

## Objectif du projet

Réalisation d'une application mobile pour optimiser les visites touristiques du festival d'Avignon

### I. Contexte

Le Big Data représentent ces données massives générées par les activités quotidiennes (emails, transactions bancaires, enregistrements vidéos, signaux GPS, téléphonie...) qui amplifient exponentiellement. Et dans le Big Data, il y a l'Open Data, une information publique brute, qui a vocation à être librement accessible et réutilisable. L'Open Data, c'est surtout une philosophie, une volonté citoyenne, de considérer l'information publique comme un bien commun. En Europe, la donnée ouverte (Open Data) est encadrée par des réglementations légiférées au Parlement européen depuis 2003. Les applications et les secteurs de l'Open Data sont illimités. Parmi les secteurs les plus touchés par l'utilisation de l'open data, nous pouvons citer le domaine de la culture. Nous allons axer le projet sur ce dernier et plus particulièrement sur le Festival d'Avignon<sup>1</sup>.

Le festival d'Avignon, Fondé en 1947 par Jean Vilar, le Festival d'Avignon est aujourd'hui l'une des plus importantes manifestations internationales du spectacle vivant contemporain. Chaque année, en juillet, Avignon devient une ville-théâtre, transformant son patrimoine architectural en divers lieux de représentation, majestueux ou étonnants, accueillant des dizaines de milliers d'amoureux du théâtre de toutes les générations (plus de 130 000 entrées). Souvent en vacances et venus de loin, beaucoup de spectateurs séjournent plusieurs jours à Avignon et voient quelques-uns des spectacles parmi la quarantaine d'œuvres de théâtre, de danse, et aussi d'arts plastiques ou de musique.

L'objectif de ce projet est de permettre à ces spectateurs d'assister aux spectacles de manière intelligente. Plus concrètement, le projet doit fournir, au spectateur, en fonction des données fournies par ce dernier (catégorie du spectacle, moyen de transport, période de visite, éventuellement, le budget, ...) une *Smart-Visit* et devant indiquer :

#### 1. Le nombre de spectacles à voir.

L'application doit fournir le nombre de spectacles en fonction du budget. Les spectacles doivent être choisis en fonction de la catégorie renseignée par l'utilisateur.

---

<sup>1</sup> <https://www.festival-avignon.com/fr/histoire>

## 2. L'ordre dans lequel il est préconisé de voir ces spectacles.

En cette période de festival, le nombre de spectateurs est très important, ce qui engendre des attentes de plus en plus longues et difficilement supportables sous la chaleur du mois de juillet. L'application, via son algorithme de recommandation, doit donc être capable de recommander à l'utilisateur un ordre de visites de sorte à **minimiser le temps d'attente**. Plusieurs éléments peuvent être pris en considération pour la détermination de l'ordre des visites :

- Le taux de fréquentation de chaque spectacle et par créneau.
- Le nombre de places disponibles
- Le nombre de personnes faisant la queue sur place, si on doit par exemple, satisfaire une demande pour une visite instantanée.

## 3. Le trajet à emprunter

L'application doit être capable de recommander le meilleur trajet à l'utilisateur. Les trajets doivent être recommandés selon les moyens de transports possibles :

- *Voiture personnelle* : dans ce cas, l'application doit indiquer et surtout préconiser à l'utilisateur **le parking de stationnement** le plus proche du lieu du premier spectacle. Cette question reste ouverte et son traitement peut suivre différentes logiques. Par exemple, pour une personne âgée, probablement fatiguée en fin de journée, le parking de stationnement ne doit pas être très loin du lieu du dernier spectacle vu.
- *Vélo* : dans ce cas indiquer la station où l'utilisateur peut emprunter/laisser le vélo. Indiquer éventuellement le coût de l'emprunt.
- Transport en commun : bus ou tramway. Indiquer station (départ/arrivée) et heure de passage.

## 4. L'empreinte écologique.

Pour chaque trajet recommandé, l'application doit sensibiliser le spectateur sur les enjeux de l'écologie. Un système de point est à mettre en place. Ainsi,

- o Pour un trajet en véhicule, et en considérant éventuellement le type de moteur (électrique, essence, ...), l'émission de dioxyde de carbone (CO<sub>2</sub>), relatif au trajet, est indiquée et par conséquent des points sont retirés.
- o Pour un trajet en vélo, indiquer la quantité évitée du d'oxyde de carbone. Et par conséquent, des points sont affectés.

## II. Les données à utiliser dans le cadre de ce projet.

Les données sont les données ouvertes mises à disposition par la ville d'Avignon et récupérables sous différents formats (csv, xml, ...). Nous nous intéressons à différents datasets que vous pourriez compléter avec d'autres sources :

- Les parkings : <https://trouver.datasud.fr/dataset/avignon-parkings>  
Emplacements des parkings sur le territoire d'Avignon pendant les périodes : Festival D'Avignon, hors festival.
- Stationnement vélos : <https://trouver.datasud.fr/organization/avignon>
- Stationnement pour personne à mobilité réduite : <https://trouver.datasud.fr/dataset/avignon-pmr>

- Le programme des spectacles : <https://trouver.datasud.fr/dataset/avignon-festival-inoff-1972vs2015/resource/22379fa4-d080-413a-afc1-1ccc62209c87>  
Une version pdf de l'édition 2019 se trouve à l'url : <http://maisonpro.festival-avignon.com/fr/le-programme-2019>

Également, un tableau de bord (Dashboard) est attendu pour visualiser les données manipulées. Concrètement, vous devez définir des compteurs (dits métiers) et visualiser sous format de graphiques. Par exemple, le nombre d'utilisateurs/période/théâtre....

### III. Le déroulement du projet

Le projet peut être scindé en trois phases principales :

**Phase du traitement des données - ETL.** Dans cette phase, le premier groupe est invité à :

- Comprendre les différents jeux de données utilisés.
- Structurer/modéliser les données dans le système de stockage choisi (Cassandra, MongoDB, Neo4j, HDFS, Hive,...)
- Charger les données dans le système choisi après éventuellement une phase de traitement reposant sur Spark.

**Phase de développement de la couche applicative.** Il s'agit dans cette phase de développer le programme de recommandation. Ce dernier doit communiquer avec la base de données et l'application mobile. Dans cette phase, il faut clairement définir les formats d'entrée et de sortie des données. Le format de sortie attendu par l'application doit être réfléchi avec les membres du groupe chargé de réaliser l'application mobile.

**Phase de réalisation de l'application mobile.** Il s'agit de développer ici une application mobile permettant d'interagir avec l'utilisateur. Ce dernier peut simplement consulter la liste des spectacles, comme il peut demander une *Smar-visit* en saisissant les informations nécessaires et indiquées dans la section contexte.

NB : le but est de mettre en place une solution fonctionnelle et non un système de recommandation.

#### IV. Conception de l'architecture distribuée

Une attention particulière est accordée à la conception de l'architecture. Les étudiants doivent réfléchir à la conception de l'architecture distribuée du projet, en mettant l'accent sur les choix technologiques, leurs justifications, et les compromis identifiés au cours du développement.

Les étudiants devront :

- Concevoir une architecture distribuée cohérente répondant aux besoins du projet (volumétrie, temps réel, hétérogénéité des données).
- Comparer plusieurs technologies distribuées, et justifier les choix réalisés pour le stockage, le traitement, la visualisation et l'interface.
- Fournir un schéma d'architecture technique final annoté et commenté.
- Identifier les limites de l'architecture retenue et proposer d'éventuelles perspectives d'amélioration (scalabilité, haute disponibilité, intégration cloud, etc.).

Les choix doivent s'aligner avec :

- Les contraintes fonctionnelles (temps réel, adaptation mobile).
- Le volume et la variété des données (JSON, XML, CSV, requêtes multiples).
- L'objectif d'intégrer des composants distribués manipulés pendant le cours : Spark, Cassandra, Kafka, etc.
- La répartition en groupes : chaque groupe manipule un ou plusieurs composants indépendants mais interconnectés.

#### V. Attentes et livrables

##### **Livrable final attendu**

- Un document d'architecture distribué incluant :
  - Le diagramme global annoté.
  - Une présentation synthétique des technologies comparées.
  - Un argumentaire structuré sur les choix technologiques.
  - Une évaluation critique des limites et axes d'évolution possibles.
  - Code sources
- Présentation des travaux lors de la dernière séance
  - 30 min de présentation (POWER POINT)
  - 10 min Questions/Réponses