

# Refonte Site “Roland Garros”

*écrit par Loïc CONAN, Charles DESRAYAUD, Kellian GERMAIN,  
Florent GUERIN, Camille NICOLAS et Matthieu ROCHARD.*

*20 Mai 2022*





L'objectif du site est de recenser l'entièreté des participants et des matchs afin de permettre au public de se renseigner sur le déroulé du tournoi. Les organisateurs ont aussi, en fonction de leurs droits, la possibilité de renseigner toutes les informations comme la planification et les résultats des matchs ou la liste des participants.

Le but de ce rapport est de présenter les spécificités techniques du nouveau site.

## **Sommaire**

1. La base de données
2. MVC
  - 2.1. Principe du Pattern
3. DAO
4. Choix d'implémentation
  - 4.1. Page d'accueil
  - 4.2. Header/Footer
  - 4.3. Affichage des caractéristiques du joueur
  - 4.4. CRUD Joueur/Match



## 1. La base de données

La base de données conçu par l'équipe de développement respecte l'architecture suivante :

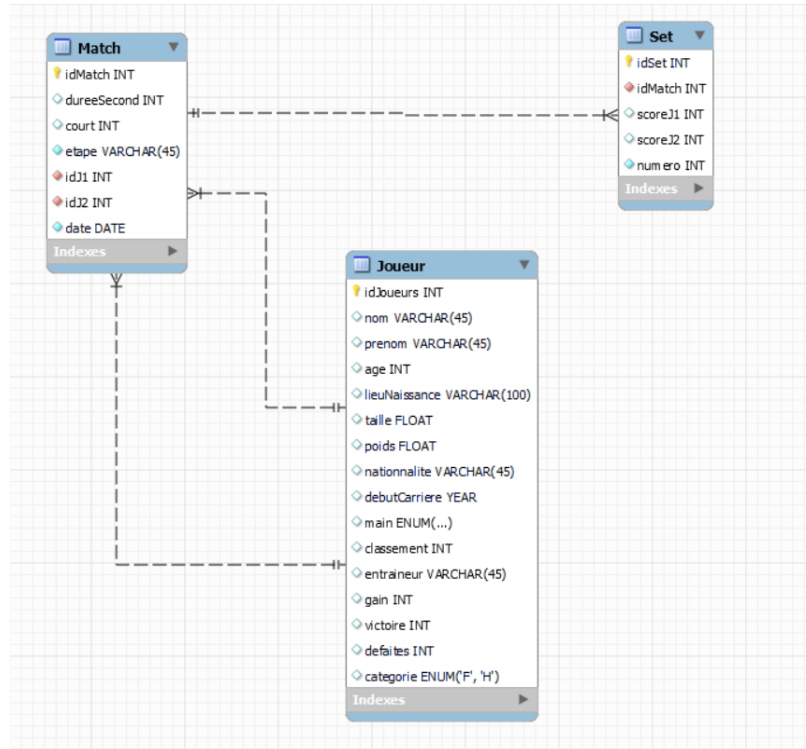


Figure 1: Diagramme d'objet de la base de données

Cette base de données comporte deux mécanismes de clés étrangères. Tout d'abord, il n'est pas possible de créer un match avec un joueur qui n'est pas dans la liste des joueurs. De même pour les sets, chaque set appartient à un match et il n'est pas possible de créer un set pour un match non existant dans la base de données.

Il y a également un mécanisme d'agrégation faible entre les joueurs et les matchs, ce qui veut dire que la suppression d'un joueur ne peut être opérée quand ce joueur participe déjà à un match.

A l'inverse, à la suppression d'un match, tous les sets associés à ce dernier seront supprimés au même moment, c'est une agrégation forte.

## 2. MVC

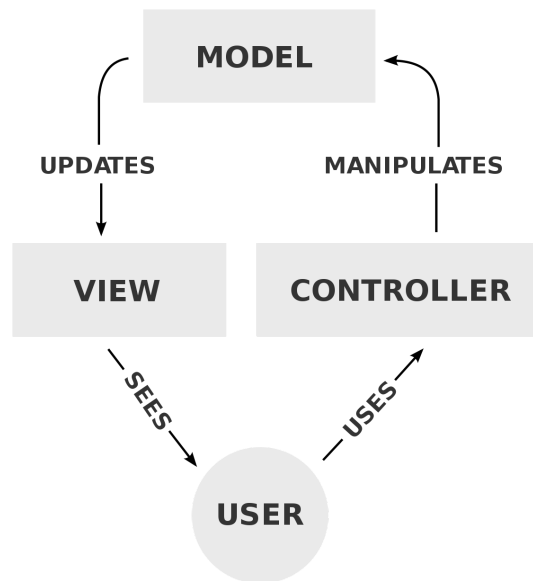
Le design pattern choisi par l'équipe est Model-View-Controller dit MVC dans la mesure où c'est le modèle le plus adapté à l'application.

Ce design pattern permet de séparer l'information vue par l'utilisateur de l'information qui est stockée. Cela nous permet de manipuler les données afin d'améliorer son apparence et proposer des affichages qu'un simple affichage de la base de données ne permettrait pas.

Ce pattern nous a aussi permis de traiter les informations passées par l'utilisateur avant son insertion en base de données.



### 2.1. Principe du Pattern



**Model:** le Model représente la donnée traitée donc dans notre cas les Joueurs, Matches et Sets.

**View:** c'est l'interface par laquelle la donnée sera affichée à l'utilisateur.

**Contrôleur:** gère les modèles et les vues en traitant l'input de l'utilisateur, gère les modèles avant de fournir une vue à l'utilisateur pour ce modèle.

## 3. DAO

L'utilisation d'un DAO permet de flexibiliser les opérations en base de données. Ici nous utilisons mySQL pour tout ce qui touche à nos données mais un DAO nous permettrait également de traiter des données sous d'autres formats type xml ou json sans grande difficulté. Nous utilisons un DAO également pour sa simplicité d'implémentation.

## 4. Choix d'implémentation

### 4.1. Page d'accueil

La page d'accueil affiche un message de bienvenue et deux images, la première permet d'accéder à la liste des matchs, la deuxième permet d'accéder à la liste des joueurs.

### 4.2. Header/Footer

Comme chacune de nos pages doit posséder un menu pour fournir l'accès aux autres pages ainsi qu'un bas de page avec un lien vers les CGU, nous avons décidé d'utiliser un header et un footer communs à toutes les pages. Pour cela, nous avons créé un fichier nommé "header.jsp" qui est ajouté au début de chaque fichier et un fichier nommé



“footer.jsp” et qui est ajouté à la fin de chaque fichier. Cela permet d’avoir une régularité dans l’aspect des pages.

Plus précisément, le header est constitué d’un titre et d’un menu déroulant permettant d’accéder aux différentes pages du site. Pour le public, l’accès à la liste des joueurs et des matchs. Ils permettent également d’accéder aux pages de gestion pour les personnes autorisées. De plus, on peut cliquer sur le menu en haut pour retourner à la page d’accueil.

De plus, le footer est un bandeau restant sur le bas de la page, il indique le copyright et donne accès à la page des conditions générales d’utilisation (CGU).

### 4.3. Affichage des caractéristiques du joueur

De nombreuses informations à propos des joueurs sont inscrites dans la base de données, telles que leur nom et prénom, leur âge, leur nombre de victoires et défaites ou leur main dominante. Lors de l’affichage des informations d’un joueur, si une information n’est pas fournie par la base de données, le mot “inconnu” est affiché à la place si une chaîne de caractère était attendue, ou bien le chiffre 0 si un nombre était attendu. Cela permet d’afficher les informations disponibles sans créer de problème à cause de l’absence de certaines informations.

### 4.4. CRUD Joueur/Match

Des formulaires d’ajout de matchs et de joueurs se caractérisent par l’envoi de méthodes *get* et *post* qui permettent d’envoyer toutes les informations nécessaires pour l’enregistrement en base de données. Il en va de même pour les actions de suppression et de modification, mis à part qu’il était nécessaire de construire un affichage clair pour un maximum de lisibilité et d’ergonomie.