



AudioWire

Ce document à pour but de présenter la réalisation de ce projet plus en détail. Il présente les différentes parties du développement, les éventuels besoin matériel ainsi que l'organisation générale du projet.

Maximilien Brunelle, Huai Xie, Stéphane Rose
Guillaume Feraud, Augustin Pasquini, Hugo Defrance,
Vivien Meilhac

Description du document

Titre	AudioWire – Cahier des Charges
Date	04/04/2012
Auteurs	Huai Xie, Stéphane Rose, Guillaume Feraud, Maximilien Brunelle, Augustin Pasquini, Hugo Defrance, Vivien Meilhac
E-Mail	audiowire_2014@labeip.epitech.eu
Sujet	Cahier des Charges
Mots-clés	
Version	1.0

Tableau des révisions :

Date	Auteurs	Section(s)	Commentaire
04/04/12	L'ensemble du groupe	Toutes	Première Version du Cahier des Charges

Table des matières

1. Rappel de l'EIP.....	4
a. Qu'est-ce qu'un EIP et Epitech ?.....	4
1. Epitech, l'école de l'innovation et de la passion informatique.....	4
2. Epitech Innovative Project.....	4
3. Les objectifs.....	5
b. Sujet de l'EIP.....	6
2. Présentation de l'environnement de réalisation	7
a. Environnement de réalisation.....	7
b. Environnement matériel.....	8
c. Architecture technique	9
d. Composants existants.....	9
e. Gestion de la sécurité	10
f. Points sensibles	10
3. Description des différentes parties du programme à réaliser	11
4. Description de la base de données	13
5. Description des tests de premier niveau	18
6. Organisation du projet	20
7. Annexes.....	21

1. Rappel de l'EIP

a. Qu'est-ce qu'un EIP et Epitech ?

1. Epitech, l'école de l'innovation et de la passion informatique

L'Epitech, est une école d'expertise en informatique créée en 1999 dans la mouvance de l'Ecole pour l'Informatique et les Techniques Avancées (Epita) pour accueillir les bacheliers passionnés qui souhaitent apprendre l'informatique par la pratique et non la théorie. Le cursus se déroule sur 5 ans et fourni un diplôme d'expert en technologies de l'information reconnu en tant que diplôme de niveau I par la CNCP.

La particularité et la force de cette école repose principalement sur le fait que les étudiants sont amenés à apprendre par eux-mêmes les notions recherchées dans le cadre de mini-projets de groupes et de projets plus ambitieux comme le PFA (Projet de Fin d'Année) et l'EIP (Epitech Innovative Project). En effet, contrairement à l'enseignement scolaire classique où l'on amène la connaissance à l'élève afin qu'il puisse résoudre des problèmes, un étudiant à l'Epitech devra rechercher de lui-même les notions dont il a besoin avec son groupe de travail pour résoudre un problème ambiguë dans lequel il n'aura pas à l'origine les bases nécessaires. C'est de ce principe que naîtra l'émergence du comportement que recherche l'Epitech, à savoir la capacité d'adaptabilité par rapport à une problématique et des concepts ainsi que la gestion d'une équipe de travail.

2. Epitech Innovative Project

L'Epitech Innovative Project appelé aussi EIP, est un projet de groupe conséquent et innovant dans le domaine de l'informatique. Il doit être réalisé sur une période de 18 mois et est constitué de 5 élèves au minimum.

L'EIP illustre parfaitement la pédagogie enseignée par l'école car il permet de responsabiliser les étudiants sur la recherche et le développement d'une idée novatrice dans le but de l'amener à la fin de sa réalisation dans les délais impartis, aussi bien sur le plan technique que sur le plan administratif.

Ce projet représente une étape majeure dans le cursus d'un élève à l'Epitech car il lui permet de passer du statut d'étudiant à celui de professionnel confirmé. En effet, la réussite de ce projet permettra à la plupart des étudiants de s'installer sur le marché professionnel en créant leur propre entreprise, ou alors en rejoignant une firme déjà existante qui serait intéressée par le concept.

3. Les objectifs

Il faut savoir que l'EIP n'est pas seulement un projet destiné à montrer les connaissances que l'on a acquises lors de notre passage à l'Epitech. En effet, la technologie et plus particulièrement le domaine de l'informatique est un domaine qui évolue constamment. C'est pourquoi la connaissance à l'état pur n'est pas vraiment une plus-value et n'est que passagère car très vite obsolète. Un des buts principaux de l'EIP est d'apprendre aux étudiants à dépasser les aspects techniques d'un projet en informatique afin de développer la partie documentation et communication extérieure.

Cet aspect pédagogique du projet est très important étant donné que ce sont des correcteurs extérieurs qui jugeront le projet et le travail fournis pendant ces deux années lors de la présentation orale de la soutenance.

b. Sujet de l'EIP

Notre projet AudioWire est un lecteur de musique évolué.

Il présente les fonctionnalités suivantes :

- Gestionnaire de bibliothèque
- Gestion des Playlists
- Playlists "Intelligente" (Algorithme d'apprentissage en fonction des goûts et des envies de l'utilisateur) avec des algorithmes liés au data-mining.
- Lecture synchronisée sur le réseau local (deux postes à deux endroits jouent le même morceau en même temps - le morceau n'étant pas forcément sur les deux postes)
- Visualizer
- Possibilité d'écouter des radios et des podcasts

De plus ce logiciel proposera des fonctionnalités plus axées "web" :

- Un chat et une liste d'amis
- Partage de fichiers
- Partage des goûts (Playlists, Playlists intelligentes ...)
- Streaming inter-utilisateurs, c'est-à-dire la possibilité d'écouter en même temps une même musique entre plusieurs utilisateurs.

AudioWire sera compatible Windows, Mac et Linux. Nous avons également prévu de le porter sur iOS et Android.

2. Présentation de l'environnement de réalisation

a. Environnement de réalisation

Afin de réaliser notre EIP, nous comptons utiliser les outils suivants :

- **Weka 3** : il s'agit d'un logiciel que nous comptons utiliser lors de la phase de développement du projet afin de réaliser du data-mining dans des fichiers audio. Cette prospection de données nous permettra de développer notre intelligence artificielle.
- **C++** : ce langage sera majoritairement utilisé au cours du développement de la partie fonctionnelle des clients Windows, Mac et Linux.
- **Python** : python est un langage de script, qui est à la fois facile à prendre en main et permet d'obtenir un code rapidement fonctionnel et puissant. Ce langage est majoritairement utilisé pour les serveurs.
- **Java** : langage portable grâce à l'émulation de son propre environnement. Java sera utilisé pour le développement du client Android.
- **Cocoa/objective-C** : ces deux langages seront utilisés pour l'interface graphique du client Mac ainsi que pour la totalité du client iPhone.
- **C#/.net**: ce langage ainsi que cette API serviront à la réalisation de la partie graphique pour le client Windows.
- **Virtualenv** : ceci est un outil permettant d'isoler un environnement de travail en Python permettant ainsi aux développeurs de travailler sur un environnement commun, indépendamment de la machine hôte. Il sera utilisé sur le serveur gérant l'accès à la base de données.

Pour assurer une bonne communication au sein du groupe, il sera indispensable d'utiliser les outils de communications suivant :

- **Skype** : Pour la communication interne au groupe.
- **Microsoft Live Meeting** : Pour la communication entre le laboratoire EIP et le groupe.

Comme dans tous projets, notre groupe sera confronté à différents types de contraintes :

- **Contrainte géographique** : les développeurs seront localisés sur différents continents, ce qui impliquera donc des problèmes de décalage horaires.
- **Projet évolutif** : étant donné qu'une intelligence artificielle est, de par sa nature, toujours évolutive, il est fort probable que cette dernière soit encore perfectible après la date de rendu du projet.
- **Contrainte de ressource** : le projet étant portable sur de nombreuses plateformes, certains développeurs seront dans l'incapacité de travailler sur certaines d'entre-elle à cause d'un problème de ressources limitées.
- **Contrainte législative** : la diffusion de musique par le réseau ne pourra se faire qu'au sein d'un réseau local dans un souci de protection des droits d'auteurs. De plus, étant donné que notre produit permettra la création d'un compte donnant accès à des listes d'amis notamment, conformément à la loi en vigueur, il sera possible pour l'utilisateur d'exercer un droit de rétraction et de modification sur ces propres données.
-

b. Environnement matériel

Afin de mener à bien le projet, nous pouvons distinguer deux types d'outils : les outils matériels, et les outils logiciels.

- Les outils matériels sont un serveur gérant la base de données et le site web ainsi que les différents ordinateurs et téléphones nécessaires aux développeurs pour travailler.
- Les outils logiciels nécessaires aux développeurs afin de réaliser les différents clients de notre projet sont des IDE (Integrated development environment). Ces derniers seront Eclipse pour le développement du client Android, Microsoft Visual Studio pour le développement du client Windows et xCode pour le développement des clients Mac et iPhone. Pour le développement sous linux, les développeurs seront libres d'utiliser leur IDE préféré.

c. Architecture technique

Le lecteur multimédia sera utilisable sur différentes plateformes (Android, iPhone, Windows, Mac et Linux). Ces clients communiqueront par l'intermédiaire d'un serveur qui assurera la liaison avec la base de données.

Un site web est également prévu pour assurer la communication externe autour du projet et permettre aux utilisateurs d'utiliser les fonctions sociales du produit. Ce site web communiquera de façon directe avec la base de données.

Il existera également un serveur de développement qui sera une copie conforme du serveur de production. Celui-ci permettra de tester toutes les nouvelles fonctionnalités avant la mise en production. Ce qui nous garantira un serveur de production toujours fonctionnel avec une version stable du projet.

d. Composants existants

Afin de concentrer nos efforts sur l'intelligence artificielle, le corps de notre lecteur multimédia sera basé sur le corps d'un projet déjà existant.

De plus, nous utiliserons des algorithmes d'intelligence artificielle de traitement des données audio. Ces dernières serviront au fonctionnement des options d'intelligence du lecteur.

e. Gestion de la sécurité

Afin de préserver la confidentialité des données, tous les transferts d'information jugés sensibles seront chiffrés par le protocole Transport Layer Security (TLS). Ce protocole garanti un chiffrement des données transmises grâce à un système de clé privée/publique.

En ce qui concerne le site internet, les connections non sécurisées (HTTP) seront automatiquement redirigées en connections sécurisées (HTTPS).

f. Points sensibles

Du fait de la nature du projet qui est largement basé sur des algorithmes d'intelligence artificielle, il est possiblement envisageable que certaines options du lecteur multimédia s'avèrent finalement irréalisable. Il est donc important de bien évaluer les possibilités offertes par les algorithmes existants afin de pallier aux éventuels problématiques rencontrées au plus vite.

La manière dont les données sensibles sera transférées et stockées devra faire l'objet d'une attention toute particulière afin de pouvoir garantir la confidentialité et la sécurité de celles-ci.

3. Description des différentes parties du programme à réaliser

Le projet AudioWire se compose de différentes parties.

- L'intelligence artificielle :

L'intelligence artificielle est la partie principale de notre EIP. C'est pourquoi nous avons mis en place un partenariat avec le lab Acsel en plus du lab EIP, auquel nous avons à rendre des comptes mensuellement. L'IA sera capable de classifier les données en fonction de différents facteurs qui vont être implémentés :

- o L'humeur de l'utilisateur : qu'il sélectionne manuellement ou alors que le programme arrive à déterminer.
- o L'analyse des fichiers audios, que ce soit des flux audios, ou des métadonnées.
- o Le recoupement des informations est aussi prit en compte : quel temps fait-il, quelle période de l'année sommes-nous, que s'est-il passé récemment dans l'actualité, suis-je tout seul ou en soirée ?

Nous allons utiliser le logiciel WEKA afin d'analyser les flux audio, y extraire les indices musicaux.

La récupération des flux audios comprend est une conséquente partie de l'IA, et comprend différentes données à traiter :

- o Les fonctions de hachage qui permettent de comparer les différents flux avec leur hach.
- o Le BPM.
- o Les métadonnées.
- o Le tempo (rythme).
- o Les breaks.
- o La gestion des styles simple.

- Le client :

L'EIP AudioWire est un projet très technique. Cependant, il est primordial de développer une interface client agréable à utiliser, "user friendly", moderne.

Les deux membres du groupe qui s'occuperont de cette partie devront mettre en place les fonctionnalités du projet développées du côté de l'intelligence artificielle.

Le développement du client devra respecter de un plan d'action définit dans les UML et diagrammes de transitions à venir. Le client donnera accès à un profil détaillé de l'utilisateur, et permettra d'utiliser toutes les fonctionnalités de l'IA de manière ergonomique.

- Les applications mobiles :

Une importante partie d'AudioWire va se développer sur mobile. En effet, il est prévu de développer une application iPhone et une application Android de lecture de la musique, de manière intelligente. Elles seront disponibles gratuitement au téléchargement sur leur market respectif. Les applications intégreront toutes les fonctionnalités principales du projet, notamment la lecture de musiques en fonction de nombreux paramètres, définit par l'utilisateur et l'intelligence artificielle.

- La base de données :

Le projet va nous amener à stocker un très grand nombre de données, que ce soit au niveau des musiques que des utilisateurs. Notamment les métadonnées des musiques (genre, album, artiste, etc), ou les infos récupérées par l'intelligence artificielle (décrites dans la partie dédiée). Le lab EIP serait d'accord pour que l'on emprunte un de leur serveurs afin de stocker toute ces informations.

Cette base sera d'abord enrichie par un corpus de musique fourni par les développeurs, puis peu à peu par les utilisateurs.

4. Description de la base de données

La base de données est un point clé à ne pas négliger dans la réussite de notre EIP car c'est elle qui va permettre la gestion de toutes les informations dont on aura besoin aussi bien en termes d'utilisateur que d'intelligence artificielle. En effet, les données renseignées par les clients seront directement enregistrées afin de pouvoir être utilisées par les différents algorithmes de recherches de gout. Il va sans dire que plus la base sera importante, plus la recherche sera riche et les résultats intéressants. Etant donné que le but premier de notre EIP n'est pas la gestion de base de données mais bien la réalisation d'algorithmes perfectionnés dans la recherche de goûts musicaux, il nous faut choisir un éditeur qui soit relativement simple à utiliser et qui nous permette de passer le moins de temps possible à sa conception. De plus il est nécessaire que la recherche et le traitement de ces données soit un tant soit peu optimisé dans le sens où plus le temps d'exécution des requêtes seront courtes et plus les résultats de nos algorithmes seront percutants.

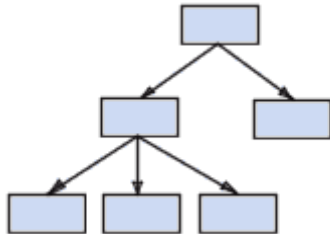
Dans un premier temps nous avons exposé les différents types de bases de données existantes ainsi que leurs caractéristiques puis nous allons déterminer qu'elle choix nous paraît le plus juste dans le cadre d'AudioWire.

Les modèles de bases de données

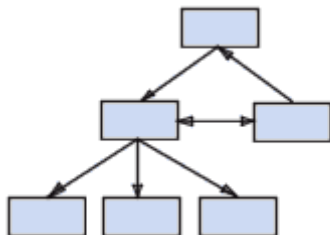
En informatique, une base de données est un lot d'informations stockées dans un dispositif de stockage. Aujourd'hui, elles accompagnent quasiment tout projet informatique commercial car c'est elle qui va permettre de garder en mémoire toutes les informations nécessaires au fonctionnement de l'application. Il est possible de modifier et d'interpréter ces données à l'aide de différents langages plus ou moins complexes en fonction des bases. Ces derniers seront expliqués un peu plus bas. Il existe plusieurs modèles de base de données.

Le modèle hiérarchique :

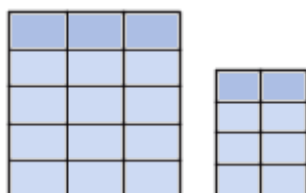
Les données sont classées hiérarchiquement, selon une arborescence descendante. Ce modèle utilise des pointeurs entre les différents enregistrements. Il s'agit du premier modèle de SGBD

**Le modèle réseau :**

Comme le modèle hiérarchique ce modèle utilise des pointeurs vers des enregistrements. Toutefois la structure n'est plus forcément arborescente dans le sens descendant

**Le modèle relationnel (SGBDR, Système de gestion de bases de données relationnelles) :**

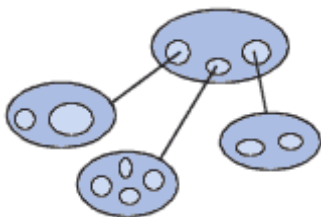
Les données sont enregistrées dans des tableaux à deux dimensions (lignes et colonnes). La manipulation de ces données se fait selon la théorie mathématique des relations



Le modèle déductif :

Les données sont représentées sous forme de table, mais leur manipulation se fait par calcul de prédicats

Le modèle objet (SGBDO, *Système de gestion de bases de données objet*) : les données sont stockées sous forme d'objets. Les champs sont des instances de ces classes.



A la fin des années 90, les bases relationnelles sont les bases les plus répandues dans le monde (environ $\frac{3}{4}$ des bases de données). Nous allons donc nous intéresser à ce modèle précis de base de données. D'un point de technique, il est plus apte à répondre à nos attentes pour notre EIP, c'est-à-dire qu'il y a toujours une relation entre les données en base, par exemple, il y aura toujours un lien entre un artiste et un album ou une chanson.

Modèle relationnel

Selon le modèle relationnel, il peut y avoir plusieurs relations connexes. Dans une base de données relationnelle chaque enregistrement d'une table contient un groupe d'informations relatives à un sujet et les différents sujets sont connexes. Les liens existants entre les informations sont stockés dans les champs des enregistrements (composants des n -uplets) sous forme de clé primaire et clé étrangère. Le n -uplet est une collection de n objets tel qu'il soit

possible de dire exactement celui qui est le premier élément, le second élément, ..., le n ème. Les éléments sont aussi appelés composantes.

Une composante N d'un n -uplet est une clé primaire lorsque par nature la valeur de cette composante est différente pour chaque n -uplet d'une relation, et que pour une valeur donnée, il n'existe qu'un seul n -uplet dans toute la relation dont la composante a cette valeur (absence de redondance).

Une composante N d'un n -uplet est une clé étrangère lorsque les valeurs de cette composante sont des références à une clé primaire. Il y a une situation d'intégrité référentielle lorsqu'à chaque valeur de la clé étrangère A correspond une valeur de la clé primaire référencée B .

Ce modèle utilise l'algèbre relationnelle, théorie de mathématiques et d'informatique, qui dit que les informations sont classées dans des relations : des [matrices](#) contenant un ensemble de [n-uplets](#) (un n -uplet est un groupe de N valeurs). Des opérations permettent à partir d'une relation d'obtenir une autre relation ainsi que de combiner deux relations et obtenir une troisième relation.

On distingue les opérations possibles en deux catégories, selon qu'elles soient unaires ou binaires.

- Les **opérations unaires** qui mettent en œuvre un seul opérande: la projection, le renommage et la sélection.
- Les **opérations binaires**: la différence, la division, l'intersection, les jointures, le produit cartésien et l'union.

C'est là qu'intervient le langage SQL (Structured Query Language), langage d'interrogation et de manipulation de tous les SGBDR actuels (Oracle, PostgreSQL, MySQL, MS SQLServer, MS Access et tous les autres). Actuellement, Oracle détient plus de la moitié du marché des SGBD avec MySQL et Oracle Database.

Le Langage SQL

Le SQL (Structured Query Language) est un langage de définition de données, un langage de manipulation de données, et un langage de contrôle de données, pour les bases de données relationnelles.

SQL est un langage de définition de données (LDD), c'est-à-dire qu'il permet de créer des tables dans une base de données relationnelle, ainsi que d'en modifier ou en supprimer.

SQL est un langage de manipulation de données (LMD), cela signifie qu'il permet de sélectionner, insérer, modifier ou supprimer des données dans une table d'une base de données relationnelle.

Il est possible avec SQL de définir des permissions au niveau des utilisateurs d'une base de données. On parle de DCL (Data Control Language).

Il est aussi possible d'inclure des requêtes SQL dans un programme écrit dans un autre langage, ainsi que d'envoyer directement les requêtes SQL telles quelles au SGBD.

MySQL ou Oracle ?

Suite à ces recherches, notre choix se porterait donc vers un modèle SGBD MySQL ou Oracle car ce sont les deux modèles le plus courants sur le marché et surtout multiplateformes. Cependant, MySQL serait plus intéressant étant donné la nature de notre projet. En effet, Oracle est un modèle très puissant et très complexe pour le traitement de grosses bases de données mais il demande un coût trop élevé et serait un investissement trop important surtout dans le cadre d'un EIP, alors que MySQL reste un SGBD open source et plus simple d'utilisation.

5. Description des tests de premier niveau

AudioWire se décompose en deux grandes parties : la première est le lecteur lui-même, la seconde est l'algorithme d'intelligence artificielle qui permettra une lecture appropriée aux goûts de l'utilisateur.

Les premiers tests se feront sur la partie lecteur car celle-ci devra être rapidement fonctionnelle afin de nous laisser le temps suffisant de coder l'algorithme d'Intelligence Artificielle performant.

Nous devons donc tester les fonctionnalités suivantes :

- Gestionnaire de bibliothèque
- Gestion des Playlists
- Visualizer

Pour ces fonctionnalités, de simples tests suffiront. Il nous faudra vérifier leur stabilité mais également surveiller leur consommation en ressource (memory leak, consommation CPU, ...).

De plus, AudioWire comportera une partie réseau :

- Lecture synchronisée
- Possibilité d'écouter des radios et des podcasts
- Un chat et une liste d'amis
- Partage de fichiers
- Streaming inter-utilisateurs (Ecouter un morceau en même temps par exemple)

Il nous faudra, pour ces fonctions réseau, ajouter aux tests décrits ci-dessus une autre batterie de tests afin de vérifier que tous les cas d'utilisations seront gérés.

La mise en place de scripts sera sûrement nécessaire et permettra d'évaluer et

donc d'optimiser la consommation en bande passante. En effet, permettre une lecture synchronisée demandera de faire de nombreux tests et nous mènera probablement à adapter le protocole de communication réseau.

Enfin, la seconde partie des tests sera liée à l'algorithme d'IA. Tout au long de l'EIP, nous devrons vérifier la viabilité de nos choix techniques auprès des membres du laboratoire ACSEL.

Comme un algorithme d'IA peut nécessiter une phase dite « d'entraînement » qui permet de le rendre efficace, nos tests seront réalisés en même temps par le biais de petits programmes/scripts.

6. Organisation du projet

Tout d'abord nous allons établir deux états de l'art qui nous permettront de réaliser la partie intelligence artificielle de notre logiciel. Ceux-ci présenteront le Data Mining (à quoi il sert, ses implémentations, etc.) et l'analyse de fichier Audio (Que peut on récupérer ou déduire à partir d'un signal audio, etc.) Cette partie, en partenariat avec le laboratoire ACSEL, durera jusqu'au départs a l'étranger des différents membres du groupe. Nous commencerons a ce moment la partie développement du projet.

En ce qui concerne la réalisation du projet, nous comptons, dans un premier temps, réaliser une application pour une des plateformes non mobile, Windows.

Cette stratégie va nous permettre de construire une application stable que nous allons pouvoir par la suite distribuer à nos proches. En effet, la majeure partie de notre projet se base sur l'intelligence artificielle, et plus précisément sur du « Data Mining ». C'est ceci qui va nous permettre d'établir quels sont les goûts des utilisateurs, et qui va nous permettre de leur proposer d'autres morceaux.

Il est donc primordial que nous récupérions les informations des utilisateurs afin de tester notre algorithme et de proposer des résultats meilleurs et mieux adaptés aux utilisateurs.

La plateforme Windows s'impose presque d'elle même pour cette première version de notre logiciel puisqu'elle reste globalement la plateforme la plus utilisée. En choisissant celle-ci, nous nous assurons que nous allons pouvoir faire circuler la version « Beta » de notre logiciel à de nombreuses personnes, et donc de récupérer beaucoup d'informations.

Cette période de « beta » nous permettra non seulement de récupérer des informations sur les goûts des utilisateurs, mais également de trouver les quelques bugs de notre logiciel afin de pouvoir les régler pour la version finale.

Une fois notre version Windows stable, nous entamerons la deuxième étape qui consistera à réaliser d'un coté les versions osX et Linux et de l'autre les version Andoird et iOS pour les plateformes mobiles. Nous repartirons le groupe sur ces quatre taches afin d'optimiser le temps de travail.

Tout comme pour la plateforme Windows, ces plateformes mobiles et non-mobiles nécessitent également des phases de tests intensifs, que ce soit par les membres du groupe, ou par des gens extérieurs au groupe.

7. Annexes

Sources concernant les bases de données :

<http://webtic.free.fr/sql/>

http://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es http://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_relationnelle

<http://www.commentcamarche.net/contents/bdd/bddtypes.php3>

<http://www.commentcamarche.net/contents/sql/sqlintro.php3>

Weka :

http://en.wikipedia.org/wiki/Weka_%28machine_learning%29

Voir Diagramme de Gantt