

# ADA - Atribuição de Aulas

**Ana Paula Moura Messias de Souza, Gustavo Santos Costa Soares, Isabella Valerio Mazará,  
Josineudo das Chagas Arruda, Paulo Kenji Yokota Muneischi**

<sup>1</sup>Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP)  
São Paulo – SP – Brasil  
Curso Técnico em Informática Integrado ao Ensino Médio  
PDS - Prática para Desenvolvimento de Sistemas

**Abstract.** *Este projeto, Atribuição de Aulas, como o próprio nome sugere, coordena e executa o processo de atribuição de aulas aos professores do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) - Campus São Paulo. Como objetivo visa automatizar o processo atual, em Excel, para tornar o preenchimento e a leitura dos dados do Formulário de Preferência de Atividade (FPA) funcional e descomplicado. Destinado a atender a necessidade final dos docentes, o projeto também será gerido por outros funcionários, cujos cargos ficarão responsáveis pela atribuição naquele ano. Para executar todos os processos, será utilizado o framework Django, baseado na linguagem python, em conjunto com o PostgreSQL.*

**Resumo.** *Este meta-artigo descreve o estilo a ser usado na confecção de artigos e resumos de artigos para publicação nos anais das conferências organizadas pela SBC. É solicitada a escrita de resumo e abstract apenas para os artigos escritos em português. Artigos em inglês deverão apresentar apenas abstract. Nos dois casos, o autor deve tomar cuidado para que o resumo (e o abstract) não ultrapassem 10 linhas cada, sendo que ambos devem estar na primeira página do artigo.*

## 1. Introdução

No IFSP Câmpus São Paulo, são mais de 350 docentes, com 56 na área de DIT. Logo, é essa a quantidade que participa da atribuição de aulas semestral e/ou anual, indicando seu extenso e intrincado processo. Além da referida quantidade, o que dá essas características à atribuição são os critérios a serem seguidos, desde a ordem de prioridade dos docentes até as preferências colocadas por cada um no FPA e os regramentos presentes na resolução vigente – em 2023, ainda rege a Resolução nº109/2015, de 4 de novembro de 2015.

Todo esse processo é realizado manualmente, através da entrega do FPA, da ferramenta Excel e da comunicação particular constante entre o administrador daquele ano e o docente, principalmente no caso de permuta. Assim, procedem adversidades, conflitos interpessoais e atrasos, relatados semestralmente pelos docentes, principalmente pelos que ficam no final da fila de prioridade (os substitutos), e comentados, após contato da equipe, pelo administrador das atribuições da DIT atual e antigo, os quais enfatizaram sobre a consequente sobrecarga em suas funções, ao tentar equilibrar a vontade de todos e, ao mesmo tempo, cumprir com a lei.

Em decorrência disso, surge a necessidade da automatização de parte dos processos, que tem como objetivo a aprimoração do andamento do fluxo de trabalho, trazendo como resultado o aumento da produtividade e a redução de custos e de erros.

E, à vista do que foi exposto, o projeto retratado propõe a elaboração de um sistema que automatize os principais processos da atribuição - as seleções do FPA e a permutação - e, simultaneamente, cumpra o exigido na Resolução, e nos outros critérios estabelecidos hoje (como a prioridade da escolha do docente na atribuição) e que podem ser posteriormente adicionados. Esse sistema é o ADA.

### **1.1. Objetivo**

O ADA visa apresentar uma solução e uma aprimoração às problemáticas da atribuição de aulas. Logo, oferecer um sistema Web responsivo de Single Application Page (SPA) aos funcionários, que automatize essa atribuição e a respectiva e consecutiva permutação (caso habilitada), sem a necessidade de organização manual e de negociações individuais e extraoficiais.

Com processos correspondentes aos problemas centrais, terá a implementação do login, pelo e-mail oficial do Instituto no Google, tratando do gerenciamento geral dos docentes e dos administradores; da automatização do FPA, tratando da dificuldade de estruturação da grade horária seguindo todos os critérios e os regramentos; e da automatização das negociações à permuta, tratando dos atritos e da dificuldade de comunicação gerados.

Consequentemente, o sistema proporcionará, a princípio, um ambiente em que o administrador superior e os subadministradores consigam controlar e ordenar os critérios às suas área e subáreas, respectivamente, e habilitar funções como a permutação e a desativação de um em determinada matéria. Ademais, proporcionará um ambiente em que o docente consiga selecionar todas as suas preferências e solicitar suas permuta (caso habilitadas) em um único local, sem demasiadas complicações e processos.

Uma operação antes com responsabilidades individuais e organização manual, a qual incorre de mais erros devido a subjetividade e os problemas humanos, passará a ser uma operação tecnológica mais limpa e funcional, com menos erros.

## **2. Entrega Parcial**

A fim de implementar um sistema que trate dos problemas citados e consiga atingir os objetivos propostos, são necessários processos e a utilização de determinadas tecnologias, citadas no subtópico .

Os processos principais são três, o cadastramento dos usuários, a automatização do FPA e a possibilidade de habilitação de outros processos, por parte dos administradores, com destaque à permutação dos horários já atribuídos e à desativação de um docente em determinada matéria.

### **2.1. Cadastramento dos usuários**

Preliminar à qualquer utilização do ADA, o Administrador Superior (Admin), será cadastrado pelos próprios programadores e terá o maior nível de acesso, podendo realizar quaisquer alterações e controlar quais serão os Administradores (Staff).

Então, os outros funcionários receberão um link para acessarem o ADA via Google, pelo e-mail institucional - o que evita acessos não permitidos, e serão atribuídos instantaneamente ao papel de Professor (professor); como mencionado, a mudança desse

nível de acesso para o de Staff é realizada pelo Admin. E acessos posteriores poderão ser através do Google ou do prontuário e senha.

## **2.2. Configuração do ambiente**

A configuração do ambiente é um subprocesso, em que o Admin será responsável por habilitar a possibilidade de permuta e de desativação do docente em uma disciplina; prazos limites à organização; e definição ou atualização dos critérios da atribuição - baseados na legislação vigente e na ordem de prioridade de escolha das disciplinas.

E o Staff será responsável pela subárea, conseqüentemente, por subir a grade horária; determinar prazos específicos; autorizar a permutação e se deseja participar da aprovação das permuta; controlar os docentes desativados; e adicionar<sup>1</sup> os que participam de sua subárea.

## **2.3. Automatização do FPA**

Finalizada a organização do sistema pelos administradores e todos os docentes cadastrados nas subáreas, eles poderão acessar o sistema e iniciar o processo de escolha da disponibilidade de horários e da preferência de aulas (prioritária e secundária) e de atividades. Conforme é realizado esse processo, o ADA verifica se cada escolha segue os regramentos, e impossibilita a escolha de disciplinas em conflito; igualmente, informa com uma mensagem breve caso o docente selecione uma em que foi desativado.

A determinação da preferência de atividades poderá ser modificada dentro do prazo de entrega estabelecido pelo Staff. Porém, ao encerrar o prazo, o ADA percorre a lista de docentes, em ordem decrescente, e atribui as aulas de acordo com o selecionado. O processo é interrompido - e é armazenado o que já foi feito - caso haja conflito com uma disciplina já escolhida; assim, aquele docente receberá uma solicitação para alterar sua escolha dentro de determinado prazo.

## **2.4. Permutação**

A permutação é aberta, caso habilitada com a conclusão da grade pelo sistema. De modo geral, é feita com a solicitação de um docente pela troca de sua aula por uma específica do outro, selecionada na grade. É impossibilitada mais de uma solicitação, ao mesmo tempo, para uma mesma aula; Apenas é liberada quando essa for aceita ou recusada. Igualmente é impossibilitada a solicitação de alguma que descumpra o regramento. Caso o Staff seja moderador, ele terá que aprovar a aceitação da permuta pelo segundo docente.

Por fim, é gerada a grade horária final, onde os docentes e os administradores conseguem visualizar e salvar a atribuição de aulas da subárea. Além da possibilidade de gerar o FPA com essa grade pronta.

## **3. Tecnologias e ferramentas aplicadas**

Em vista do desenvolvimento do ADA de maneira concisa e eficaz, a implementação de tecnologias e suas respectivas ferramentas se faz necessária. Além disso, repositórios de controle de versão e Integrated Development Environment (IDE) deverão, e serão, utilizados.

---

<sup>1</sup>Essa adição será manual e de acordo com a prioridade escolhida. Portanto, um subprocesso, onde o Staff colocará os docentes na ordem e, igualmente, poderá alterá-la em caso de erro ou modificações futuras.

### 3.1. Tecnologias

A seguir estão as tecnologias utilizadas, suas características principais e, assim, porque foram escolhidas. A finalidade principal desse conjunto é escrever a aplicação de forma rápida e eficiente, em acordo com o conhecimento da equipe, concentrando toda a energia no desenvolvimento e na aplicação da lógica, e, logo, poupando tempo em funcionalidades básicas.

#### 3.1.1. Django

É um framework Web open source e de alto nível, desenvolvido em Python, que se baseia no padrão MTV, apresentando semelhança com o MVC. Assim, segue o princípio dry<sup>2</sup>, é moderadamente opinativo<sup>3</sup> e apresenta suporte para erros comuns de segurança. Além desses benefícios, foi escolhido devido a sua aplicação em grandes empresas (como Mozilla e Pinterest) e, principalmente, no SUAP do IFSP, o que permite manter o padrão de tecnologias no Instituto.

É uma mudança da separação anterior entre *front* e *backend*, pelo framework Angular e Spring Boot, respectivamente. Foi realizada visando uma maior produtividade, porque, em comparação, a configuração do ambiente do Spring é mais complexa, pois necessita implementar diversas dependências com o Maven. Isso difere do Django, que cria o projeto inicial em uma única linha de código. Ademais, o desenvolvimento se torna mais complicado com os dois framework, afinal, são dois serviços funcionando ao mesmo tempo, o que demanda muito processamento dos computadores do IFSP, tornando-os lentos e atrapalhando o processo de construção da aplicação, já que a equipe utilizará eles para programar.

**MTV** Derivação da arquitetura de software MVC, de três camadas, altera a nomenclatura e a relação entre os arquivos. O Model permanece o mesmo, como um canal de conexão entre os tipos de dados e como serão armazenados no Banco de Dados, e a exibição ao ter requisição à View. Essa é responsável, então, pelo gerenciamento das requisições e a lógica de negócio, com a formatação dos dados enviados pelo Model. Por fim, o Template é a interação com o usuário, através de uma exibição estática ou inserção de sintaxe de conteúdo dinâmico, com a renderização dos dados entregues pela View.

#### 3.1.2. Python

É uma linguagem de programação *open source* e de alto nível, interpretada em *scripts* e com Orientação a Objetos (OO), que apresenta tipagem dinâmica<sup>4</sup> forte. Logo, prioriza

---

<sup>2</sup>Permite que as aplicações sejam desenvolvidas com a maior quantidade de aproveitamento de código possível.

<sup>3</sup>Flexibilidade que o framework dá aos desenvolvedores à resolução dos problemas. Opinativo, já possui uma maneira correta de resolvê-los, sem margens; não-opinativo, não possui essas regras e deixa livre para resolvê-los como quiser. Django equilíbrio entre soluções prontas e arquitetura desacoplada com liberdade na resolução de erros.

<sup>4</sup>Tipo do dado é determinado no tempo de execução, de acordo com o valor do dado, não a partir da sua variável.

a agilidade por meio de sua sintaxe menor e simplificada, e sem muitas exigências gramaticais. E é por isso que foi escolhida, uma ótima opção que supriu de forma excelente a necessidade de uma aprendizagem rápida pela equipe e fácil codificação nos dispositivos do IFSP, além de poder ser facilmente integrada a outras linguagens de programação populares, caso seja necessário no decorrer do projeto.

É uma mudança, ao invés do Java, a linguagem de programação anterior. Ao realizar a troca do framework Spring Boot pelo Django, as linguagens foram substituídas também, pois cada framework tem a sua específica.

### 3.1.3. AJAX

O ajax é uma técnica de desenvolvimento *web*, caracterizada pela criação de aplicações interativas através de requisições ao servidor. Uma junção das funcionalidades do Javascript com a troca dos dados, armazenados e transmitidos, nesse caso, pelo JSON (mais próximo do JS). Foi escolhido justamente por servir como um canal de comunicação independente entre o cliente e o servidor.

### 3.1.4. JavaScript

É uma linguagem de programação de alto nível e interpretada em *scripts*, com recursos de OO e API, que apresenta tipagem dinâmica. Assim, por meio de um funcionamento assíncrono<sup>5</sup>, usa trechos dos códigos HTML para renderizar funções que proporcionem uma interação dinâmica local com o conteúdo da página. Foi escolhida para proporcionar essa dinamicidade em tempo real, recarregamento automático, em conjunto com o AJAX.

### 3.1.5. HTML

O HTML é uma estrutura responsável pela exibição dos dados no navegador *web*, caracterizado por seus elementos hierarquizados e sua marcação que abriga elementos como tags. Na aplicação ADA, é utilizada nos templates, explicados no parágrafo 2.4.1.1.1 .

### 3.1.6. CSS

O CSS é uma linguagem de marcação, responsável pela estilização de elementos HTML. Foi escolhido a fim de ajudar na formatação dos templates em detalhes específicos que, por vezes, não são compreendidos pelo framework, pois este é mais genérico.

### 3.1.7. Bootstrap

É um framework *front-end*, logo, voltado à estilização, e *open source*. Foi escolhido e mantido devido à agilidade no desenvolvimento da página para o usuário, característica

---

<sup>5</sup> A programação assíncrona é uma técnica na qual o programa inicia uma tarefa e ainda é capaz de executar simultaneamente outros eventos, ao invés de bloquear processos para esperar o término da execução.

nos framework, e, principalmente, devido à responsividade proporcionada.

### **3.2. Hospedagem**

A fim de ter uma instância em nuvem e conseguir fazer a hospedagem da aplicação, foi utilizada a AWS. É uma plataforma que disponibiliza diversos serviços de computação em uma rede de servidores remotos. Assim, é possível criar instâncias de máquinas com sistema operacional Windows ou Linux, de modo que a aplicação funcione constantemente, sem necessitar que um dispositivo pessoal fique ligado.

Somente com ela já é disponibilizada a aplicação na *web*. Todavia, o acesso é difícil, pois aparecerá somente o endereço IPv4 público da máquina virtual criada. Para resolvê-lo, foi comprado o domínio `mottarios.cloud` no *website* Hostinger.

É importante ressaltar que a equipe já tinha experiência com os serviços da Amazon, então, não foram testados outros serviços de hospedagem. Ademais, a AWS fornece um ano de gratuidade em alguns dos seus serviços, o que foi crucial para a escolha.

Apêndices

### **References**