

Chat Ibilce

Uma aplicação de chat em tempo real para a
comunidade do IBILCE

Sumario

- Intrudução
- Motivação
- Tecnologias envolvidas
 - Plataforma utilizada
 - Linguagens de programação
 - Frameworks
 - Hospedagem
 - Banco de dados
 - Versionamento
- Conclusão

Introdução

- Aplicação de chat em tempo real
- Voltado para a comunidade interna no Ibilce;
- Estrutura de salas, com diversas conversas em simultâneo;
- Sistema com cadastro de usuário;
- Disponível para mobile e desktop.

Motivação

- Forma de comunicação rápida e informal;
- Possibilidade de conhecer novas pessoas;
- Salas de discussões sobre tópicos da faculdade;
- Comunicação de forma anônima.

Tecnologias envolvidas

- Plataformas escolhidas - Web App e Mobile (híbrido);
- Linguagem de desenvolvimento - Javascript na plataforma Node.js;
- Frameworks utilizados - Express, Socket.io e Passport.js;
- Hospedagem do serviço - Heroku;
- Banco de dados utilizado - MongoDB;
- Controle de versionamento - git e github
- Desenvolvimento Mobile: Cordova

Node.js

- Node.js é um runtime environment de Javascript baseado na V8 Machine do Google;
- Ele possibilita o uso de Javascript no server-side;
- Utiliza o NPM como gerenciador de pacotes, que facilita muito a instalação e gerenciamento dos pacotes a serem usados;
- NPM fez muito sucesso e é utilizado por vários projetos hoje em dia, contendo MUITOS pacotes disponíveis (têm uma galáxia inteira!)

Node.js (2)

- Linguagem altamente adotada no mercado;
- Usando Javascript no back-end permite usar uma stack inteira de Javascript;
- Ótima performance para tarefa que dependem de entrada e saída, devido a natureza assíncrona do Node.js

Express

- Framework de API REST mais famoso do Node.js, dá nome a principal stack de software do Node, MEAN(MongoDB Express Angular Node.js);
- Framework minimalista focado em programar as rotas, explicitando os endpoints e especificando sua lógica;
- Possui vários módulos que ajudam no uso do framework, como Morgan(logging), express-session(armazenamento de sessões) e outros;
- Trabalha com vários tipos de templating engine, no caso foi usado EJS

Socket.io

- Framework de Web Socket para aplicação em tempo real;
- Fácil utilização, utiliza estratégia de eventos para comunicar o servidor com o cliente Web Socket;
- Não trabalha só com Strings, também funciona com BLOBs de tipos variados.

Passport.js

- Framework utilizado para facilitar a autenticação;
- Utiliza strategies para fazer a autenticação, tendo Local, Facebook, GitHub e vários outros;
- No nosso caso, foi usado a strategy Local para fazer a autenticação de nossos usuários, salvando a sessão no nosso banco.

Hospedagem - Heroku

- Serviço de hospedagem de aplicações;
- PaaS - platform as a service;
- Suporta diversas linguagens;
- Integração com diversos aplicativos;
- Fácil gerenciamento e monitoramento;
- Possui um plano gratuito “ilimitado”;
 - Contas verificadas possuem 1000 horas de execução;
 - Aplicações “adormecem” após 30 minutos de inatividade.

Hospedagem - Heroku(2)

- Facilidade na hospedagem;
- Diversas informações de logging;
- Aplicação sempre disponível;
- Integração com serviços de banco de dados;
- Integração com github;
- Gratuito para sempre.

Banco de dados - MongoDB (mLab)

- Integração entre Heroku e mLab;
- Armazena as credenciais de acesso ao programa;
 - Utiliza formato de hash bcrypt para as senhas.
- Armazena as sessões de usuários logados no sistema;
- Interface web de fácil gerenciamento das coleções;
- Plataforma de banco de dados gratuita;
 - Plano compartilhado de até 500MB.

Versionamento - Git

- Protocolo de controle de versionamento de código;
- Permite o trabalho conjunto no desenvolvimento;
- Estimula o desenvolvimento de código open source;
- Utilizado no heroku para fazer deploy das aplicações.

Versionamento - Github

- Plataforma gratuita de controle de versionamento de código;
- Rede social colaborativa de desenvolvimento open source;
- Possui integração com o Heroku;
- Todo commit na branch master gera um novo deploy da aplicação.

Desenvolvimento Mobile - Cordova

- Desenvolvimento de Aplicativos Híbridos
- Possibilita o uso de tecnologias web com aparência de app
 - CSS/HTML
 - Javascript
- Interação com o celular
- Emula o sistema android e cria a APK

Conclusão

A partir de um conjunto de ferramentas e tecnologias abertas, foi possível desenvolver uma aplicação que contribua com a comunidade interna do Ibilce.

Se trata de uma aplicação simples e de fácil utilização.

Não necessita instalação (versão web).

Aplicação é de fácil manutenção e funcionamento estável.

Através do uso de serviços gratuitos, pretende-se deixar a aplicação disponível por tempo indeterminado.