

# Cinemis: projeto de sistema web para facilitar o cadastro de filmes no Museu da Imagem e do Som de Campinas

Patrícia Linhares Areias<sup>1</sup>, Fernando Sambinelli<sup>1</sup>

<sup>1</sup>Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas Instituto Federal de São Paulo – (IFSP) - Campus Hortolândia – SP - Brasil

a.patricia@aluno.ifsp.edu.br, sambinelli@ifsp.edu.br

**Abstract.** *For more than 30 years, the Museu da Imagem e do Som de Campinas has had an audio-visual trusteeship program that is self-managed by the public that frequents the space. The movie sessions are voluntarily managed and publicized. The analysis of this process of managing and disseminating the sessions verified that it generates costly voluntary work as the information is manually collected, and there is no optimization between data collection, its chronological ordering, and its dissemination. Considering this context, this work presents a Web Application proposal to start the process of optimizing and automating the entire flow of work at the museum, aiming to guarantee the necessary requirements for data collection and management of debate and film sessions in a satisfactory manner.*

**Resumo.** *Há mais de 30 anos o Museu da Imagem e do Som de Campinas possui uma programação de áudio visual auto gerenciada pelo público frequentador do espaço. As exibições são voluntariamente gerenciadas e divulgadas. Na análise deste processo de gerenciar e divulgar as sessões verificou-se que ele gera um trabalho voluntário dispendioso na medida em que as informações são manualmente coletadas, e não há otimização entre a coleta dos dados, sua ordenação cronológica, e sua divulgação. Considerando esse contexto, este trabalho apresenta uma proposta de Aplicação Web para dar início ao processo de otimização e automatização de todo o fluxo de trabalho no museu, visando garantir os requisitos necessários para coleta de dados e gestão das sessões de debates e filmes de maneira satisfatória.*

## 1. Introdução

Há mais de 30 anos o Museu da Imagem e do Som de Campinas promove, gratuitamente, exibições e debates de filmes. A programação do Circuito MIS de Cinema é feita com a participação do público, e busca trazer aos espectadores diferentes linguagens e olhares cinematográficos. As exibições dos filmes são seguidas de debates preservando assim, o espírito cineclubista [SANTAROSA 2015].

De acordo com Marcela Servano (2023): “Os cineclubes são espaços democráticos, educativos, políticos, sem fins lucrativos que contribuem na formação de público, porque não só estimulam as pessoas a assistirem a obras audiovisuais, como também promovem rodas de discussões”. Ela ainda cita que os principais objetivos dos cineclubes são:

(...)refletir sobre a linguagem e do cinema, possibilitar a experiência fílmica como ferramenta de educação, estimular o desenvolvimento do pensamento crítico e viabilizar ações concretas de intercâmbio entre cineclubistas, realizadores, pesquisadores, críticos e pessoas enxergam o cinema como uma arte transformadora.

As curadoras e curadores dos filmes a serem exibidos no MIS Campinas exercem o trabalho de curadoria de forma voluntária. Da mesma forma (voluntária) é feito o trabalho de administração e divulgação da agenda mensal de exibição dos filmes.

Todo o trabalho de coleta dos dados dos filmes que serão exibidos por cada curadoria e de divulgação das exibições dos filmes é feito de forma manual, sem o auxílio de tecnologia para agrupar as informações de forma otimizada, e sem que haja uma conexão automática entre as etapas do fluxo de trabalho.

A utilização de sistemas integrados para padronizar rotinas de trabalho facilita operações e otimiza processos de forma geral. De acordo com Rafael Zambelli (2022) “A automatização de processos significa operacionalizar atividades por meio da tecnologia. É uma ação diretamente relacionada a ganhos de produtividade e eficiência.”

O objetivo deste trabalho é dar início ao processo de automatização do fluxo, que hoje é inteiramente manual, através da proposta do desenvolvimento da aplicação *web Cinemis*.

O artigo está estruturado da seguinte maneira: a Seção 2 apresenta o referencial teórico; na Seção 3 são descritos os trabalhos correlatos do projeto; a Seção 4 descreve as metodologias utilizadas na construção do projeto; a Seção 5 discorre sobre a fase de desenvolvimento, e a Seção 6 mostra a conclusão.

## **2. Referencial Teórico**

Nesta seção são abordados os principais conceitos e aspectos técnicos quanto ao desenvolvimento deste trabalho:

### **2.1. Aplicação Web**

Algumas décadas atrás, a engenharia de *software* era voltada à criação de aplicações para funcionar em computadores. Então, estudavam-se formas de interação da aplicação com os recursos de *hardware* e dos sistemas operacionais das máquinas. Além disso, havia também a preocupação em manter os sistemas atualizados e seguros [MARTINS 2023].

Por ser uma aplicação de *software* que funciona com base na infraestrutura da internet, em vez de funcionar com base em sistemas operacionais, a aplicação *web* muda completamente a forma de pensar a relação com as tecnologias, o que a torna uma alternativa fácil de acessar, sem demandar manutenção por parte do usuário (como manutenções de segurança e de atualizações, por exemplo).

A decisão pelo desenvolvimento da aplicação *Cinemis* como uma aplicação do tipo *web* baseou-se no fato de que uma aplicação feita para a internet está livre da necessidade de ser compatível com elementos específicos de um computador ou dispositivo, por ela operar de forma descentralizada e móvel.

Considerando a diversidade entre curadores(as) em relação ao tipo de dispositivo tecnológico usado, optou-se pela aplicação *web* justamente por ser menos restritiva quando comparada a aplicações cujo uso está condicionado a uma instalação prévia. Lembrando que aplicações que precisam ser instaladas para serem usadas condicionam seu uso à disponibilidade de memória do dispositivo e aos requisitos dos sistemas operacionais (tipo e versão, por exemplo).

## 2.2. Arquitetura Modelo-Visão-Controle (Model View Controller)

Os padrões de arquitetura foram criados como uma maneira de apresentar, compartilhar e reutilizar conhecimento sobre sistemas. Eles descrevem a organização de um sistema que foi bem sucedida em sistemas anteriores [SOMMERVILLE 2018].

Segundo Pressman (2011): “A arquitetura de *software* de um programa ou sistema computacional é a estrutura ou estruturas do sistema, que abrange os componentes de *software*, as propriedades externamente visíveis desses componentes e as relações entre eles” (p.230).

No início do desenvolvimento de *software*, um *software* era desenvolvido para rodar em uma única máquina, os aplicativos possuíam apenas uma camada, e ela suportava códigos fonte, os eventos dos usuários, a lógica de negócios e os acessos aos dados, o que dificultava a programação e a manutenção deste *software*. Estas aplicações receberam o nome de aplicação monolítica [BAPTISTELLA 2009].

Com a necessidade de se compartilhar a lógica de acesso a dados entre vários usuários simultaneamente, surgiu a necessidade de criar outra camada para o acesso aos dados, estas aplicações passaram a ser chamadas de aplicações em duas camadas ou aplicações cliente-servidor, onde a camada de acesso aos dados ficava em uma máquina específica e o *software* era instalado do lado cliente, contendo a lógica de negócio e a interface com o usuário.

A arquitetura de *software* do tipo cliente-servidor é uma arquitetura de aplicação distribuída, que divide a lógica de uma aplicação em dois ou mais componentes: o cliente e o servidor. Em uma rede, programas são distribuídos em diferentes locais: os fornecedores de recursos e que são responsáveis pela manutenção da informação (servidores), e os requerentes dos recursos e que obtêm os dados ou serviços (clientes). O cliente não compartilha nenhum de seus recursos com o servidor, no entanto ele solicita alguma função do servidor, sendo ele, o cliente, responsável por iniciar a comunicação com o servidor [LEMOS *et al.* 2013].

A arquitetura cliente-servidor é muito utilizada na computação e permite que os usuários acessem serviços e informações de qualquer lugar, pois a interação entre o cliente e o servidor é baseada em requisições HTTP, que são compostas por um método, um caminho (URI) e uma versão do protocolo.

O problema da arquitetura em duas camadas era que aplicativos dos usuários precisavam ser atualizados quando alterações eram feitas. Para resolver esse problema, criou-se uma terceira camada que passou a abrigar a lógica de negócio separadamente da camada de interface com o cliente e da camada de dados. Uma camada intermediária é criada entre o cliente e o servidor, que armazena as regras do negócio e a lógica da aplicação. O cliente fica responsável apenas pela interface com o usuário [LEMOS *et al.* 2013].

O padrão arquitetural Model-View-Controller (MVC) utiliza-se de três camadas: a modelo (*model*), a de visualização (*view*) e o controlador (*controller*).

O componente *Model* é o objeto de aplicação, contém a comunicação com os dados armazenados que serão visualizados na *View*, podendo estar armazenado em um banco de dados, em um arquivo XML ou em qualquer meio para tal funcionalidade. É somente na *Model* que as operações de *create*, *read*, *update* e *delete* (CRUD), operações básicas em um banco de dados (que cria, lê, atualiza e deleta dados), podem ocorrer.

A *View* é a interface visualizada pelo usuário, não importando quais dados e de qual lugar

tenham vindo, mas, sim, de como serão exibidas essas informações.

A *Controller* é responsável por administrar todo o fluxo da aplicação, tem a lógica para trabalhar com os dados de entrada da *View* e resolver qual operação utilizará da camada *Model* [LEMOS *et al.* 2013].

Abaixo, na Figura 1, uma representação gráfica descritiva de uma arquitetura de padrão web usando o padrão MVC:



**Figura 1. Arquitetura de padrão web usando o padrão MVC. Fonte: Sommerville (2018, p.156).**

O padrão de *software* MVC é então utilizado quando há várias maneiras de visualizar e interagir com os dados, e também quando os requisitos futuros para interação e apresentação dos dados são desconhecidos [SOMMERVILLE 2018].

### 2.3. Sistema de Banco de Dados Relacional

Um banco de dados é uma coleção logicamente coerente de dados, com um significado inerente, que representa algum aspecto do mundo real. Isso significa que ele é projetado, construído e populado com dados para uma finalidade específica, tendo um grupo definido de usuários e algumas aplicações de interesse desses usuários, previamente definidas, de modo que as alterações dos eventos do mundo real que ele representa sejam refletidas no banco de dados. Além disso, é necessário que a organização e gerenciamento das informações seja feita de modo que os usuários possam consultar, recuperar e atualizar os dados quando necessário [ELMASRI e NAVATHE 2010].

Nesse sentido, outra importante decisão arquitetural a ser tomada em um projeto de desenvolvimento de *software* é a definição do tipo de estrutura de armazenamento de dados que será utilizada. Atualmente existem dois paradigmas bem distintos de gerenciamento e armazenamento de dados: Banco de Dados do tipo Relacional (SQL), e Banco de Dados do tipo Não Relacional (NoSQL).

Considerando que o Banco de Dados armazena um conjunto de dados que possuem uma relação entre si, o Sistema de Gerenciamento de Banco de Dados (SGBD) é um sistema de *software* de uso geral que facilita a definição, construção, manipulação e compartilhamento de banco de dados entre diversos usuários e aplicações. Em outras palavras, é um sistema computadorizado de manutenção e gerenciamento desses registros. Essa manutenção e gerenciamento terá características diferentes de acordo com o paradigma escolhido.

Os SGBDs devem possuir características para evitar problemas tais como redundância e inconsistência dos dados, acesso indevido e problemas de atomicidade. Dentre as principais características de um SGBD estão o controle de concorrência, segurança, recuperação de falhas,

gerenciamento dos mecanismos de armazenamento de dados e controle das restrições de integridade do Banco de Dados. Além disso, eles são responsáveis pelo gerenciamento de transações, ou seja, pelo conjunto de operações de leitura ou escrita que são realizadas no banco de dados [LÓSCIO et al. 2011].

A execução dessas transações em um SGBD deve obedecer a algumas propriedades a fim de garantir o correto funcionamento do sistema e a respectiva consistência dos dados. Essas propriedades são: atomicidade (todas as operações da transação são executadas, ou nenhuma será), consistência (após a transação o banco de dados permanece consistente satisfazendo as restrições de integridade previamente assumidas), isolamento (duas transações que estão sendo executadas concorrentemente têm seus efeitos isolados um do outro) e durabilidade (o efeito de uma transação não poderá mais ser desfeito, mesmo em caso de falha).

O modelo de banco de dados do tipo relacional representa o banco de dados como uma coleção de relações, em que cada relação é uma tabela de valores com registros do tipo plano por terem uma estrutura linear. Cada linha da tabela representa um fato correspondente a uma entidade ou relacionamento do mundo real, e a tabela é então uma coleção de valores de dados relacionados [ELMASRI e NAVATHE 2010]. Se uma tabela chama 'Filme', por exemplo, cada linha dessa tabela representa fatos sobre uma entidade particular de filme (nome do filme, direção, ano de lançamento, etc). Conceitualmente diz-se que um banco de dados do tipo relacional traduz um modelo Entidade-Relacionamento (ER) pois, descreve os dados como entidades, relacionamentos e atributos.

O modelo Relacional é muito bem aceito devido a sua padronização de conceitos, sua base formal, e a facilidade de uso da linguagem SQL (*Structured Query Language*) - linguagem padrão para consultas e manipulação de dados relacionais.

O modelo relacional apresenta uma grande organização dos dados. A relação entre tabelas torna a estrutura muito prevenida contra possíveis falhas como a duplicidade de informações (por bloquear a redundância de dados nas inserções de registros), por exemplo, assim como também pode-se ter a certeza de que determinada tabela não ficará vazia, por meio de configurações.

Detalhando um pouco mais as restrições existentes numa estrutura de banco de dados relacional, ela permite dois tipos de restrições sobre os dados para além da aplicação das restrições provenientes dos programas de aplicação: as restrições que são inerentes a este modelo (relacional), e as que são expressas diretamente nas especificações da linguagem de definição de dados (DDL) – a DDL mais utilizada no banco de dados relacional é a linguagem SQL [ELMASRI e NAVATHE 2010].

O fato dos registros dos valores ser do tipo plano, ou seja, serem representados por uma linha da tabela, faz com que as relações entre os dados possam representar não só fatos sobre entidades, mas também fatos sobre relacionamentos, o que possibilita determinar quais as combinações de valores possíveis, de modo que outros tipos de combinações de valores sejam falsas, e portanto não aplicáveis a uma transação dentro do banco de dados modelado [ELMASRI e NAVATHE 2010].

A aplicação das restrições definidas pela linguagem de definição de dados (DDL) do modelo relacional são restrições de esquema e incluem restrições de domínio (tipos de dados possíveis para cada atributo), de chave, de NULLs (qual atributo pode ter valor vazio), de

integridade de entidade (nenhuma chave primária pode ser NULL) e de integridade referencial (as relações entre as linhas das tabelas devem ser válidas - o que é garantido pelas restrições das chaves estrangeiras, que garante que uma chave estrangeira de uma relação coincida com uma chave primária da tabela “pai”, impedindo que haja valores duplicados em colunas específicas e limitando os valores aceitos por uma ou mais colunas) [ELMASRI e NAVATHE 2010].

De acordo com Jugurta Lisboa Filho (2001): "Modelagem de dados é o processo de abstração onde somente os elementos essenciais da realidade observada são enfatizados, descartando-se os elementos não essenciais" (p. 13).

Na seção 5.6 é apresentada a modelagem do banco de dados a ser utilizada pela aplicação para a manipulação de dados pelo back-end do sistema.

## 2.4. Desenvolvimento Web Responsivo

Com a evolução da tecnologia, o uso dos dispositivos tecnológicos (principalmente os móveis) está cada vez mais presente no dia a dia das pessoas, e diferentes tipos de usuários passam a ter acesso cada vez maior a cada vez mais dispositivos eletrônicos, que possuem uma ampla diversidade de características de interfaces.

Durante o processo de desenvolvimento de *software* é necessário definir qual ou quais plataformas são mais apropriadas para garantir sua utilização, considerando principalmente que o *software* desenvolvido será incompatível com plataformas diferentes daquela para o qual foi projetado.

Uma aplicação *web* funciona em qualquer dispositivo que possua um navegador compatível, pois é uma solução desenvolvida para ser acessada através de um navegador da *web*, sem a necessidade de instalação da aplicação no dispositivo do usuário. Ela é desenvolvida usando tecnologias *web* como HTML, CSS e JavaScript e pode ser projetada para fornecer uma experiência de usuário rica e interativa, semelhante à de um aplicativo de *desktop* ou *mobile*. [HERMANO 2024].

Porém, para garantir que os usuários tenham uma experiência consistente, independentemente do dispositivo que estão usando, é necessário fazer ajustes no código CSS de customização (referente à parte relacionada ao *layout* da aplicação), para que o conteúdo da aplicação se adapte de acordo com o tamanho da tela dos dispositivos em que ela será acessada.

*Design* responsivo é uma técnica de estruturação HTML e CSS, em que a página *web* se adapta à resolução do navegador sem precisar definir diversas folhas de estilos (CSS) para cada resolução.

Assim, através de um mesmo código fonte, é possível originar aplicações para diferentes plataformas a partir da construção do *layout* da interface utilizando-se de *fluid grids* - que possibilitam a alteração dinâmica da localização dos elementos da página de acordo com o tamanho da tela -, e das *media queries* - que altera o tamanho dos componentes da página de acordo com o tamanho da tela. [SANTOS 2022].

## 2.5. Design de Interfaces e Prototipagem

Ao desenvolver um *software* é importante considerar que a interface da aplicação com a qual o usuário tem contato transmite a ele informações, por isso a maneira como dados e informações são apresentados tem um papel importante.

De acordo com Valquiria Santos Segurado (2016):

O design de sistemas interativos não é apenas uma questão de design de interfaces. A interação humano-computador como um todo tem de ser levada em consideração, bem como a interação humano-humano que é frequentemente propiciada por meio dos sistemas. (...) O design de sistemas interativos se preocupa, em última instância, com a criação de experiências interativas para as pessoas. Ser centrado no humano é colocar as pessoas em primeiro lugar; é projetar sistemas interativos que favoreçam as pessoas e dos quais elas possam usufruir (p. 6).

Existe inclusive uma área de estudo voltada exclusivamente para isso chamada *design* de informação. O objetivo do *design* de informação é assegurar a efetividade da comunicação a partir da facilitação de processos de percepção, leitura, compreensão, memorização e uso da informação apresentada [QUINTÃO e TRISKA 2014].

A utilização de elementos visuais reduz o tempo que um usuário leva para entender e responder a uma informação. O *design* de informação pode exercer um papel social fundamental, no sentido de fazer com que tais informações sejam disponibilizadas de maneira clara e objetiva para aqueles que, seja por necessidade ou desejo, têm contato com elas [QUINTÃO e TRISKA, 2014].

O trabalho do *designer* de informação se inicia com a estruturação dos conjuntos de dados. O *design* de informação examina, por exemplo, como os conteúdos são visualizados por meio de seleção, ordenamento, hierarquização, conexões e distinções visuais.

O *International Institute for Information Design – IIID*, fundado na Áustria em 1986, caracteriza o design de informação como “a definição, planejamento e modelagem dos conteúdos de uma mensagem e do ambiente em que ela é apresentada, com a intenção de satisfazer às necessidades de informação dos destinatários” [QUINTÃO e TRISKA 2014].

O *design* produz interfaces gráfico-digitais voltadas para a interação do usuário com o produto.

A Interface *Web* é composta por um conjunto de elementos que tornam possível o usuário ver, ouvir e interagir com as informações. Ela tem a função de “permitir ao usuário obter uma visão panorâmica do conteúdo, navegar na massa de dados sem perder a orientação e, por fim, mover-se no espaço informacional de acordo com seus interesses”. (BATISTA, p.45, 2008)

A técnica da prototipagem tem por objetivo identificar qual a melhor alternativa de interface para o usuário. Sua finalidade é avaliar os requisitos. Por isso é importante construir uma prova de conceito, um protótipo da aplicação antes de iniciar o seu desenvolvimento [CENTENARO 2014].

Segundo David Banyon (2011): “Um protótipo é uma representação ou implementação concreta, porém parcial, do design de um sistema (...) Um protótipo pode ter a intenção de destacar apenas a interface, ou algum aspecto crucial da funcionalidade” (p. 121).

Com a prototipação, os clientes e outros interessados no projeto, podem analisar como todas as funcionalidades estão distribuídas, bem como se a organização do *layout* está correta, se o sistema oferece uma boa experiência para o usuário, permitindo que validações sejam feitas antes da implementação [DIAS 2019].

Existem dois tipos principais de prototipação: baixa fidelidade (*lo-fi*) e alta fidelidade (*hi-fi*).

O protótipo de alta fidelidade, por ser desenvolvido quando o projeto já está adiantado, com ideias firmes sobre a solução a ser desenvolvida, é útil para avaliações detalhadas dos principais elementos de *design*, e pode ser utilizado em estudo de usabilidade. Normalmente serve como documento final de *design* antes da implementação da aplicação.

Os protótipos hi-fi são semelhantes ao produto final esperado, em termos de aparência e sensação, embora não necessariamente, em termos de funcionalidade. Eles são produzidos em software, seja em um ambiente de desenvolvimento que será usado para a implementação ou em pacotes que permitirão que efeitos interativos sejam facilmente simulados (BANYON, p. 121-122, 2011).

O protótipo de baixa fidelidade captura ideias iniciais do *design* (ideias amplas e fundamentais do *design*, como conteúdo, forma e estrutura), sendo rapidamente produzidos e por isso facilmente descartados. A forma mais comum desse tipo de protótipo é uma série de ‘capturas de tela’. O objetivo é o baixo custo da construção do protótipo [BANYON 2011].

Neste trabalho utilizaremos a prototipação de baixa fidelidade, do tipo descartável.

## 2.6. Frameworks

Um *framework* de desenvolvimento é um conjunto de códigos que facilita o desenvolvimento de *sites*, aplicações e serviços *web*. Os *frameworks* de desenvolvimento *web* podem otimizar o desenvolvimento de *software* através da aplicação de estruturas, comandos e funcionalidades prontas, da reutilização de código, e da utilização de uma abordagem genérica para resolver problemas recorrentes [GUIMARÃES 2019].

*Frameworks* são, em outras palavras, estruturas de código que ajudam a organizar e padronizar o desenvolvimento de aplicações ao oferecerem soluções para tarefas comuns como criação de interfaces de usuário dinâmicas, integração com APIs, roteamento de páginas, manipulação de estado da aplicação, manipulação e interação com o banco de dados.

O desenvolvimento é mais eficiente uma vez que, ao seguir os padrões estabelecidos, economiza-se tempo por não ser necessário escrever tudo "do zero".

Os *frameworks* podem ser divididos em dois grupos: os de *back-end* e os de *front-end*.

### 2.6.1. Framework Back-End Express (Node.js)

No *back-end*, *frameworks* contribuem com a construção da lógica presente no servidor (criação do servidor, criação das rotas de API), gerenciamento do banco de dados, processamento de requisições e gerenciamento de erros.

*Express* é um *framework* que se utiliza de *Node.js* (que utiliza de bibliotecas responsáveis pelo tempo de execução) para executar a linguagem de programação JavaScript fora do navegador, que permite que os desenvolvedores criem um ambiente de *back-end* com menos código e fornecendo um grande número de recursos, como roteamento, *middleware*, motores de



modelo (*template engines* - que facilitam a criação de páginas HTML por permitir o uso de recursos das linguagens de programação em páginas HTML) e tratamento de erros [PESSÔA 2022].

A principal característica do *Node.js* é ser um ambiente de execução assíncrono, isto é, cada requisição terá uma instância de um único processo - delegando os processos demorados a um segundo plano -, ao invés de enfileirar as requisições e depois as processar, uma a uma, não permitindo múltiplos processamentos [PEREIRA 2014].

Por fim, é importante dizer que o *Node.js* possui uma estrutura orientada a eventos e uma forma de entrada e saída de dados (I/O - *input/output*), que faz com que ele seja leve e eficiente pois, ao executar o Javascript usando o mecanismo V8 desenvolvido pela Google do lado do servidor (ao invés de dentro do navegador), ele proporciona um ambiente de tempo de execução muito rápido do lado do servidor. Essas características são essenciais para um melhor desempenho quando diante de um intenso tráfego de rede e para aplicações em tempo real [IMPERES FILHO 2023].

### 2.6.2. Framework Front-End Bootstrap

Não é simples desenvolver ou adaptar um *site* para ser responsivo, pois trabalhar com *layouts* fluidos requer cálculos e análises para os mais variados tipos e tamanhos de telas.

O *Bootstrap* disponibiliza uma coleção de vários elementos visuais (botões, grades, títulos, textos, imagens) de uma página e seus efeitos de transição e estado (como o posicionar o *mouse* sobre um botão, o carregamento de uma página com efeito de barra de progresso), e funções personalizáveis para projetos da *web*, empacotados previamente em uma única ferramenta. Esses elementos personalizáveis contidos no *Bootstrap* são uma combinação de HTML5, CSS3 e JavaScript.

Ao projetar um *site* com o *Bootstrap*, os desenvolvedores podem escolher quais elementos querem usar de modo que os elementos escolhidos não conflitarão entre si e serão totalmente responsivos funcionando tanto para *desktop* como quaisquer outros tamanhos de tela. Como toda a estrutura do CSS já vem definida, basta procurar o componente necessário e adicionar ao código [TOMAZINI *et al* 2015].

Assim, seu conjunto de ferramentas facilita o desenvolvimento responsivo de *sites* e sistemas *web*.

## 3. Trabalhos Correlatos

Esta seção tem o objetivo de descrever as principais funcionalidades de um aplicativo e um *site* existentes relacionados, que serviram de base para a proposta do desenvolvimento deste trabalho.

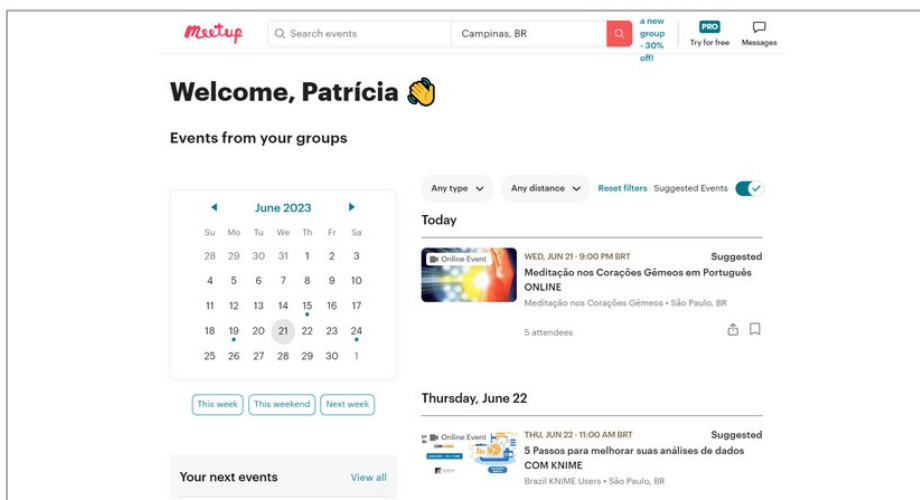
### 3.1 Meetup

*Meetup* é uma aplicação multiplataforma, podendo ser utilizada tanto na *web* quanto em dispositivos móveis, que tem por objetivo divulgar eventos (dos mais diversos e nas mais diversas cidades) criados por usuários da aplicação [MEETUP 2023].

Ela incentiva a criação de comunidades locais baseadas em assuntos de interesse mútuo. A necessidade de haver uma comunidade para a criação de um evento ou atividade é similar ao

fato de uma sessão de exibição do MIS estar associada a uma curadoria.

A Figura 2 mostra a interface de exibição cronológica dos eventos, e a possibilidade de filtrar os eventos de acordo com uma data escolhida no calendário:



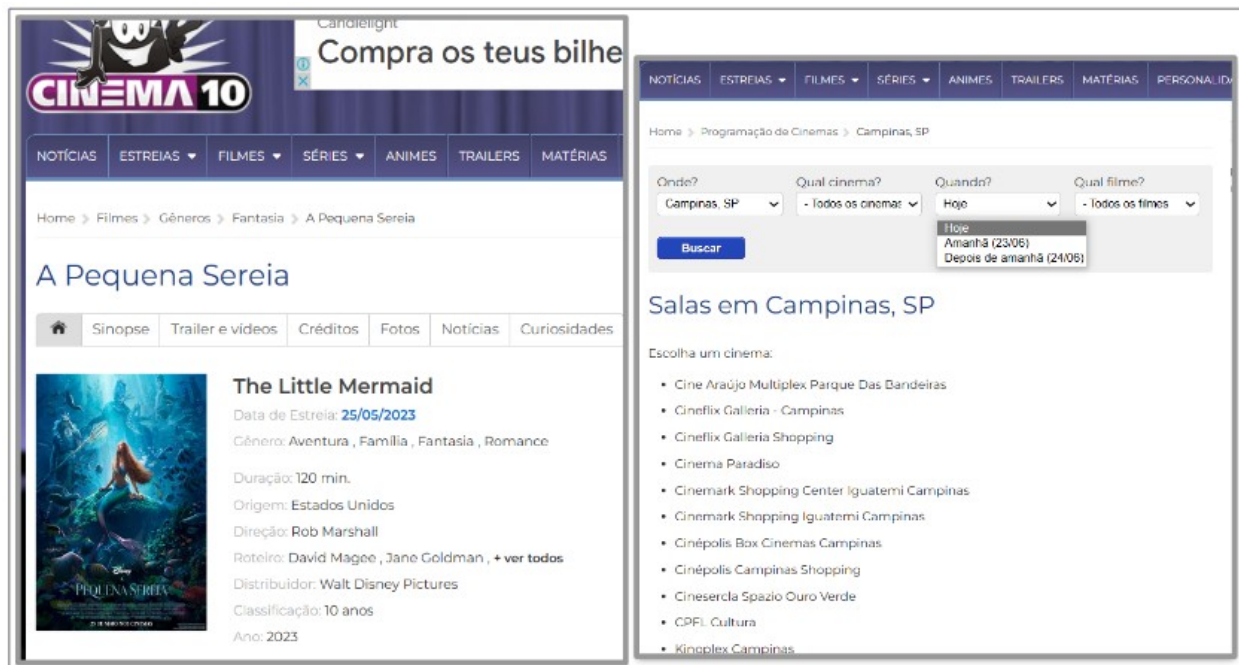
**Figura 2. Tela de eventos do Meetup [MEETUP LCC 2023].**

A utilização da aplicação é grátis para quem busca informações sobre os eventos, porém é paga para quem os divulga (para as comunidades). Para utilizar a aplicação é necessário fazer um cadastro, e para constituir uma comunidade de divulgação de eventos é preciso efetuar o pagamento do pacote de divulgação desejado. É possível buscar eventos filtrando por localização e/ou por tema. Além do *site web* existe um aplicativo para ser utilizado em dispositivos móveis.

### 3.2 *cinema10.com.br*

O portal *cinema10.com.br* é um *site web* que divulga sessões de cinema. É possível buscar salas de cinema por localidade, e sessões por sala. Ele permite a busca pelo nome do filme. É grátis para quem consulta. A diferença em relação ao *Cinemis* é que ele é dedicado às exibições presentes no circuito comercial de cinema.

A Figura 3 mostra as interfaces de filtragem das sessões, e de detalhamento de uma sessão, com as informações pertinentes a um filme:



**Figura 3. Telas de busca com filtros e de detalhamento da sessão [cinema10.com.br 2023].**

### 3.3. Comparativo entre trabalhos

A Tabela 1 apresenta um comparativo das semelhanças e diferenças entre os trabalhos correlatos e a proposta da aplicação Cinemis.

**Tabela 1. Comparativo entre *Cinemis* e aplicações correlatas.**

Funcionalidades	<i>Meetup</i>	<i>cinema10.com.br</i>	<i>Cinemis</i>
Multiplataforma	sim	não	sim
Comunidade Cria Evento	sim	não	sim
Grátis para quem publica o evento	não	Não informado	sim
Focado em sessões de cinema	não	sim	sim
Toda a programação divulgada é grátis	não	não	sim

### 4. Metodologia

Este trabalho é um estudo de caso do fluxo já implementado de gestão das sessões de exhibições dos filmes no MIS Campinas. Inicialmente, foi efetuada a elicitação de requisitos através da diagramação deste fluxo.

Para a elicitação dos requisitos coletamos informações através das técnicas de observação (etnografia) e de entrevista. Tivemos acesso ao grupo de *whatsapp* que abrange a comunicação entre curadoras e curadores com a administradora, da administradora com a pessoa responsável pela diagramação da programação de bolso, e da administradora com a pessoa responsável pela divulgação das sessões nas mídias sociais. Também por esse grupo tivemos acesso ao produto da divulgação das sessões: a programação de bolso e os links de divulgação das sessões nas redes sociais. Frequentamos as sessões de exhibições cineclubistas participando dos debates e a partir daí tivemos conversas informais com curadores e curadoras, e também com a administradora. Por fim, marcamos uma entrevista com a administradora onde expusemos a diagramação do

fluxo de gestão das sessões, e os requisitos funcionais da aplicação, pedindo para que ela avaliasse o material, indicando se o problema tinha sido corretamente mapeado, e se a solução agregaria valor ao seu trabalho, deixando-a livre para propor alterações. As sugestões de melhorias foram então implementadas de modo a integrar o produto final desse trabalho.

A partir da diagramação do fluxo de gestão das sessões foi criado o diagrama de Entidade e Relacionamento, e a definição dos Requisitos Funcionais e Não Funcionais.

Após isso, para o refinamento dos requisitos, foram criadas histórias de usuário, juntamente com protótipos de baixa fidelidade, com a ferramenta de *design* “Canva”, a fim de validar entradas de dados e estrutura das interfaces. Definidos os requisitos, foi feito um esquema da solução técnica para decidir as tecnologias a serem utilizadas no desenvolvimento do projeto, assim como a sua arquitetura.

A base de dados foi modelada com informações a serem utilizadas em uma futura implementação do *back-end*.

Para a implementação do projeto foi proposta a utilização do *framework Bootstrap*, as linguagens HTML, CSS e Javascript para o desenvolvimento do código do *front-end*, e a utilização do *framework Express* (com *Node.js*), e a linguagem de programação Javascript para o desenvolvimento do código do *back-end*.

## 5. Desenvolvimento

Esta seção apresenta o desenvolvimento realizado neste trabalho, com descrições de cada tarefa realizada nas diferentes fases de sua elaboração.

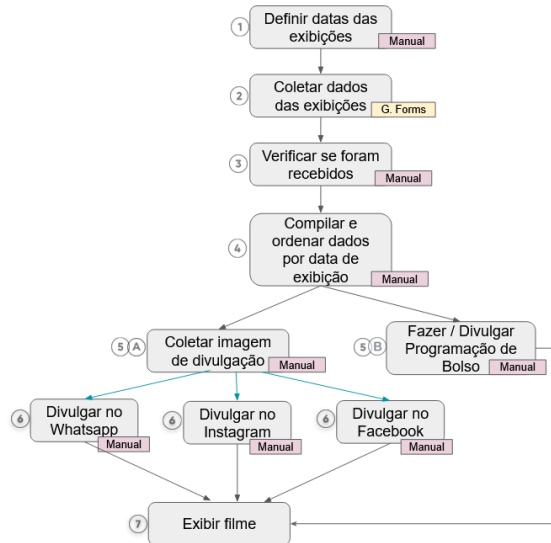
### 5.1. Elicitação de Requisitos

De acordo com Ian Sommerville (2018), "os objetivos do processo de elicitação de requisitos são compreender o trabalho que os *stakeholders* realizam e entender como usariam um novo sistema para apoiar o trabalho deles" (p. 96).

A finalidade da elicitação de requisitos é saber mais sobre o domínio da aplicação, as atividades envolvidas no trabalho, os serviços e as características que o sistema desejado precisará ter. Durante a fase de elicitação de requisitos buscou-se compreender como as pessoas trabalham na administração das sessões e na curadoria delas, o que elas produzem, como usam outros sistemas e o que precisariam alterar para acomodar um novo sistema.

Para este trabalho estabeleceu-se como requisito de finalização da fase de elicitação de requisitos a criação de uma documentação de registro do fluxo de trabalho de administração e curadoria das sessões cineclubistas realizadas no Museu da Imagem e do Som de Campinas para que, em seguida, fosse possível definir os requisitos inicialmente necessários à implementação da solução.

Na Figura 4 temos a representação simplificada do fluxo do trabalho de administração e divulgação das sessões:



**Figura 4. Representação simplificada do fluxo de trabalho do circuito MIS de cinema.**

Passaremos agora à descrição detalhada de cada uma das etapas desse fluxo, analisando a decisão de incluir ou não a etapa em questão como requisito da primeira versão da aplicação *Cinemis*.

### 5.1.1. Definir datas das exibições

Por se tratar de uma instituição pública, gerida pelo município de Campinas é necessário que a administradora das sessões verifique junto à administração do MIS Campinas se todas as datas / horários que ela planejou para exibir filmes estarão disponíveis pois, existem casos em que ou o prédio estará fechado, ou a sala de exibição estará alocada para outra atividade que não o cinema cineclubista.

Nessa fase, após confirmar as datas com a administração do museu, a administradora passa a confirmar as datas com as curadoras e curadores - pois aqui pode ser que algum curador queira efetuar a troca de data de sua exibição, ou não esteja disponível para realizá-la naquele mês - já solicitando que eles encaminhem todas as informações sobre o filme a ser exibido (próxima etapa do fluxo).

Esse trabalho de definição de datas é normalmente realizado com periodicidade mensal.

Essa etapa integrou o requisito funcional “cadastrar sessão”.

### 5.1.2. Coletar dados das exibições

Para que seja possível realizar a divulgação da sessão do cinema cineclubista, existem algumas informações importantes a serem informadas tais como o nome do filme, o ano do filme, a classificação indicativa, direção, país, se é legendado, se o áudio é original, se é colorido ou preto e branco, além de uma imagem para divulgação. O papel dos curadores e curadoras é fornecer esses dados para a administradora.

Esse é um ponto que exige uma organização manual bastante grande por parte da administradora na medida em que ela precisa gerir a entrega completa dessas informações pela curadoria sempre no início do mês anterior ao mês da exibição. Devido ao grande volume de sessões, e ao grande número de pessoas envolvidas nesse processo, esse controle manual acaba

sendo passível de falhas na medida em que as informações podem ser entregues de maneira incompleta (gerando um retrabalho ou uma divulgação incompleta das informações do filme), podem ser entregues fora do prazo, ou podem ser entregues mas acabarem esquecidas ou repassadas com alguns erros.

Essa etapa integrou os requisitos funcionais “cadastrar sessão” e “editar sessão”.

### **5.1.3. Verificar se os dados das exhibições foram recebidos**

Essa é a etapa de conferência manual de que todos os dados, de todas as sessões, foram recebidos dentro do prazo e estão corretos.

Essa etapa integrou os requisitos funcionais “detalhar sessão” e “baixar sessões”.

Além desses requisitos funcionais, futuramente poderão ser descritos outros requisitos que determinem ao sistema notificar a administração e a curadoria quando os dados da sessão estiverem incompletos, ou quando os dados ainda não tiverem sido fornecidos dentro do prazo, por exemplo.

### **5.1.4. Compilar e ordenar dados por data de exibição**

Como atualmente o controle das sessões é feito através de um documento do tipo texto, existe um trabalho manual em organizar os dados recebidos em ordem cronológica pois, mesmo havendo um formulário criado no *Google Forms* para que a curadoria envie as informações, nem todos os curadores o utilizam.

Organizar as sessões em ordem cronológica é fundamental para que futuramente seja realizada a divulgação tanto nas redes sociais como impressa (programação de bolso).

Essa etapa integrou o requisito funcional “Listar Sessões Futuras”.

#### **5.1.5.A. Coletar imagem de divulgação**

A coleta de imagem de divulgação da sessão é feita separadamente, já que não há um campo para o usuário fornecer a imagem no formulário do *Google Forms*, isso gera um trabalho extra pois o “mapeamento” entre imagem e dados do filme é feito de maneira manual.

Esta etapa juntamente com a próxima etapa ocorrem simultaneamente e integram a fase de divulgação das sessões em mídias distintas: a imagem do filme é pré-requisito para que a sessão seja divulgada nas mídias sociais, enquanto que a programação de bolso é feita para ser impressa e distribuída na sala de exhibições.

Essa etapa integrou os requisitos funcionais “cadastrar sessão” e “editar sessão”.

#### **5.1.5.B. Fazer / Divulgar programação de bolso**

Uma vez que os dados de todas as sessões estão devidamente inseridos no arquivo de controle das sessões, uma pessoa é responsável por, manualmente, realizar a diagramação dessas informações de forma que a programação das sessões do mês possa ser impressa.

Essa etapa integrou o requisito funcional “baixar sessões”.

### **5.1.6. Divulgar nas plataformas digitais whatsapp, instagram e facebook**

Uma vez que os dados de todas as sessões estão devidamente inseridos no arquivo de controle das sessões, uma pessoa é responsável por verificar se as imagens dos filmes foram fornecidas pela curadoria para então, as sessões dos filmes que tiverem imagem, serem

manualmente divulgadas nas mídias sociais do MIS: *whatsapp*, *instagram* e *facebook*.

Essa etapa não integrou os requisitos funcionais iniciais da aplicação devido a complexidade de sua implementação, por depender da automação de um processo que precisará se comunicar com APIs externas (de outros sistemas). Futuramente essa etapa poderá ser descrita como um requisito.

#### **5.1.7. Exibir o filme**

A exibição do filme encerra o fluxo de trabalho de administração e divulgação das sessões. Ainda que a exibição do filme em formato cineclubista seja a finalidade de todo o fluxo de trabalho descrito, ela não faz parte do escopo da aplicação *Cinemis*.

## **5.2. Requisitos funcionais e não funcionais**

O objetivo principal do desenvolvimento de *software* é atender a requisitos que contemplem necessidades de determinados usuários para resolver determinado problema. Os requisitos de um sistema descrevem os serviços que ele deve prestar e as restrições de sua operação [SOMMERVILLE 2018].

A Engenharia de Requisitos é uma disciplina que integra a Engenharia de *Software* fornecendo técnicas para cobrir atividades de obtenção, documentação e manutenção de um conjunto de requisitos para *software* que atendam aos objetivos de negócio e sejam de qualidade: requisitos com especificações completas, consistentes e não ambíguas que descrevam o quê o produto de *software* irá fazer ou executar [VAZQUEZ e SIMÕES 2016], [TURINE e MASIERO 1996].

Os requisitos de negócio são elaborados anteriormente à criação do projeto, e não envolvem portanto especificações técnicas que serão utilizadas no desenvolvimento da solução.

Enquanto os requisitos funcionais descrevem, em linguagem natural somada a diagramas, o comportamento que o *software* deve ter em relação às tarefas ou serviços do usuário, os requisitos não funcionais descrevem as características de especificação ou restrição do sistema como um todo, que podem gerar limitações gerais ao funcionamento dos requisitos funcionais estabelecidos para o *software* [SOMMERVILLE 2018].

Os requisitos, portanto, não descrevem o desenho da arquitetura de solução, embora sejam profundamente afetados por ela.

A Tabela 2 exhibe os requisitos funcionais e a Tabela 3 exhibe os requisitos não funcionais da aplicação *Cinemis*.

**Tabela 2. Requisitos funcionais da aplicação Cinemis.**

Identificação	Nome	Descrição
RF001	Registrar-se	Permitir que administrador e curador se cadastrem no sistema.
RF002	Efetuar Login	Permitir que administrador e curador tenham um perfil de acesso ao sistema.
RF003	Cadastrar Sessão	Permitir que administrador e curador cadastrem uma sessão no sistema.
RF004	Editar Sessão	Permitir que administrador e curador editem uma sessão no sistema.
RF005	Publicar Sessão	Permitir que administrador publique uma sessão no sistema.
RF006	Excluir Sessão	Permitir que administrador e curador excluam uma sessão no sistema.
RF007	Detalhar Sessão	Exibir detalhes de uma sessão selecionada
RF008	Listar Sessões Futuras	Listar todas as sessões futuras cadastradas
RF009	Listar Sessões Filtradas	Listar todas as sessões de acordo com os filtros selecionados
RF010	Baixar Sessões	Fazer o <i>download</i> de todas as sessões listadas

**Tabela 3. Requisitos não funcionais da aplicação Cinemis.**

Identificação	Nome	Descrição
RNF01	Acesso a Internet	O usuário deve possuir acesso à Internet para utilizar o sistema.
RNF02	Responsividade	Garantir que o sistema se adapte aos diferentes tamanhos e resoluções de tela.
RNF03	Segurança	Garantir que somente um usuário previamente logado no sistema possa realizar o cadastro, edição, publicação ou exclusão de uma sessão.

A técnica de modelagem de casos de uso leva em consideração que todo *software* deve ter um comportamento em resposta a eventos promovidos por seus usuários ou promover eventos que façam esses usuários agirem. Ela especifica o comportamento associado às funcionalidades do *software* por meio da descrição de casos de uso do *software* pelos seus usuários, denominados atores na terminologia de quem define seu padrão, a UML (*Unified Modeling Language* - linguagem de modelagem unificada) [VAZQUEZ e SIMÕES 2016].

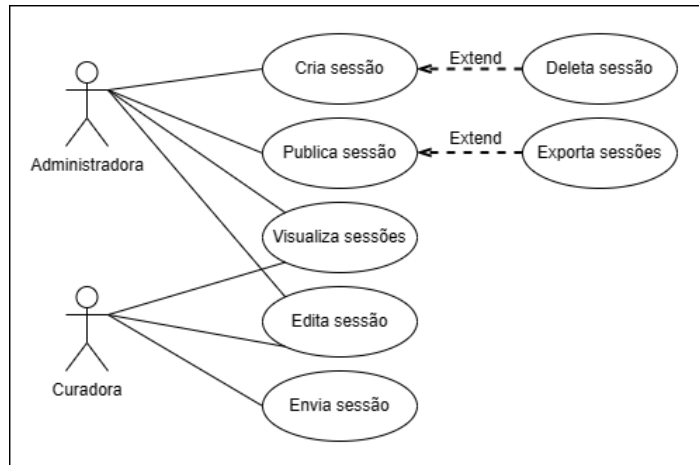
O diagrama de casos de uso demonstra de forma gráfica quais funcionalidades do *software* atenderão a quais usuários específicos, e representa graficamente os papéis que os usuários (denominados atores) desempenham e a inter-relação desses elementos. As funcionalidades estão representadas pelos relacionamentos da interação de um ator com um caso de uso, ou na interação de casos de uso entre si.

Estes modelos expressam os requisitos descritos no documento de requisitos, possibilitando um maior entendimento do domínio da aplicação, servindo para determinar se a especificação está completa, consistente e precisa, fornecendo uma transição para a fase de projeto [TURINE e MASIERO 1996].



A Figura 5 ilustra o diagrama de casos de uso da aplicação Cinemis.

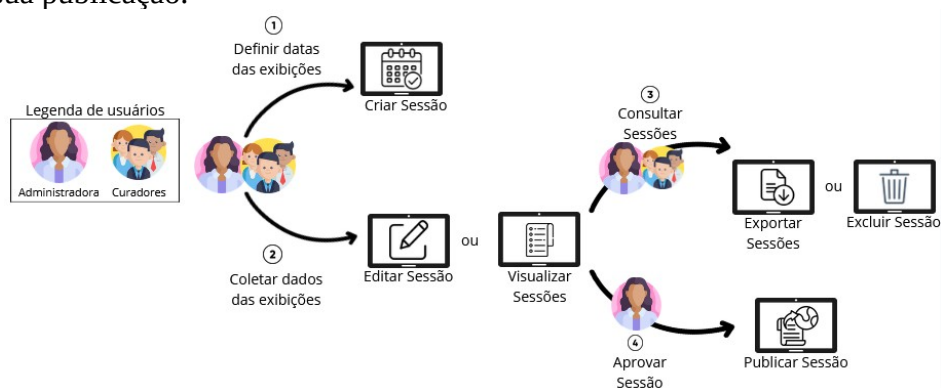
Na modelagem UML, é possível utilizar um relacionamento de extensão para especificar que um caso de uso (extensão) estende o comportamento de outro caso de uso (base). Enquanto o caso de uso base é definido independentemente e é significativo por si próprio, o caso de uso de extensão não é, ou seja, o caso de uso estendido representa um comportamento opcional que o caso de uso que o estende possui, que ainda assim pode acessar e modificar os atributos do caso de uso base. No diagrama, um relacionamento de extensão é representado como uma seta pontilhada que se origina no caso de uso de extensão em direção ao caso de uso base [IBM, 2021].



**Figura 5. Diagrama de casos de uso da aplicação Cinemis.**

Após o processo de compreensão dos requisitos do sistema, foi definido e criado um fluxo da visão geral de negócio da aplicação para representar sua utilização pelos usuários (administradora e curadores). A Figura 6 demonstra o fluxo de negócio criado.

O fluxo se inicia quando um usuário (ou administradora ou curadores) acessa o sistema a fim de definir as datas das sessões (passo 1) para poder criar uma nova sessão. Após criar a sessão será feita a coleta dos dados das exibições (passo 2), onde administradora e curadores poderão editá-las para acrescentar todos os dados da exibição ou visualizá-las. Ao consultar as sessões inseridas no sistema (passo 3), administradora e curadores poderão exportá-las (em formato pdf) ou excluí-las. Por fim, caberá à administradora aprovar a sessão (passo 4) realizando sua publicação.



**Figura 6. Esquema de visão de negócio elaborado.**

### 5.3. Prototipação de interfaces e Histórias de Usuários

Para apresentar uma ideia das telas da aplicação, foram produzidos os protótipos das interfaces.

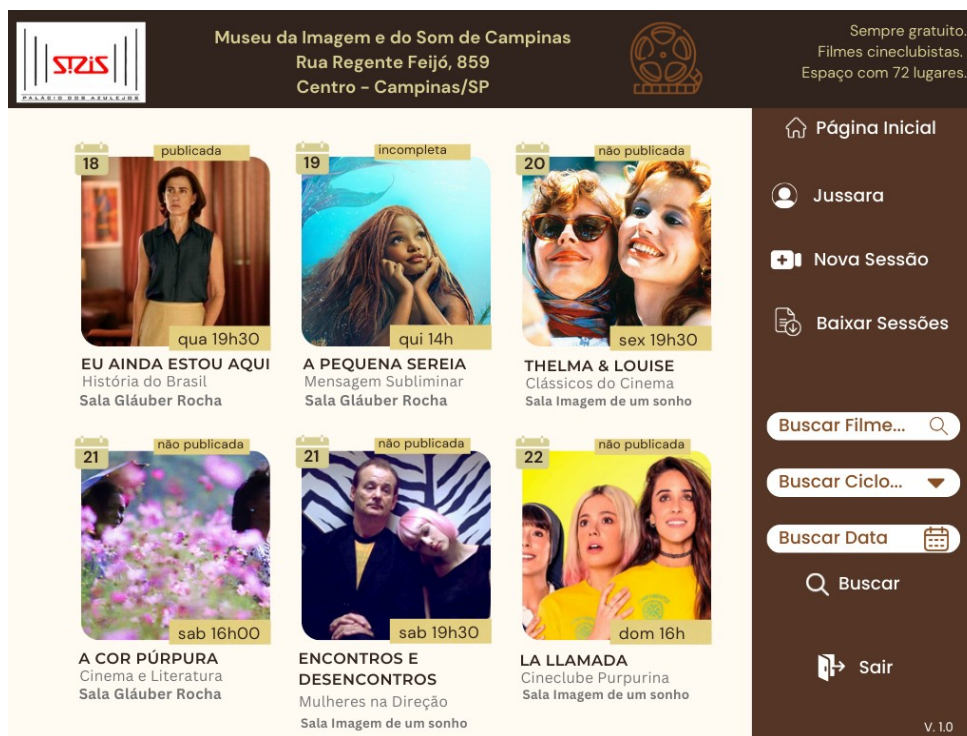
Juntamente com o desenvolvimento dos protótipos, foram sendo criados e definidos os critérios de aceitação e as histórias de usuário para cada uma das interfaces.

As Histórias de Usuário capturam como as pessoas lidam, ou imaginam que poderiam lidar, com determinadas situações. A partir dessa narrativa mais concreta, a partir de exemplos da vida real, fica mais fácil compreender os requisitos de sistema mais especificamente.

Dessa forma, elas descrevem como o sistema pode ser utilizado em alguma tarefa em particular. Elas são escritas como texto narrativo e apresentam uma descrição de alto nível do uso do sistema [SOMMERVILLE 2018].

As figuras abaixo demonstram exemplos de interfaces prototipadas. Outros protótipos das interfaces da aplicação estão disponíveis no Apêndice A deste trabalho.

A Figura 7 é o protótipo da tela principal da aplicação onde são listadas na divisão principal todas as sessões cadastradas indicando seus estados (não publicada, incompleta, publicada). Em um menu lateral são exibidas as opções de editar os dados cadastrais, criar nova sessão e baixar sessões. Ele contém ainda opções de filtragem das sessões por nome do filme ou direção, ciclo ou curadoria, e data. Por fim um cabeçalho dispõe informações estáticas sobre a localização do Museu, o tipo de exibição dos filmes (formato cineclubista), bem como sua capacidade de acomodação de pessoas.



**Figura 7. Protótipo da tela principal com a listagem das sessões, cabeçalho e menu lateral.**

A História de Usuário correspondente a esta tela (Exibir sessões cadastradas cronologicamente) encontra-se abaixo. As outras histórias estão disponíveis no Apêndice B deste trabalho.

### 5.3.1. Exibir sessões cadastradas cronologicamente

Como administradora ou curadora, quero visualizar os dados das exibições ordenados cronologicamente

Pré-requisito: As sessões devem ter sido previamente cadastradas;

Critérios de aceite:

1. As sessões devem estar ordenadas cronologicamente (por data e horário), tomando como data inicial para condição de exibição a data atual, ou seja, serão exibidas as sessões que tenham dia e horário igual ou maior à data atual;
2. Todos os dados mínimos obrigatórios da sessão deverão ser exibidos: nome do filme / nome da curadoria / data da exibição / horário da exibição. O estado da sessão deverá ser exibido: publicado / incompleto / não publicado;
3. A página deve ser 'escrolável';

A Figura 8 é o protótipo da tela de detalhamento de uma sessão, com todos os campos do formulário sendo exibidos, além da exibição do estado da sessão, e as possibilidades de excluir, editar ou publicar a sessão, respeitando os requisitos (apenas a administradora pode publicar uma sessão, apenas a administradora pode deletar ou editar uma sessão já publicada).



**Figura 8. Protótipo da tela de detalhamento de uma sessão.**

A História de Usuário correspondente a esta tela (Exibir detalhamento da sessão

selecionada) encontra-se abaixo.

### 5.3.2. Exibir detalhamento da sessão selecionada

Como administradora ou curadora, quero visualizar os dados detalhados de uma exibição cadastrada

Pré-requisito: A sessão deve ter sido previamente cadastrada;

Critérios de aceite:

1. Ao selecionar uma das sessões listadas, devem ser visíveis as informações de todos os campos presentes no formulário de cadastro da sessão;
2. Todos os dados mínimos obrigatórios da sessão deverão ser exibidos: nome do filme / nome da curadoria / data da exibição / horário da exibição;
3. Se a sessão for uma sessão de estado 'publicada', todos os dados do formulário de cadastro da sessão devem ser exibidos preenchidos;
4. Se a sessão for uma sessão de estado 'incompleto', os campos não preenchidos deverão aparecer destacados indicando que estão em branco;
5. Para usuário do tipo curadoria deverão aparecer as opções de editar e excluir a sessão;
6. Para usuário do tipo administradora além das opções de editar e excluir a sessão, deverá aparecer a opção de publicar a sessão;
- 6.a. Se todos os dados da sessão estiverem preenchidos, a opção de publicar a sessão deverá estar habilitada;
- 6.b. Se nem todos os dados da sessão estiverem preenchidos, a opção de publicar a sessão deverá estar desabilitada;

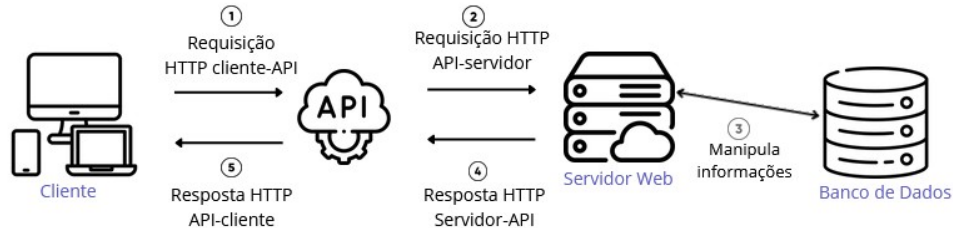
## 5.4. Visão Geral da Arquitetura

A arquitetura do *software* identifica os principais componentes estruturais em um sistema e as relações entre eles, determinando a estrutura geral do sistema e como ele deve ser organizado. Ela deverá ser planejada estando fundamentalmente vinculada à engenharia de requisitos, de modo que as especificações do sistema viabilizem o desenvolvimento de seus requisitos, previamente determinados. Ao definir as características do sistema, afeta seu desempenho, robustez, capacidade de distribuição e manutenibilidade, ou seja, seus requisitos não funcionais [SOMMERVILLE 2018].

Ao projetar a arquitetura da aplicação *Cinemis* foi considerado o reaproveitamento do conhecimento de arquiteturas de aplicações do tipo *web*, considerando que sistemas do mesmo domínio de aplicação costumam ter arquiteturas similares. Optou-se assim por adotar o padrão de arquitetura cliente-servidor, do tipo *Model View Controller* (Modelo, Visão e Controle), conforme descrito no item 2.2 deste trabalho.

A representação gráfica do padrão de arquitetura deste projeto é exibido pela Figura 16. Nela é possível observar a comunicação entre programas dentro da aplicação *Cinemis*: o programa no lado do cliente se comunica com uma API (*Application Programming Interface* -

Aplicação Programada para Interface) através de uma requisição HTTP (*Hypertext Transfer Protocol*). Essa requisição é enviada pela API ao servidor. Ao realizar o processamento da requisição, o servidor poderá comunicar-se com o banco de dados do sistema para consultar e/ou manipular informações. Uma requisição de resposta é então enviada pelo servidor à API, que a retorna para o cliente através de uma resposta HTTP.



**Figura 9. Representação visual da comunicação entre programas do Cinemis.**

## 5.5. Modelagem da base de dados

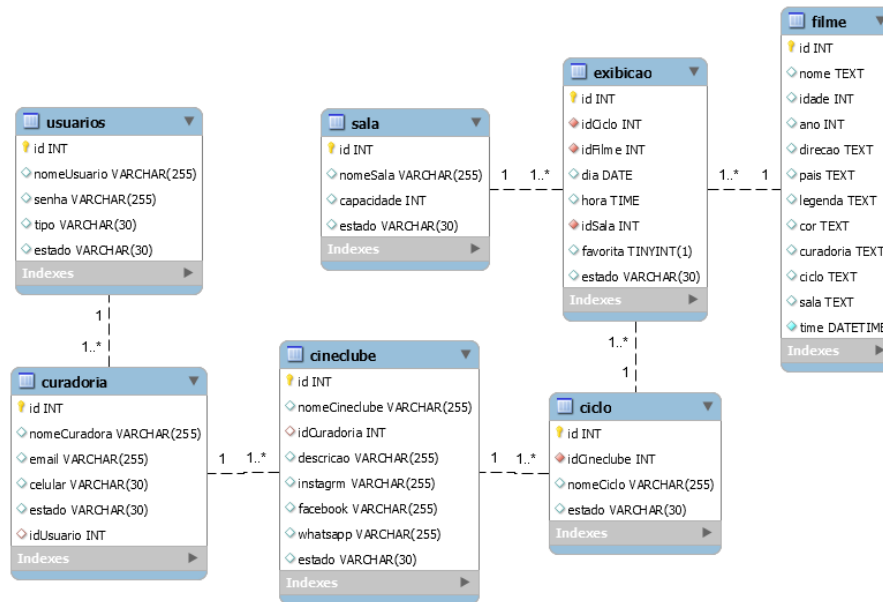
Nesta seção é apresentada a modelagem do banco de dados a ser utilizada pela aplicação para a manipulação de dados pelo *back-end* do sistema.

O diagrama entidade relacionamento (DER) elaborado para o sistema é exibido na Figura 17 abaixo. Nela é possível observar as estruturas das tabelas a serem utilizadas, bem como as relações entre elas.

A tabela 'exibição' é a principal tabela do diagrama, tendo uma forte associação com os dados armazenados nas tabelas 'filme' (que contém todos os dados referentes ao filme), 'sala' (que contém os dados das salas de exibição), e 'ciclo' (que determina a qual ciclo a exibição está associada).

A tabela 'curadoria', por sua vez, associa-se à tabela 'usuários' de modo que todo o usuário que não seja do tipo 'administrador', será necessariamente associado à tabela 'curadoria'. Nesta tabela estão presentes os dados dos curadores e curadoras das sessões. Nem todo o curador ou curadora cadastrado no sistema será um usuário da aplicação.

Por fim, os curadores e curadoras compõem um cineclube (tabela 'cineclube'), e o cineclube é responsável pela realização de um ciclo de exibição de filmes (tabela 'ciclo').



**Figura 10. Diagrama de Entidade Relacionamento.**

## 6. Conclusão

Este trabalho fez o mapeamento do fluxo do trabalho de administração e divulgação das sessões cineclubistas de cinema do MIS Campinas. A partir daí identificou todos os pontos de atuação manual que poderão ser futuramente automatizados e deu início ao processo de construção de uma aplicação *web*, que viabilizará essas automações futuras, através do levantamento dos requisitos funcionais e não funcionais, definição geral de arquitetura, e construção de protótipos de algumas das telas da aplicação.

Os requisitos necessários foram descritos em formato de casos de uso, foram definidas as tecnologias que seriam utilizadas na construção da aplicação e foi modelado o banco de dados.

Foi documentado ainda que o desenvolvimento inicial da aplicação *Cinemis* deverá realizar o armazenamento concentrado dos dados das sessões, possibilitando seu acesso por diversas pessoas ao mesmo tempo. Além disso, possibilitará também que diferentes pessoas possam inserir e editar os dados de exibição no mesmo local, pulverizando assim a responsabilidade de alimentação das informações das sessões (no repositório da aplicação) ao invés de concentrar essa carga de trabalho em uma única pessoa.

As disciplinas de Banco de Dados, Engenharia de Software, Interação Humano-Computador, Arquitetura de Software, Metodologia de Pesquisa Científica e Tecnológica, Projeto de Sistemas, Gestão de Projetos, realizadas durante o curso possibilitaram o desenvolvimento deste trabalho.

Como perspectivas para trabalhos futuros, propõe-se o desenvolvimento da aplicação com a implementação de todos os requisitos descritos neste estudo, bem como a documentação de novos requisitos, marcando o início de uma segunda fase de desenvolvimento. Entre esses novos requisitos, destaca-se a criação de abas de navegação específicas para a categorização de filmes em exibições futuras, passadas e marcados como favoritos, sendo estes últimos armazenados localmente durante o acesso às informações públicas disponibilizadas pela aplicação, tanto em navegadores *web* quanto em dispositivos móveis. Além disso, sugere-se a

implementação de uma funcionalidade que permita a publicação automática da sessão desejada em redes sociais, com especial atenção às plataformas *Instagram*, *Facebook* e *WhatsApp*.

## Referências

- BAPTISTELLA, Antônio José. (2009) Abordando a arquitetura MVC, e Design Patterns: Observer, Composite, Strategy. Disponível em: <http://www.linhadecodigo.com.br/artigo/2367/abordando-a-arquitetura-mvc-e-design-patterns-observer-composite-strategy.aspx>. Acesso em: 19 nov. 2024.
- BATISTA, Claudia Regina. (2008). Modelo e Diretrizes para o processo de design de interface web adaptativa. Disponível em: <https://repositorio.ufsc.br/xmlui/handle/123456789/91727>. Acesso em: 19 nov. 2024.
- BENYON, David. Interação humano-computador. 2. ed. São Paulo: Pearson, 2011. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 14 dez. 2024.
- BOOTSTRAP. About. Disponível em: <http://getbootstrap.com/about/> Acesso em: 22 Nov. 2024.
- CENTENARO, Jonas. (2014). Desenvolvimento de um software web para gerenciamento de requisitos de software. Disponível em: [https://repositorio.utfpr.edu.br/jspui/bitstream/1/20053/3/FB\\_DESIDM\\_I\\_2014\\_09.pdf](https://repositorio.utfpr.edu.br/jspui/bitstream/1/20053/3/FB_DESIDM_I_2014_09.pdf). Acesso em: 19 nov. 2024.
- DIAS, Ricardo. (2019) Prototipagem de Software. Disponível em: <https://medium.com/contexto-delimitado/prototipagem-de-software-7ac07027e6d8>. Acesso em: 24 nov. 2024.
- ELMASRI, R.; NAVATHE, S. B. Sistema de banco de dados. 6. ed. São Paulo, SP: Pearson, 2010. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 15 dez. 2024.
- FRANÇA, Fabiano dos Santos. (2015). Web design responsivo: caminhos para um site adaptável. Interfaces Científicas-Exatas e Tecnológicas 1.2: 75-84. Disponível em: <https://periodicos.grupotiradentes.com/exatas/article/view/2220/1295>. Acesso em: 17 nov. 2024.
- GUIMARÃES, Ana Luiza. (2019).O que são frameworks para desenvolvimento WEB. Disponível em: <https://blog.configr.com/frameworks-para-desenvolvimento-web>. Acesso em: 19 nov. 2024.
- HERMANO, Pedro. (2024). Entenda o que são Web Apps e conheça alguns exemplos. Disponível em: <https://www.attri.com.br/blog/web-apps-o-que-sao-e-quais-sao-os-principais-exemplos>. Acesso em: 16 nov. 2024.
- IBM. (2021). Diagramas de caso de uso. Disponível em: <https://www.ibm.com/docs/pt-br/dmrt/9.5?topic=diagrams-extend-relationships>. Acesso em: 30 nov. 2024.
- LEMOS, M F de. OLIVEIRA, P C de. RUELA, L C. SANTOS, M da S. SILVEIRA, T C. (2013). Aplicabilidade da arquitetura MVC em uma aplicação web (WebApps). Disponível em: <https://revistas.unifenas.br/index.php/RE3C/article/view/54>. Acesso em: 19 nov. 2024.
- LISBOA FILHO, Jugurta. (2001). Estruturação e modelagem de bancos de dados. Disponível em: <https://aut5826.fau.usp.br/gisbr2001.pdf>. Acesso em: 02 dez. 2024.

- LÓSCIO, B F. OLIVEIRA, H R de. PONTES, J C de S. (2011). NoSQL no desenvolvimento de aplicações Web colaborativas. Disponível em: [https://www.researchgate.net/profile/Bernadette-Loscio/publication/268201466\\_NoSQL\\_no\\_desenvolvimento\\_de\\_aplicacoes\\_Web\\_colaborativas/links/576aa72008aef2a864d1ef8c/NoSQL-no-desenvolvimento-de-aplicacoes-Web-colaborativas.pdf](https://www.researchgate.net/profile/Bernadette-Loscio/publication/268201466_NoSQL_no_desenvolvimento_de_aplicacoes_Web_colaborativas/links/576aa72008aef2a864d1ef8c/NoSQL-no-desenvolvimento-de-aplicacoes-Web-colaborativas.pdf). Acesso em: 23 nov. 2024.
- LULA, Mariana Meirelles de Mello. (2011). Um processo ágil para especificação de requisitos em programas interativos com foco em roteiros de TV. Disponível em: <https://repositorio.ufpb.br/jspui/handle/tede/6064>. Acesso em 30 Nov 2024.
- MARTINS, Diana. Aplicação web: o que é, diferença para website, como funciona e mais! Disponível em: <https://rockcontent.com/br/blog/aplicacao-web/>. Acesso em 19 nov. 2024..
- PEREIRA, Caio Ribeiro. (2014). Node.js: aplicações web real-time com node.js. São Paulo, SP: Casa do Código, E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 19 nov. 2024.
- PESSÔA, Camila. (2022). Node.JS: definição, características, vantagens e usos possíveis. Disponível em: <https://www.alura.com.br/artigos/node-js-definicao-caracteristicas-vantagens-usos>. Acesso em: 19 nov. 2024.
- PRESSMAN, Roger. (1995). Engenharia de software. 3. ed. São Paulo: Makron Books.
- QUINTÃO, Fernanda de Souza. TRISKA, Ricardo. (2014). Design de informação em interfaces digitais: origens, definições e fundamentos. Disponível em: <https://infodesign.emnuvens.com.br/infodesign/article/view/243>. Acesso em: 19 nov. 2024.
- SANTAROSA, Marcelo. (2023). Quem cura o MIS? Projeto final apresentado no Curso de Produção de Documentário DOC 360<sup>0</sup>/2015, 40 anos do MIS Campinas. Disponível em: <https://www.youtube.com/watch?v=DjcwYseY-bc&pp=ygUPcXVlbSBjdXJhIG8gbWlz>. Acesso em 22 dez.
- SANTOS, Ezequias Silva dos. (2022). Desenvolvimento de uma aplicação web de gestão dos processos de distribuição de bebidas, usando o conceito de sistema responsivo. Disponível em: <https://www.proquest.com/openview/3f441582cd5f1d35dc4cdb87c878ffd4/1?pq-origsite=gscholar&cbl=2026366&diss=y>. Acesso em: 17 nov. 2024.
- SEGURADO, Valquiria Santos (org.). Projeto de interface com o usuário. 1. ed. São Paulo, SP: Pearson, 2016. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 14 dez. 2024.
- SERVANO, Marcela. (2023). Cineclube: um espaço político, pedagógico e de formação de público. Disponível em: <https://institutodecinema.com.br/mais/conteudo/cineclube-um-espaco-politico-pedagogico-e-de-formacao-de-publico>. Acesso em: 14 mai. 2023.
- SILVA, Gilmar José da. FERREIRA, Júlio Cesar Oliveira. (2017). Análise Comparativa de Desempenho de Consultas entre um Banco de Dados Relacional e um Banco de Dados Não Relacional. Disponível em: <http://dspace.uniube.br:8080/jspui/handle/123456789/178>. Acesso em: 17 nov. 2024.
- SILVA, Rafael Barbosa da. (2023). Mestre Sala: Desenvolvimento de um sistema web para gerenciamento de disponibilidade e reserva de salas. Disponível em: XXX. Acesso em: 30 Nov. 2024.



- SOMMERVILLE, Ian. (2018). Engenharia de software. 10. ed. São Paulo, SP: Pearson. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 17 nov. 2024.
- TOMAZINI, Marcos. LOPES, Luiz Fernando Braga. (2015). Web design responsivo-Bootstrap. Disponível em: [https://unicv.edu.br//admin/assets/repositorio\\_arquivo/014529452c44398122c6a634ad4e1755.pdf](https://unicv.edu.br//admin/assets/repositorio_arquivo/014529452c44398122c6a634ad4e1755.pdf). Acesso em: 19 nov. 2024.
- TURINE, Marcelo Augusto Santos. MASIERO, Paulo Cesar. (1996). Especificação de requisitos: uma introdução, Disponível em: <https://repositorio.usp.br/bitstreams/499f05d2-f00c-4f6a-8edc-b8939b6ac2fc> . Acesso em: 28 nov. 2024.
- UNIVERSIDADE ESTADUAL PAULISTA DE SÃO PAULO. (2005). Prototipação de Software. Disciplina: Engenharia de Software. Disponível em: [http://disciplinas.lia.ufc.br/es062/arquivos/Captulo\\_8-Prototipacao.pdf](http://disciplinas.lia.ufc.br/es062/arquivos/Captulo_8-Prototipacao.pdf). Acesso em: 19 nov. 2024.
- VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira. (2016). Engenharia de requisitos: software orientado ao negócio. 1. ed. Rio de Janeiro: Brasport. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 26 nov. 2024.
- ZAMBELLI, Rafael. (2022). 9 passos para implementar a automatização de processos na sua empresa. Disponível em: <https://blog-pt.checklistfacil.com/automatizacao-de-processos/>. Acesso em: 02 julho 2023.

## APÊNDICE A - PROTÓTIPOS

### A.1. Protótipo da tela de cadastro de usuário.



Protótipo da tela de cadastro de usuário. A interface é dividida em duas partes principais: uma barra de cabeçalho escura e uma área de formulário clara. À esquerda, há uma barra decorativa amarela com uma ilustração de uma fita de filme preta. O cabeçalho contém o título "Criar conta nova" e um link "Entrar" para quem já tem uma conta. O formulário solicita: nome (preenchido com "Jussara Martins"), e-mail (preenchido com "jumartins@gmail.com"), senha (representada por "\*\*\*\*\*") e o tipo de usuário (menu suspenso com "Curadora" selecionado). Um botão "CRIAR CONTA" está localizado na base do formulário.

**Criar conta nova**  
Já tem conta? [Entrar](#)

Digite seu nome  
Jussara Martins

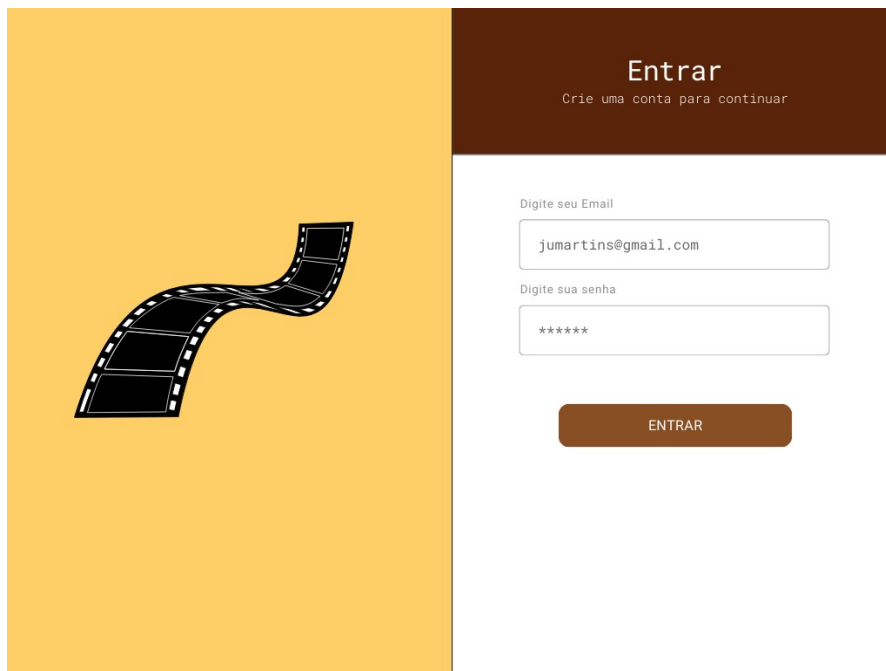
Digite seu Email  
jumartins@gmail.com

Digite sua senha  
\*\*\*\*\*

Tipo de usuário  
Curadora

**CRIAR CONTA**

### A.2. Protótipo da tela de login.



Protótipo da tela de login. A interface é dividida em duas partes principais: uma barra de cabeçalho escura e uma área de formulário clara. À esquerda, há uma barra decorativa amarela com uma ilustração de uma fita de filme preta. O cabeçalho contém o título "Entrar" e o texto "Crie uma conta para continuar". O formulário solicita: e-mail (preenchido com "jumartins@gmail.com") e senha (representada por "\*\*\*\*\*"). Um botão "ENTRAR" está localizado na base do formulário.

**Entrar**  
Crie uma conta para continuar

Digite seu Email  
jumartins@gmail.com

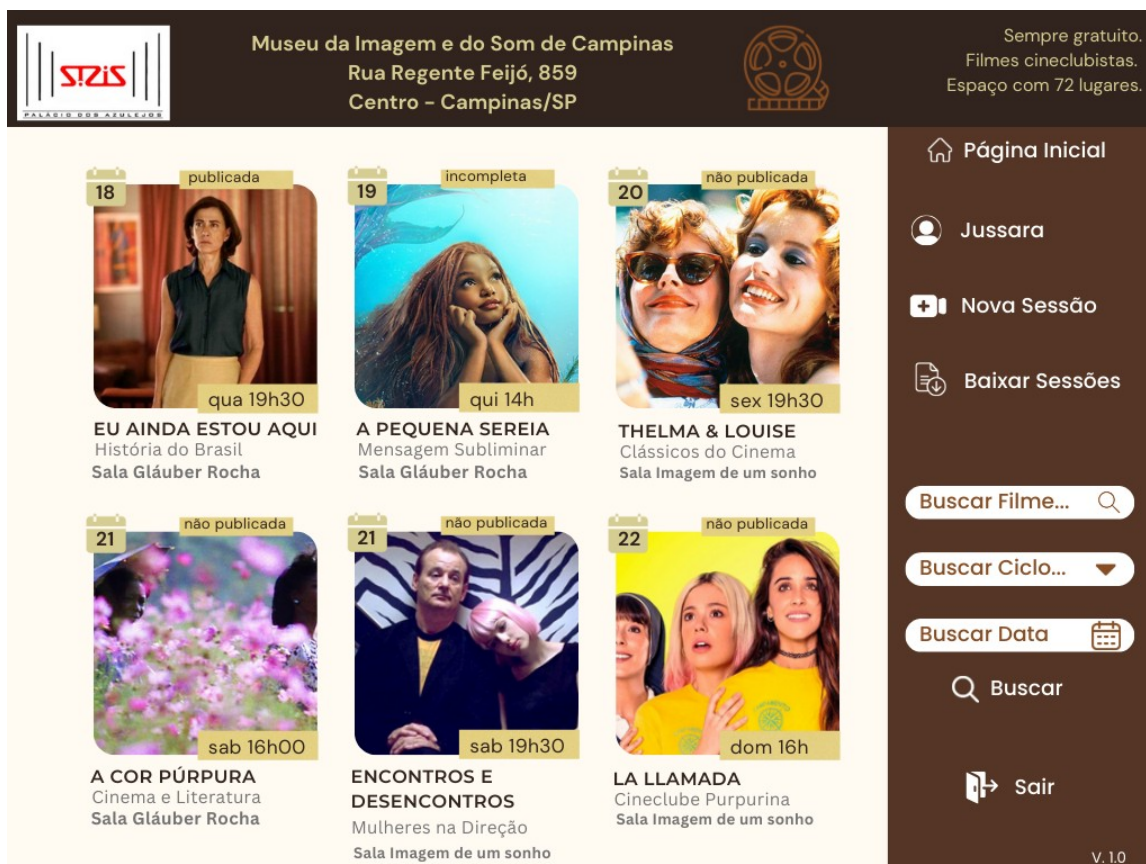
Digite sua senha  
\*\*\*\*\*

**ENTRAR**

### A.3. Protótipo da tela de recuperação de senha.



A.4. Protótipo da tela principal da aplicação contendo todas as sessões cadastradas, seu estado, as opções de criar nova sessão e baixar sessões, um menu lateral com as opções de filtragem das sessões, e cabeçalho com informações estáticas do Museu e a exibição dos filmes.



A.5. Protótipo da tela de detalhamento de uma sessão, que exibe todas as informações do filme e da sessão, e as opções de excluir, editar ou publicar a sessão.



Museu da Imagem e do Som de Campinas  
Rua Regente Feijó, 859  
Centro - Campinas/SP



Sempre gratuito.  
Filmes cineclubistas.  
Espaço com 72 lugares.

publicada



**DOMINGO, 15 DEZ 2024**

Filme: Eu Ainda Estou Aqui  
Ciclo: História do Brasil  
Curadoria: Cláudia Bortoloto  
Duração: 2h17m  
País: Brasil  
Direção: Walter Salles  
Classificação: 14 anos  
Ano: 2024  
Sala: Gláuber Rocha

**SINOPSE**

Em 1971, durante a Ditadura Militar, uma mãe (Fernanda Torres) precisa se reinventar para cuidar dos cinco filhos e ao mesmo tempo lutar por seus direitos como ativista, enquanto o marido foi preso e está sob a custódia da polícia militar.

Sem notícias e sem ao menor saber o motivo de sua prisão, ela nunca desiste de encontrá-lo e acabar com aquela injustiça.

**EXCLUIR**      **EDITAR**      **PUBLICAR**

 Página Inicial

 Jussara

 Nova Sessão

 Baixar Sessões







 Buscar

 Sair

V. 1.0

A.6. Protótipo da primeira tela de cadastro da sessão, com os campos do formulário a serem preenchidos com os dados da sessão.

Museu da Imagem e do Som de Campinas  
Rua Regente Feijó, 859  
Centro - Campinas/SP

Sempre gratuito.  
Filmes cineclubistas.  
Espaço com 72 lugares.

### Cadastro de sessão

Dados da Sessão - Passo 1 de 3

Escolha a data  
23/12/2024

Curadoria  
Ricardo Pereira

Ciclo  
Cinema e Literatura

Sala  
Gláuber Rocha

SALVAR CONTINUAR

Página Inicial  
Jussara  
Nova Sessão  
Baixar Sessões  
Buscar Filme...  
Buscar Ciclo...  
Buscar Data  
Buscar  
Sair

V. 1.0

A.7. Protótipo da segunda tela de cadastro da sessão, com os campos do formulário a serem preenchidos com os dados do filme.

Museu da Imagem e do Som de Campinas  
Rua Regente Feijó, 859  
Centro - Campinas/SP

Sempre gratuito.  
Filmes cineclubistas.  
Espaço com 72 lugares.

### Cadastro de sessão

Dados do Filme - Passo 2 de 3

Nome do Filme  
A Cor Púrpura

Faixa Etária Indicativa  
14 anos

Direção  
Steven Spielberg

Áudio  
Inglês

Ano de Lançamento  
1985

Áudio / Legenda  
Legendado

Pais de Origem  
EUA

SALVAR CONTINUAR

Página Inicial  
Jussara  
Nova Sessão  
Baixar Sessões  
Buscar Filme...  
Buscar Ciclo...  
Buscar Data  
Buscar  
Sair

V. 1.0

A.8. Protótipo da terceira tela de cadastro da sessão, a ser preenchida com a sinopse e a imagem de divulgação do filme.



## **APÊNDICE B - HISTÓRIAS DE USUÁRIO**

### **B.1. Cadastrar sessão**

Como curadora ou administradora do sistema, quero cadastrar todos os dados referentes:

1. Ao filme a ser exibido;
2. À sessão em que o mesmo será exibido;

Pré-requisito: Os dados dos cineclubes e dos curadores devem ter sido previamente cadastrados no banco de dados;

Critérios de aceite:

1. No formulário de cadastro os nomes dos cineclubes devem ser apresentados (em formato lista, tipo dropdown) para serem selecionados;
2. No formulário de cadastro haverá duas opções: Cancelar / Salvar;
3. Ao clicar no botão 'salvar' os dados devem ser inseridos no banco de dados e estarem disponíveis na tela de listagem das sessões para administração e curadorias;
4. Ao clicar no botão 'cancelar' o usuário deve ser informado que os dados serão perdidos;
5. Não deve ser possível cadastrar uma sessão sem preencher os campos mínimos obrigatórios (nome do filme / nome da curadoria / data da exibição / horário da exibição);
6. Não deve ser possível cadastrar mais de um filme para o mesmo dia e horário e sala.

### **B.2. Exibir sessões cadastradas cronologicamente**

Como administradora ou curadora, quero visualizar os dados das exibições ordenados cronologicamente

Pré-requisito: As sessões devem ter sido previamente cadastradas;

Critérios de aceite:

1. As sessões devem estar ordenadas cronologicamente (por data e horário), tomando como data inicial para condição de exibição a data atual, ou seja, serão exibidas as sessões que tenham dia e horário igual ou maior à data atual;
2. Todos os dados mínimos obrigatórios da sessão deverão ser exibidos: nome do filme / nome da curadoria / data da exibição / horário da exibição. O estado da sessão deverá ser exibido: publicado / incompleto / não publicado;
3. A página deve ser 'escolável';

### **B.3. Exibir detalhamento da sessão selecionada**

Como administradora ou curadora, quero visualizar os dados detalhados de uma exibição cadastrada



Pré-requisito: A sessão deve ter sido previamente cadastrada;

Critérios de aceite:

1. Ao selecionar uma das sessões listadas, devem ser visíveis as informações de todos os campos presentes no formulário de cadastro da sessão;
2. Todos os dados mínimos obrigatórios da sessão deverão ser exibidos: nome do filme / nome da curadoria / data da exibição / horário da exibição;
3. Se a sessão for uma sessão de estado 'publicada', todos os dados do formulário de cadastro da sessão devem ser exibidos preenchidos;
4. Se a sessão for uma sessão de estado 'incompleto', os campos não preenchidos deverão aparecer indicando que estão em branco;
5. Para usuário do tipo curadoria deverão aparecer as opções de editar e excluir a sessão;
6. Para usuário do tipo administradora além das opções de editar e excluir a sessão, deverá aparecer a opção de publicar a sessão;
- 6.a. Se todos os dados da sessão estiverem preenchidos, a opção de publicar a sessão deverá estar habilitada;
- 6.b. Se nem todos os dados da sessão estiverem preenchidos, a opção de publicar a sessão deverá estar desabilitada;

#### **B.4. Editar sessão**

Como administradora ou curadora, quero poder editar os dados de uma exibição cadastrada

Pré-requisito: A sessão deve ter sido previamente cadastrada;

Critérios de aceite:

1. Ao selecionar uma das sessões listadas, deve haver a opção de editá-la;
2. Deve ser possível editar qualquer um dos dados do formulário de cadastro da sessão;
3. Se a sessão for uma sessão de estado 'publicada', apenas a administradora poderá editar a sessão;
4. Deve ser possível cancelar as edições realizadas sem que as informações originais da sessão sejam alteradas/perdidas;
5. Ao cancelar a edição uma mensagem de que todas as alterações serão perdidas deverá ser exibida;
6. Deve ser possível salvar as edições realizadas de modo que as informações originais da sessão sejam sobrescritas pelas novas informações;
7. Ao confirmar a edição uma mensagem de que todas as alterações foram salvas deverá ser exibida;
8. Ao tentar salvar as edições com valores inválidos, uma mensagem deverá ser exibida informando que as alterações não foram salvas, indicando especificamente qual valor é inválido e porque;

- 8.a. É considerado inválido deixar em branco um dos campos mínimos obrigatórios (nome do filme / nome da curadoria / data da exibição / horário da exibição);
- 8.b. É considerado inválido cadastrar mais de um filme para o mesmo dia e horário e sala;
- 8.c. É considerado inválido cadastrar uma sessão para data passada;
- 9. Deve ser possível fechar a tela de edição, e então será exibida a tela de listagem de sessões;

### **B.5. Publicar sessão**

Como administradora, quero poder publicar uma sessão, para que ela fique disponível para ser visualizada por qualquer pessoa que acesse a aplicação

Pré-requisito: A sessão deve ter sido previamente cadastrada e ter todos os dados preenchidos;

Critérios de aceite:

1. Ao clicar na opção de publicar a sessão, uma mensagem de confirmação deverá aparecer, com as opções de prosseguir ou cancelar;
2. A mensagem de confirmação deverá informar que ao publicar uma sessão, suas informações se tornarão públicas;
3. Ao confirmar a publicação o estado da sessão deverá ser atualizado para 'publicado', e uma mensagem de sucesso deverá ser exibida;
4. Ao cancelar a publicação da sessão, nada deverá ser alterado;

### **B.6. Filtrar sessão**

Como administradora ou curadora, quero poder filtrar sessões, para que apenas as sessões que tenham os parâmetros do filtro sejam exibidas

Pré-requisito: Deve haver pelo menos uma sessão previamente cadastrada;

Critérios de aceite:

1. Deve haver três opções de filtro: um que receba texto, um que liste os nomes dos cineclubes/ciclos, e um que disponibilize as datas do calendário;
2. O filtro do tipo texto deverá buscar correspondência nos campos nome do filme e direção;
3. O filtro de listagem deve listar todos os nomes de cineclubes e de ciclos, devendo ser possível escolher um ou mais itens da lista para a filtragem;
4. O filtro de calendário deve exibir todas as datas, devendo ser possível filtrar por um dia ou por um intervalo de dias;
5. Para realizar a filtragem o usuário poderá preencher um, dois ou os três filtros com dados para serem filtrados;
6. A sessão deverá aparecer na listagem de sessões filtradas caso tenha correspondência com todos os filtros utilizados;
7. Caso nenhuma sessão corresponda aos parâmetros do(s) filtro(s), deverá ser exibida uma

mensagem informando que a busca não retornou resultados;

### **B.7. Excluir sessão**

Pré-requisito: A sessão deve ter sido previamente cadastrada;

Critérios de aceite:

1. Ao selecionar uma das sessões listadas, deve haver a opção de excluí-la;
2. Se a sessão for uma sessão de estado 'publicada', apenas a administradora poderá excluir a sessão;
3. Ao clicar no botão 'excluir', uma mensagem de confirmação da exclusão deverá ser exibida informando que a sessão será permanentemente excluída da aplicação;
4. Ao clicar no botão 'excluir', na mensagem de confirmação da exclusão deverá haver duas opções: cancelar ou confirmar a exclusão;
5. Ao cancelar a exclusão nada deverá ser alterado;
6. Ao confirmar a exclusão uma mensagem de que a sessão foi excluída deverá ser exibida;
7. A sessão excluída deverá ser removida da listagem de sessões;
8. A sessão excluída não deverá ser exibida na listagem de sessões filtradas;

### **B.8. Baixar sessões**

Como administradora ou curadora, quero poder baixar (em meu dispositivo) as informações de todas as sessões listadas

Pré-requisito: Deve haver pelo menos uma sessão previamente cadastrada;

Critérios de aceite:

1. A página inicial deverá conter um botão para o download da listagem das sessões publicadas;
2. Caso nenhuma sessão esteja cadastrada, o botão de download deverá estar desabilitado;
3. Ao clicar no botão de baixar as sessões, finalizado o download, o usuário deverá receber uma mensagem indicando que o mesmo foi finalizado;
4. Havendo erro para baixar o arquivo, o usuário deverá receber uma msg indicando o motivo da falha;
5. Todas as informações cadastradas para todas as sessões atualmente listadas devem ser incluídas no arquivo;
6. Caso a listagem atual seja de sessões filtradas, apenas as sessões filtradas deverão estar presentes no arquivo;

# Documento Digitalizado Público

## Versão Final do Artigo de TCC

**Assunto:** Versão Final do Artigo de TCC  
**Assinado por:** Fernando Sambinelli  
**Tipo do Documento:** Formulário  
**Situação:** Finalizado  
**Nível de Acesso:** Público  
**Tipo do Conferência:** Documento Digital

Documento assinado eletronicamente por:

- **Fernando Sambinelli, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 06/02/2025 10:00:34.

Este documento foi armazenado no SUAP em 06/02/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

**Código Verificador:** 1930418

**Código de Autenticação:** 30cf74ffb9

