faut retenir de ce guide d'implémentation se sont les dernières étapes qui permettent de constituer le référentiel de données. Ce sont donc trois éléments du master data qui sont découverts grâce à cette marche à suivre. À savoir les clients, les produits et les écritures au journal comptable nécessaires à l'ouverture des comptes. Techniquelement, cela se traduit par importer des données dans les modèles :

- **res.partner** pour les clients.
- **product.template** pour les produits.
- **account.move** pour les écritures dans la comptabilité.



Complément d'informations :

Les informations techniques sont récupérées grâce au mode développeur actif.

Ces éléments sont à récupérer à partir de la description du scénario. En voici le contenu :

- Les clients : Les 220 revendeurs.
- Les produits : Les matières premières, les produits commercialisés – c'est-à-dire les quatre sortes de bières – et finalement la nomenclature.
- Les écritures comptables : Toutes celles nécessaires à la mise en place du bilan.

Les résultats à obtenir doivent s'apparenter à ceux qui sont présentés ci-dessous :

<input type="checkbox"/> [Suisse Est] Détailleur N°1
<input type="checkbox"/> [Suisse Est] Détailleur N°2
<input type="checkbox"/> [Suisse Est] Détailleur N°3
<input type="checkbox"/> [Suisse Est] Détailleur N°4
<input type="checkbox"/> [Suisse Est] Détailleur N°5
<input type="checkbox"/> [Suisse Est] Détailleur N°6
<input type="checkbox"/> [Suisse Est] Détailleur N°7
<input type="checkbox"/> [Suisse Est] Détailleur N°8

Figure 078 : Exemple master data des détaillants pour la région Suisse Est.

Name	Cost	Weight	Internal Category	Product Type	Quantity On Hand	Forecasted Quantity
<input type="checkbox"/> Honing Beer 0.25	2.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Honing Beer 0.50	2.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Lager Beer 0.25	2.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Lager Beer 0.50	2.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Lager Beer BIO 0.25	2.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Lager Beer BIO 0.50	2.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Spice Beer 0.25	2.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Spice Beer 0.50	2.00	0.00	All	Stockable Product	0.000	0.000

Figure 079 : Master data des produits commercialisés.

Name	Cost	Weight	Internal Category	Product Type	Quantity On Hand	Forecasted Quantity
<input type="checkbox"/> Bouquet d'épices	8.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Eau traitée	0.15	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Houblon	18.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Houblon BIO	27.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Levure à bière	250.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Levure à bière BIO	275.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Malt d'orge	0.60	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Malt d'orge BIO	0.75	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Malt de fromant	0.45	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Miel	10.00	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Récipient format 25cl bouteille en verre foncé	0.05	0.00	All	Stockable Product	0.000	0.000
<input type="checkbox"/> Récipient format 50cl cannette d'aluminium	0.01	0.00	All	Stockable Product	0.000	0.000

Figure 080 : Master data des matières premières.

Product	Reference	Product Quantity
<input type="checkbox"/> Lager Beer 0.25		1.000
<input type="checkbox"/> Lager Beer 0.50		1.000
<input type="checkbox"/> Honing Beer 0.25		1.000
<input type="checkbox"/> Honing Beer 0.50		1.000
<input type="checkbox"/> Lager Beer BIO 0.25		1.000
<input type="checkbox"/> Lager Beer BIO 0.50		1.000
<input type="checkbox"/> Spice Beer 0.25		1.000
<input type="checkbox"/> Spice Beer 0.50		1.000

Figure 081 : Master data des nomenclatures. Une par produit commercialisé.

JOURNAL ITEMS

Account	Partner	Label	Debit	Credit
1500 Machines et appareils	VUBS SA	Achat d'une partie de l'installation de production	50,000.00 CHF	0.00 CHF
1600 Immeubles d'exploitation	VUBS SA	Achat immeuble et terrain	1,000,000.00 CHF	0.00 CHF
2100 Dettes bancaires	VUBS SA	Octroi du crédit initial	0.00 CHF	1,200,000.00 CHF
10201 Bank	VUBS SA	Sàn	150,000.00 CHF	0.00 CHF
			1,200,000.00	1,200,000.00

JOURNAL ITEMS

Account	Partner	Label	Debit	Credit
2800 Capital-actions, capital social, capital de fondation	Breweries Holding Corp.	Apport en capital de la Holding	0.00 CHF	400,000.00 CHF
1500 Machines et appareils	VUBS SA	Achat d'une partie de l'installation de production	400,000.00 CHF	0.00 CHF
			400,000.00	400,000.00

Figure 082 : Master data des écritures comptables dans deux journaux différents.

À ce stade, l’instance est prête à être employée pour les besoins du scénario. Il faut encore définir les collaborateurs de la société en ajoutant des données dans les utilisateurs. Techniquelement, cela se fait dans le modèle **res.users**.

Complément d’informations :

i Il est évident que l’insertion des 220 revendeurs est rendue difficile de manière manuelle. La phase de configuration du simulateur se charge de les entrer.

14.3. Inventory Management

Le scénario impose la mise en place de la logistique selon la stratégie flux-poussé. Dans le progiciel, cela se traduit par la configuration de l’application et au paramétrage des produits.

Le processus de planification s’appuie sur l’application **Inventory Management** (stock).

14.3.1. Configuration

Cette application doit être configurée pour les besoins du scénario. Il est nécessaire de modifier le mode de valorisation comptable du stock. Etant donné que le jeu simule des jours en quelques secondes, la valorisation doit se faire en temps réel.

La seconde manipulation à entreprendre est celle nécessaire au bon fonctionnement du calcul des besoins. Elle est nécessaire pour gérer la production sur prévisions.

14.3.1.1. Valorisation des stocks

Il est nécessaire de se rendre dans l'application **Inventory / Configuration / Settings** et dans la rubrique **Accounting** remplacer **Periodic inventory valuation (recommended)** par **Perpetual inventory valuation (stock move generates accounting entries)**. Comme cela, chaque mouvement de stock est comptabilisé financièrement et l'évaluation des résultats financiers en est grandement facilitée.

Finalement, les mouvements de stock sont relatifs aux produits que la société manipule. Pour que la valorisation soit effective, il faut modifier la catégorie de produit **All** du modèle **product.category** de la manière suivante :

Account Stock Properties			
Inventory Valuation		Stock Input Account	3901 Variation des stocks de produits finis
Costing Method	Standard Price	Stock Output Account	3901 Variation des stocks de produits finis
Inventory Valuation	Perpetual (automated)	Stock Valuation Account	1260 Stocks de produits finis
		Stock Journal	Stock Journal (CHF)

Figure 083 : Paramètres de la catégorie de produit All.

Ces modifications sont nécessaires afin d'attribuer les mouvements d'entrées, de sorties et de valorisation à des comptes.

! Limites :

Dans cette configuration, les matières premières et les produits finis sont valorisés dans les mêmes comptes. Il serait nécessaire de privilégier la diversification des deux natures de produit. Tout d'abord en créant deux catégories distinctes avec des propriétés comptables différentes.

14.3.1.2. Calcul des besoins

Pour que le calcul des besoins puisse fonctionner, il est nécessaire d'apporter des modifications aux objets du modèle **product.template**.

D'une part, les articles manufacturés doivent être modifiés. Les propriétés ci-dessous doivent être modifiées de la sorte :

- Champ **Can be Solde** (techniquement : **sale_ok**) avec comme valeur **true**.
- Champ **Routes** (techniquement : **route_ids**) avec comme valeur uniquement **Manufacture**.

Les produits à acheter, c'est-à-dire les matières premières doivent être modifiées comme ceci :

- Champ **Routes** (techniquement : **route_ids**) avec comme valeur **Buy** et **Make To Order**.

En paramétrant les produits de la sorte, le MRP est capable d'aller comparer les besoins en produits finis pour créer les ordres de fabrication et les besoins en matières premières pour créer des demandes d'achats.

! Limite :

En paramétrant les matières premières en MTO, le scénario n'est pas totalement cohérent. Car les matières premières ne sont par conséquent pas entreposées sous forme de stock mais uniquement mises en attente du lancement des OF pour lesquels elles sont commandées.

14.4. Purchase Management

Cette application qui porte techniquement le nom **purchase** est utilisée pour supporter le processus d'approvisionnement.

14.4.1. Configuration

Il n'y a pas de configuration spécifique à mettre en œuvre dans cette application. Il est cependant nécessaire que les produits qui peuvent être achetés soient paramétrés correctement :

- Champ **Can be Purchased** (techniquement : **purchase_ok**) avec comme valeur **true**.
- Champ **Cost** (techniquement : **standard_price**) avec un prix d'achat.

i Complément d'informations :

La liste des fournisseurs dans les produits achetables est nécessaire à la bonne exécution du MRP et de la création des demandes d'achat.

14.5. Manufacturing

Cette application est nécessaire au pilotage de la production. Elle porte comme nom technique **mrp** et aucun paramétrage spécifique n'est effectué. Cependant, l'analyse des possibilités offertes par cette application n'a pas été faite en profondeur. Une des lacunes dans la configuration actuelle est que les lignes de production et leur capacité ne sont nulle part configurées.

Ce qui pose un problème dans le contexte Brewery & Co. En effet, si les règles d'approvisionnements automatiques sont paramétrées avec des quantités excessives, le MRP créé des ordres de fabrication en conséquence et cela n'est plus cohérent avec le scénario.

Rappel :

Le scénario impose des contraintes sur les capacités maximales de production. La configuration actuelle ne permet pas de gérer cela.

14.6. Sales Management

Cette application est nécessaire pour gérer le processus des ventes. Elle porte le nom technique **sale** et doit être configurée pour les besoins du scénario Brewery & Co.

Principalement, pour pouvoir attribuer des ventes aux régions et ainsi pouvoir créer des analyses pertinentes afin de mesurer l'importance de chacune d'entre elles. Mais aussi pour mesurer les préférences pour les produits des marchés. C'est un point déterminant pour créer des stratégies gagnantes dans le jeu.

14.6.1. Configuration

Afin d'améliorer les possibilités de reporting, il est nécessaire d'ajouter des équipes de vente. Il est conseillé de créer une équipe de vente par région.

Ceci s'effectue grâce au modèle **crm.team** et impacte directement le master data :

The screenshot shows a user interface for managing sales teams. At the top, it says "Sales team" and "Swiss East Area". Below that, there are two checked checkboxes: "Quotations" and "Invoices". Under "Team Leader", there is a dropdown menu set to "Administrator". There is also a "Code" field below it. At the bottom of the screen, there are two tabs: "TEAM MEMBERS" and "MORE INFO", and a green "ADD" button.

Figure 084 : Ajout d'une équipe de vente pour la région Suisse Est.

Pour les besoins de Brewery & Co, il est recommandé de créer trois équipes de ventes, respectivement :

- **Swiss East Area, Swiss Middle Area et Swiss West Area.**

Selon le même principe présenté par la figure 084.

14.7. Guide des opérations

Sur la base de la chaîne de valeur ci-dessous, ce chapitre permet de découvrir les objets et leurs différents états tout au long des processus. Il constitue un support obligatoire à la conception du jeu OdooSIM.

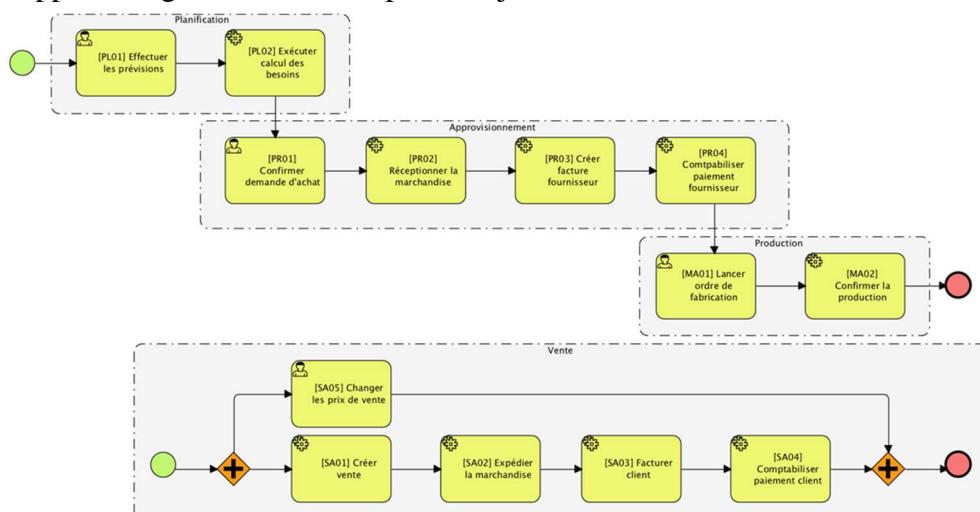


Figure 085 : Processus d'affaires pour Brewery & Co.

Les quatre processus sont présentés avec pour chacun d'eux les activités principales à effectuer.

Les chapitres qui suivent donnent le détail de chacune des activités, les objets du modèle manipulés et l'impact de sa réalisation sur les deux processus de support que sont la gestion des stocks et la comptabilité.

Complément d'informations :



Les activités sont identifiées par un code logique composé d'un préfixe signifiant le processus anglophone : PL pour Planning, PR pour Procurement, MA pour Manufacturing et SA pour Sale. Ensuite, un numéro fait office d'identificateur chronologique.

14.7.1. Planification

Pour présenter les activités de ce processus, il est nécessaire de se baser sur ce qui suit :

Hypothèse :

Les prévisions ont été faites et les cadres logistiques ont défini que dans une vision à court terme, le produit **BW01 0.25** est demandé à hauteur de 30%.

La société qui produit quatre sortes de bière dans deux formats 0.25 et 0.50 dispose d'une zone de stockage pouvant abriter 10'000 articles conditionnés et prêt à la vente.

La stratégie retenue est de ne pas sous-traiter le stockage des articles à un tiers. Par conséquent, il est nécessaire de ne jamais dépasser la limite quantitative. En finalité et d'après les calculs prévisionnels, ce stock doit se composer de 3'000 articles **BW01 0.25** (30%).

Les données des ventes consolidées font état d'une demande quotidienne de 1'750 unités. La capacité productive totale est de 5'000 articles quotidiens. Il est nécessaire de mettre en place la règle d'approvisionnement automatique suivante :

- Pour le produit BW01 0.25 cl, stock sécurité de 1'800 et stock maximal 3'000.

* Ces informations sont données à titre d'exemple et permettent de disposer d'un cadre pour ce chapitre.

14.7.1.1. Effectuer les prévisions PL01

Cette activité s'effectue de la manière suivante. Tout d'abord, il faut se rendre dans l'application **Inventory / Inventory Control / Reordering Rules** et puis effectuer l'insertion telle que présentée :

Name	OP/00001		
Product	[BW01A] Lager Beer 0.25		
Rules		Misc	
Minimum Quantity	1800.000	Active	<input checked="" type="checkbox"/>
Maximum Quantity	3000	Lead Time	5
Quantity Multiple	1.000	Day(s) to get the product	

Figure 086 : Règle d'approvisionnement pour le produit BW01A.

Les impacts sur les processus de gestion des stocks et de la comptabilité sont néants lorsqu'on ajoute ces règles d'approvisionnement.

Complément d'informations :

La création d'une règle d'approvisionnement est utilisée par le calcul des besoins pour créer des ordres de fabrication dans l'état brouillon et des demandes d'achats dans à l'état de devis.



Le MRP déclenche la création de ces deux éléments une fois que le stock virtuel (qui est égal au stock réel – toutes les commandes fermes – les réservations) est en-dessous de la valeur sécuritaire, alors le MRP calcule la différence entre la valeur maximale et ce stock virtuel pour créer une demande d'approvisionnement (OF + OA).

14.7.1.2. Exécuter le calcul des besoins PL02

Une fois que les règles d'approvisionnement sont mises en place, il faut exécuter le calcul des besoins. Pour exécuter cette action, il faut se rendre dans l'application **Inventory / Schedulers** et cliquer sur **Run Reordering Rules** et répondre par **Compute Stock** à la question **Wizard checks all the stock minimum rules and generate procurement order**.

Odoo n'affiche rien. Cependant, il effectue le calcul présenté précédemment pour créer les ordres de fabrication (OF) et les demandes d'achats (OA) selon les besoins.

Au niveau de la comptabilité, rien ne s'effectue à ce moment-là. Cela est somme toute très logique vu qu'il n'y a aucun mouvement réellement effectué.

La gestion des stocks est quant à elle impactée. En effet, des transferts de stock sont créés en rapport avec l'ordre de fabrication généré, pour les visualiser, application **Inventory / Reports / Stock Moves** :

Stock Moves						
Search... ✖						
Reference	Source Document	Product	Quantity	Date	Expected Date	Status
<input type="checkbox"/>	MO00001	[BW01A] Lager Beer 0.25	3,000.000	06/21/2016 02:00:00	06/21/2016 02:00:00	Waiting Availability
<input type="checkbox"/>	MO00001	Eau traitée	750.000	06/21/2016 02:00:00	06/21/2016 02:00:00	Waiting Another Move
<input type="checkbox"/>	MO00001	Houblon	3.000	06/21/2016 02:00:00	06/21/2016 02:00:00	Waiting Another Move
<input type="checkbox"/>	MO00001	Malt d'orge	99.000	06/21/2016 02:00:00	06/21/2016 02:00:00	Waiting Another Move
<input type="checkbox"/>	MO00001	Levure à bière	6.000	06/21/2016 02:00:00	06/21/2016 02:00:00	Waiting Another Move
<input type="checkbox"/>	MO00001	Récipient format 25cl bouteille en verre foncé	3,000.000	06/21/2016 02:00:00	06/21/2016 02:00:00	Waiting Another Move

Figure 087 : Liste des mouvements de stock après exécution du calcul des besoins.

On voit bien la source MO00001 comme étant l'élément déclencheur de tous ces mouvements.

Autres indications

Un autre indicateur est impacté par ce calcul des besoins. En effet, la valeur du stock virtuel des matières premières est recalculée en prenant en compte la prévision de l'engagement quantitatif dans l'ordre de fabrication. Ainsi, en prenant une des matières premières entrant en composition dans la réalisation d'un ordre de fabrication, les détails de la matière première et plus particulièrement l'indicateur **Forcasted** prend en compte ce mouvement futur :

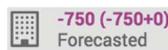


Figure 088 : Indicateur Forcasted dans le détail d'un produit. Pour les produits consommés, application Purchases / Purchase / Products et sélection du produit voulu. Pour les articles fabriqués, application Sales / Sale / Products et sélection du produit voulu.

14.7.2. Approvisionnement

Dans l'ordre chronologique, l'exécution de ce processus vient directement après celui des prévisions. En effet, comme présenté précédemment, pour que l'ordre de fabrication puisse être lancé, il faut que les requis dépendants nécessaires soient disponibles. Pour se faire, il est nécessaire de les acheter auprès des fournisseurs.

14.7.2.1. Confirmer les demandes d'achat PR01

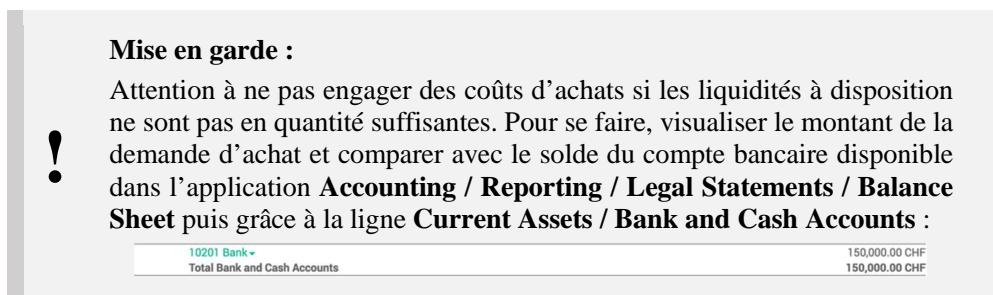
En se rendant dans l'application **Purchases / Purchase / Requests for Quotation**, une liste de devis dans l'état brouillon **Draft PO** sont visualisables. Ces devis sont générés automatiquement par le calcul des besoins effectué par l'activité PL02. Ils sont en relation avec les besoins en requis dépendants pour constituer le requis indépendant à fabriquer dans l'ordre de fabrication concerné :

Requests for Quotation							
		Search...					
CREATE		IMPORT					
Reference	Order Date	Vendor	Scheduled Date	Source Document	Untaxed	Total	Status
PO00002	06/21/2016 02:00:00	Maltonne SA	06/22/2016 02:00:00	M000001	1,632.78	1,763.40	Draft PO
PO00001	06/21/2016 02:00:00	Ourseau SA	06/22/2016 02:00:00	M000001	104.17	112.50	Draft PO

Figure 089 : Demande d'achat à l'état brouillon. Générée par le calcul des besoins et en relation avec l'ordre de fabrication M000001.

Deux demandes d'achats sont créées car les composants ne proviennent pas du même fournisseur. Le document d'origine M000001 est visualisable, c'est-à-dire l'ordre de fabrication pour lequel ces demandes sont créées.

Pour effectuer l'activité PR01, il est nécessaire d'afficher le détail de chacune des deux demandes en cliquant dessus tour à tour puis en confirmant à l'aide du bouton **Confirm Order**.



Cette opération de confirmation des demandes d'achats n'a aucun impact sur la comptabilité car à ce moment-là, aucun flux monétaire n'a lieu.

Au niveau de la gestion des stocks, deux éléments se créent. Un ordre de prélèvement et un mouvement de stock par produit acheté sont créés :

WH/IN/00001

Partner	Maltonne SA	Scheduled Date	06/22/2016 02:00:00
		Source Document	P000002
OPERATIONS		INITIAL DEMAND	ADDITIONAL INFO
Product		To Do	Done
Houblon		3.000	0.000
Malt d'orge		99.000	0.000
Levure à bière		6.000	0.000
Récipient format 25cl bouteille en verre foncé		3.000.000	0.000

Figure 090 : Prélèvement de stock des matières premières achetées.

Dans le prélèvement de stock, deux choses peuvent être visualisées, les opérations faites et celles restantes grâce à la liste **Operations**. Les mouvements de stock liés sont listés dans la liste **Initial Demand**. La figure ci-dessous présente les mouvements de stock relatifs à chaque produit en entrée. En affichant les mouvements de stock du prélèvement précédent avec la liste **Initial Demand** les mêmes lignes de mouvement sont retournées. Cela représente bien un lien :

<input type="checkbox"/> WH/IN/00002 P000001	Eau traitée	750.000	06/21/2016 02:00:00	06/22/2016 02:00:00	Available
<input type="checkbox"/> WH/IN/00001 P000002	Houblon	3.000	06/21/2016 02:00:00	06/22/2016 02:00:00	Available
<input type="checkbox"/> WH/IN/00001 P000002	Malt d'orge	99.000	06/21/2016 02:00:00	06/22/2016 02:00:00	Available
<input type="checkbox"/> WH/IN/00001 P000002	Levure à bière	6.000	06/21/2016 02:00:00	06/22/2016 02:00:00	Available
<input type="checkbox"/> WH/IN/00001 P000002	Récipient format 25cl bouteille en verre foncé	3.000.000	06/21/2016 02:00:00	06/22/2016 02:00:00	Available

Figure 091 : Mouvement de stock des matières premières achetées.

14.7.2.2. Réceptionner la marchandise PR02

Les demandes d'achats étant confirmées, il ne reste qu'à attendre la livraison des marchandises selon les délais impartis. Dans la pratique, la main est passée aux employés du département réception marchandise. Par ailleurs, en se rendant dans l'application **Inventory / Dashboard** l'observation qui peut être faite est que le système affiche deux opérations de réception à effectuer :

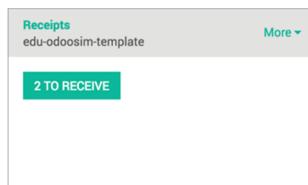


Figure 092 : Tableau de bord de la gestion des stocks, tâches en attentes.

En cliquant sur le lien **2 To Receive**, les deux réceptions de marchandise en attentes sont affichées. Une fois qu'elles arrivent à quai, il est nécessaire de les vérifier grâce au bon de livraison et d'effectuer le transfert de stock et les mouvements associés.

Pour entrer les marchandises dans les stocks de la société et dans le PGI, il faut ouvrir les prélèvements en attentes, application **Inventory / Dashboard / 2 To Receive** ou **Inventory / Operations / All Transfers** puis afficher leur détail :

Stock Operations / WH/IN/00001		Print ▾ Action ▾		2 / 2 < >	
EDIT	CREATE	DRAFT	WAITING AVAILABILITY	PARTIALLY AVAILABLE	AVAILABLE
VALIDATE	PRINT	CANCEL			DONE
WH/IN/00001					
Partner	Maltonne SA	Scheduled Date	06/22/2016 02:00:00	Source Document	PO00002
<u>OPERATIONS</u>	INITIAL DEMAND	ADDITIONAL INFO			
Product	To Do	Done			
Houblon	3.000	0.000			
Malt d'orge	99.000	0.000			
Levure à bière	6.000	0.000			
Récipient format 25cl bouteille en verre foncé	3.000.000	0.000			

Figure 093 : Prélèvement de stock pour la demande d'achat PO00002.

Un clic sur **Validate** et la confirmation du message **You haven't set processed quantities. If you click apply, Odoo will process all quantities to do** suffisent pour effectuer l'entrée en stock de l'entier de la quantité dans le système. En faisant référence au prélèvement, ce dernier est dans l'état terminé **Done**. De même que pour la liste des mouvements de stock, ceux relatifs au prélèvement **WH/IN/00001** sont tous grisés et par conséquent dans le statut terminé **Done** :

<input type="checkbox"/> WH/IN/00001 PO00002	Houblon	3.000	06/20/2016 09:46:50	06/22/2016 02:00:00	Done
<input type="checkbox"/> WH/IN/00001 PO00002	Malt d'orge	99.000	06/20/2016 09:46:50	06/22/2016 02:00:00	Done
<input type="checkbox"/> WH/IN/00001 PO00002	Levure à bière	6.000	06/20/2016 09:46:50	06/22/2016 02:00:00	Done
<input type="checkbox"/> WH/IN/00001 PO00002	Récipient format 25cl bouteille en verre foncé	3.000.000	06/20/2016 09:46:50	06/22/2016 02:00:00	Done

Figure 094 : Mouvements de stock et leur état suite à l'activité PR02.

Un élément important est généré à ce moment-là, un **Quant** par mouvement de stock. Pour s'en apercevoir, il faut se rendre dans le détail d'un des mouvements terminés et se référer à la liste **Moved Quants** :

Product	Houblon	Picking Type	edu-odoosim-template: Receipts	
Quantity	3.000	Priority	Normal	
Description	Houblon	Date	06/20/2016 09:46:50	
Origin				
Source Document PO00002				
Transfer Reference WH/IN/00001				
Creation Date 06/20/2016 08:58:18				
Procurement Group PO00002				
Moved Quants				
Product	Quantity	Location	Incoming Date	Inventory Value
Houblon	3.00	WH/Stock	06/20/2016 09:46:49	54.00

Figure 095 : Détails d'un mouvement de stock dans l'état terminé.

Au niveau de la comptabilité, l'impact est aussi mesurable. Chaque mouvement de stock a généré une écriture comptable. Pour s'en apercevoir, il est nécessaire de se rendre dans l'application **Accounting / Adviser / Journal Entries** et de visualiser les entrées qui concerne le journal **Stock Journal (CHF)** et plus particulièrement le prélèvement **WH/IN/00001** :

Journal Entries						
<input type="button" value="CREATE"/> <input type="button" value="IMPORT"/> <input type="text" value="Miscellaneous Operations X"/> <input type="text" value="Search..."/> <input type="button" value="Q"/>						
<input type="checkbox"/>	Date	Number	Partner	Reference	Journal	Amount
<input type="checkbox"/>	06/20/2016	STJ/2016/0004	Maltonne SA	WH/IN/00001	Stock Journal (CHF)	1,500.00 CHF Posted
<input type="checkbox"/>	06/20/2016	STJ/2016/0003	Maltonne SA	WH/IN/00001	Stock Journal (CHF)	54.00 CHF Posted
<input type="checkbox"/>	06/20/2016	STJ/2016/0002	Maltonne SA	WH/IN/00001	Stock Journal (CHF)	150.00 CHF Posted
<input type="checkbox"/>	06/20/2016	STJ/2016/0001	Maltonne SA	WH/IN/00001	Stock Journal (CHF)	59.40 CHF Posted
<input type="checkbox"/>	06/19/2016	/	Breweries Holding Corp.		Miscellaneous Operations (CHF)	400,000.00 CHF Posted
						401,763.40

Figure 096 : Liste des écritures comptables.

En affichant le détail d'une écriture, le comportement du système comptable dans Odoo est identifiable :

					View reconciled entries
STJ/2016/0004					
Journal	Stock Journal (CHF)	Reference	WH/IN/00001		
Date	06/20/2016				
JOURNAL ITEMS					
Account	Partner	Label	Debit	Credit	
3901 Variation des stocks de produits finis	Maltonne SA	Levure à bière	0.00 CHF	1,500.00 CHF	
1260 Stocks de produits finis	Maltonne SA	Levure à bière	1,500.00 CHF	0.00 CHF	
			1,500.00	1,500.00	

Figure 097 : Détails d'une écriture comptable liée à une entrée de stock.

Les comptes impactés sont présentés. Le **3901 pour la variation de stocks** et le **1260** pour la valorisation effective.

i Complément d'informations :
Ce sont les comptes qui sont paramétrés au niveau de la catégorie de produit

Actuellement, le processus d'achat n'est pas totalement terminé. En effet, il reste à payer le fournisseur. De ce fait, notre compte d'exploitation – accessible dans l'application **Accounting / Reporting / Legal Statements / Profit and Loss** – affiche un bénéfice alors qu'aucune vente n'a été effectuée :

GROSS PROFIT		
▼ Income		
3901 Variation des stocks de produits finis▼		1,875.90 CHF
Total Income		1,875.90 CHF
Other Income		0.00 CHF
Cost of Revenue		0.00 CHF
Total GROSS PROFIT		1,875.90 CHF
 Expenses		
Expenses		0.00 CHF
Depreciation		0.00 CHF
Total Expenses		0.00 CHF
 NET PROFIT		 1,875.90 CHF

Figure 098 : Compte pertes et profits après entrée en stock suite à un achat.

Ce fonctionnement est tout à fait normal. Car en réalité à ce moment précis la société s'est enrichie de la valeur des matières acquis sans les avoir encore payées. La balance s'équilibre par la suite lorsque le fournisseur est payé.

Au niveau du bilan – accessible dans l'application **Accounting / Reporting / Legal Statements / Balance Sheet** –, la valeur des matières premières acquises est aussi comptabilisée :

Current Assets	
▼ Bank and Cash Accounts	
10201 Bank ▾	150,000.00 CHF
Total Bank and Cash Accounts	150,000.00 CHF
Receivables	0.00 CHF
▼ Current Assets	
1260 Stocks de produits finis ▾	1,875.90 CHF
1500 Machines et appareils ▾	450,000.00 CHF
1600 Immeubles d'exploitation ▾	1,000,000.00 CHF
Total Current Assets	1,451,875.90 CHF
Prepayments	0.00 CHF
Total Current Assets	1,601,875.90 CHF

Figure 099 : Bilan suite à la réception des matières premières achetées.

Le gestionnaire de stock peut aussi visualiser l'impact de ces entrées de stock en affichant un rapport accessible via l'application **Inventory / Reports / Inventory Valuation** :

Inventory Valuation		Internal Locations	Search...	Q		
		Product	Quantity	Location	Incoming Date	Inventory Value
▼ Houblion (1)			3.00			54.00
▼ WH/Stock (1)			3.00			54.00
		Houblion	3.00	WH/Stock	06/20/2016 09:46:49	54.00
» Levure à bière (1)			6.00			1,500.00
» Malt d'orge (1)			99.00			59.40
» Récipient format 25cl bouteille en verre foncé (1)			3,000.00			150.00

Figure 100 : Affichage d'un rapport d'évaluation des stocks.

Complément d'informations :



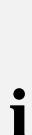
Dans la pratique, les employés du stock n'ont pas accès à l'application **Accouting** et inversement, les comptables n'ont pas accès à l'application **Inventory**. De ce constat, la valorisation des stocks peut être visualisée à partir des deux applications.

14.7.2.3. Créer la facture fournisseur PR03

Dans la majeure partie des cas de figure, le bon de livraison s'accompagne d'une facture. Elle est normalement envoyée au service comptabilité de la société et enregistrée dans le système informatique.

Pour se faire, il faut privilégier sa création à partir de l'application **Accounting / Purchases / Vendor Bills**.

Complément d'informations :



Odo propose de le faire directement à partir de l'application **Purchases** et dans le détail de la commande d'achats. Cependant, dans la réalité ce ne sont pas les acheteurs qui comptabilisent les factures mais les employés de la comptabilité. Par conséquent, il est préférable d'effectuer l'activité PR03 et PR04 à partir de la bonne application. Cela coïncide aussi avec les droits d'accès au PGI. Un comptable n'a probablement pas accès à l'application **Purchases** et inversement un acheteur n'a pas l'accès à l'application **Accounting**.

Une fois dans la bonne vue, il faut utiliser le bouton **Create** pour enregistrer une nouvelle facture fournisseur. Le système est assez intelligent pour offrir la possibilité de choisir une des commandes d'achats dans le statut **Purchase Order**. Cela s'effectue à partir de la liste déroulante **Add Purchase Order**. Il faut donc ajouter les bonnes références aux demandes d'achats et si l'il n'y a pas de modification à faire valoir, un clic sur le bouton **Validate** suffit à enregistrer la facture dans le système dans l'état facture ouverte **Open**.

Comptablement parlant, au niveau du bilan, deux choses sont comptabilisées, une créance dans le compte passif **2000 Créditeurs** est créditee et un actif dans le compte **1170 Impôt préalable : TVA...** est débité. Au niveau des charges, le compte **4200 Achats de marchandises destinées à la vente** est débité du montant hors taxes. Cette opération ne modifie aucunement l'état des stocks.

14.7.2.4. Comptabiliser le paiement fournisseur PR04

Dernière opération dans la réalisation du processus d'approvisionnement, le paiement des factures ouvertes.

Pour se faire, il suffit d'ouvrir les factures créées en amont et d'enregistrer un paiement avec un certain compte d'actif :

Payment Method	Bank (CHF)	Payment Date	06/20/2016
Payment Amount	1,763.40 CHF	Memo	BILL/2016/0002

Buttons at the bottom: VALIDATE (highlighted in green), CANCEL.

Figure 101 : Enregistrement d'un paiement.

Une fois qu'il est validé, ce paiement impacte directement plusieurs éléments. Premièrement, la facture passe dans un état payé **Paid**. Puis au niveau de la comptabilité, les écritures au journal Bank (CHF) ont lieu de la manière suivante :

BNK1/2016/0001

Journal	Bank (CHF)	Reference		
Date	06/20/2016	BILL/2016/0002		
JOURNAL ITEMS				
Account	Partner	Label	Debit	Credit
10201 Bank	Maltonne SA	SUPROUT/2016/0001	0.00 CHF	1,763.40 CHF
2000 Crédanciers	Maltonne SA	Vendor Payment: BILL/2016/0002	1,763.40 CHF	0.00 CHF
			1,763.40	1,763.40

Figure 102 : Détail de l'impact d'un paiement sur les comptes d'actif et de passif.

La créance est annulée en débitant le compte **2000 Crédanciers** et le compte de liquidité est crédité de la même somme. En regardant de plus près les deux éléments importants de la comptabilité, le bilan et le compte de pertes et profits, le constat est qu'à ce stade la société est bénéficiaire de la taxe sur la valeur ajoutée TVA car elle dispose d'un nouveau débiteur. En effet, la Confédération doit rembourser ce montant en fin d'année.

14.7.3. Production

Une fois que l'achat des matières premières est effectué et qu'elles sont disponibles en stock, la production peut débuter. La page qui liste les ordres de fabrication en attente, application **Manufacturing / Manufacturing Orders**, permet de visualiser les ordres de fabrication :

The screenshot shows a table titled "Manufacturing Orders" with the following data:

Reference	Scheduled Date	Product	Product Quantity	Total Hours	Total Cycles	Source Document	Status
MO00001	06/21/2016 02:00:00	[BW01A] Lager Beer 0.25	3,000.000	0.00	0.00	OP/00002	Ready to Produce

Below the table, there is a summary bar with the values: 3,000.000, 0.00, and 0.00.

Figure 103 : Affichage des ordres de fabrication.

Dans ce processus, il s'agit de lancer les ordres de fabrication qui sont dans l'état **Ready to Produce**, c'est la responsabilité de l'activité MA01. Ensuite, une fois que le temps nécessaire à la transformation des matières premières en produits finis est atteint, il faut valider la production. La confirmation se fait grâce à l'activité MA02.

14.7.3.1. Lancer des ordres de fabrication MA01

C'est seulement lorsque toutes les matières premières sont disponibles que l'ordre de fabrication passe du statut **Awaiting Raw Materials** à **Ready to Produce**. À ce moment-là, il devient possible d'effectuer les opérations de production. Pour lancer l'ordre de fabrication, il est nécessaire de se rendre dans le détail de l'OF via l'application **Manufacturing / Manufacturing Orders** et en cliquant sur l'ordre en question. À partir de ce formulaire, plusieurs points sont très intéressants :

- Les produits consommés, c'est-à-dire la liste des matières premières et la quantité associée.
- Les produits fabriqués, c'est-à-dire la liste des produits finis.

The screenshot shows two tables under the headings "Products to Consume" and "Products to Produce".

Product	Quantity
Eau traitée	750.000
Houblon	3.000
Malt d'orge	99.000
Levure à bière	6.000
Récipient format 25cl bouteille en verre foncé	3,000.000

Product	Quantity
[BW01A] Lager Beer 0.25	3,000.000

Figure 104 : Détails d'un ordre de fabrication.

Ces éléments sont directement liés aux mouvements de stock que l'on peut visualiser à la figure 087.

Pour renseigner le système sur le fait que la production est lancée, il faut uniquement appuyer sur le bouton **Mark As Started** dans le menu supérieur. L'effet de cette action est uniquement de passer l'ordre de fabrication de **Ready To Produce** à **Production Started**. Les stocks et la comptabilité ne sont pas impactés par cette opération.

14.7.3.2. Confirmer la production MA02

Après un certain laps de temps défini par la productivité des lignes à produire l'ordre de fabrication, ce même ordre doit être confirmé comme étant terminé.



Complément d'informations :

Le temps imparti à la production est défini en tant que contrainte au niveau du scénario utilisé.

Cela s'effectue en accédant à l'ordre de fabrication concerné, application **Manufacturing / Manufacturing / Manufacturing Orders** et sélection de l'ordre de fabrication pour lequel les opérations touchent à leurs fins. À partir de là, il est possible d'effectuer un clic sur le bouton **Produce** qui va actionner une demande de confirmation sur les éléments produits et ceux consommés :

Product	Quantity (in default UoM)
Eau traitée	750.000
Houblon	3.000
Malt d'orge	99.000
Levure à bière	6.000
Récipient format 25cl bouteille en verre foncé	3,000.000

Figure 105 : Confirmation de la production d'un ordre de fabrication. En confirmant la production avec le bouton Confirm, l'ordre de fabrication passe au statut terminé Done.

En terme comptable, plusieurs effets peuvent être identifiés. Tout d'abord, les écritures suivantes sont effectuées dans le journal **Stock Journal (CHF)** :

Date	Number	Partner	Reference	Journal	Amount	Status
06/20/2016	STJ/2016/0011			Stock Journal (CHF)	6,000.00 CHF	Posted
06/20/2016	STJ/2016/0010			Stock Journal (CHF)	150.00 CHF	Posted
06/20/2016	STJ/2016/0009			Stock Journal (CHF)	1,500.00 CHF	Posted
06/20/2016	STJ/2016/0008			Stock Journal (CHF)	59.40 CHF	Posted
06/20/2016	STJ/2016/0007			Stock Journal (CHF)	54.00 CHF	Posted
06/20/2016	STJ/2016/0006			Stock Journal (CHF)	112.50 CHF	Posted

Figure 106 : Liste des écritures comptables après production.

L'exemple ci-dessus fait apparaître les transactions **STJ/2016/0006 à 0010** qui concerne des écritures d'augmentation du compte **1260 Stocks de produits finis**. En effet, les matières premières disparaissent car elles sont consommées. Il est donc normal que le système diminue la valeur du compte

de valorisation. Ensuite, la transaction **STJ/2016/0011** débite le même compte de la valeur des produits fabriqués.

Limite :

! Réellement, cette opération a d'autres coûts financiers. En effet, de la main d'œuvre est nécessaire pour effectuer ces opérations. Dans l'état actuel de la configuration, cela n'est pas comptabilisé.

14.7.4. Vente

À présent, les produits vendables sont disponibles dans les stocks de la société. Les forces de vente peuvent prospectionner auprès de leurs clients pour réaliser de nouvelles ventes.

14.7.4.1. Changer les prix de vente SA05

Cette activité se réalise normalement avant de débuter les ventes et par la suite selon deux cas de figure :

- Les objectifs de vente ne sont pas réalisés.
- La production n'arrive pas à suivre la cadence.

Si l'un de ces deux évènements survient, il est nécessaire de changer les prix de ventes des articles.

Pour se faire, les vendeurs peuvent accéder à l'application **Sales / Sales / Products** et sélectionner l'article pour lequel un changement de prix de vente est nécessaire :

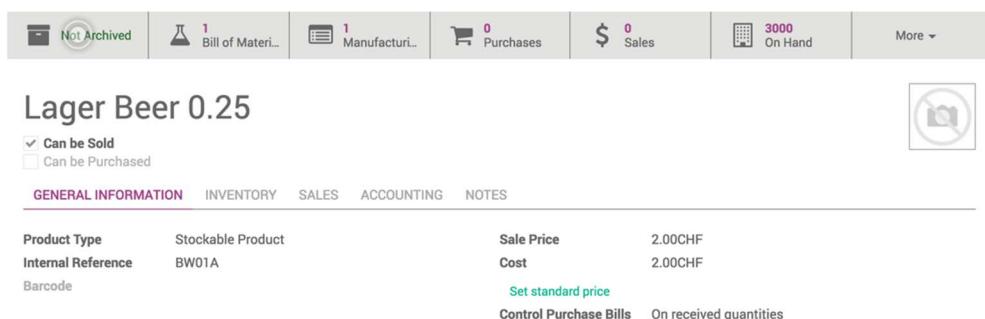


Figure 107 : Paramètres d'un produit vendable, l'article Lager Beer 0.25.

Pour effectuer un changement du prix de vente, il faut modifier la valeur du champ **Sale Price** en CHF.

14.7.4.2. Crée une vente SA01

Hypothèse :

En se référant au chapitre **Guide des opérations**, une vente fait suite à un devis et sa confirmation par le client.

Dans les exemples qui suivent, la vente est directement insérée dans le système en tant que commande ferme.

Pour enregistrer une vente, il faut utiliser l'application **Sales / Sales / Sales orders** et effectuer un clic sur le bouton de création :

New

Figure 108 : Entête d'une nouvelle commande effectuée pour le client [Suisse Est] Détailant N°1.

Odoo crée une nouvelle commande où il est nécessaire de renseigner le client dans l'entête et éventuellement les conditions et l'expiration de l'offre.

Suite à cela, il faut choisir les articles que le client convoite à l'aide du détail accessible dans la partie **Order Lines** :

ORDER LINES		OTHER INFORMATION					
Product	Description	Ordered Qty	Delivered	Invoiced	Unit Price	Taxes	Subtotal
+ [BW01A] Lager Beer 0.25	[BW01A] Lager Beer 0.25	100.000	0.000	0.000	4.00	TVA due à 8.0% (Incl. TN)	370.37 CHF
Add an item							
Setup default terms and conditions in your company settings.							
Untaxed Amount : 370.37CHF Taxes : 29.63CHF Total : (update) 400.00CHF							

Figure 109 : Liste d'articles dans une vente.

Finalement, il faut attribuer la vente à la bonne équipe en se rendant dans le détail **Other Information** :

Figure 110 : Autres informations dans une vente.

Dans cet exemple, le client étant un détaillant dans la zone Suisse Est, logiquement l'équipe qui conclue la vente est la **Swiss East Area**.

À ce stade, la vente peut être confirmée à l'aide du bouton **Confirm Sale**.

Une fois que la confirmation a lieu, seule la gestion du stock est impactée. Évidemment car les articles commandés doivent être livrés au client. Comme pour l'achat, un ordre de prélèvement est créé :

WH/OUT/00001

Figure 111 : Prélèvement de stock lié à une vente (SO001).

Le client **Partner** et le document d'origine **Source Document** y figurent. Suite à cela, les opérations et les mouvements de stock sont visualisables respectivement dans les listes **Operations** et **Initial Demand** du même formulaire. En se rendant dans les mouvements de stock, application **Inventory / Reports / Stock Moves**, le mouvement de stock est présent :

Product	[BW01A] Lager Beer 0.25	Picking Type	edu-odoosim-template: Delivery Orders
Quantity	100.000	Priority	Normal
Description	[BW01A] Lager Beer 0.25	Expected Date	06/20/2016 12:35:28
Origin			
Source Document S0001			
Transfer Reference WH/OUT/00001			
Creation Date 06/20/2016 12:45:38			
Procurement Group S0001			
Reserved Quants			
Availability 100.0 (reserved)			

Figure 112 : Détails d'un mouvement de stock lié à un prélèvement Transfer Reference et une vente Source Document.

14.7.4.3. Expédier la marchandise SA02

Dans l'optique où les stocks sont constitués à l'avance, il ne devrait pas y avoir de pénurie. Pour autant que l'analyse prévisionnelle ait été faite correctement. La livraison des articles peut donc être directement effectuée.

Hypothèse :



La livraison ne se gère pas au niveau des ventes. Par conséquent, le bon de livraison et le traitement de cette dernière doivent être effectués par le département de l'expédition.

Pour se faire, il faut accéder à l'application **Inventory / Operations / All Transfers** et repérer celui qui se réfère à la vente :

<input type="checkbox"/> WH/OUT/00001	Partner Locations/Customers	[Suisse Est] Détailant N°1	06/20/2016 12:35:28	S0001	Available
---------------------------------------	-----------------------------	----------------------------	---------------------	-------	-----------

Figure 113 : Prélèvement de stock pour la livraison relative à une vente.

Comme cité ci-dessus, la marchandise est normalement disponible **Available** et peut être expédiée. Pour le faire, un clic sur le bouton **Validate** dans le détail du prélèvement et une confirmation au message **You haven't set processed quantities. If you click apply, Odoo will process all quantities to do suffisent** à effectuer le transfert de la marchandise des entrepôts de la société vers le client.

Au niveau de la comptabilité, de nouvelles écritures s'effectuent :

STJ/2016/0012

Journal	Stock Journal (CHF)	Reference	WH/OUT/00001	
Date	06/20/2016			
JOURNAL ITEMS				
Account	Partner	Label	Debit	Credit
1260 Stocks de produits finis	[Suisse Est] Détailleur N°1	[BW01A] Lager Beer 0.25	0.00 CHF	200.00 CHF
3901 Variation des stocks de produits finis	[Suisse Est] Détailleur N°1	[BW01A] Lager Beer 0.25	200.00 CHF	0.00 CHF
			200.00	200.00

Figure 114 : Journal du stock et écritures comptables suite à une expédition d'articles chez un client.

À ce moment-là, en se référant au bilan et au compte de résultats, l'entreprise semble plus pauvre que précédemment. Ceci est tout à fait normal. Le système vient de créditer le compte d'actif **1260 Stocks de produits finis** et débiter le compte de produits **3901 Variation des stocks de produits finis** ce qui représente uniquement des charges. La société est donc plus pauvre du montant des écritures. Dans cet exemple, 200.00 CHF.

Les stocks eux aussi sont impactés par l'envoi des articles chez le client. Tout d'abord, il suffit de visualiser les états des différents éléments générés par l'étape précédente (SA01). Le prélèvement est dans l'état terminé **Done**. Le mouvement de stock est aussi terminé **Done**. Si on affiche le détail du mouvement **WH/OUT/00001**, le constat est que tout comme pour la réception marchandise, un Quant est produit :

Product	[BW01A] Lager Beer 0.25	Picking Type	edu-odoosim-template: Delivery Orders
Quantity	100.000	Priority	Normal
Description	[BW01A] Lager Beer 0.25	Date	06/20/2016 14:19:46
Origin			
Source Document	S0001		
Transfer Reference	WH/OUT/00001		
Creation Date	06/20/2016 12:45:38		
Procurement Group	S0001		
Moved Quants			
Product	Quantity	Location	Incoming Date
[BW01A] Lager Beer 0.25	100.00	Partner Locations/Customers	06/20/2016 12:00:29
			200.00

Figure 115 : Détails d'un mouvement de stock pour la livraison d'articles.

Complément d'informations :

Comme présenté par la figure 115, le mouvement **WH/OUT/00001** créé un **Quant**. En inspectant cette ligne, La constatation est que dans la colonne Location, un emplacement est renseigné par le système. Il s'agit là, d'un emplacement virtuel **Partner Locations/Customers** vers lequel tous les articles qui partent en expédition sont transférés. C'est le principe de double inventaire qu'Odoo a mis en place dans la gestion des stocks.



14.7.4.4. Facturer le client SA03

À l'expédition s'adjoint la facturation. Il s'agit là de la créer et de l'envoyer.



Hypothèse :

La facturation s'effectue par le département de la comptabilité et non pas par les ventes.

Pour effectuer cette opération, le comptable doit se rendre dans l'application **Accounting / Sales / Customers** et récupérer le bon client. Une fois qu'il est sélectionné, il suffit d'afficher son détail et d'utiliser le lien :

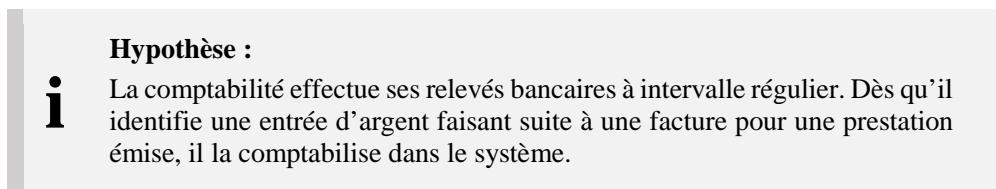


Figure 116 : Nombre de vente pour un client donné.

Là on accède à toutes les ventes qui lui sont liées. Il suffit de visualiser celles qui portent le statut **To Invoice** et cliquer dessus et dans le détail appuyer sur le bouton qui crée la facture **Create Invoice** et répondre à la question **What do you want to invoice ? Par Invoiceable lines (deduct down payments)**. Suite à cela, il suffit d'aller visualiser le rendu pour se rendre compte de la facture qui est créée par le système. Si l'on laisse la facture dans cet état, c'est-à-dire en brouillon **Draft**, rien ne s'effectue dans la comptabilité. Il est nécessaire de la valider avec le bouton prévu à cet effet **Validate**.

Dès qu'elle est validée, la somme hors-taxe va se mettre au débit du compte **1100 débiteurs**, les 8.0% de TVA au crédit du compte de dettes **2200 TVA due** et le compte **3200 Ventes de marchandises** est crédité de la valeur hors taxe. Au niveau des stocks, rien ne change à ce moment-là.

14.7.4.5. Comptabiliser le paiement client SA04



Pour comptabiliser le paiement dans le système, le service comptabilité de la société doit se rendre dans l'application **Accounting / Sales / Customer Invoices** et rechercher la facture correspondante à l'aide de la référence qui se trouve aussi sur le relevé bancaire. Pour se faire, il est tout à fait possible d'utiliser la barre de recherche que l'outil propose. Dès que la facture en question est repérée, il suffit d'afficher son détail en cliquant dessus et utiliser le bouton **Register Payment** :

Payment Method	Bank (CHF)	Payment Date	06/20/2016
Payment Amount	400.00	CHF	Memo INV/2016/0001

Developer Tools General Action View

VALIDATE CANCEL

Figure 117 : Comptabilisation d'un paiement client.

Ce formulaire enregistre un paiement avec un moyen de paiement spécifique **Payment Method** et pour une somme donnée **Payment Amount**. Pour le comptabiliser, il suffit de valider.

Cette opération affecte uniquement la comptabilité. Elle annule le compte **1100 débiteurs** en le créditant et débite le compte courant **10201 Bank** de la même somme. On se retrouve avec les mêmes chiffres au bilan.

14.8. Limites et recommandations

En parallèle des opérations, l'API est utilisée pour mettre directement à jour les objets manipulés. Dans un cadre comme celui-ci, les résultats démontrent qu'il est tout à fait réalisable de travailler ainsi. Par conséquent, il faut se baser sur le référentiel Analyse des objets manipulés et les recommandations faites pour l'implémentation des activités simulées Analyse de l'aspect pédagogique au niveau du simulateur.

15. Conclusion

Cette partie conclue ce travail de Bachelor. Elle passe en revue l’expérience acquise, les limites et recommandations générales et donne la parole à l’auteur qui fait part de sa conclusion personnelle.

15.1. Retour d’expérience

Après quatorze semaines d’analyse et de développement, plusieurs éléments sont qualifiables de spécifications. Lesquelles peuvent servir de bases et d’appuis pour les prochaines étapes menant à la réalisation finale du jeu sérieux OdooSIM.

Le scénario axé PME est développé et présenté au travers d’un descriptif complet présent dans cet ouvrage au chapitre Brewery & Co.. Il est aussi présenté au format Power Point d’une manière plus simple et plus ludique. Cette présentation est accessible dans les annexes Votre Job chez Brewery & Co.. Le scénario est baptisé Brewery & Co. et sert de contexte à l’exécution du jeu. Il présente une société qui fabrique des produits brassicoles et les commercialisent dans trois régions de Suisse.

Le simulateur OdooSIM en version proof of concept est développé et fonctionnel. Il met en œuvre une partie limitée aux ventes. Cette version est architecturée de sorte à configurer le PGI par rapport au scénario, c’est-à-dire de configurer le master data. Elle gère la simulation des jours de jeu durant lesquels elle génère des ventes intelligentes. En effet, les demandeurs sont sensibles aux effets de variation de prix. Elle classe les équipes par rapport au chiffre d’affaires qu’elles génèrent et restitue le résultat au professeur, lequel peut faire un retour aux participants. Cette partie des ventes peut être reprise et implémentée dans la version complète. Toutefois, certaines recommandations existent et sont largement invoquées dans cet ouvrage.

Les problématiques soulevées dans ce travail sont exposés clairement et plusieurs pistes de qualités sont données afin d’orienter les futurs candidats à la réalisation d’OdooSIM.

15.2. Limites

Ce travail a soulevé plusieurs limites à la démarche de réalisation d'un nouveau jeu sérieux sur Odoo. Ces limites sont catégorisées au travers des chapitres relatifs à l'accès au service, l'API et aux performances.

15.2.1. Accès SaaS

Il est démontré que cette méthode d'accès n'est pas la meilleure possible dans l'utilisation qui est faite de la solution. Le fait que les bases de données ne peuvent être gérées par programmation impose trop de travail répétitif au corps professoral. De plus, comme toutes activités répétitives, le taux d'erreurs est trop important.

De plus, ce mode de consommation ne permet pas de disposer d'environnement d'intégration. En effet, le simulateur possède beaucoup de portion de code dont le mapping avec les éléments du PGI sont renseignés en dur. Au vu du fait que les montées de version ne sont pas gérées par les concepteurs du simulateur, aucune possibilité de tester les nouvelles fonctionnalités et la compatibilité avec la version du simulateur n'est réalisable. Ce qui peut provoquer d'un jour à l'autre un fonctionnement biaisé du simulateur.

15.2.2. Web Service API

La nature de cette API repose sur le protocole HTTPS et XMLRPC ne convient pas forcément au besoin de rapidité qu'OdooSIM nécessite. Il est démontré dans le chapitre Mesure temps des I/O que le temps nécessaire à l'accès aux informations est rendu trop long par rapport au nombre de transactions à effectuer durant un cycle opérationnel.

15.2.3. Performances

Les performances de l'API ne sont pas assez bonnes pour espérer entrer dans des temps abordables. Il est nécessaire de trouver des solutions afin de diminuer drastiquement les échanges et d'accélérer le temps nécessaire à l'exécution de toutes les activités que le simulateur doit gérer.

15.3. Recommandations

Ces recommandations d'ordre général sont ici en tant que complément à celles qui se trouvent dans les différents chapitres de cet ouvrage. Il convient de les lire en tant que complément.

15.3.1. Accès au service Odoo

Il est recommandé d'héberger personnellement la solution **Odoo Community** pour les besoins d'OdooSIM. De ce fait, l'accès au service se fait en mode **On-premise**. En effet, même si les résultats obtenus par les tests ne parlent pas en la faveur de la version **Community** sur un réseau local, la capacité du serveur qui hébergeait Odoo au moment des tests ne peut qu'être tenu responsable des mauvais temps calculés.

Le fait d'héberger la solution permet d'avoir un contrôle sur les montées de version et de décider lorsqu'on met en production une nouvelle version d'Odoo. Ceci après avoir effectué des essais dans un environnement d'intégration où l'on peut tester la nouvelle version avec le simulateur et corriger les bugs éventuels.

De plus, des essais ont été faits sur cette version d'Odoo et les principaux processus dont a besoin Brewery & Co sont intégralement gérés.

15.3.2. Performances

L'utilisation qui est faite de l'API dans le cadre d'OdooSIM ne semble pas correspondre avec sa raison d'être. En effet, lorsque les appels avec l'API sont effectués quasiment en temps réel, le résultat est que le temps nécessaire à chaque échange est trop lent. De ce fait, il convient de trouver une alternative.

Plusieurs pistes peuvent servir de solution. Premièrement, tester la mise en place d'un Thread par équipe. Ce Thread a pour mission d'exécuter indépendamment des autres la chaîne de valeur pour une équipe. Il y a autant de Thread qu'il y a d'équipes. Ainsi, les opérations sont exécutées parallèlement et le temps des opérations n'est donc plus multiplié par le nombre d'équipes. Pour que cela soit réalisable, il faut garder une base de données par équipe.

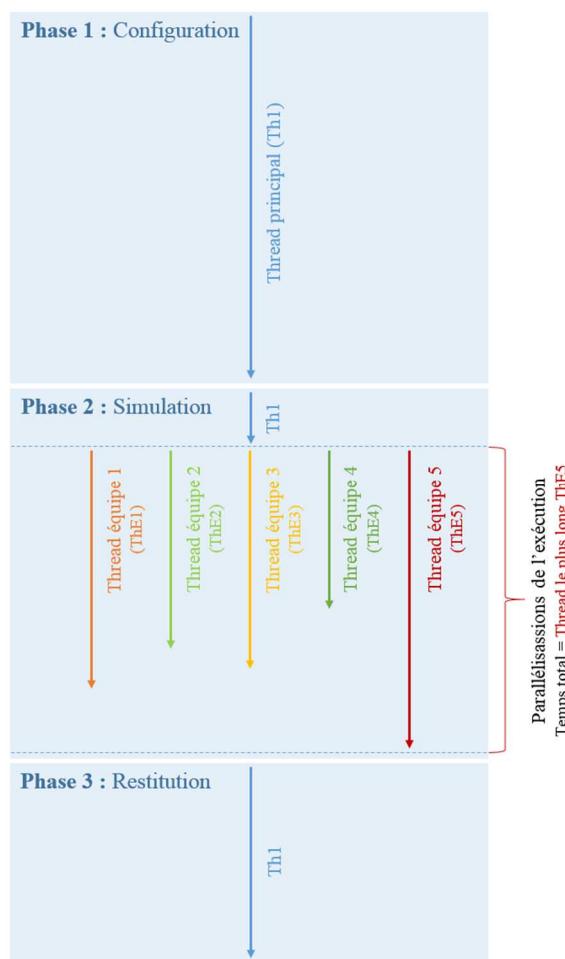


Figure 118 : Programmation concurrente au niveau de la phase 2.

De cette manière, la phase 2 (où le temps d'exécution est critique) prend le temps de l'exécution de Th1 pour créer communément la demande et l'offre. Puis l'exécution est parallélisée en autant de Thread qu'il y a d'équipe. Le but étant de raccourcir le temps global de la phase 2.

En effet, dans ce contexte-là, le Thread de l'équipe ayant le plus d'opérations à effectuer dicte le temps d'exécution tandis que les autres attendent qu'il termine. Une fois que le Thread le plus lent est terminé, une nouvelle itération peut débuter. La figure 118 ci-dessus, l'économie de temps observée peut être de :

$$\text{Gain de temps} = (\text{Th1} + \text{ThE1} + \text{ThE2} + \text{ThE3} + \text{ThE4} + \text{ThE5}) - (\text{Th1} + \text{ThE5})$$

Cette solution peut amener une réelle plus-value à OdooSIM. C'est la piste la plus sérieuse qui est recommandée. D'autant plus, que des tests ont pu démontrer que les appels à l'API peuvent être réalisés de manière simultanée sans que cela plante.

La seconde est de développer un middleware responsable de traiter directement avec le PGI. Il remplace l'API et utilise un protocole plus rapide qu'HTTPS. Par exemple, grâce à des sockets. Cette recommandation est purement hypothétique. De plus, pour que cela fonctionne, il faut pouvoir agir directement sur le code du PGI, chose qui est réalisable uniquement avec une version installée localement, c'est-à-dire **Odoo Community**.

15.3.3. Utilisation des workflows

Bien qu'Odoo semble faire disparaître les workflows dans les nouvelles moutures de son progiciel, ce sont des solutions qui doivent permettre d'apporter la solution au problème de performance.

En effet, les workflows sont représentés comme étant des graphes où un nœud représente une activité et où les arcs sont des transitions logiques. De ce fait, il est possible de réaliser des workflows qui soient capables de faire tout ce que l'on désire. Exécuter des tâches successivement et gérer les transitions grâce à des conditions logiques telles qu'elles existent dans tous langages informatiques.

Ces éléments peuvent soulager drastiquement le simulateur. Ils peuvent contenir toute la logique chronologique des activités et gérer les transitions dans le processus. Par exemple, les dates peuvent très bien être gérées par ce biais-là. Avec cette solution, le simulateur ne fait que de créer les objets puis les workflows associés s'exécutent côté serveur :

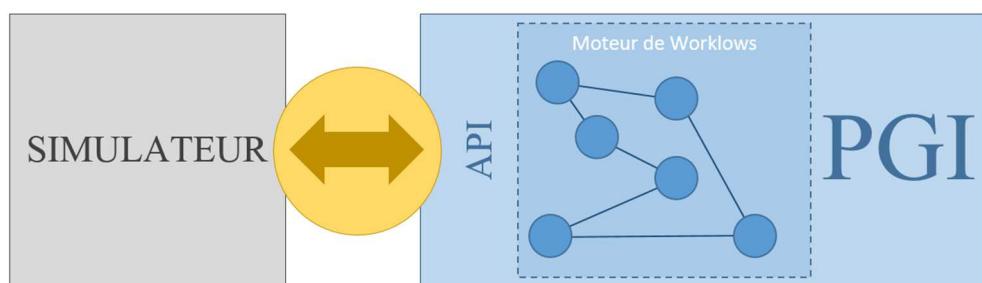


Figure 119 : Le moteur de workflows se trouve sur le PGI.

Dans Odoo, les workflows sont accessibles grâce au mode développeur actif. Le workflow associé à l'objet **mrp.product**, qui représente les ordres de fabrication présenté de la manière suivante :

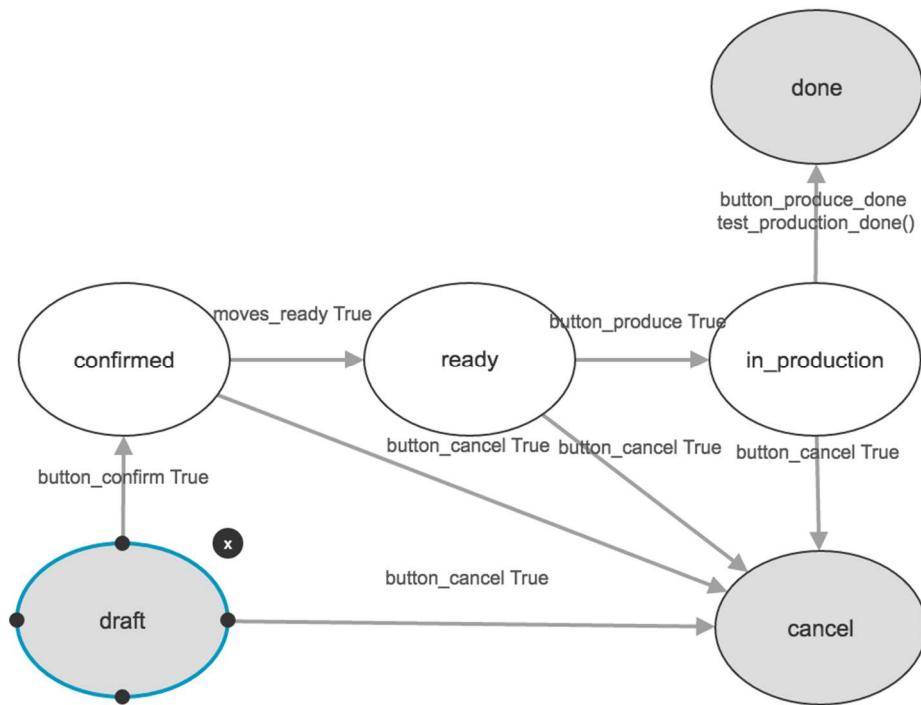


Figure 120 : Workflow associé à l'objet `mrp.production`.

Le processus de fabrication tel qu'il est présenté au chapitre La production est bien visualisable. Toutes les étapes correspondent à un nœud. Les transitions sont identifiables grâce aux arcs. Les labels apposés dessus informent au sujet de la condition de passage d'un nœud à l'autre.

En ouvrant le détail du nœud **in_production**, les propriétés et les transitions IN/OUT sont visualisables :

The screenshot shows the 'Open: Activity' dialog for an activity named 'in_production'. The top section displays basic properties: Name (in_production), Workflow (mrp.production.basic), Kind (Function), and Flow Start/Stop checkboxes. Below this, there are tabs for 'PROPERTIES' and 'TRANSITIONS'. The 'PROPERTIES' tab shows a Subflow section with a 'Signal (subflow.*)' field and a Conditions section with Split Mode (Xor) and Join Mode (Xor). The 'TRANSITIONS' tab lists three outgoing transitions:

Source Activity	Signal (Button Name)	Condition
ready	button_produce	True
done	button_produce_done	test_production_done()
cancel	button_cancel	True

A tooltip for the 'button_produce' signal provides details: Field: out_transitions, Object: workflow.activity, Type: one2many, Relation: workflow.transition.

Figure 121 : Détail de l'activité in_production.

Ce qui en résulte est qu'il est tout à fait possible de modifier le comportement d'Odoo. Par exemple, pour soulager le simulateur, il peut être utile de créer un workflow qui lors de la création de l'objet **sale.order** va effectuer d'une traite le processus de vente. Ainsi, les transactions actuellement nécessaires à la concrétisation de ce processus sont réduites à une seule, celle de création de la vente.

Dans cette nouvelle version d'Odoo Online, les seuls workflows restants sont ceux liés aux modèles **mrp.product** et **account.invoice**. Il est recommandé de ce basé sur ces derniers pour développer ceux nécessaires à OdooSIM.

15.3.4. Gestion des dates virtuelles

Comme il en est fait état au chapitre Dates virtuelles versus dates réelles, les dates virtuelles posent problèmes car elles ne reflètent pas la réalité du temps configuré sur le serveur. Pour gérer cette problématique, il convient de modifier toutes les dates faisant du sens et ce pour chaque objet généré. Cette constatation est en parfaite contradiction avec les précédentes recommandations. En effet, un des futurs enjeux du projet OdooSIM est de diminuer au maximum les échanges avec l'API durant la phase de simulation. Hors, si la responsabilité de changer toutes les dates doit être gérée à ce moment-là, le simulateur n'est pas assez rapide et le jeu inutilisable.

15.4. Postface personnelle

Ce travail aura suscité beaucoup d'incertitudes et une énorme débauche d'énergie. Malgré tout, je suis content d'avoir pu travailler sur un projet intéressant et au spectre très large. Pour rappel, je n'avais pas eu le choix quant au sujet attribué et je ne l'aurais pas sélectionné d'une manière intentionnelle. Il me semblait trop abstrait. Mais afin de rebondir sur ma première affirmation, je ne regrette en rien d'être tombé dessus. Le fait d'avoir dû mettre à profit toutes sortes de compétences était très appréciable. Je suis passé par de la comptabilité, de la micro-économie, de la gestion d'entreprise, du PGI, du BPMN et de la programmation.

Pour parler du parcours, j'ai débuté en ne possédant que peu de compétences dans le domaine business. J'avais suivi les cours d'ERP que l'école dispense mais ce n'était que très peu semblable à ce que j'ai pu découvrir en effectuant mes recherches dans le cadre d'OdooSIM. J'ai investi beaucoup de temps à essayer de comprendre de quelle manière les entreprises de production fonctionnent. Finalement, je crois pouvoir affirmer que mes compétences sont sorties grandies avec un nouvel intérêt pour l'analyse métier.

En termes de points positifs, je pense avoir mené à bien ce travail. Souvent dans le trouble, régulièrement en train de ressentir du désespoir par rapport à certains moments difficiles, j'ai réussi à trouver les solutions nécessaires à l'évolution dans les démarches. J'ai beaucoup pu compter sur les gens de mon entourage qui m'ont permis de repousser mes limites. Mon directeur de travail, Boris Fritscher, que je remercie une nouvelle fois, m'a régulièrement aiguillé et donné des retours positifs. Au final, j'ai réussi à aboutir à une version du simulateur fonctionnelle avec une bonne vision d'ensemble que j'ai essayé au mieux de retrancrire dans les différents chapitres de cet ouvrage. Je pense donner les pistes nécessaires à toute personne qui continuera le projet OdooSIM. Je suis juste déçu de ne pas pouvoir continuer à travailler sur ce projet car dans les derniers temps du travail, j'ai vraiment eu un déclic qui j'en suis sûr me permettrait d'atteindre très vite de nouveaux résultats de qualité. De ce fait, c'est frustrant de ne pas pouvoir accorder encore d'avantage de temps pour réaliser de nouvelle version de ce jeu.

Principalement, l'outil de développement que j'ai utilisé dans Google Chrome est très puissant et donne accès à l'exactitude des transactions que chaque action utilisateur génère. C'est dommage que je n'ai pas trouvé cela avant. Mais comme le travail consistait à explorer ce domaine, et de ce fait à soulever des problématiques plus profondes, je me dis que c'est toute à fait normal d'avoir évolué ainsi.

Cependant, les erreurs commises permettent de tirer des leçons et de ne plus les reproduire. Certes, elles sont à éviter mais je ne prétends pas être à un niveau qui me permette de ne pas en faire. De ce fait, je reviens sur les éléments qui selon moi n'ont pas fonctionné durant ce travail. Je pense avoir perdu trop de temps à me questionner sans forcément produire du concret. Certes, j'ai énormément utilisé le papier et le crayon mais ce n'était que des croquis et des notes que je n'ai pu reprendre pour constituer de la documentation. Ceci vient du fait que je suis de nature perfectionniste. De ce fait, être plongé dans un environnement que je ne connaissais pas m'a beaucoup freiné. L'exemple le plus marquant qui me revient, est celui du scénario. Son développement m'a tellement posé de problèmes tant il y avait des éléments de questions dont je n'avais pas les réponses qu'il m'a fallu énormément de temps pour aboutir sur un résultat.

Cependant, d'une manière générale, je suis relativement satisfait de ce que j'ai réussi à produire durant ce travail. Je voudrais encore ajouter quelque chose pour mettre un point d'honneur à cette belle et exigeante expérience. Une citation qui m'est venue suite aux leçons que j'ai eu la chance de retirer tout au long de ce travail :

« La perfection ne peut être atteinte du premier coup. Toute idée peut être discutée, repensée et améliorée. Cependant, fixes-toi des échéances et sois confiant qu'en appliquant cette démarche, ton travail sera toujours de qualité supérieur car sans cesse autocritiqué ! ».

16. Bibliographie

- ALLAL-CHERIF, Oihab, MAKHLOUF, Mohamed et BAJARD, Armand, 2014. Les serious games au service de la gestion des ressources humaines : une cartographie dans les entreprises du CAC40. In : *Systèmes d'information & management* [en ligne]. 2014. Vol. 19, n° 3, p. 97-126. DOI 10.3917/sim.143.0097. Disponible à l'adresse : <http://www.cairn.info/revue-systemes-d-information-et-management-2014-3-page-97.htm>.
- ASSOCIATION SUISSE DES BRASSERIES, 2016a. La fabrication de la bière. In : [en ligne]. 2016. [Consulté le 11 mai 2016]. Disponible à l'adresse : <http://bier.ch/fr/autour-de-la-biere/le-brassage/fabrication/>.
- ASSOCIATION SUISSE DES BRASSERIES, 2016b. Le marché suisse de la bière en chiffres. In : [en ligne]. 2016. [Consulté le 11 mai 2016]. Disponible à l'adresse : <http://bier.ch/fr/chiffres-cles/saviez-vous/>.
- ASSOCIATION SUISSE DES BRASSERIES, 2016c. Les matières premières. In : [en ligne]. 2016. [Consulté le 12 mai 2016]. Disponible à l'adresse : <http://bier.ch/fr/autour-de-la-biere/le-brassage/ingredients/>.
- BAUDET, Cédric, TERMINE, Francesco et PLUMEZ, Julien, 2015. Co-design multidisciplinaire : un cas de ludification touristique selon une approche de Design Science. In : . 2015. p. 1-20.
- BRIOL, Patrice., 2008. *BPMN : the business process modeling notation pocket handbook*. S.l. : Patrice Briol. ISBN 9781409202998.
- FRITSCHER, Boris, DE SANTO, Alessio et HES-SO, Haute Ecole de Gestion Arc, 2016. Description du travail de Bachelor OdooSIM. In : . 2016. p. 1.
- GALEUCHET, Stève, JEANNERET, Loïc et HES-SO, Haute Ecole de Gestion Arc, 2014. Découverte des ERP. In : . 2014. p. 51.
- GENDRON, Bruno, 2012. *L'essentiel de la Micro-économie*. Lextenso é. S.l. : s.n. ISBN 978 - 2 - 297 - 02506 - 5.
- IONESCU, Calin, 2014. Microéconomie 1. In : *Support de cours filière EE*. 2014. p. 286.
- LÉGER, Pierre-Majorique, FELDSTEIN, Harvey D, BABIN, Gilbert, CHARLAND, Patrick, ROBERT, Jacques et LYLE, Derick, 2011. Business simulation training in information technology education: Guidelines for new approaches in IT training. In : *Journal of Information Technology Education* [en ligne]. 2011. Vol. 10, p. 39-53. Disponible à l'adresse : <https://dbproxy.udallas.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=60635511&site=ehost-live&scope=site>.
- LOGISTIQUE POUR TOUS.FR, 2016. Flux poussé ou Flux tiré ? That is the question ! In : [en ligne]. 2016. [Consulté le 11 mai 2016]. Disponible à l'adresse : <http://logistique-pour-tous.fr/flux-pousse-ou-flux-tire/>.
- MICHAUD, L. & ALVAREZ, J., 2008. Seriousgames: advergaming, edugaming, training. In : . 2008.
- ODOO S.A, 2016. Comparatif des ERPs pour PME: Microsoft Dynamics, Sage & Odoo. In : . 2016. p. 16.

PARAÎTRE, De, 2010. « SERIOUS GAME révolution pédagogique ». In : . 2010.

THOMMEN, Jean-Paul, 2007. *Introduction à la gestion d'entreprise*. 4e édition. S.l. : s.n. ISBN 9783039090280.

WIKIPÉDIA, 2015. Licence publique générale limitée GNU. In : 2015-07-15 [en ligne]. 2015. [Consulté le 19 juin 2016]. Disponible à l'adresse : https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_limit%C3%A9e_GNU.

WIKIPÉDIA, 2016. Odoo. In : 2016-05-11 [en ligne]. 2016. [Consulté le 23 mai 2016]. Disponible à l'adresse : <https://fr.wikipedia.org/wiki/Odoo>.

Partie Administrative

1. Modalités

Les modalités peuvent être lues dans les annexes :

[**/Administratifs/TB.AT.Modalités.pdf**](#)

2. Descriptif du mandat

Le descriptif du travail de Bachelor OdooSIM Simulation d'entreprise sur PGI peut être lu dans les annexes :

[**/Administratifs/TB.AT.Descriptif.pdf**](#)

3. Demande de ratification

La problématique, la démarche et les résultats à atteindre ont été définis dans la demande de ratification. Elle est accessible dans les annexes :

[**/Administratifs/TB.AT.Demande de ratification.pdf**](#)

4. Rapport hebdomadaire de travail

Les rapports hebdomadaires de travail ne comportent que très peu d'informations. Cependant, il convient de les lire en tant qu'éléments complémentaires au journal de travail. Ils sont accessibles dans les annexes :

[**/Administratifs/TB.AT.RHTx.pdf.**](#)

5. Journal de travail

Tout le détail des journées de travail est inscrit dans le document :

[**/Administratifs/TB.AT.JT.pdf**](#)

Partie Annexe

1. Guide de déploiement du simulateur

Pour déployer le simulateur OdooSIM, la procédure se situe dans un document à part disponible dans les annexes :

/Annexes/TB.AT.OdooSIM.pdf

2. Guide du professeur

Ce guide pratique permet à n'importe quel professeur de configurer une nouvelle session de jeu et de la lancer grâce au simulateur OdooSIM.

2.1. Créer le template

Afin d'être opérationnel plus rapidement, il est nécessaire de créer une base de données modèle.



Complément d'informations :

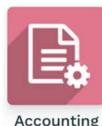
Pour créer une nouvelle base de données par équipe, il suffit de dupliquer la base de données modèle sans avoir à passer par le processus d'activation.

2.1.1. Crédit de la base de données

Connectez-vous à votre compte Odoo affilié au programme éducatif. Une fois que vous y êtes, rendez-vous à la page vous permettant de gérer vos bases de données : <https://accounts.odoo.com/my/databases/manage> et utilisez le bouton :

+ CREATE A NEW DATABASE

En tant que première application, sélectionnez **Accounting** :



Entrez les informations générales de sorte à ce que le formulaire s'apparente à celui-ci :

Odoo Accounting
Instant access.

Name
Brewery & Co. - Modèle OdooSIM

Email
votre-adresse-email@he-arc.ch

Company Name
edu-breweryandco-i

Phone Number

Country
Switzerland

Language
English

Company size
< 5 employees

Primary Interest
I am a teacher or a student

START NOW

Complément d'informations :

Le champ **Company Name** doit être modifié par un nom qui n'est pas utilisé sur la plateforme odoo.com. Il est recommandé d'incrémenter le nombre **i**. De plus, il est indispensable que le préfixe **edu-** soit utilisé.



Afin que la base de données soit valide durant une période de maximum 10 mois, il vous faut vous affiliez au programme éducatif. Pour se faire, veuillez-vous référer au document **EN_Odoo 9 Guidelines Education Oct15.pdf** disponible dans les annexes.

Patinez quelques instants afin que la configuration de la nouvelle base de données soit effective :

Sit tight! We're creating your account.

You will be redirected in a few seconds.

Initializing your company

2.1.2. Activation

Une fois que le processus de création est terminé, vous vous trouverez sur la page d'accueil de la nouvelle base de données. Vous devez apercevoir le message suivant :

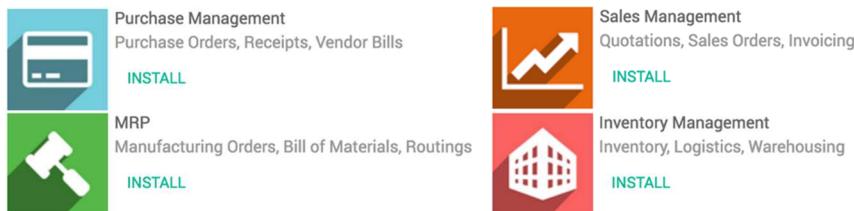
 Activation pending! Your database will expire in 4 hours [Activate by email](#)

Connectez-vous à votre e-mail (celui que vous avez précédemment renseigné) et cliquez sur le lien vous permettant d'activer la base nouvellement créée. Le message est remplacé. Le nouveau indiquant que la base de données est gratuite pour autant que vous utilisez une seule application. Ne prenez pas en compte cette information. En effet, si vous êtes enregistré au programme éducatif et que vous avez prefixé la nouvelle base à l'aide de **edu-** vous avez accès gratuitement à toutes les applications.

2.1.3. Configuration des applications

Pour que le modèle soit compatible avec les besoins du scénario, il vous faut installer les applications nécessaires.

Allez dans l'application **Apps** et installez à tour de rôle les applications suivantes en cliquant simplement sur le bouton **Install** :



2.1.4. Configuration de la valorisation des stocks

Le simulateur dans sa version proof of concept ne gère malheureusement pas la configuration de la valorisation financière des stocks de l'entreprise. Il faut donc le faire manuellement.

Rendez-vous dans les paramètres de l'inventaire, application **Inventor / Configuration / Settings** et modifiez le paramètre **Accounting / Inventory Valuation** :

Accounting	
Landed Costs	<input checked="" type="radio"/> No landed costs <input type="radio"/> Include landed costs in product costing computation
Inventory Valuation	<input type="radio"/> Periodic inventory valuation (recommended) <input checked="" type="radio"/> Perpetual inventory valuation (stock move generates accounting entries)

Enregistrez grâce au bouton **Apply** se situant dans la partie supérieure de la vue.

Actuellement, la valorisation ne fonctionne pas encore. Il faut se rendre une nouvelle fois dans l'application **Inventory / Configuration / Products / Product Categories** et éditer celle qui porte le nom **All**. Dans la partie **Account Stock Properties**, veuillez entrer les paramètres qui suivent :

Account Stock Properties	
Stock Input Account	3901 Variation des stocks de produits finis
Stock Output Account	3901 Variation des stocks de produits finis
Stock Valuation Account	1260 Stocks de produits finis
Stock Journal	Stock Journal (CHF)

La partie **Inventory Valuation** doit aussi être paramétrée de la manière suivante :

Inventory Valuation	
Costing Method	Standard Price
Inventory Valuation	Perpetual (automated)

Comme tous les produits sont par défaut attachés à la catégorie **All**, tous les mouvements de stock sont à présent comptabilisés.

2.1.5. Configuration du compte d'administration

La configuration touche presque à sa fin. Il reste à configurer le mot de passe du compte d'administration. Pour se faire, allez dans l'application **Settings / Users / Users** et cochez le compte nommé **Administrator**. Puis, à l'aide du menu, vous devez faire **Action / Change Password** :

Name	Login	Language	Latest connection
<input type="checkbox"/> Administrator	@he-arc.ch	English	06/24/2016 10:18:45

Dans le champ **New Password** renseignez un mot de passe à retenir :

User Login	New Password
@he-arc.ch	*****

Dès que vous êtes sûr de votre choix et que vous vous en souvenez, cliquez sur **Change Password**.

Complément d'informations :

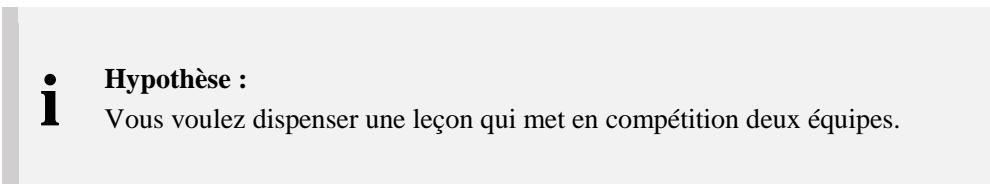


Ce mot de passe est nécessaire pour le fonctionnement du simulateur. En effet, lorsque le simulateur appelle l'API, il s'authentifie auprès d'odoo.com grâce à ce compte.

2.2. Configurer une nouvelle partie

Lorsque vous voulez débuter une nouvelle session de jeu, vous devez commencer par accéder à la gestion de vos bases de données sur le portail odoo.com.

2.2.1. Dupliquer le modèle



Une fois que vous êtes connecté sur votre espace privé et que vous vous êtes rendu sur la page de gestion des bases de données, vous devez voir apparaître le modèle précédemment créé :

The screenshot shows a web-based interface titled "Manage Your Databases". At the top, there's a header bar with the text "Manage Your Databases". Below the header, there's a table-like structure with one row. The first column contains the database name "edu-breweryandco-0" and its version "9.saas-10e". The second column shows the status "edu-breweryandco-0". The third column contains three buttons: "DUPLICATE", "RENAME", and "BACKUP".

Utilisez le bouton Duplicate deux fois pour copier le modèle et créer les bases de données nécessaires à votre nouvelle session de jeu. Lors de la duplication, odoo.com vous demande le nom qu'il doit attribuer à la copie. Il est recommandé de nommer les nouvelles bases de données de la manière qui suit :

- **edu-votrenomvotreprenom-ddmmmyy-classe-équipe**

Par exemple :

- **edu-anthonytomat-240616-3igptb-teama**
- **edu-anthonytomat-240616-3igptb-teamb**

Dès que le nom est spécifié, il est juste nécessaire de valider la duplication et après quelques instants, la base de données est prête à l'emploi. Répétez cette action autant de fois qu'il y a d'équipe qui prennent part à la session de jeu.

2.2.2. Paramétrer le fichier de configuration

Cet élément est obligatoire pour que le simulateur puisse fonctionner. Il est nécessaire de le remplir avec rigueur.

2.2.2.1. Les informations d'authentification

Le compte qui est utilisé par le simulateur pour atteindre l'API est celui qui est associé à votre compte d'administration des bases de données.

Il faut le spécifier à cet endroit dans le fichier de configuration **settings.xml** :

```
<odoor protocole="https" dns="odoo.com">
<account>VOTRE ADRESSE EMAIL</account>
<password>VOTRE MOT DE PASSE</password>
```

2.2.2.2. Les équipes et les participants

Pour chaque équipe, il est nécessaire de dupliquer à l'intérieur du nœud **companies**, le nœud **company** autant de fois qu'il y a d'équipe.

```
<company name="">
  <erp database="" />
  <players>
    <player login="">
      <name></name>
      <roles>
        <role>
          <name></name>
          <value></value>
        </role>
      </roles>
    </player>
  </players>
  <bank>
    <name>VUBS SA</name>
  </bank>
  <shareholder>
    <name>Breweries Holding Corp.</name>
  </shareholder>
</company>
```

Dès que cela est fait, vous devez entrer le nom de l'équipe dans l'attribut **name** du nœud **company**, l'adresse de l'instance dans l'attribut **database** du nœud **erp** et autant de fois qu'il y a de participant, dupliquer la balise **player** à l'intérieur de **players**.

Pour les besoins de l'exemple, vous devriez arriver à ce résultat :

```
<companies>
    <company name="Team A">
        <erp database="edu-anthonytomat-240616-3igptb-teama"/>
        <players>
            <player login="anthony.tomat@gmail.com">
                <name>Anthony Tomat</name>
                <roles>
                </roles>
            </player>
        </players>
        <bank>
        <shareholder>
    </company>
    <company name="Team B">
        <erp database="edu-anthonytomat-240616-3igptb-teamb"/>
        <players>
            <player login="boris.fritscher@he-arc.ch">
                <name>Boris Fritscher</name>
                <roles>
                </roles>
            </player>
        </players>
        <bank>
        <shareholder>
    </company>
</companies>
```

2.2.3. Briefe les participants

Une fois arrivé à cette étape, il est nécessaire de donner vos instructions aux participants. Il est recommandé de procéder ainsi :

1. Distribuer le document **TB.AT.Brewery and Co.pdf** (se référer au chapitre Jouer au proof of concept).
2. Répondre aux éventuelles questions.
3. Donner les informations sur le nombre de rounds, le nombre de jours ainsi que le temps de simulation par jour.
4. Vérifier une ultime fois que tout le monde est prêt à démarrer.

Une fois que ces étapes sont effectuées, vous pouvez démarrer le simulateur.

2.2.4. Démarrer le simulateur

Le simulateur se démarre en effectuant un double-clic sur le fichier odoosim.jar. Une fois que cela est fait, vous obtiendrez des instructions à l'écran qui vous permettront de lancer successivement les différentes phases en accord avec ce que vous avez spécifié dans le fichier de configuration **settings.xml**.

3. Votre Job chez Brewery & Co.

La description du scénario est disponible au chapitre Brewery & Co. de cet article ou en annexe dans le document :

/Annexes/TB.AT.Brewery and Co.pdf

4. Jouer au proof of concept

L'aide aux opérations pour la version proof of concept peut être trouvée dans le document en annexe :

/Annexes/TB.AT.JobAids Brewery and Co.pdf

FIN