**TCC – Site para produção de textos narrativos com enredos alternativos.**

**TCC – Barber Shop App (Cancelado)**

Aluno: Carlos Henrique Camilo da Fonseca

Orientador: Fernando Cesar Balbino

Coorientador: André Violin\*

**Diário de Bordo**

1º dia – 05/08/2018

Olá, primeiramente gostaria de deixar registada aqui minha ideia de trabalho de conclusão de curso, irei desenvolver uma aplicação para gerenciar agendamentos de cortes de cabelos. Será uma aplicação simples onde os próprios usuários agendarão seus cortes através de seus celulares ou tabletes. Essa ideia surgiu no semestre passado, na disciplina de Empreendedorismo, ministrada pelo professor Pedro Fonseca, onde fizemos um estudo de viabilidade para a implantação dessa ideia. Essa ideia não está limitada apenas a barbearias, ela pode ajudar também manicures, pedicures, lava-rápidos etc. Todos os prestadores de serviço que precisam agendar um horário de atendimento para seus clientes podem ser beneficiados. Vamos restringir aqui o nosso escopo apenas para as barbearias. Já conversei com duas pessoas: uma amiga minha que é cabelereira em Batayporã/MS, ela topou testar a aplicação em seu comércio, e o outro é cabelereiro em Nova Andradina/MS, também topou testá-la.

Vou aproveitar o desenvolvimento desse trabalho como projeto também da disciplina de Desenvolvimento de Aplicações para Dispositivos Móveis (DADM), ministrada pelo professor André Violin. Assim, poderei gastar mais energias em um único projeto.

Hoje, domingo, comecei a modelar as interfaces do nosso aplicativo, para isso estou usando o Adobe XD. O professor André que nos indicou essa ferramenta para prototipação, é a primeira vez que estou usando ela. Achei ela bem funcional, com ela podemos criar animações dos eventos dos cliques.

2º dia – 13/08/2018

Semana passada conversei com o professor Fernando Cesar Balbino, ele topou orientar esse projeto. Fiquei muito feliz por ele ter aceitado meu convite, apresentei a ele duas ideias, essa da barbearia e uma outra que seria desenvolver uma aplicação para ajudar pessoas com depressão. Abandonei essa segunda ideia por enquanto por que ela exigirá muita pesquisa na área de psicologia, e esse tempo nós não temos no momento, quem sabe em um mestrado?

Hoje, criei o projeto com react onde será desenvolvida nossa aplicação. Vou começar a desenvolver os componentes que iremos usar. Ainda não sentei com o professor Fernando para acertamos os detalhes do projeto, mas essa semana faremos isso.

3º dia – 14/08/2018

Hoje foi o primeiro dia de desenvolvimento dos componentes que iremos utilizar. Não foi possível desenvolver muita coisa devido a conexão de internet da ufms estar muito ruim.

**Obs.: Instalar esses plugin de auto complete e importação automática para react quando estiver conectado em um wifi.**

**- Full react/react native/react router/redux/...**

**- Simple react snippets ...**

4º dia – 15/08/2018

Deu carunjo no algodão, kkkkkkkkk. Ontem conversei com o professor Fernando sobre o tcc, foi uma conversa de corredor, não deu para estender muito o assunto, mas ele me disse que na sexta-feira dessa semana ele vai me apresentar outra ideia de tcc para desenvolvermos. Ela está relacionada com tecnologia e educação. Não sei ainda se vou trocar ou não, se a ideia dele servir também para o projeto do professor André Violin posso até trocar, caso contrário continuarei com a minha ideia da barbearia.

5º dia – 07/09/2018

Hoje, feriado da independência do Brasil, e eu aqui mexendo com TCC. Troquei de tema. Agora temos: “DESENVOLVIMENTO DE UM SISTEMA WEB PARA A PRODUÇÃO DE TEXTOS NARRATIVOS COM ENREDOS ALTERNATIVOS”. Essa é a ideia do professor Fernando. Aceitei o desafio, irei desenvolver duas aplicações distintas para as disciplinas de TCC (Trabalho de Conclusão de Curso) e de DADM (Desenvolvimento de Aplicações para Dispositivos Móveis).

A ideia consiste basicamente em montar uma plataforma onde os usuários possam criar estórias que possam ter vários tipos de desfecho. Dentro dessas estórias, os autores podem criar links para finais diferentes, de acordo com o gosto do leitor. Nosso principal objetivo com esse projeto é poder incentivar mais a escrita e a leitura nos alunos do ensino médio e fundamental.

O professor Fernando já começou a montar a documentação desse projeto, eu não peguei ele 100% cru, e me parece que o professor possui um carinho especial por essa ideia e quer ver seu projeto em uso. Pensando nisso devemos ter alguns cuidados.

Ele me pediu como primeira missão que eu montasse um projeto com as interfaces da nossa plataforma. Estou usando o Adobe XD, a mesma ferramenta que eu usei para modelar as telas do BarberAppShop, que por sinal, ficaram ótimas, kkkk. Continuo trabalhando nesse projeto na disciplina de DADM.

Eu pensei que essa missão seria de boa, mas não é, pois devemos sempre pensar no nosso usuário final. E pensando nisso, temos que tomar muito cuidado com as cores que iremos utilizar. É sabido por muitos que as cores possuem a capacidade de influenciar inconscientemente nossas emoções, devido a isso, optamos por usar cores neutras e calmas, que não afetem as ideias dos escritores. Nossos autores dos textos também podem passar horas e horas escrevendo, pensando nisso também optamos por cores que não cansem muito a visão.

Só para nota de esclarecimento, vou explicar como funcionará os acessos do sistema. Os textos podem ser privados ou públicos. Teremos dois perfis de usuários: Leitores e Escritores (ou Autores). Os leitores apenas leem os textos do tipo público e escolhem os tipos de enredos que preferem. Em uma narração podem haver mais de um escritor, porém cada autor é responsável pelo que escreveu.

Prosseguindo com as interfaces, optamos por tons de verdes e azuis:

Verde: emana o instinto de algo natural e orgânico, o que faz com que seja uma cor agradável aos olhos. Ela costuma estar relacionada à uma sensação de relaxamento, harmonia e segurança.

Azul: transmite sentimentos de paz e confiança. É uma das cores mais comuns na web, e inconscientemente diminui o apetite, o que no nosso caso, pode ajudar os escritores a manter o foco no texto em produção.

Fonte: <https://pt.wix.com/blog/2012/12/usando-as-cores-certas-para-o-seu-site/>

Sabendo das cores que vamos utilizar, fomos a procura dos melhores tons dessas cores para nosso site. Em uma pesquisa pela internet encontramos algumas paletas de cores que nos agradou. Depois entramos no site coolos.co e escolhemos sobre tons das cores escolhidas.

Cores escolhidas: #77C9D4, #57BC90, # 015249, # 1D3461, # A5A5AF.

Fonte: <https://coolors.co/77c9d4-57bc90-015249-1d3461-a5a5af>

Fonte: <https://www.canva.com/pt_br/aprenda/cores-para-sites-50-paginas-impactantes/> Tópico 20. Degradê e cores desbotadas.

Após escolhidas as cores, começamos a modelagem das interfaces.

=>Documentação das atividades realizadas no sexto semestre

6º dia – Dia 10/03/2019

Voltando a todo vapor para a conclusão desse trabalho. Já temos o projeto de interface concluído e iniciamos o projeto frontend da nossa aplicação, que vai ser um site para ajudar na produção de textos narrativos dinâmicos.

Duarante as férias aproveitei para conhecer e estudar as tecnologias que iremos utilizar durante esse projeto, elas são: NodeJS, MongoDB, Mongoose, Express, React, Redux. Fiz alguns cursos da Alura e da Udemy para servirem de apoio a esse projeto, estou lendo também o livro Node.js (Aplicaçõess web real-time com Node.js) de Caio Ribeiro Pereira, esse livro peguei emprestado com meu colega de classe Jhonatan Nobre. Agora, com uma boa bagagem de aprendizado podemos iniciar a aplicação com mais confiança.

Iremos dividir em três blocos principais as tarefas a serem realizadas daqui em diante.

1 – Correção do texto;

2 – Backend da aplicação;

3 – Frontend da aplicação.

7º dia – Dia 11/03/2019

Hoje, segunda-feira, terminei de ler o livro que peguei emprestado com meu amigo Jhonatan, esse livro é bem conceitual e explica muito bem quais são os passos necessários para colocar uma aplicação Node.js em modo de produção, foi uma excelente leitura, recomendo.

No meu computador eu utilizo o sistema operacial Linux Manjaro, ele é baseado no Arch Linux, tenho essa preferencia por software livre, particularmente os dois sistemas operacionais que eu mais gosto são os Linux Mint e Linux Manjaro. Não gosto muito do Windows, principalmente dos Windows 10 e 8, o Windows xp e 7 até que eu gostava mais, mas as atualizações do sistema operacional da Microsoft não me agradaram muito, principalmento o 10 com um processo chamado Cortana que vigia tudo o que o usuário faz e nao tem como matar esse processo. Escolhi o Manjaro simplesmente pelo otimo gerenciador de pacotes que ele possui, o Pacman. Contudo, embora eu não curta muito os sistemas da Microsotf, eles possuem um software que eu gosto muito, estou falando do pacote Office, para poder utiliza-lo em meu computador estou usando o PlayOnLinux (Wine), esse programa é como se fosse uma máquina virtual ou um emulador que permite instalar aplicativos da plataforma Windows na plataforma Linux, nele eu instalei o Word, Excel e PowerPoint para editar os textos que estou produzindo ao longo do desenvolvimento deste trabalho. Outro programa que eu precisava instalar no meu computador era o Adobe XD, porem não consegui, tive que utilizar um pc de um colega para poder visualizar os projetos de interface, já que a suite da Adobe não executa na plataforma Linux.

Os **primeiros passos** a serem tomados nesse semestre em relação ao desenvolvimento desse TCC serão: **Iniciar a aplicação Backend e Frontend com a parte de autendicação, cadastro de usuário e recuperação de senha.** Ao longo dessa semana tranalharemos com esse objetivo.

8º dia – Dia 13/03/2019

Comecei a implementação do backend da nossa aplicação. Como já é sabido, estamos usando o node.js e o mongoDB, mas para ganharmos tempo e melhorarmos requisitos de segurança do nosso servidor estamos usando alguns plugins como dependências:

{

"name": "tcc-backend",

"version": "1.0.0",

"description": "",

"main": "src/loader.js",

"scripts": {

"dev": "nodemon",

"production": "pm2 start src/loader.js --name tcc-backend"

},

"keywords": [],

"author": "",

"license": "ISC",

"dependencies": {

"bcrypt": "^3.0.4",

"body-parser": "^1.18.3",

"express": "^4.16.4",

"express-query-int": "^3.0.0",

"jsonwebtoken": "^8.5.0",

"lodash": "^4.17.11",

"mongoose": "^5.4.19",

"mongoose-paginate": "^5.0.3",

"node-restful": "^0.2.6",

"pm2": "^3.4.0"

},

"devDependencies": {

"nodemon": "^1.18.10"

}

}

Acima, está toda a configuração do nosso projeto, ela está na pasta raiz do projeto dentro do arquivo package.json. As dependências existentes em nosso servidor são:

bcrypt: plugin para criptografar senhas;

body-parser: middleware do node.js, ele analisa corpos de requisição de entrada em um middleware antes de seus manipuladores, disponíveis sob a propriedade req.body;

express: framework para node.js, ele é projetado para construir aplicativos web e api’s, fornecendo a estrutura básica de um servidor node.js;

express-query-int: middleware que converta cadeias de consulta em números para aplicativos express / connect;

jsonwebtoken: plugin que gera uma chave token para controle de sessão de usuário logado ao sistema;

lodash: biblioteca para exportar módulos. O Lodash torna o JavaScript mais fácil eliminando o incômodo de trabalhar com matrizes, números, objetos, strings, etc. Os métodos modulares do Lodash ajudam à: Iterando matrizes, objetos e cordas; Manipulando e testando valores; Criando funções compostas;

mongoose: é uma ferramenta de modelagem de objetos do MongoDB projetada para funcionar em um ambiente assíncrono.

mongoose-paginate: biblioteca de paginação personalizada baseada em cursor para o Mongoose com rótulos personalizáveis.

node-restful: registra os recursos do mongoose e as rotas REST padrão automaticamente.

pm2: plugin para monitorar o funcionamento do servidor quando estiver em produção, ele exibe o uso de mémoria RAM, o tamanho da aplicação, o status de funcionamento, etc. Para levantarmos nosso servidor usando o pm2 devemos usar o comando **npm run production**.

Temos também uma dependência de desenvolvimento, que será usada apenas no ambiente de produção:

nodemon: O nodemon é uma ferramenta que ajuda a desenvolver aplicativos baseados em node.js reiniciando automaticamente o servidor quando as alterações de arquivo no diretório são detectadas.

Até agora, implementamos o CRUD (Creat, read, update e delete) para o cadastro de textos narrativos e de usuários. O nosso servidor está cadastrando e listando os documentos que representan os textos escritos pelos autores, mas as opções de atualizar e deletar não estão funcionando, provavelmente é algum conflito entre versões do node. A parte do CRUD do usuário ainda não foi testada. Após todos os erros serem corrigidos implementares a comunicação entre esses dois tipos de documentos, textos narrativos e usuários.

9º dia – Dia 15/03/2019

Ontem conversei durante o intervalo com o professor Fernando, falei que comecei o desenvolvimento da aplicação mas ainda não parei para mexer com as correções do texto da dissertação, ele disse para eu focar na produção do nosso site e deixar para fazer as correções do texto mais próximo da data da apresentação final do tcc.

Para melhorar nossa comunicação vou compartilhar os projetos backend, frontend e o diário de bordo no GitHub. Dessa forma, vamos poder acompanhar melhor o nosso progresso e teremos sempre um backup de segurança, caso ocorra algum imprevisto com nossos computadores não perderemos nossos arquivos.

10º dia – Dia 16/03/2019

Compartilhei a aplicaçãono GitHub, fiz o primeiro commit ontem mesmo. O link para acessá-la no GitHub é: <https://github.com/FonsecaCarlos/site-tcc>.

Depois, criei os arquivos README.md, usei o site <https://dillinger.io/> para poder fazer uma pré-visualização de como eles irão ficar no GitHub. Foram criados três arquivos desse, o primeira está na pasta raiz, ele exibe um resumo e esclarrece o objetivo geral da aplicação; o segundo está no diretório do projeto backend, ele mostra a configuração necessária para que o servidor funcione; e o terceiro está no diretório do projeto frontend, ele também mostra a configuração necessária para que a aplicação funcione corretamente. Para configurar corretamente esses aquivos utilizei as instruções passadas por esse usuário do GitHub -> <https://github.com/urielcaire/aprendamd> .

Meu amigo de classe, Jhonatan Nobre, me indicou um programa para controlar o versionamento dos meus projetos no Git, o nome dele é GitKraken (<https://www.gitkraken.com/>). Fiz o download dele ontem mas ainda não utilizei ele na prática, mas pelo que o Jhonatan me mostrou ele é muito ágil e por possuir uma interface bem intuitiva acaba facilitando o controle de versionamento.

11º dia – 02/04/2019

Antes de ontem, domingo, terminei de configurar a parte de autenticação do sistema. Tanto o backend como o frontend estão funcionando perfeitamente. O cadastro de novo usuário e a criação de uma chave token estão funcionando da forma esperada. Após a autenticacao o usuário é redirecionado para a página inicial da aplicação onde estarão listadas as suas histórias, por enquanto essa página ainda não foi implementada, esse será nosso próximo passo.

Estamos usando duas api’s no backend, a primeira pode ser acessada por qualquer pessoa, estamos chamando ela de openApi, pois ela é aberta. A openApi dá acesso a três rotas: /login, /signup e /validateToken. A outra api usada é a protectedApi, ela só pode ser acessada com um token válido, nela podemos acessar as rotas de listagem, cadastro, edição e remoção dos textos narrativos.

Estou usando o GitKraken para gerenciar esse projeto, ele deixa mais intuitivo o versionamento do código, fiz alguns textes com ele na atividade de criptografia do professor Fábio e gostei do modo do fluxo de trabalho.

12º dia – 08/04/2019

Nesse final de semana comecei a desenvolver o interface inicial do sistema, onde serão listadas todas as história públicas e todas as histórias de autoria do usuário autenticado. Não estamos utilizando nenhuma bibliotca CSS, apenas código puro. Para montar o layout do nosso site estamos utilizando muito um novo recurso do CSS3, que é o display flex. Usamos como guia a documentação criada pela Origamid (<https://origamid.com/projetos/flexbox-guia-completo/>). Por enquanto a fonte utilizada é a Molengo, estamos utilizando ela diretamente da api do Google Fonts (<https://fonts.google.com/specimen/Molengo?selection.family=Molengo>).

Foram realizadas também algumas melhorias na aplicação backend. Criamos uma nova rota /narrativeText/index . Nela serão exibidas todas as histórias do sistema por meio de paginação, estamos usando o plugin Mongoose.paginate para isso, nela serão exibidos no máximo 15 histórias por página, escolhemos essa quantidade por se encaixar no design do site, nele serão exibidas 3 histórias por linha.

Adicionamos alguns atributos no nosso modelo de narrativeText, são eles: alternativeText – array com os id’s dos enredos alternativos e isPublic – bolleano que registra se a história é pública ou não.