PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA

UNIDADE EDUCACIONAL PRAÇA DA LIBERDADE

Bacharelado em Engenharia de Software

Ana Luiza Ribeiro Martins Pertussati

Bryann Bueno Fonseca

Jose Vitor da Silva Sousa

Lucas Gabriel Padrão Rezende

Victor Menezes Caldeira

Salão de festas – Patati Patata

Belo Horizonte

2020

Ana Luiza Ribeiro Martins Pertussati

Bryann Bueno Fonseca

Jose Vitor da Silva Sousa

Lucas Gabriel Padrão Rezende

Victor Menezes Caldeira

Salão de festas – Patati Patata

Trabalho interdisciplinar de fundamentos de engenharia de software e algoritmos e estrutura de dados.

Professores: Ivre Marjorie Ribeiro Machado e Maria Augusta Vieira Nelson

Belo Horizonte

2020

SUMÁRIO

1. Apresentação

O intuito do nosso programa é sanar alguns dos problemas que já aconteceram no salão de festas – Patati Patata. Realizamos um sistema que cadastre os clientes, fornecedores, funcionários e as festas para que não ocorra conflitos de datas e a perda de informações.

2. Backlog

A figura 1 apresenta uma visão geral de como organizamos a divisão de tarefas para cada membro do grupo. O quadro foi dividido em 4 colunas, sendo elas: “Coisas para fazer”, “Em andamento”, “Concluído” e “Documentação”. Decidimos fazer 4 Sprints com duração de 4 dias cada para conseguirmos entregar todas as atividades até o final do prazo estipulado identificando quem é o responsável de cada questão.

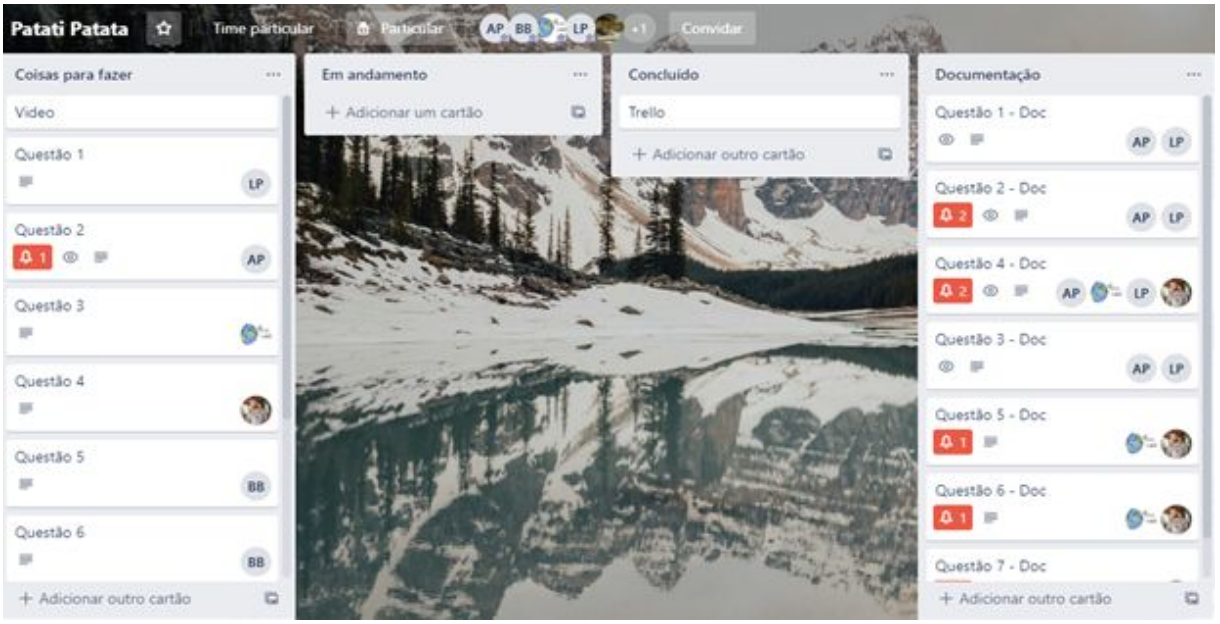

Figura 1 – Início do projeto.

Na primeira Sprint focamos mais no código, cada membro do grupo fez uma questão. Totalizando 5 atividades ao final, como mostra na Figura 2. Nos 4 dias

seguintes (Sprint 2), designamos poucas tarefas pois estávamos testando o código e arrumando alguns erros (Figura 3).

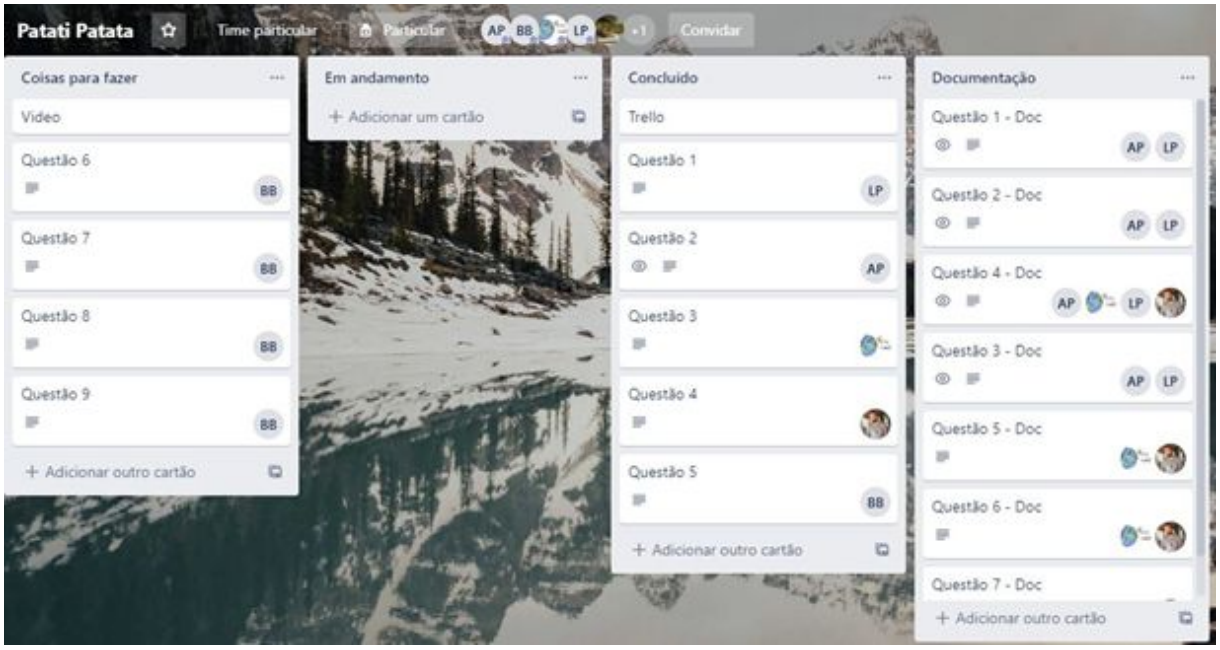

Figura 2 – Final da Sprint 1.

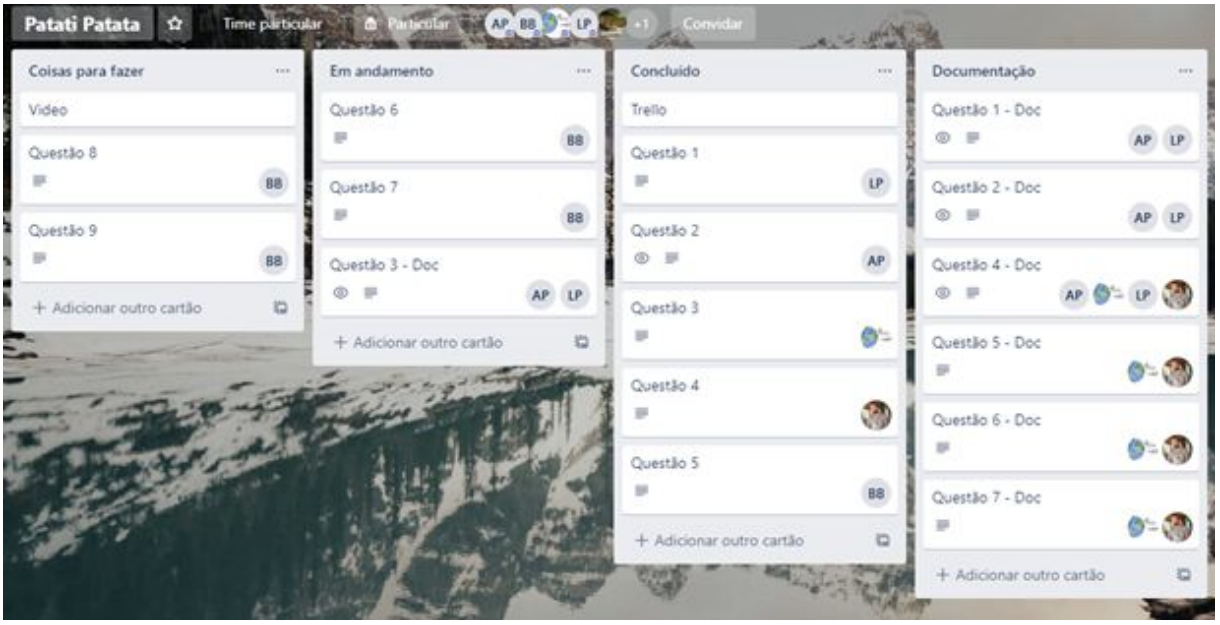

Figura 3 – Inicio da Sprint 2.

Chegando nos últimos dias da Sprint 2, conseguimos apenas realizar os códigos, a documentação ficou para os próximos dias com o início da nossa penúltima Sprint (Figura 4).

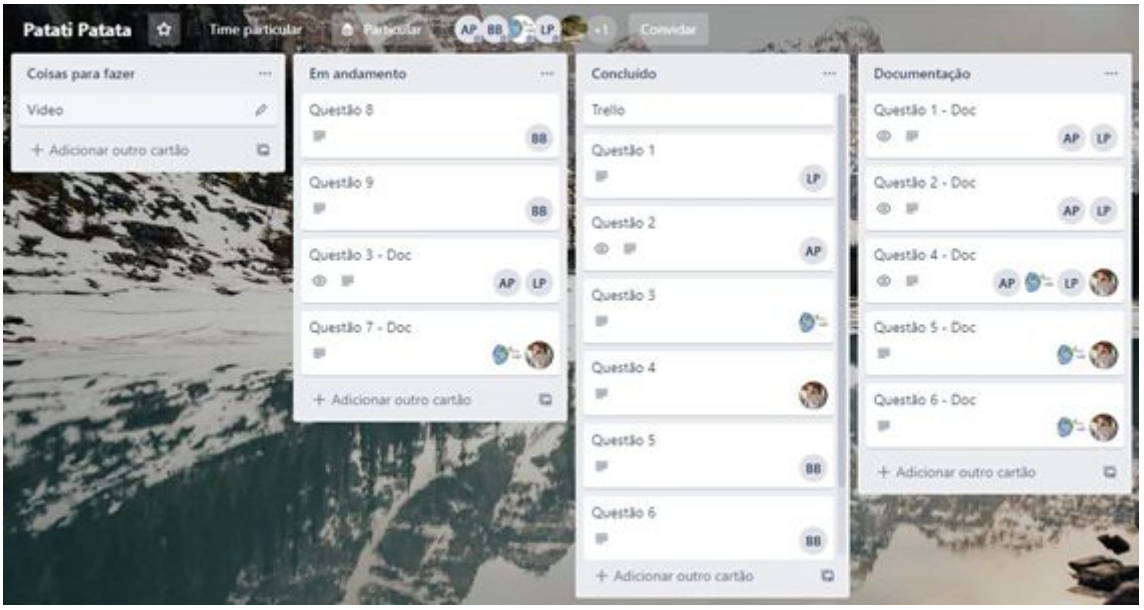

Figura 4 – Final da Sprint 2 e início da Sprint 3.

Com a entrega se aproximando, finalizamos o código como planejado. Na nossa última Sprint (4), focamos na documentação e a apresentação do vídeo. (Figura 5). Dessa forma, concluímos todas as atividades e entregamos nosso sistema funcionando perfeitamente (Figura 6).Ademais, foi feito o reajuste do Trello, explicitando melhor a conclusão das tarefas e o que foi codificado por cada membro(Figura 7).

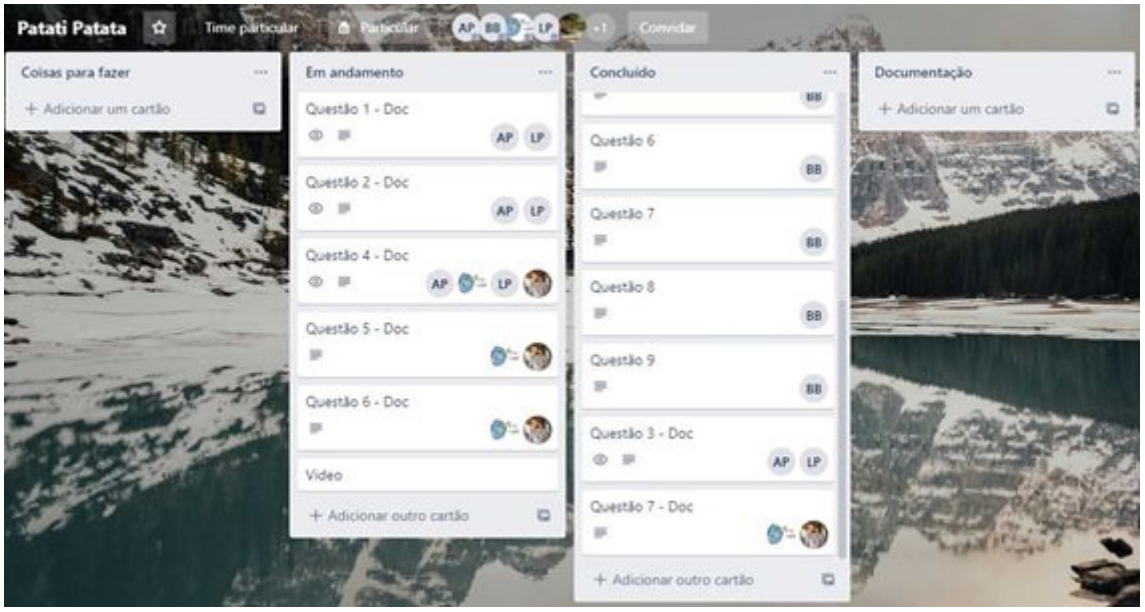

Figura 5 – Final da Sprint 3 e início da Sprint 4.

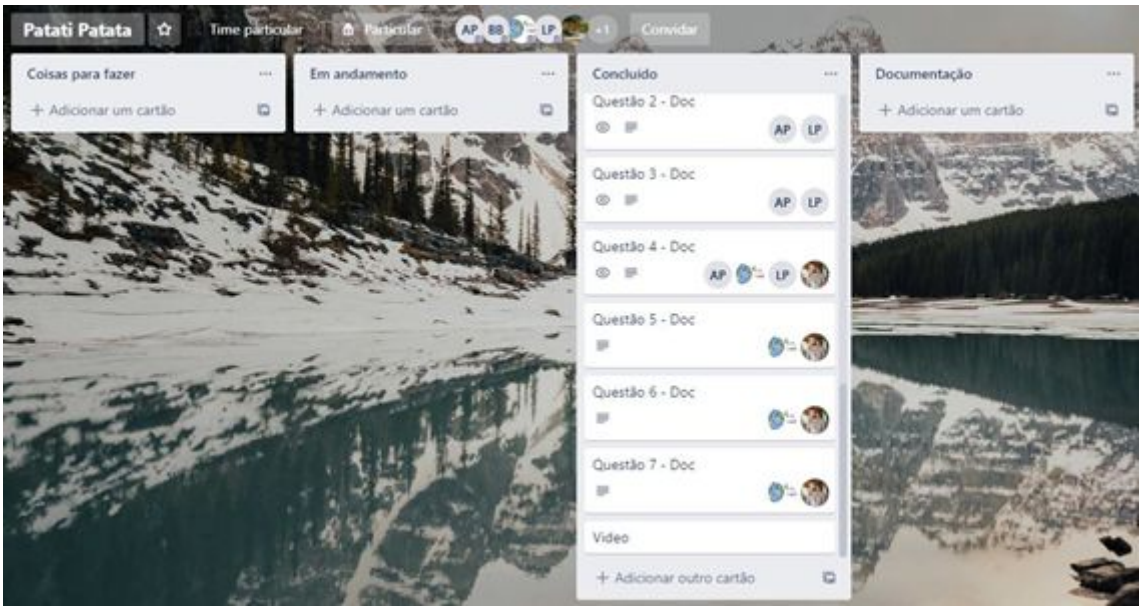

Figura 6 – Final das atividades.

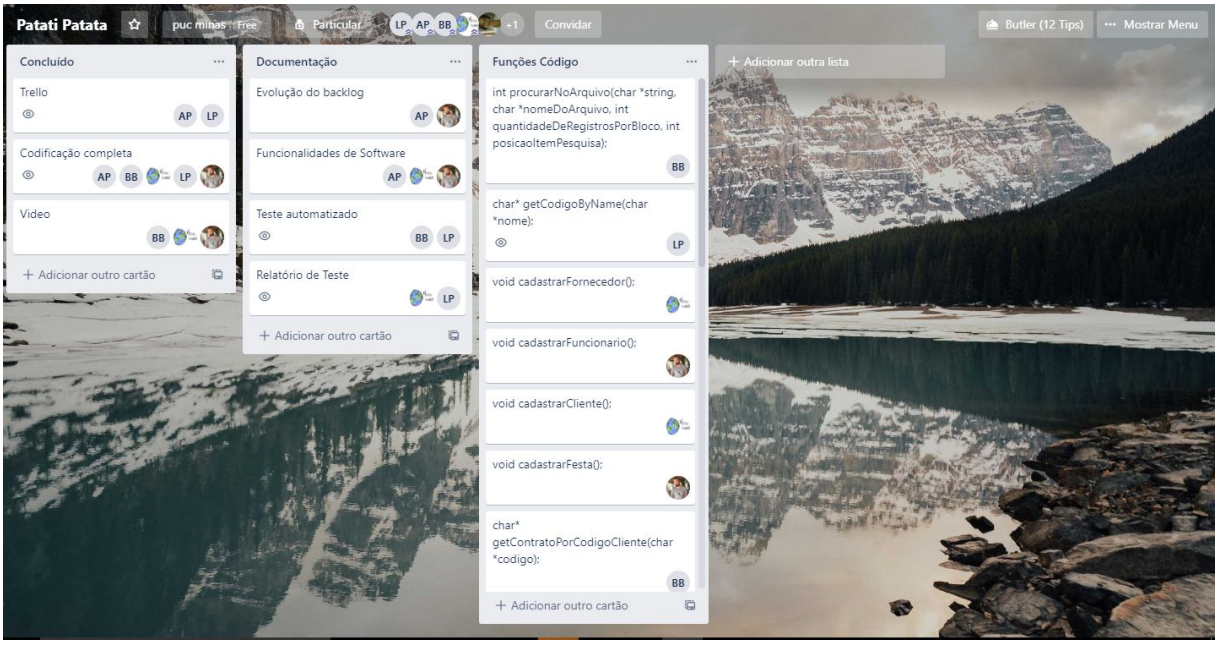

Figura 7 – Remanejo Trello.

3. Lista de assinaturas das funções e parâmetros

As funções utilizados no software foram:

3.1. Void cadastrarFornecedor()

Função para cadastrar os fornecedores no arquivo.

3.2. Void cadastrarFuncionario()

Função para cadastrar os funcionários no arquivo.

3.3. Void cadastrarCliente()

Função para cadastrar os clientes no arquivo.

3.4. Void cadastrarFesta()

Função para cadastrar festas no arquivo.

3.5. Char* diaSemanaFesta(int codigo)

Função para buscar o dia da festa cadastrado no arquivo, recebe como parâmetro o código da festa e retorna o dia da semana cadastrado para a festa.

3.6. Char* quantidadePessoasFesta(int codigo)

Função que procura a quantidade de convidados de uma determinada festa, recebe como parâmetro o código da festa e retorna à quantidade de convidados cadastrados

3.7. Int relatorioDia()

Função para mostrar o relatório de festas cadastradas de determinado dia no arquivo.

3.8. Void listarFestasCliente()

Função para verificar quantas e quais festas determinado cliente possui cadastrado no arquivo.

3.9. Void pesquisarClientes()

Função que busca os dados dos clientes cadastrados no arquivo.

3.10. Void pesquisarFuncionarios()

Função que busca os dados dos funcionários cadastrados no arquivo.

3.11. Void pesquisarFornecedores()

Função que busca os dados dos fornecedores cadastrados no arquivo.

3.12. Void cadastrarContrato()

Função para cadastrar o contrato no arquivo

3.13. Void alterarContrato()

Função para alterar o status do contrato de um cliente.

3.14. Int verificaContrato(int codigo)

Função para verificar se o contrato existe, possui o código do contrato como parâmetro

3.15. Char* getFestaPorData(char *data);

Função que recebe como parâmetro o dia da festa como ponteiro, que tem como finalidade identificar o registro da festa no dia passado e retornar os dados da festa criada

3.16. Char* getCodigoByName(char *nome);

Função que recebe como parâmetro o nome do cliente, que tem como finalidade identificar o código pelo nome e retorna o código, caso não abrir e fechar o arquivo é retornado Erro

3.17. Int procurarNoArquivo(char *string, char *nomeDoArquivo, int quantidadeDeRegistrosPorBloco, int posicaoItemPesquisa);

Função que recebe como parâmetro o dado pesquisado, o arquivo, a quantidade de dados do cliente e o tipo do dado, que tem como finalidade buscar um dado dentro de um arquivo e retorna o número 3 caso não haja o dado inserido.

3.18. Char* getContratoPorCodigoCliente(char *codigo);

Função recebe como parâmetro o código da festa, que tem como finalidade conseguir as informações do contrato por meio do código da festa e retorna as informações do contrato da festa do código.

3.19. Char* pesquisarPorNome(char* nome, char* arquivo, int quantidadeLinhasRegistro, int posicaoLinhaNome, int tipoConsulta);

Função recebe como parâmetros o nome, o arquivo, a quantidade de dados que o arquivo tem para cada usuário, a posição do nome no arquivo e o tipo de usuário que pode ser fornecedor, funcionário ou cliente, que tem como finalidade buscar os dados de um usuário e retorna “Erro” caso não consiga abrir o arquivo.

3.20. Void setLinha(int linha);

Função que recebe como parâmetro a linha do código cliente sob a festa determinada, que tem como finalidade listar as festas do cliente.

4. Testes

O processo de realização dos testes da solução desenvolvida está documentado na seção que se segue e traz o plano de testes e o registro dos testes realizados.

Plano e Registro de Testes

Os testes funcionais a serem realizados no aplicativos estão ilustrados na tabela a seguir e está disponível na Planilha de testes.

Entrada	**Classes Válidas**	**Resultado Esperado**	**Classes Inválidas**	**Resultado Esperado**
Código Cliente, funcionário, fornecedor e festa: Utilização de números e letras.	Código é válido se não houver nenhum cadastro semelhante.	Código válido, segue para as próximas etapas de cadastro de informações.	Código inválido se existir mais de um cliente com o mesmo código.	Mensagem avisando que já existe um cliente ou funcionário ou fornecedor ou festa com esse código e pede para inserir um novo código.

Tipo de funcionário: selecionar 1 para fixo ou 2 para temporário.	Número 1 ou 2 é válido.	Tipo de funcionário válido, segue para as últimas etapas de cadastro.	Tipo de funcionário inválido se for inserido algo diferente de 1 ou 2.	Mensagem avisando que deu erro e pede para inserir uma nova opção.
Quantidade de convidados: Utilização de números menores que 100 pessoas.	Número de convidado inferior a 100 pessoas é válido.	Número válido, segue para as outras etapas de cadastramento da festa.	Número inválido se a festa for superior a 100 pessoas.	Mensagem avisando que o número de convidados é superior a capacidade do estabelecimento e pede um número válido.
Data da festa: Inserir a informação em formato DD/MM/AAAA	Data da festa é válido se não houver festa no mesmo dia.	Data da festa válida, segue para as próximas etapas de cadastro.	Data da festa é inválido se já houver uma festa programada no mesmo dia.	Data da festa é inválida, solicitando uma nova data ao usuário.

Dia da festa: Inserir um número de 1 a 7 informando o dia da semana.	Dia da festa é válido se inserir números entre 1 e 7.	Dia da festa válida, segue para as próximas etapas de cadastro.	Dia da festa é inválido inserir número inferior a 1 e superior a 7.	Dia da festa é inválida, solicitando um novo dia ao usuário.

5. Video

A apresentação dinâmica do software, mostrando todos as funções está disponível para a visualização em :

- https://www.youtube.com/watch?v=O0Cj25djIKw