

# TESTE DE PERFORMANCE 09



**Nome:** Mauricio dos Santos Magnani Junior

**Disciplina:** Projeto de Bloco: Desenvolvimento .NET [22E1\_5]

**Professor(a):** Diego Tadeu Martins Acioly Ribeiro Dias

**Data:** 06/06/2022

[Introdução](#)

[Propósito do Documento](#)

[Escopo do Projeto](#)

[Requisitos de Usuário \( Awesome Tomatoes - Rede social para Cinéfilos \)](#)

[Comparativo Maturidade CMMI](#)

[Descrição do Sistema](#)

[Requisitos de Usuário e Sistema](#)

[Matriz de Rastreabilidade](#)

[Diagrama de Arquitetura em Camadas](#)

[Diagrama de Componentes](#)

[Impacto da Utilização do Modelo em Cascata](#)

[Código Fonte do Projeto](#)

[Referências](#)

## Introdução

### Propósito do Documento

Esse documento tem por objetivo a especificação de requisitos relacionados a rede social **Awesome Tomatoes**. Sendo esse documento se destina todos os profissionais envolvidos no desenvolvimento dessa solução.

Histórico de Alterações		
Nome	Data	Razão
Mauricio Magnani	14/03/2022	Criação do documento
Mauricio Magnani	06/06/2022	Atualização do conteúdo

### Escopo do Projeto

O sistema irá permitir aos usuários cadastrados na rede a possibilidade de criar reviews/avaliações de filmes e séries. Além disso será possível interagir com outros usuários e também ler/escrever artigos relacionados ao mundo do cinema.

### Requisitos de Usuário ( Awesome Tomatoes - Rede social para Cinéfilos )

**RF1:** Atualização de status/humor

**Ator:** Usuário

**Descrição:** Os usuários podem postar uma atualização textual de seu humor ou status atual relacionado de alguma forma com expressões utilizadas em filmes, séries, etc.

**RF2:** Privacidade da atualização de status/humor

**Ator:** Usuário

**Descrição:** Os usuários podem escolher se a postagem descrita no *RF1* será compartilhada com amigos, todos ou ninguém além de si mesmo.

**RF3:** Visualização de atualizações/conteúdo

**Ator:** Usuário

**Descrição:** Os usuários são apresentados a uma lista de postagens feitas por pessoas de quem são amigos ou postagens feitas para Todos (Verificar *RF2*).

**RF4:** - Filtro de atualizações/conteúdo

**Ator:** Usuário

**Descrição:** As postagens que aparecem nas listas do *RF3* são filtradas para conter apenas aquelas que o usuário achar relevante ou interessante.

**RF5** - Interação com as atualizações/conteúdo

**Ator:** Usuários

**Descrição:** Os usuários podem reagir a todos as postagens utilizando emoticons (definidos nos requisitos funcionais).

**RF6** - Postagem “review”/avaliação de filme

**Ator:** Usuário

**Descrição:** Os usuários podem realizar a postagem das suas “reviews”/avaliações sobre algum filme que tenham assistido.

**RF7:** Privacidade “review”/avaliação de filme

**Ator:** Usuário

**Descrição:** Os usuários podem escolher se as “reviews”/avaliações descrita no *RF6* será compartilhada com amigos, todos ou ninguém além de si mesmo.

**RF8:** Interação com outro usuários (chat)

**Ator:** Usuário

**Descrição:** Os usuários podem entrar em contato com outras pessoas através de um sistema de comunicação disponível na plataforma.

**RF9:** Filtro do sistema de comunicação (chat)

**Ator:** Usuário

**Descrição:** Os usuários podem criar uma lista de pessoas bloqueadas que serão impedidas de entrar em contato através do sistema de comunicação disponível na plataforma (Verificar *RF8* )

**RF10:** Publicação de Artigos

**Ator:** Usuário

**Descrição:** Os usuários podem criar artigos relacionados a curiosidades do cinema, filme, atores e assuntos afins. Esses artigos serão fixados no perfil de cada usuário.

## Comparativo Maturidade CMMI

O CMMI apresenta um modelo (níveis) de maturidade que foi adotado por diversas empresas na avaliação da maturidade dos seus processos internos.

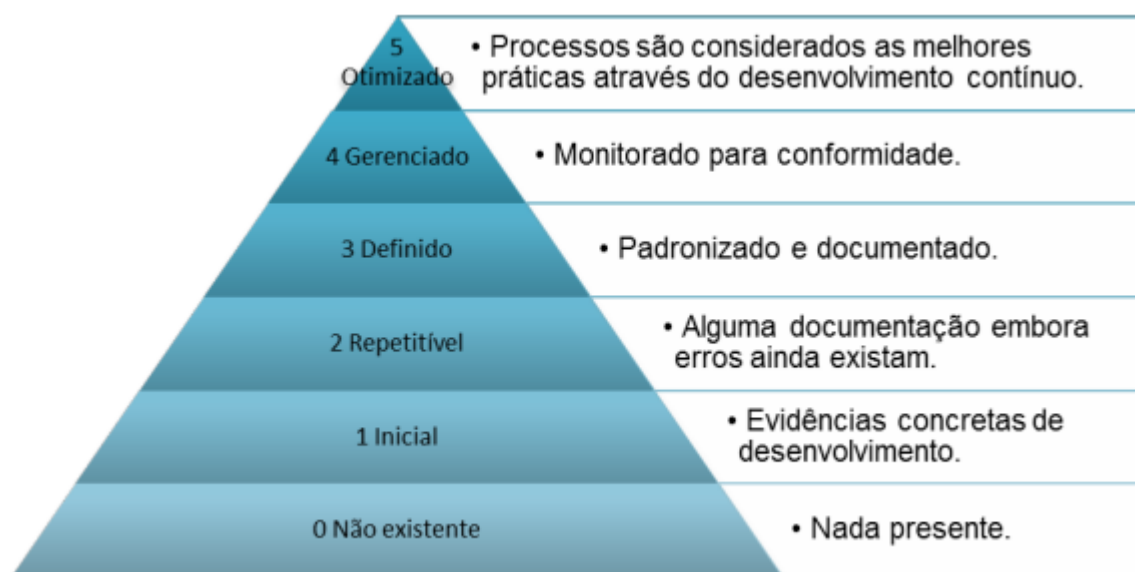
**Nível 1 - Inicial:** O processo de software é caracterizado como “ad hoc” e ocasionalmente pode ser caótico. Poucos processos estão definidos e o sucesso depende de esforços individuais.

**Nível 2 - Repetível:** Os processos básicos de gerenciamento estão estabelecidos para controlar custo, cronograma e funcionalidade. A disciplina necessária dos processos permite repetir o sucesso em outros projetos com aplicações similares.

**Nível 3 - Definido:** O processo de software para as atividades de gerenciamento e de engenharia é documentado, padronizado e integrado em um processo padrão de software para organização.

**Nível 4 - Gerenciado:** Medições detalhadas do processo de software e da qualidade do produto são coletadas. Tanto o processo de software quanto o produto de software são quantitativamente entendidos e controlados.

**Nível 5 - Otimizado:** A melhoria continua do processo é feita através do “feedback” quantitativo dos processos e das aplicações de novas idéias e tecnologias.



Os níveis de maturidade 1 e 2 são facilmente atingíveis. Por outro lado, atingir os níveis 3, 4 e 5 demandam forte comprometimento dos executivos, uma abordagem detalhada e sistematizada de implantação e uma gestão adequada das mudanças organizacionais.

## Descrição do Sistema

Awesome Tomatoes é uma rede social para cinéfilos que foi inspirada no Rotten Tomatoes, um website americano, agregador de críticas de cinema e televisão. O objetivo do Awesome Tomatoes é se tornar a rede social preferida dos cinéfilos, no qual eles poderão criar reviews de filmes e séries, conversar com outros apaixonados por cinema e também ficar por dentro das ultimas novidades no mundo da sétima arte.

## Requisitos de Usuário e Sistema

**RF1:** O usuário deve ser capaz de realizar uma atualização de status/humor

**Atores:**

- Usuário
- Sistema

**Pré-condições:**

- O usuário deve estar cadastrado na rede social Awesome Tomatoes
- O usuário deve estar com uma sessão ativa (logado).

**Fluxo básico:**

1. O usuário acessa a página principal.
2. O usuário clica em “Iniciar uma nova publicação”
3. O sistema abre uma caixa de texto.
4. O usuário digita a sua atualização de status/humor.
5. O usuário clica em publicar atualização de status/humor.
6. O sistema exibe a publicação para todos os seguidores desse usuário.

**Fluxo Alternativo:**

5a. O usuário clica um ícone “X” para cancelar descartar a atualização de status/humor antes de ser publicada.

**Regras de Validação:**

- Nenhuma

**Pós-Condições:**

- Nenhuma

**RF2:** O usuário deve ser capaz de configurar a privacidade de uma atualização de status/humor

**Atores:**

- Usuário
- Sistema

**Pré-condições:**

- O usuário deve estar cadastrado na rede social Awesome Tomatoes
- O usuário deve estar com uma sessão ativa (logado).
- Deve existir uma publicação ( Verificar *RF1* )

**Fluxo básico:**

1. O usuário acessa a página de privacidade e publicações.
2. O usuário clica em gerenciar privacidade
3. O sistema exibe uma listagem relacionada a privacidade da publicação.
4. O usuário seleciona um grupo/item da lista para tornar a publicação visível somente para aquele grupo/item.
5. O sistema atualiza a privacidade da publicação.
6. O sistema exibe a publicação somente para o grupo/item escolhido.

**Fluxo Alternativo:**

- 3a. O usuário clica “Cancelar” para descartar a alteração de privacidade.

**Regras de Validação:**

- Nenhuma

**Pós-Condições:**

- Nenhuma

**RF3:** O usuário deve ser capaz de interagir com outros usuários (chat)

**Atores:**

- Usuário
- Sistema

**Pré-condições:**

- O usuário deve estar cadastrado na rede social Awesome Tomatoes
- O usuário deve estar com uma sessão ativa (logado).
- Outro usuário deve estar cadastrado na rede social Awesome Tomatoes
- Outro usuário deve estar com uma sessão ativa (logado).

**Fluxo básico:**

1. O usuário acessa o chat
2. O usuário busca pelo amigo com o qual deseja de comunicar.
3. O usuário digita um texto para iniciar a comunicação.

**Fluxo Alternativo:**

- 2a. O usuário não encontrar nenhum amigo “online”.

**Regras de Validação:**

- Nenhuma

**Pós-Condições:**

- Nenhuma

**RF4:** O usuário deve ser capaz de configurar filtros para o sistema de comunicação (chat)

**Atores:**

- Usuário
- Sistema

**Pré-condições:**

- O usuário deve estar cadastrado na rede social Awesome Tomatoes
- O usuário deve estar com uma sessão ativa (logado).
- RF3

**Fluxo básico:**

1. O usuário acessa a página de privacidade e chat.
2. O usuário clica em gerenciar privacidade
3. O sistema exibe uma listagem relacionada a privacidade do chat.
4. O usuário seleciona um grupo/item da lista para que seja bloqueado no chat.
5. O sistema atualiza a privacidade do chat.

**Fluxo Alternativo:**

- 3a. O usuário clica “Cancelar” para descartar a alteração de privacidade.

**Regras de Validação:**

- Nenhuma

**Pós-Condições:**

- Nenhuma

**RF5:** O usuário deve ser capaz de publicar novos artigos relacionados a cinema

**Atores:**

- Usuário
- Sistema

**Pré-condições:**

- O usuário deve estar cadastrado na rede social Awesome Tomatoes
- O usuário deve estar com uma sessão ativa (logado).

**Fluxo básico:**

1. O usuário acessa a página de artigos.



2. O usuário clica em “Escrever um novo artigo”
3. O sistema abre uma caixa de texto.
4. O usuário digita o artigo.
5. O usuário clica em publicar artigo.
6. O sistema exibe o artigo para todos os usuários do Awesome Tomatoes.

**Fluxo Alternativo:**

- 5a. O usuário clica um ícone “X” para cancelar descartar a atualização

**Regras de Validação:**

- Nenhuma

**Pós-Condições:**

- Nenhuma

## Matriz de Rastreabilidade

	RF01	RF02	RF03	RF04
RF01		X		
RF02				
RF03				
RF04			X	

## Diagrama de Arquitetura em Camadas

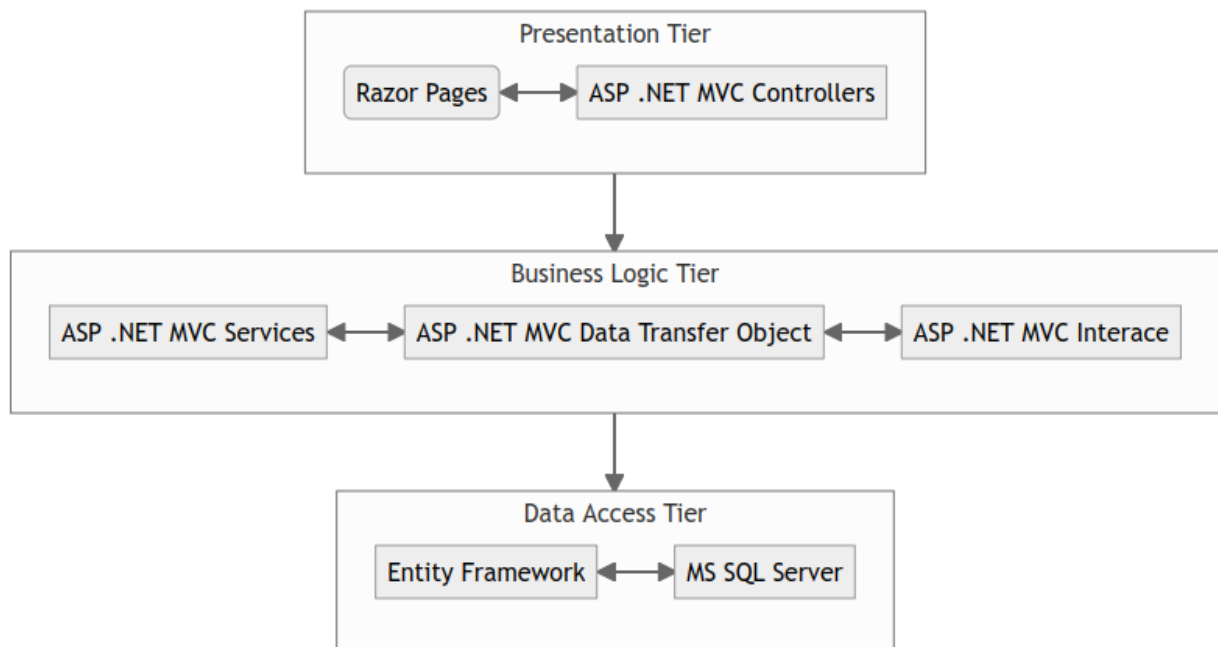
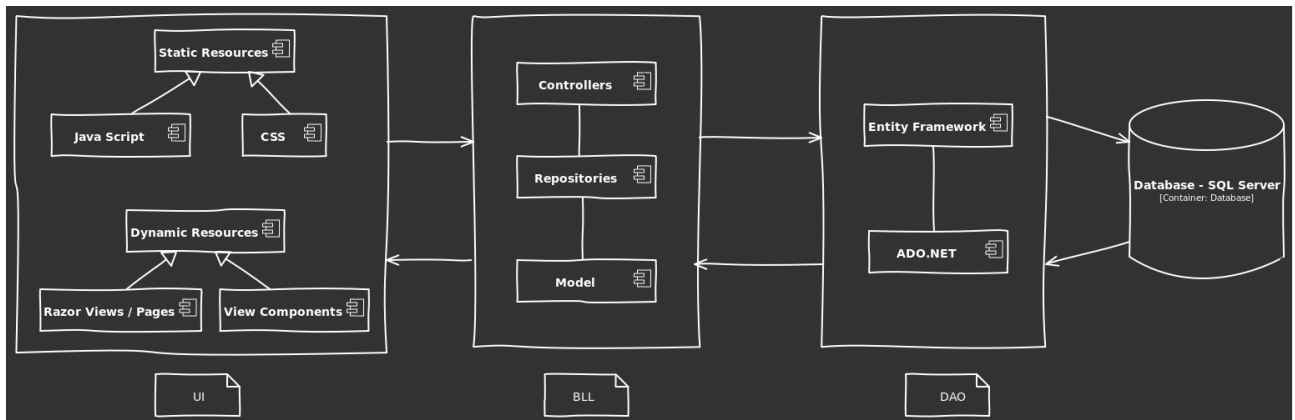
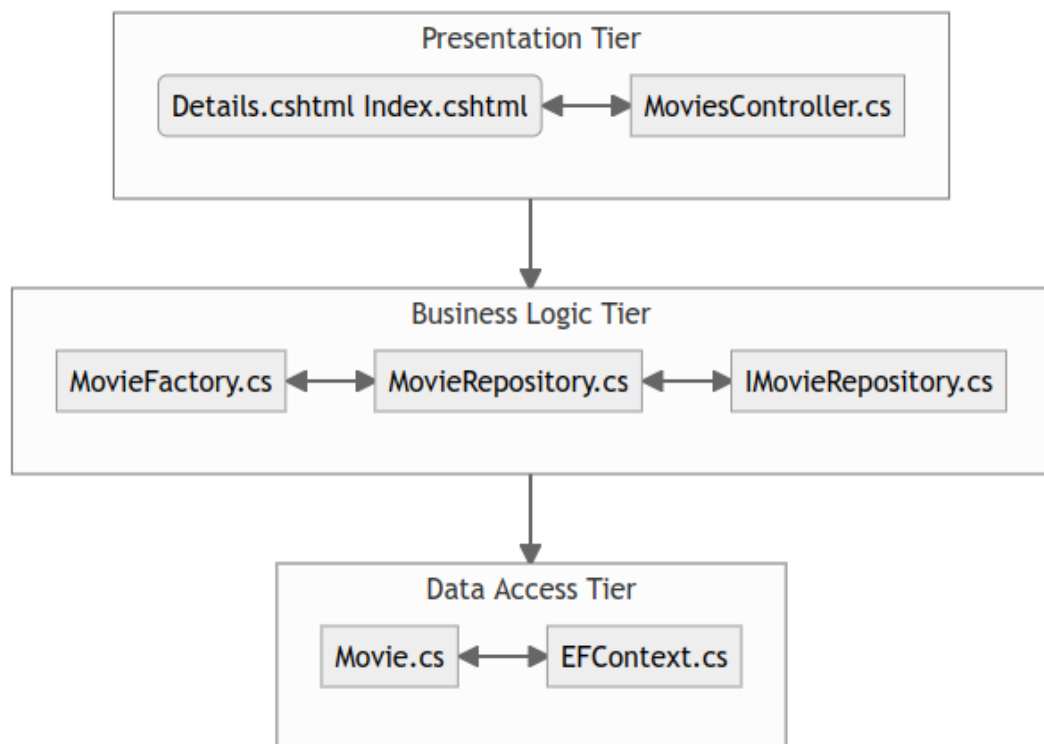


Diagrama de Componentes



Código desses diagramas no github:

<https://raw.githubusercontent.com/msmagnanijr/edc-dotnet-csharp/main/project-de-bloco/TP05/README.md?token=GHSAT0AAAAAABNEJ6DF2A3XKAOL5GW542YAYSXNO4A>

## Impacto da Utilização do Modelo em Cascata

Durante o desenvolvimento desse projeto foi utilizado o modelo em cascata no qual as funcionalidades são desenvolvidas em sequência. A utilização desse modelo foi um pouco prejudicial ao projeto pois muitos requisitos foram descobertos durante o desenvolvimento das funcionalidades, o que ocasionou o retrabalho em algumas ocasiões.

## Código Fonte do Projeto

Ver anexo **Mauricio\_Magnani\_TP09.zip**

## Referências

Sommerville, I.(2019) “Engenharia de Software”, 10.ed. São Paulo: Pearson