

## **Sistema de Recomendação de Disciplinas - “Sistema\_Matriculas\_Racket”**

### **Introdução**

O contexto do desenvolvimento do projeto é o de um sistema de matrículas com muita liberdade e variáveis, como o da oferta de disciplinas, o das disciplinas já cursadas pelo estudante e o curso específico o qual pretende cursar, além da possibilidade de desvios do quadrimestre ideal.

Assim, o problema que inspirou o grupo foi o da complexidade para o aluno realizar sua matrícula a cada quadrimestre e a possibilidade de cometer erros e os desgastes que sofre no processo.

Atualmente não há ferramentas que auxiliem nesse processo, acreditamos que isso ocorre para que o aluno tenha total responsabilidade e consciência da grade que estará montando. Por este motivo, o projeto visa apenas auxiliar e propor uma grade adequada, mantendo a decisão por segui-la ou não, nas mãos do aluno.

Portanto, o grupo pretende desenvolver um sistema que receba as disciplinas que o aluno já cursou, e processe as disciplinas disponíveis para o próximo quadrimestre. Com essas informações, também deve considerar a conexão entre as disciplinas e, também, o curso específico do aluno (computação, no caso), e lhe forneça uma sugestão otimizada de grade para o próximo quadrimestre.

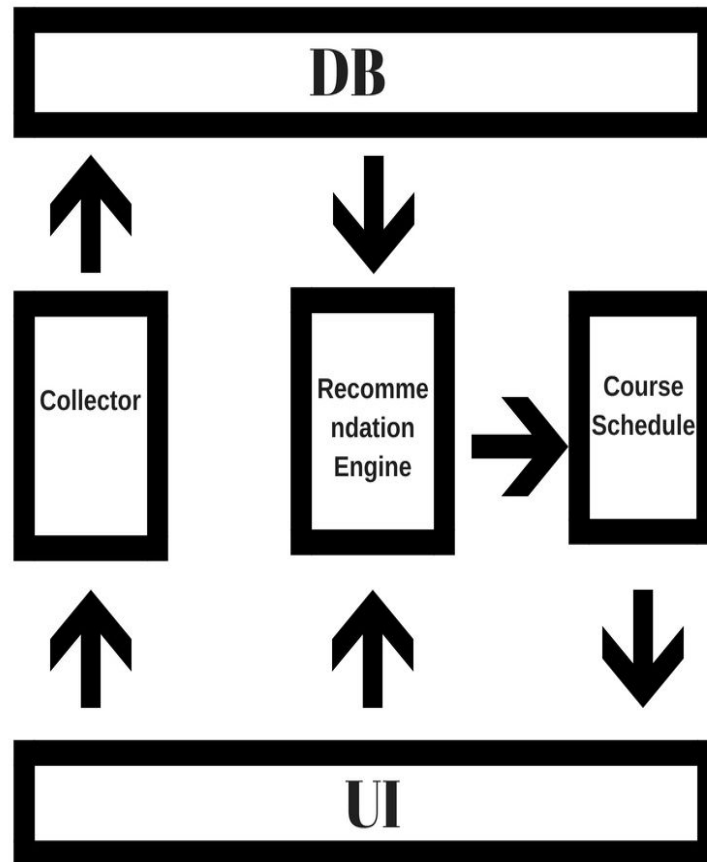
Essa abordagem é adequada pois fornece ao aluno uma grande personalização para a sua necessidade específica e, também, pois o ajuda a diminuir a chance de escolher uma grade não-ótima para o próximo quadrimestre. Além disso, é uma abordagem original pois ainda não existe um sistema assim para alunos da UFABC e seria difícil utilizar outro sistema de outra universidade pois a UFABC se diferencia no Brasil pelo seu projeto pedagógico diferenciado.

### **Metodologia**

#### **\* Quais são os principais componentes previstos para a arquitetura da aplicação?**

O projeto consistirá em 5 principais componentes e módulos: database, collector, recommendation engine, course schedule e UI. A arquitetura terá um coletor que pegará as matérias já cursadas anteriormente pelo aluno (usuário) e gravará no banco de dados. A engine de recomendação coletará os inputs do usuário e fará a query do banco de dados

para gerar uma lista de matérias. No final, essa lista irá para o módulo course schedule, no qual haverá o output dos resultados através da UI.



**\* Que tipos de dados serão processados pela aplicação?**

A aplicação irá tratar apenas dados no formato de texto, como entrada, as matérias já cursadas pelo aluno e como base para processamento da sugestão, as matérias ofertadas no quadrimestre em questão.

**\* Quais são as aplicações similares ou relacionadas com a que você está propondo?**  
**Citar adequadamente, especificar vantagens e desvantagens.**

Para a UFABC, nenhuma aplicação conhecida possui as mesmas características que a aplicação proposta, o que encontramos atualmente são plugins para navegadores de internet que auxiliam de alguma forma o aluno no momento da matrícula, trazendo informações de outras fontes para o portal de matrículas e incluindo novos filtros.

O problema que pretendemos resolver consiste em um Course Scheduling Program (CSP) [1] que envolve o problema de satisfação de restrições (Constraint Satisfaction Problem) [1, 2], presente nos campos de Inteligência Artificial e Machine Learning, entre outros. Problemas de “university scheduling” (US) atraem grande atenção por parte de pesquisadores das áreas de IA e pesquisa operacional [3]. Normalmente nesses problemas, um conjunto de eventos (que podem ser representados pelos exames ou cursos), é atribuído a um certo número de horários disponíveis para se montar a grade. Em nosso caso, o sistema se baseará nas disciplinas já cursadas pelo aluno e da grade existente de disciplinas no momento (para o curso de Ciência da Computação). Considerando as peculiaridades da oferta de disciplinas e das grades sugeridas da UFABC, bem como as diferenças de aluno para aluno ao longo de suas trajetórias acadêmicas, nosso sistema se diferencia dos demais que se propõem a resolver problemas semelhantes, apresentando vantagens relacionadas à personalização da grade, levando-se em conta as opções do aluno.

#### **– Como será a avaliação da aplicação? Quais conjuntos de dados serão utilizados?**

##### **Será necessário testar com usuários?**

Os testes da aplicação serão realizados pelos próprios integrantes do grupo, como cada um possui boa experiência com o sistema atual de matrículas, será possível identificar possíveis falhas e pontos de melhoria.

A avaliação da aplicação será realizada comparando-se a grade sugerida com as possíveis grades de acordo com as matérias já cursadas, conflitos de horários e ordem de prioridades das disciplinas.

O conjunto de dados a ser utilizado na aplicação provêm dos arquivos de ofertas de disciplinas disponibilizados pela universidade, uma vez que esses arquivos são disponibilizados em documentos PDF. Antes, será necessário extrairmos essa informação para um ambiente adequado como banco de dados.

**– Link para o repositório do projeto**

[https://github.com/beatrizmayumi/sistema\\_matriculas](https://github.com/beatrizmayumi/sistema_matriculas)

**Funcionalidades executadas**

Funcionamento por meio do cruzamento com dados do pdf de oferta de disciplinas com base no código das disciplinas.

Início do desenvolvimento do código em racket, apresentando componentes da arquitetura que serão desenvolvidos posteriormente.

**Referências**

[1] Course Scheduling. Disponível em:

<<https://web.stanford.edu/class/cs221/assignments/scheduling/>> Acesso em 30 jun 2018.

[2] Constraint Satisfaction Problems (CSPs). Disponível em:

<<https://web.stanford.edu/class/cs221/2017/lectures/section6.pdf>> Acesso em 30 jun 2018.

[3] Yazdani, M., Naderi, B., & Zeinali, E. (2017). Algorithms for university course scheduling problems. *Tehnicki Vjesnik-Technical Gazette*, 24, 241-247.