

RÉPUBLIQUE DU SÉNÉGAL



Un Peuple - Un But - Une Foi

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR,  
DE LA RECHERCHE ET DE L'INNOVATION



*Département génie informatique et télécommunications*

# Projet Modélisation Stochastique

DIC1 GIT 2020 - 2021



**File d'attente**

## Membres du projet

Falilou Fall	Elève-ingénieur à l'E.P.T	ffalilou@ept.sn
Mohamet Tall	Elève-ingénieur à l'E.P.T	tmohamet@ept.sn
Mame Diarra Sow	Elève-ingénieur à l'E.P.T	smamediarra@ept.sn

Professeur :

Mr Thiongane

Ce projet entre dans le cadre des projets de classe en Modélisation Stochastique dans le Génie Informatique et Télécommunication (G.I.T.) de l'Ecole Polytechnique de Thiès (E.P.T).

# Sommaire

<b>Partie 1 : Problématique .....</b>	<b>4</b>
<b>Partie 2 : Solutions .....</b>	<b>5</b>
<b>Partie 3 : Gestion Client A .....</b>	<b>6</b>
<b>Partie 4 : Gestion Client B.....</b>	<b>8</b>
<b>Partie 5 : Outils et Bibliothèques.....</b>	<b>10</b>
<b>Partie 6 : Conclusion.....</b>	<b>11</b>

## Partie 1 : Problématique

Il s'agit de modéliser une succursale d'une banque travaillant de 10H à 16H par période de 2heures. Nous avons deux types de clients, ceux de type A qui se présentent à un guichet pour une transaction et sont servis par des Caissiers ou Conseillers dans certains cas et ceux de type B qui ont un rendez-vous avec un conseiller bien donné. Les clients de type A arrivent suivant un processus de poisson et leurs durées de service suit une loi lognormale. Les clients de type B doivent avoir pris un rendez-vous avant le début de la journée (12 plages de 30min). Ces types de Clients peuvent se présenter avec une probabilité de  $p$  et leur durée de service suit également une loi lognormale. Les clients A sont placés dans une file d'attente si tous les caissiers sont occupés, ceux de type B aussi peuvent être placés dans une file d'attente au cas où le conseiller en question est occupé à l'heure du rendez-vous. Les conseillers servent un client de type A s'il reste  $s$  secondes au moins avant son prochain rendez-vous et qu'il y ait un client A dans la file d'attente.

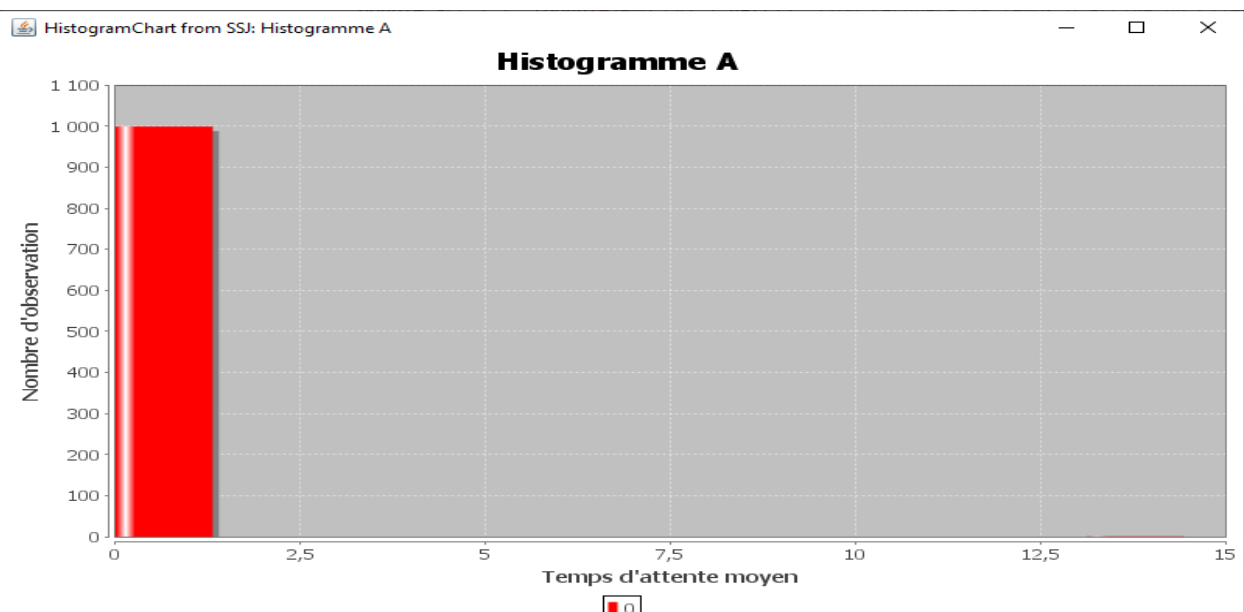
## Partie 2 : Solutions

Il s'agira de modéliser le système en java en utilisant nos propres classes et d'autres provenant de la bibliothèque SSJ pour ainsi renvoyer à la fin  $W_a$  et  $W_b$ , qui sont respectivement les temps d'attentes moyennes par jour des clients A et B à long terme. Soient  $W_{ia}$  et  $W_{ib}$  la somme des temps d'attente des clients de types A et B pour le jour  $i$ , le programme va simuler ce système et calculer ces valeurs, qui sont des réalisations de variables aléatoires, construire deux histogrammes, pour les valeurs des temps d'attente.

## Partie 3 : Gestion Client A

Pour le client A on a eu à implémenter la classe Bank\_A qui simule l'arrivée et le service des clients A dans le système. Elle est composée des classes et méthodes suivantes :

- **Service\_A** : Où on génère le temps de service et on gère le client directement si le nombre d'agents occupés est inférieur au nombre de caissiers dans le cas contraire il est placé sur une file d'attente.
- **NextPeriod** : qui étend la classe Event qui permet de simuler le passage d'une période à l'autre.
- **Arrival A** : c'est une classe qui étend également Event, elle permet de générer les arrivés qui suivent une loi de poisson de taux  $\lambda[j]$ .
- **Service Completion** : Cette classe étend la classe Event et simule la fin d'un Service d'un client A de ce fait le caissier devient à nouveau libre pour servir à d'autres.
- **CheckQueue** : C'est une méthode qui permet de gerer de nouveaux clients A si le les caissiers sont libres et que la liste d'attente des clients A n'est pas vide.
- **CheckQueueB** : c'est une méthode qui permet aux conseillers de servir des clients A lorsqu'ils ont du temps libre avant leur prochain rendez-vous.
- **SimulateOneDay** : cette méthode est utilisée pour simuler le système pour une journée.
- **generPatience** : cette méthode génère un temps de patience de 0 en fonction de la probabilité  $p$  ce qui signifie que le client est directement servi ou bien elle génère une valeur qui suit une certaine loi ExponentialDist.
- **startA** : elle permet de simuler le système sur 1000 jours, affiche le rapport final pour les clients de type A et enfin trace les histogrammes.
- **readData** : elle permet de lire les valeurs à partir d'un fichier externe.



# Rapport final A

Num. service expected = 83.0

Report for Bank A:

	num obs.	min	max	average	variance	std. dev.	conf. int.	
Number of arrivals per day	1000	40.000	351.000	164.477	2786.013	52.783	90.0% (	161.729, 167.225)
Average waiting time per customer	1000	0.000	0.297	0.021	1.0E-3	0.032	90.0% (	0.019, 0.023)
Waiting times within a day	213	0.000	0.000	0.000	0.000	0.000	90.0% (	0.000, 0.000)

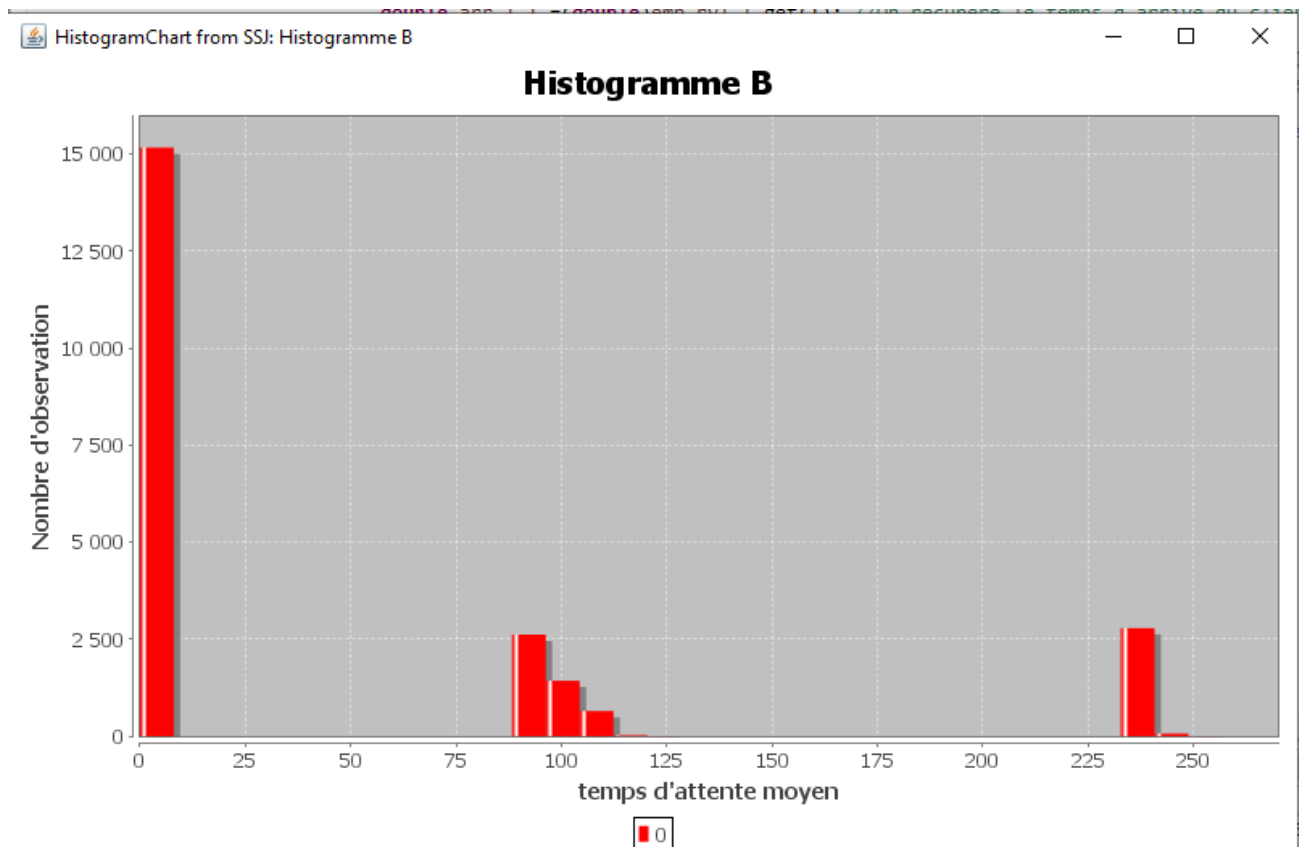
Pour une expérience ayant duré 1000 jours pour les clients de type A, nous avons obtenu les résultats suivant :

- Temps d'attente minimum : 0
- Temps d'attente maximum : à 0.297
- Une moyenne de 0.021

## Partie 4 : Gestion Client B

Pour le client B on a eu à implémenter la classe BankBf qui simule l'arrivée et le service des clients B dans le système. Pour le Client B, les conseillers ne sont pas des objets mais on a utilisé un Hashmap pour pouvoir répertorier pour un conseiller d'une période l'ensemble de des heures d'arrivées et de rendez-vous. Pendant son temps libre, le conseiller peut servir des BankBf est composée des classes suivantes :

- **Service\_B** : où on gère le client de type B dans le système en fonction du conseiller avec lequel il a un rendez-vous, de la période et de la plage.
- **startB()** : exécute les étapes suivantes  
Simule la banque pendant un nombre de jours donné ici 1000 en considérant les périodes,  
Affiche le rapport de la simulation pour les clients de type B de la banque,  
Trace l'histogramme en utilisant le collecteur statistique,  
Temps d'attente moyen en abscisse,  
Nombre d'observation en ordonnée
- **Simulate** : Permet de simuler par période. Elle retourne la méthode Service\_B avec la période en input.





## Rapport final B

```
Bank B
REPORT on Tally stat. collector ==> Temps d'attente moyen par client
  num. obs.      min      max      average      variance      standard dev.
    22750      0.000    256.792     49.571     6402.183       80.014
```

Pour une expérience ayant duré 1000 jours pour les clients de type A, nous avons obtenu les résultats suivant :

- Temps d'attente minimum : 0
- Temps d'attente maximum : à 256.792
- Une moyenne de 49.571

## Partie 5 : Outils et Bibliothèques

### I. Java

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable. Des ordinateurs portables aux centres de données, des consoles de jeux aux superordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts.

### II. Bibliothèque SSJ

SSJ est une bibliothèque Java, développée au Département d'Informatique et de Recherche Opérationnelle (DIRO) de l'Université de Montréal, par Pierre L'Ecuyer et son groupe de recherche. Elle fournit des outils pour la simulation stochastique, en particulier pour générer des valeurs aléatoires uniformes et non uniformes, calculer différentes quantités liées à des lois de probabilité, effectuer des simulations de type Monte Carlo et quasi-Monte Carlo, recueillir et traiter des statistiques, et programmer des simulations à événements discrets.

## Conclusion

Ce projet est très intéressant dans la mesure où il nous permet de simuler des systèmes de la vie réelle telle qu'une banque et de pouvoir collecter les statistiques qui sont assez proches de celle d'une banque de la vie réelle. Donc ainsi avec la modélisation stochastique on peut expérimenter, tester de nouvelles choses, faire des ajustements sur ces systèmes simulés au lieu de les effectuer sur le système réel et de prendre le risque d'en subir les conséquences.