UNIVERSIDADE FEDERAL DO ABC
BACHARELADO EM CIÊNCIA E TECNOLOGIA

DANIEL JOSÉ GOMES DE LIMA

UFABC TRACK: Uma aplicação web para notificar disponibilidade de matrículas em disciplinas da UFABC

SANTO ANDRÉ
2022

DANIEL JOSÉ GOMES DE LIMA

UFABC TRACK: Uma aplicação web para notificar disponibilidade de matrículas em disciplinas da UFABC

Trabalho de Conclusão de Curso apresentado à disciplina de Projeto dirigido do curso Bacharelado em Ciência e Tecnologia da Universidade Federal do ABC. Orientador: Dr. Diogo Santana Martins

SANTO ANDRÉ
2022

RESUMO

O presente trabalho descreve a aplicação web denominada “UFABC Track”, desenvolvida com intuito de auxiliar os discentes da Universidade Federal do ABC durante o período de reajuste de matrículas. A ferramenta funciona notificando o estudante por e-mail caso ocorra liberação de uma disciplina previamente selecionada. Para este projeto foram utilizadas tecnologias e arquiteturas de programação como: Java, JavaScript, MVC, Spring Framework, arquitetura limpa e infraestrutura em nuvem. Esta aplicação, ainda em estado prototipal, propõe uma continuidade em projetos futuros. Algumas das dificuldades encontradas tem relação com a falta de conhecimento acadêmico prévio em ferramentas específicas, as quais poderão ser melhor desenvolvidas em trabalhos futuros.

Palavras-chave: Java; Web; Matrícula; Programação;

SUMÁRIO

1. INTRODUÇÃO

A disciplina “Projeto dirigido” presente na grade do Bacharelado em Ciência e Tecnologia (BC&T) da Universidade Federal do ABC (UFABC), lecionada no primeiro quadrimestre de 2022 pela docente Dra. Iseli Lourenço Nantes Cardoso, tem como objetivo a prática interdisciplinar vivenciada nos componentes curriculares previamente adquiridos durante a graduação. O estudante deve elaborar um projeto teórico, experimental e/ou científico sob a orientação de um docente da instituição. O projeto tem a finalidade de propor soluções para problemas aderentes a temas como: representação e simulação, processos de transformação, estrutura da matéria, humanidades e informação; através de um produto inovador ou análise técnico-científica (PRÓ-REITORIA DE GRADUAÇÃO UNIVERSIDADE FEDERAL DO ABC, 2021).

Os discentes foram orientados pela responsável a buscar individualmente por outro Doutor para orientá-los durante o desenvolvimento do projeto. Deste modo, o Dr. Diogo Santana Martins passou a coordenar e avaliar este projeto por meio de reuniões quinzenais via a ferramenta Google Meet. Essa modalidade síncrona se justifica levando em consideração a grande crise sanitária de 2020 em decorrência da Pandemia do vírus COVID-19. A Portaria 544/2020 do MEC permitiu que as Universidades retomassem as atividades de maneira não presencial seguindo os protocolos e recomendações da Organização Mundial da Saúde (CONSELHO DE ENSINO, PESQUISA E EXTENSÃO DA FUNDAÇÃO UNIVERSIDADE FEDERAL DO ABC, 2020).

O objetivo deste projeto foi desenvolver um protótipo de uma ferramenta que auxilie os discentes durante o período de matrícula nas disciplinas desta instituição. Essa demanda decorre de um processo que atualmente exige um acompanhamento intensivo dos estudantes, ao buscar disciplinas que tem alta procura de inscritos na etapa de reajuste, visto que isso é recorrente em disciplinas obrigatórias do ciclo básico da graduação.

Esse trabalho se justifica socialmente na tentativa de tornar mais igualitária a inscrição de disciplinas durante o período de reajuste, visto que a maior parte dos discentes alegam realizar atividades empregatícias, o que pode afetar a disposição de tempo para acompanhar o processo.

Denominada “UFABC *Track”*, a ferramenta foi desenvolvida como um *website* que permite aos estudantes serem notificados via e-mail caso alguma das disciplinas desejadas sejam disponibilizadas, após a desistência de outro graduando. Trata-se de uma aplicação web com arquitetura Model-View-Controller (MVC) implementada via um sistema de processamento e armazenamento em nuvem utilizando as linguagens de programação Java, em seu *backend*[1] e *JavaScript,* em seu *frontend*[2]. A “UFABC *Track”* pode ser acessada por qualquer dispositivo com acesso à internet, contudo, devido ao estado inicial de desenvolvimento, é recomendado o acesso via computadores.

Dentre as dificuldades encontradas estão o tempo reduzido do quadrimestre letivo levando em consideração a complexidade do projeto, a instabilidade do primeiro servidor utilizado tendo que recorrer a uma alternativa paga e a falta de conhecimento acadêmico prévio em infraestrutura de nuvem e desenvolvimento da interface de usuário.

2. O MODELO DE MATRÍCULA DA UFABC

É imprescindível conhecer as diretrizes e o sistema de matrículas na UFABC para compreender a relevância que este projeto pode ter, e como ele pode impactar o cotidiano dos discentes.

2.1. Método de ingresso nas disciplinas

A UFABC possui um regime quadrimestral (período de quatro meses), estrutura esta que difere da grade curricular da maioria das instituições com ensino superior brasileiras. Os discentes são encorajados a ter a autonomia na formação de seu currículo, sendo disponibilizadas disciplinas para além do bacharelado interdisciplinar obrigatório. Desta forma, é possível escolher quais das disciplinas ofertadas irão se inscrever durante os três períodos de matrícula definidos

[1] Back-end, a camada de uma aplicação na qual ocorrem o processamento das regras de negócio e manipulação de dados.

[2] Front-end, a camada de uma aplicação na qual se localizam os componentes visuais e por meio da qual o usuário pode interagir com e visualizar informações providas pelo back-end.

anualmente através do calendário acadêmico da instituição (PRÓ-REITORIA DE GRADUAÇÃO UNIVERSIDADE FEDERAL DO ABC, 2015).

A soma da carga horária prática e teórica das disciplinas é classificada como créditos e há uma quantidade máxima de créditos que um discente pode se matricular. O limite é baseado em dois parâmetros: o CP (coeficiente de progressão), isto é, porcentagem de conclusão do curso e o CR (coeficiente de rendimento) que quantifica o desempenho acadêmico. Isso é baseado nas notas individuais de cada matéria dado pela função (1), de modo que o máximo de créditos disponíveis é dado pela fórmula (2), que pode ser observado nas expressões abaixo:

$$1.\quad CR = \frac{\sum_{i=0}^{d} C_i \times D_i}{\sum_{i=0}^{d} C_i}$$

$$2.\quad L = 16 \times 5CR$$

2.1.1. Matrícula

A solicitação de matrícula do discente no quadrimestre de ingresso é realizada automaticamente pela instituição, e cabe ao mesmo manter ou não as disciplinas atribuídas, não podendo inserir novas a priori. A partir do segundo quadrimestre deve ser feita a solicitação pelo estudante ou um representante legal através do site de matrícula da instituição[3] nas datas definidas pelo calendário acadêmico do ano vigente, dadas as regras da Resolução ConsEPE nº 219 - 16/03/17 (FUNDAÇÃO UNIVERSIDADE FEDERAL DO ABC, 2017). Neste período não há um limite de discentes inscritos por disciplina. Após esta etapa passa-se por um processo de ranqueamento baseado no CP e CR para estipular quais permanecerão e quais serão removidos ou remanejados para outras turmas.

[3] O site de matrícula de disciplina da UFABC foi lançado em 2009 e está disponivel através do site www.matricula.ufabc.edu.br

2.1.2. Ajuste e reajuste

Após a reorganização das turmas na etapa anterior, as vagas remanescentes e as matrículas canceladas serão disponibilizadas para ocupação. Só poderão ser canceladas inscrições em que a quantidade de requisições for inferior a 150% das vagas disponibilizadas. As inscrições que não foram canceladas antes do processo de ajuste não podem ser liberadas pelos discentes. Porém estes são livres para adicionar e remover novas inscrições nesta etapa caso tenham o requisito de créditos necessários e o número de requisição da turma seja menor que o número de inscritos (PRÓ-REITORIA DE GRADUAÇÃO UNIVERSIDADE FEDERAL DO ABC, 2022).

O reajuste segue as mesmas diretrizes supracitadas, contudo, consiste na inscrição de vagas remanescentes após a etapa de ajuste.

2.2. Escopo social

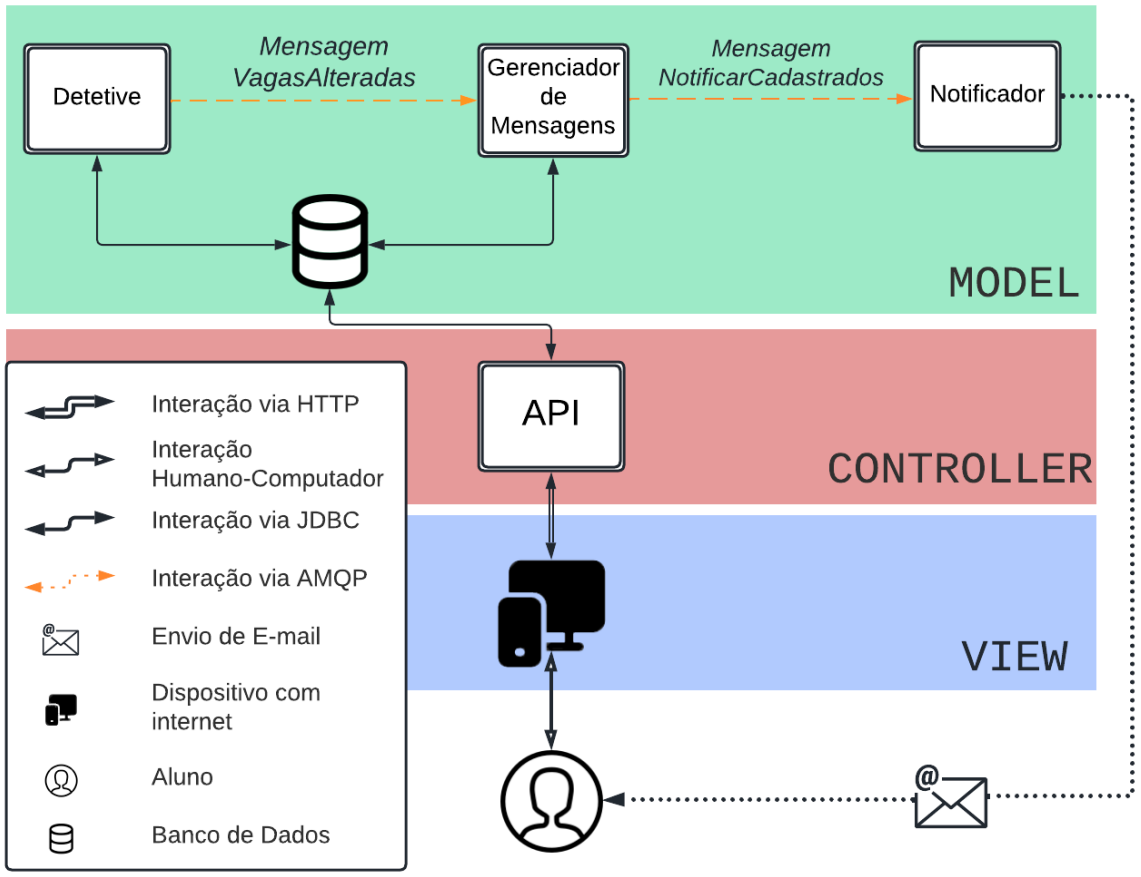
Um dos motivadores do desenvolvimento deste projeto é facilitar o processo de matrículas de alunos que não possuem a disponibilidade de acompanhar em tempo real a disposição das vagas.

De acordo com a pesquisa anual do perfil de discentes de graduação da UFABC (2021), 64% dos discentes realizam alguma atividade empregatícia remunerada ou não. Tendo em vista que a disponibilidade das vagas não tem horário pré-definido e podem acontecer de forma esporádica conforme supracitado, essa porcentagem de alunos podem vir a ser prejudicados ou menos favorecidos em relação aos demais.

3. ARQUITETURA

Um padrão arquitetural bem estabelecido é a utilização do *Model-View-Controller* (MVC) no desenvolvimento de aplicações *web*.

Figura 1: Arquitetura do projeto UFABC Track utilizando MVC



Fonte: Autor

O padrão arquitetural *Model-View-Controller* (MVC) foi formulado e divulgado em 1979 por Trygve Reenskaug, cientista da computação e professor da Universidade de Oslo, enquanto atuava como pesquisador visitante na Xerox Palo Alto Research Laboratory (PARC).

3.2. Concepção Original

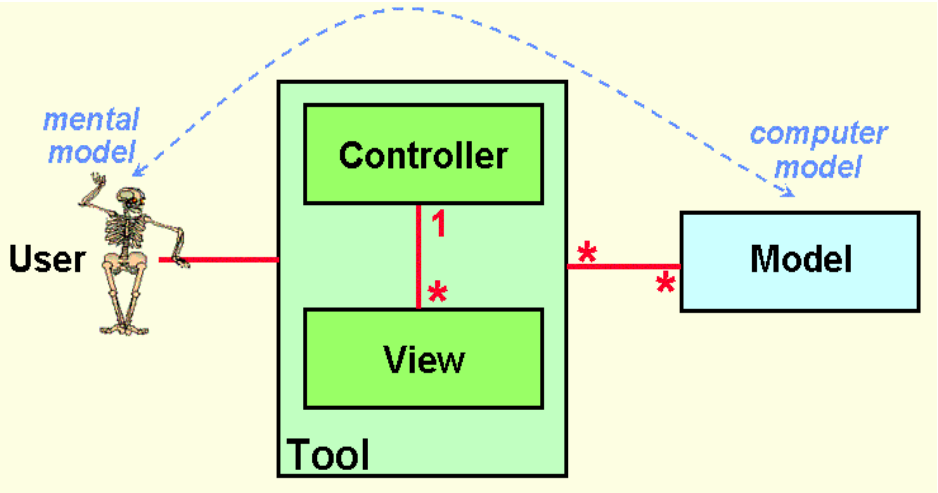
A MVC foi concebida como uma solução generalizada para permitir a manipulação de massas grandes e complexas de dados pelo usuário de modo abstraído, nomeado inicialmente de "Thing-Model-View-Editor". Em “THING-MODEL-VIEW-EDITOR an Example from a planning system", Reenskaug define cada uma das abstrações presentes no padrão.

- *Thing* é a representação abstraída do interesse do usuário com as informações.
- *Model* é descrito como a representação de uma abstração na forma de dados em um sistema, uma coletânea de dados com seus respectivos métodos necessários para o processamento das informações.
- *View* é definida de modo que para todo *Model* presente em um sistema existe uma ou mais views capazes de representá-lo graficamente, podendo realizar operações nos modelos associados a elas.
- *Editor* é a interface entre o usuário e uma ou mais views, que provê ao usuário um sistema de comandos adequado na forma de formulários e menus de interação que podem se alterar dinamicamente dependendo do contexto atual da aplicação. Fornece às *views* as coordenadas necessárias para representar os dados.

Posteriormente, Jim Althoff e outros pesquisadores da PARC implementaram outra versão denominada *“Model-View-Controller”*, termo utilizado até os dias atuais, da qual não teve participação o idealizador original. Algumas modificações foram realizadas, entre elas a definição como uma trindade e foi utilizado o termo *controller* invés de *editor*.

A MVC passou a ser uma estrutura aceita por Reenskaug e em abordagens futuras as views passaram a manipular a entrada de usuários de modo que sejam pertinentes a si e aos controladores que aceitam e lidam com entradas relevantes ao par *controller-view* como um, denominado “*tool*”.

Figura 2: Representação da arquitetura MVC por Reenskaug



Fonte: https://folk.universitetetioslo.no/trygver/themes/mvc/mvc-index.html

3.3. Aplicação contemporânea

Acompanhando o ritmo evolutivo da *internet* e processamento computacional, as aplicações aumentaram o nível de complexidade de seus sistemas, não mais restritas apenas à leitura e manipulação de dados. A arquitetura MVC permaneceu presente neste processo, fortemente vinculado às aplicações *web*, porém as tecnologias atribuídas aos itens da trindade fizeram com que eles fossem utilizados de maneiras distintas.

- A *Model* como responsável pela lógica da aplicação, encapsula gestores de armazenamento, processa, calcula e organiza as informações recebidas pelo *controller* e pode garantir a persistência dos dados para além do tempo de execução por bancos de dados externos ou arquivos brutos. (POP; ALTAR, 2014)
- A *view* como representação visual da *interface* de usuário obteve um amadurecimento no modo que as informações são disponibilizadas, utilizando métricas de qualidade de navegação para melhorar a experiência do usuário que a utiliza. Agora possui diferentes dispositivos e modos de apresentar os dados, tendo como responsabilidade diferentes tamanhos de telas que possam ser utilizadas de maneira satisfatória.

- O *controller* é responsável por lidar com os eventos realizados pelo usuário, aceitando requisições de diferentes tipos de plataformas, transcrevendo-as a *Model* e manipulando para que sejam representadas na *View* corretamente *(FREEMAN; SANDERSON, 2011).* Cabe a este lidar com as interações com aplicações externas e de usuários de forma simultânea através de *APIs* (Application Programming Interface) e *message brokers* e transmiti-las ao *Model*. Em aplicações *web* utilizam-se protocolos de troca de informação como o *REST* e *SOAP* em formatos de transferência de dados como *JSON* e *XML. Estes* são regularizados através dos *RFCs* (Request for Comments), documentos técnicos desenvolvidos e mantidos pelo *IETF* (Internet Engineering Task Force), instituição que especifica os padrões que serão implementados e utilizados na internet.

4. MÉTODO

Durante o desenvolvimento do projeto, devido a diferentes problemas técnicos foram alteradas as plataformas utilizadas no projeto. No primórdio, foi estipulada a utilização da plataforma de nuvem *Heroku* e *Vercel* para hospedagem do *back-end* e do *front-end* respectivamente, pois ambos possuem uma faixa gratuita. Porém, devido à instabilidade temporária na comunicação entre o *Heroku* com o repositório de códigos *Github*, não foi possível utilizá-lo como principal ferramenta de nuvem. Este passou a ser usado unicamente para hospedagem do módulo *API*, pois provém de forma abstraída e gratuita o certificado TLS[4], necessário para realizar as requisições HTTPS[5] entre o modelo mental e o computacional.

[4] TLS é uma sigla que representa Transport Layer Security, um tipo de segurança digital que permite a comunicação criptografada entre um domínio de site e um navegador.
[5] O HTTPS é uma extensão segura do HTTP (protocolo transferência de hipertexto). Os sites que configuram um certificado SSL/TLS podem utilizar o protocolo HTTPS para estabelecer uma comunicação segura com o servidor.

4.1. Cronograma

Tabela 1: Cronograma das etapas do processo distribuídos por semanas

Etapa	Primeiro Quadrimestre de 2022											
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
Contato e confirmação do Orientador												
Estruturação do projeto												
Fundamentação Teórica												
Móduto Detetive												
Móduto Gerenciador de Mensagens												
Módulo API												
Módulo Notificador												
Módulo Front-End												
Deploy em nuvem												
Manutenção Nuvem												
Final da prototipação												
Teste de usuários												
Entrega do projeto												

Fonte: Autor

4.2. Metodologias de desenvolvimento

No desenvolvimento de *softwares* modernos a utilização de metodologias de *design* busca facilitar o desenvolvimento, teste e manutenção de aplicações. Neste projeto foi utilizado a arquitetura limpa para padronizar o *back-end* do projeto em todos os módulos.

4.3. Microsserviços

A arquitetura foi desenvolvida com o padrão de desenvolvimento Microsserviços. Consiste na distribuição de uma aplicação em partes menores, cada pedaço é independente em seu desenvolvimento, manutenção e implantação. O UFABC *Track* foi dividido em cinco módulos distintos, estes são:

- “Detetive”, este módulo tem a função de monitorar as disciplinas disponibilizadas para matrícula e a alteração na quantidade de vagas. Utilizando os dados disponibilizados no site de matrícula através de dois *endpoints* (pontos de acesso a dados via protocolo HTTP), que são periodicamente consultados e persistidos com uso do *MySQL*. Ao ser identificada uma diferença na quantidade de vagas persistidas e o

dado disponibilizado pela instituição é enviada uma mensagem através com o tópico "vaga alterada" para o RabbitMQ.

- *“Front-end”*, este módulo foi desenvolvido com o *framework React* constituído por uma aplicação de página única. O usuário interage com o sistema por tabelas e botões, através do computador ou por dispositivos móveis. É possível visualizar as disciplinas disponíveis para matrícula e se cadastrar para receber notificações sobre ela, tal como cancelar as notificações através do link disponibilizado no *e-mail*.
- “API”, módulo responsável por transmitir as ações do usuário no *front-end* para os outros módulos do projeto e persisti-las.
- “Gerenciador de mensagens”, o módulo que consome as mensagens de alteração enviadas pelo “Detetive” e a busca pelos usuários que estão cadastrados para receber notificações. Ao definir os alvos das notificações, será enviada uma mensagem através do tópico “Notificar cadastrados" para o RabbitMQ.
- “Notificador”, o módulo que consome as mensagens de notificações enviadas pelo gerenciador de mensagens. Os usuários através de um servidor *Simple Mail Transfer Protocol* (SMTP, isto é Protocolo simples de transferência de e-mail) recebem o e-mail com os dados da disciplina que houve uma alteração na quantidade de vagas da mesma.

4.4. Arquitetura limpa

A *Clean Architecture*, ou arquitetura limpa, é uma metodologia de desenvolvimento publicada por Robert C. Martin (2017) no livro “*Clean Architecture: A Craftsman 's Guide to Software Structure and Design*”, baseado em publicações de outros autores. Como síntese dessas metodologias, busca utilizar seus conceitos para agilizar o desenvolvimento e auxiliar na manutenção de *software*. Atua utilizando camadas concêntricas com funções distintas entre si, de forma que a camada interna não deve consumir o conteúdo das camadas

exteriores, utilizando o conceito de injeção de dependência[6] para execução das ações.

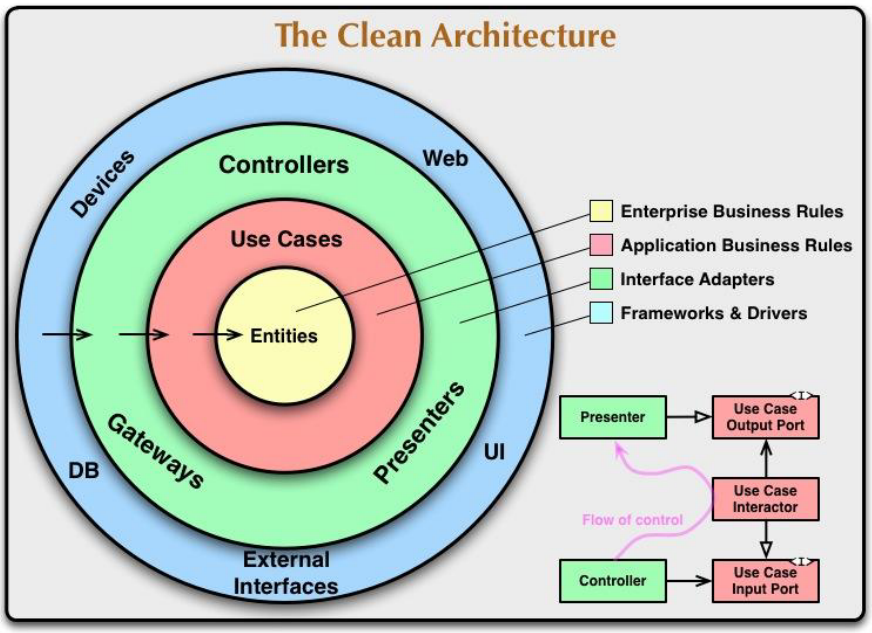


Figura 3: Estrutura concêntrica do Clear Architecture

Fonte: https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html

Robert (2012) esclarece que apesar da representação, o uso de quatro camadas não é obrigatório e novas podem ser utilizadas, desde que se mantenha dentro do uso da injeção de dependências. As principais camadas são as seguintes:

- *Entities*, onde se estabelecem os objetos e seus métodos sem que ocorra qualquer interação com o funcionamento ou regras de negócio da aplicação.
- *Use cases*, camada que agrega as regras, encapsula e implementa os casos de uso da aplicação, gerencia o fluxo de dados que tem origem e destino nas entidades.
- *Interface adapter*, conjunto de adaptadores que convertem os dados providos de frameworks externos, bancos de dados ou *web* para os *use cases.*

[6] Injeção de Dependência é um padrão de projeto usado para evitar o alto nível de acoplamento de código dentro de uma aplicação

- *Framework & Drivers*, camada externa da aplicação. É apenas neste ponto que é recomendado o uso de frameworks e interações internas da aplicação, de modo que, se qualquer alteração de tecnologia for feita nesta camada, não haverá impacto nas camadas internas (C. MARTIN, 2012).

4.5. Tecnologias utilizadas

Nesta aplicação foram utilizadas diversas tecnologias que são comumente aplicadas no desenvolvimento de *software*, cujo domínio é imprescindível para que se tenha uma aplicação de qualidade.

4.5.1. Java

Java é uma linguagem orientada a objetos criada na década de 1990 pela Sun Microsystems e atualmente sob a posse da *Oracle Corporation*. O diferencial de *Java*, entre outras linguagens, é a possibilidade de executá-la em todas as plataformas que suportam a tecnologia sem necessidade de recompilação, pois são compiladas para *Java bytecode,* linguagem intermediária lida pela *Java virtual machine* que independe da arquitetura da plataforma em que é executado. Este foi utilizado através do *Spring Framework* para estrutura e execução do *back-end* da aplicação UFABC *Track*.

4.5.2. MySQL

MySQL é um sistema de gerenciamento de banco de dados, desenvolvido pela *Oracle Corporation*, que utiliza Structured Query Language(SQL, isto é, Linguagem de Consulta Estruturada) como interface para manipular e consultar dados que foi utilizada para gerenciar os dados do projeto UFABC *Track*.

4.5.3. RabbitMQ

RabbitMQ é uma plataforma de mensagens de código aberto desenvolvida pela Erlang, que utiliza um protocolo chamado *Advanced Message Queuing Protocol* (AMQP). Através dela é possível enviar mensagens de forma

confiável e compatível com diversas tecnologias e linguagens de programação. A principal aplicação, assim como de outros servidores de mensageria, é a troca de informações entre os aplicativos distribuídos como o sistema de *microsserviços* utilizado neste projeto.

4.5.4. React

React é uma biblioteca de Javascript em código aberto mantida pela Meta, a sua principal funcionalidade é auxiliar no desenvolvimento de interface de usuários. Foi utilizado para desenvolver as telas do UFABC *Track* tanto para o acesso via computador quanto para o acesso pelo telefone celular.

4.6. *Interface* de Usuário

Buscando uma *interface* adaptável a vários dispositivos, foram utilizadas ferramentas que buscam facilitar o desenvolvimento de telas responsivas, de modo que, independente do tamanho do dispositivo, as informações estarão proporcionais e de fácil leitura para os usuários.

4.6.1. Telas em clientes desktop

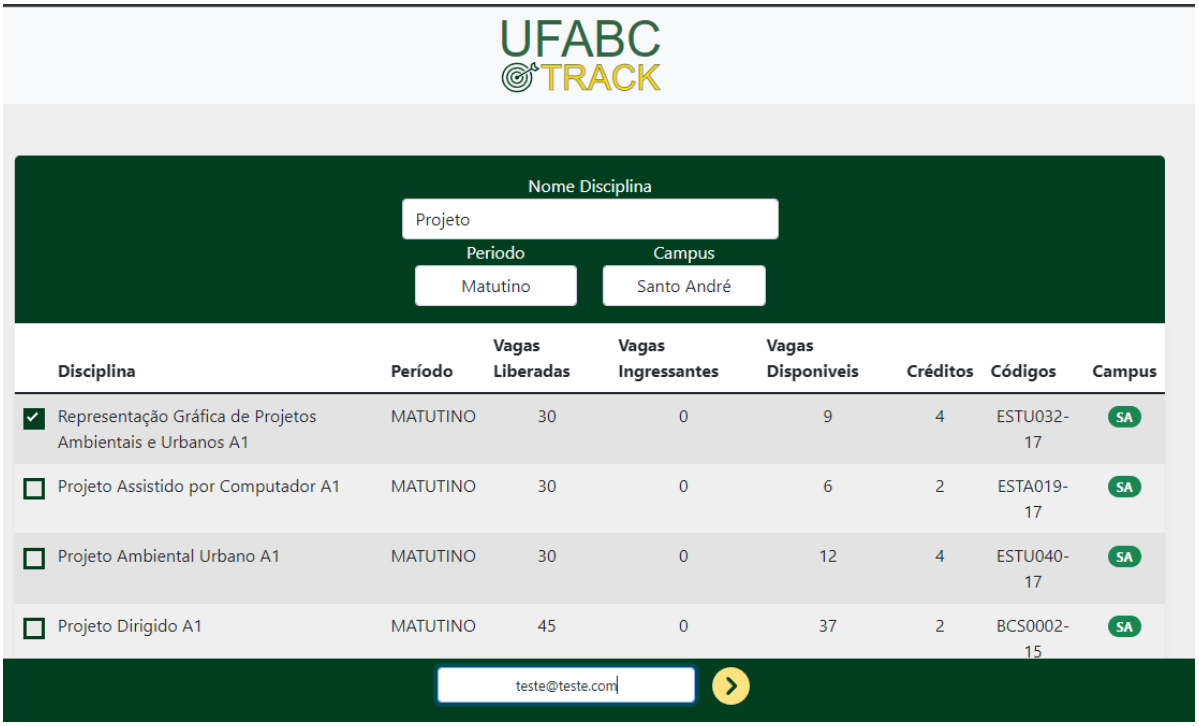
A interface consiste em uma lista com as disciplinas ofertadas junto a um formulário que facilita a busca. O usuário pode filtrar a lista de disciplinas através dos seguintes filtros:

- Localidade, o usuário pode escolher entre os câmpus Santo André, São Bernardo e ambos.
- Horário, o usuário pode escolher entre os períodos matutino, noturno e ambos.
- Nome da disciplina, o usuário pode escrever partes do nome ou o nome completo da disciplina desejada.

Após o usuário clicar nas caixas de seleção de uma ou mais disciplinas, ficará disponível um campo para preenchimento de e-mail no centro inferior da tela.

Ao indicar o e-mail e clicar no botão de prosseguir, uma nova janela será aberta confirmando as disciplinas selecionadas. O usuário pode excluir ou enviar a requisição da notificação das disciplinas.

Figura 4: Tela do aplicativo visualizada pelo computador

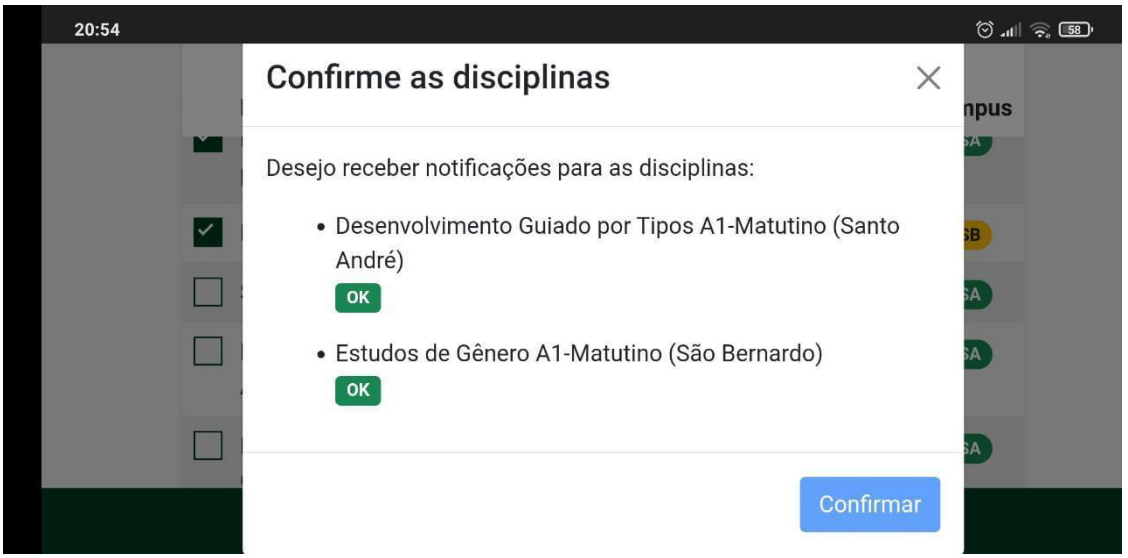


Fonte: Autor

4.6.2. Telas em dispositivos móveis

A interface para dispositivos móveis é similar à versão para computadores, porém, para dispositivos com comprimento de tela reduzido, há uma minimização nas colunas de dados apresentada ao usuário, porém o fluxo do cadastro e o modo em que é representado não se altera.

Figura 5: Tela do aplicativo visualizada com um aparelho celular no momento da inscrição



Fonte: Autor

5. AVALIAÇÕES E RESULTADOS

Para validar o funcionamento do sistema, foi realizado um teste integrado com usuários e dispositivos distintos, a fim de identificar possíveis problemas na implementação da ferramenta.

5.1. Planejamento

Foi utilizado um ambiente de teste em que os usuários se cadastraram em duas ou mais opções no site em uma lista simulada de disciplinas por meio de do computador ou dispositivo móvel. Foi disponibilizado um período de 2 dias para que tais ações fossem realizadas. Em intervalos de uma hora foi realizada a simulação de alteração de vagas na plataforma, que resultou na notificação dos usuários via e-mail conforme o roteiro.

5.2. Avaliação e conceitos

Após realizar o fluxo descrito no item 5.1, os usuários responderam o questionário com 6 perguntas descritas na tabela abaixo com 3 perguntas de

múltipla escolha e 3 perguntas quantitativas com valor de 0 a 10, utilizando a ferramenta *Google Form* para qualificar o uso da plataforma, conforme gráfico abaixo:

Tabela 2: Perguntas no formulário disponibilizado aos alunos

Pergunta	Resposta 1	Resposta 2	Resposta 3	Resposta 4	Resposta 5
Qual foi a plataforma utilizada para acessar o UFABC Track?	Celular	Tablet	Computador	Televisão	Outro
Conseguiu realizar o cadastro do alerta através da plataforma?	Sim, não houve dificuldades.	Sim, porém tive dificuldades.	Não consegui realizar o cadastro devido algum erro na plataforma.	Não consegui realizar o cadastro devido a disposição dos itens visuais na tela no dispositivo que utilizei.	Não. Outros motivos.
Recebeu os e-mails alertando a disponibilidade da disciplina?	Sim, recebi o e-mail de todas disciplinas cadastradas.	Sim, porém recebi parcialmente os alertas por e-mail.	Apesar de cadastrar a disciplina, não recebi os alertas por e-mail.	Não recebi, pois não consegui me cadastrar.	
Como classificaria a experiência utilizando a ferramenta?	-	-	-	-	-
Como classificaria os aspectos visuais da ferramenta?	-	-	-	-	-
Como classificaria a utilidade da ferramenta durante os períodos de ajuste da	-	-	-	-	-

Fonte: Autor

5.3. Resultados

Por meio do formulário divulgado em grupos de alunos da UFABC foi possível quantificar a recepção da comunidade quanto à ferramenta desenvolvida. Dos 29 alunos que responderam, temos que:

Foi utilizado apenas computador e celular nos acessos e 75% foi realizado através do computador.

Quanto à usabilidade da plataforma, 89,67% dos alunos conseguiram realizar o cadastro sem dificuldades, 6,9% realizaram com certa dificuldade e 3,45% não conseguiram realizar o cadastro devido a disposição dos itens visuais na tela.

Questionados quanto ao funcionamento do sistema, os usuários que não conseguiram receber os e-mails somam 6,9%, 10,34% receberam parcialmente os alertas e 82,76% receberam e-mail de todas disciplinas cadastradas.

A partir das três perguntas quantitativas, os usuários classificaram a experiência utilizando a plataforma e a nota média foi 9,41, os aspectos visuais obtiveram nota média 9,31 e a utilidade da ferramenta durante a graduação recebeu nota média de 9,79.

Foi identificado por alguns usuários erros no site durante o período de testes do UFABC *Track*, que poderão ser resolvidos após a etapa de prototipação.

6. CONSIDERAÇÕES FINAIS

O presente trabalho foi entregue para a disciplina “Projeto Dirigido”, a qual tem como objetivo proporcionar aos alunos a prática de desenvolver dissertações em forma de trabalho acadêmico. A prototipação da ferramenta UFABC *Track* apresenta uma aplicação *web* disponível para qualquer dispositivo com acesso à internet. Com isso foi possível ter contato com tecnologias atuais como: processamento e infraestrutura em nuvem, interface de usuário e sistema de mensageria AMQP.

Buscou-se um protótipo para proporcionar uma melhor experiência para os alunos no processo de matrícula da UFABC. Como obstáculo encontrou-se a indisponibilidade de aplicações formalizadas com dados da instituição e descrições formais do processo de reajuste de disciplinas. Como resultado obteve-se que há uma relevância do projeto na comunidade discente da UFABC e que o aplicativo já se encontra em um estado de uso funcional, apesar de algumas melhorias pendentes que podem ser implementadas posteriormente.

Existem adições que podem ser implementadas como: otimização dos sistemas já existentes, adição de ferramentas que proporcionam melhor manutenção e prevenção de erros. Serão também retrabalhados pontos que foram apontados pelos usuários durante o período de testes.

Desta forma, pode-se concluir que o conteúdo apresentado é pertinente para a formação acadêmica, pois trata-se de uma tecnologia que colabora para a comunidade. Considera-se relevante a continuidade da pesquisa, para isso, o código está disponível no repositório online *Github,* para possíveis alterações e sugestões.

Ainda não há mensuração do que o isolamento social pode ocasionar para esses alunos em diversos fatores, pois a desigualdade de acesso de dispositivos e internet pode ter afetado a agilidade em realizar a matrícula no reajuste, devido a estrutura ordinal e limitada de vagas. Isso poderá ser melhor trabalhado com o término da pandemia e a necessidade de reorganização acadêmica.

Revisão Bibliográfica

CATÁLOGO de disciplinas. *In*: PRÓ-REITORIA DE GRADUAÇÃO UNIVERSIDADE FEDERAL DO ABC (Brasil). **Catálogo de disciplinas** . 2. [*S. l.*], 2021. Disponível em: https://prograd.ufabc.edu.br/catalogos-de-disciplinas. Acesso em: 27 abr. 2022.

PRÓ-REITORIA DE GRADUAÇÃO UNIVERSIDADE FEDERAL DO ABC (Brasil). **Projeto pedagógico do curso bacharelado em ciência e tecnologia**. [*S. l.*], 2015. Disponível em: https://prograd.ufabc.edu.br/bct/pps. Acesso em: 27 abr. 2022.

FUNDAÇÃO UNIVERSIDADE FEDERAL DO ABC (Brasil). **Sistema de Matrículas**. [*S. l.*], 2009. Disponível em: https://matricula.ufabc.edu.br/. Acesso em: 27 abr. 2022.

CONSELHO DE ENSINO, PESQUISA E EXTENSÃO DA FUNDAÇÃO UNIVERSIDADE FEDERAL DO ABC. RESOLUÇÃO Nº 240/2020 - CONSEPE. Boletim de Serviço nº 963, [*S. l.*], 15 jul. 2020. Disponível em: https://www.ufabc.edu.br/images/stories/comunicare/boletimdeservico/boletim_servico_ufabc_963.pdf#page=6. Acesso em: 29 abr. 2022.

FUNDAÇÃO UNIVERSIDADE FEDERAL DO ABC. RESOLUÇÃO CONSEPE N° 219, DE 16 DE MARÇO DE 2017. **RESOLUÇÃO CONSEPE N° 219, DE 16 DE MARÇO DE 2017**, Brasil, 16 mar. 2017. Disponível em: https://www.ufabc.edu.br/images/consepe/resolucoes/resolucao-219-estabelece-as-normas-para-matricula-em-disciplinas.pdf. Acesso em: 27 abr. 2022.

PRÓ-REITORIA DE GRADUAÇÃO UNIVERSIDADE FEDERAL DO ABC. **Orientações para o reajuste de matrícula**. [*S. l.*], 2022. Disponível em: https://prograd.ufabc.edu.br/pdf/reajuste_2022.1_orientacoes.pdf. Acesso em: 27 abr. 2022.

PRÓ-REITORIA DE PLANEJAMENTO E DESENVOLVIMENTO INSTITUCIONAL DA UNIVERSIDADE FEDERAL DO ABC (Brasil). **Perfil do Discente de Graduação**. 2019. ed. [*S. l.*], 2021. Disponível em: https://propladi.ufabc.edu.br/informacoes/perfil. Acesso em: 27 abr. 2022.

POP, Dragos-Paul; ALTAR, Adam. Designing an MVC Model for Rapid Web Application Development. **Procedia Engineering**, Romania, v. 69, n. 2019, p. 1172-1179, 25 mar. 2014. DOI https://doi.org/10.1016/j.proeng.2014.03.106. Disponível em: https://www.sciencedirect.com/science/article/pii/S187770581400352X?via%3Dihub. Acesso em: 27 abr. 2022.

FREEMAN , Adam; SANDERSON , Steven. **ASP.NET MVC 3 Framework**. 3. ed. [*S. l.*: *s. n.*], 2011. 852 p. ISBN 1430234040. Disponível em: https://sd.blackball.lv/library/Pro_ASP.NET_MVC_3_Framework.pdf. Acesso em: 30 abr. 2022.

THE CLEAN Code Blog. [*S. l.*], 2012. Disponível em: https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html. Acesso em: 30 abr. 2022.

C. MARTIN, Robert. **Clean Architecture: A Craftsman ‘s Guide to Software Structure and Design**. [*S. l.*]: Pearson, 2017. 432 p. ISBN 0134494164.

REENSKAUG, Trygve. **THING-MODEL-VIEW-EDITOR an Example from a planning system.** [*S. l.*]: Pearson, 12 jun. 1979. Disponível em: https://folk.universitetetioslo.no/trygver/1979/mvc-1/1979-05-MVC.pdf. Acesso em: 1 maio 2022.