

**2022.1**  
**MATE85**

# **Tópicos em Sistemas de Informação e Web I**



**Frederico Durão**



# Apresentação

- Frederico Durão
  - [fdurao@ufba.br](mailto:fdurao@ufba.br)
- Semestre Letivo Online
  - Link: <https://conferenciaweb.rnp.br/webconf/mate85-topicos-em-sistemas-de-informacao-e-web-i>
  - <https://ufbaemmovimento.ufba.br>
  - Modo Híbrido: aulas síncronas e assíncronas
  - Todas as aulas serão gravadas e disponibilizadas

# Comunicação e Ferramentas para Aulas Online

- Lista de e-mails
  - [ufba\\_dcc\\_mate85\\_20221@googlegroups.com](mailto:ufba_dcc_mate85_20221@googlegroups.com)
- Conteúdo Programático
  - [Programa de Aulas](#) (planilha)
  - Slides e vídeo das aulas gravadas

# A disciplina

- 100% prática - **Fábrica de Desenvolvimento de Sistemas Web.**
- Não teremos aulas teóricas, mas conteúdo teórico será compartilhado na “estante virtual”.
- Cada equipe desenvolverá um Sistema Web já pré-definido.
- Todos terão a chance de aprender na prática.
- Ninguém é obrigado a ser experiente.
- Uma equipe se constrói com várias competências. É uma soma!
- No final, cada equipe entregará um sistema em um IP público.
- O sistema será de utilidade do Instituto de Computação ou UFBA.

# Avaliação

- Cada semana, as equipes farão entregas já predefinidas.
- Cada equipe fará avaliação individual de seus integrantes.
- Serão 2 notas, uma da primeira entrega e uma da segunda entrega.
- A equipe receberá uma nota, e os integrantes menos participativos terão nota decrescidas, se assim o forem.
- Sistemas que não estiverem implantados ou seja, acessíveis via um IP público terão metade da nota.
- O sistema poderá ser avaliado por usuários reais como professores, alunos ou técnicos.

# As aulas

- Encontros semanais online, todos devem estar presentes.
- Todas as aulas terão chamadas.
- Apenas **algumas** aulas serão gravadas.
- Na entrega final, a equipe fará um vídeo do que foi desenvolvido de até 3 minutos.
- Em todas as aulas as equipes terão que apresentar o que foi desenvolvido na semana anterior, seja código, melhorias no sistema, ou artefatos de software. A equipe deve controlar a produtividade de seus membros.

# Sobre os sistemas

- Foco não é o domínio do sistema, mas o desenvolvimento web.
- Escopo reduzido, requisitos não complexos, mas **úteis ao IC!**
- Requisitos definidos pelo docente.
- A equipe é responsável pelos requisitos, isso implica que vocês devem me consultar 24/7.
- Sistemas funcionais, 100% documentado, testados.
- Código versionado onde a equipe quiser.
- Equipe deve manter documentação para implantação e uso do sistema.

# Requisitos mínimos a todas as equipes

- Site do Sistema - Deve ser o mesmo do código (ex: github, gitlab).
- Documentação completa
  - requisitos, arquitetural, testes, implantação
- Todo o código deve ser documentado.
- Sistema deve estar hospedado e possuir IP público
  - ex. <https://www.heroku.com>
- Eu deverei ser user admin em todos os projetos.



# Gestão do Sistema

- Definição de software para gestão.
  - <https://clickup.com>
  - <https://about.gitlab.com>
  - Planilha Excel
- Cadastro de atividades a partir dos requisitos.
- Definição dos papéis, e gestão.
- Elaboração do cronograma de atividades.
- Associação com das atividades com desenvolvedores.

# Propriedade Intelectual

- Todo o código pertencerá ao IC de modo que outros alunos possam continuar os projetos.
- Todos os alunos podem seguir com o código para extensões de trabalhos próprios.
  - O aluno pode reutilizar, expandir, mas não cobrar propriedade.
- O cunho é único e exclusivamente educacional e de não deve ser utilizado para fins comerciais.

# A experiência

- As equipes (praticamente) não se conhecem.
- Não ensinaremos programação web.
- As aulas não serão teóricas.
- 30% dos alunos desistem antes da primeira avaliação.
- 40% dos alunos desistem antes da segunda avaliação.
- Eu não resolvo conflitos internos. Não me procurem por isso.
- As equipes devem me passar o desempenho de seus integrantes a cada semana.

# Documento Arquitetural

- Apresentar um diagrama com os módulos e interfaces do sistema.  
Essa figura deve apresentar de maneira clara cada componente que será desenvolvido.

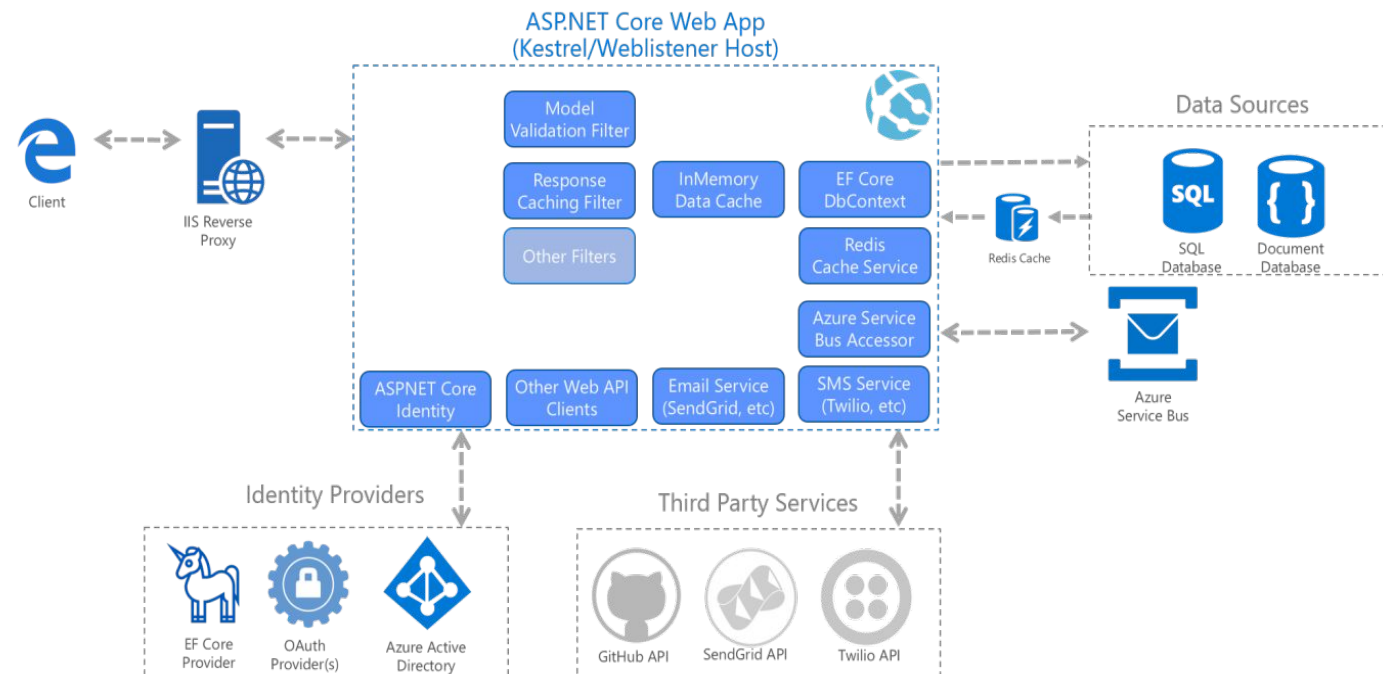
- Leitura nesse [link](#).

- [Exemplo 1](#)

- [Exemplo 2](#)

- [Exemplo 3](#)

## ASP.NET Core Architecture



# Documento de Implantação

- Sistema Operacional
- Requisitos mínimos de memória e processamento
- Softwares e versões das tecnologias macro
- Versão do banco de dados
- Cloud a ser utilizado, se aplicável
- Descrição de importação ou inicialização de dados, se aplicável.
- Endereço de Acesso (URL e Porta)
- Política de permissões, quando aplicável.

# Proposta de Testes

- [Como testar seu portal web?](#)
- Utilização alguma ferramenta ou ambiente de testes?
  - <https://jmeter.apache.org>
- Serão realizados testes unitários?
- Quem irá testar o sistema?
  - Usuário externos
  - Equipe
- Que casos de testes serão criados?
  - Documento de Caso de Teste ([Exemplo](#))

# Sugestão de Ferramentas de Gravação

- <https://atomisystems.com/download/>
  - Windows e Mac
- <https://launchpad.net/kazam>
  - Linux
- <https://zoom.us/pt-pt/meetings.html>
  - Web (independe de plataforma)

# Dicas do sucesso

- Sejam solidários e humanos, o momento exige.
- Qualquer problema, irei resolver.
- Ninguém será prejudicado por qualquer condição adversa.
- Tudo se resolve no diálogo, basta um e-mail.
- Estarei 100% disponível para ajudar.
- Microfones e vídeos desligados, no geral.
- Perguntas *preferencialmente* no chat.
- Se possível, usem fones de ouvido.



# Projeto 4 - Sistemas de Conteúdo Programático

- Login do usuário.
- Cadastro de conteúdos programáticos. Registrar nome de quem realizou a alteração e data.
- Listagem (menu lateral) e busca por código de disciplina.
- Manter histórico, log de alterações.
- Geração de [relatório](#) do conteúdo programático no formato específico.
- Link com as [ementas](#).