

# Trabalho Prático Interdisciplinar

Fundamentos da Engenharia de Software & Algoritmos e Estruturas de Dados I  
Engenharia de Software - Manhã  
**Luca Ferrari Azalim**

---

## Introdução

Neste documento, estão contidas as informações de planejamento e documentação do Trabalho Prático Interdisciplinar, cujo propósito é o desenvolvimento de um software de gestão para um salão de festas utilizando a linguagem de programação C.

## Links

Repositório no GitHub	<a href="https://github.com/lucaazalim/ti-salao-de-festas">https://github.com/lucaazalim/ti-salao-de-festas</a>
Repositório no Replit	<a href="https://replit.com/@LucaAzalim/ti-salao-de-festas">https://replit.com/@LucaAzalim/ti-salao-de-festas</a>
Vídeo de apresentação no YouTube	<a href="https://youtu.be/vg67sXewPeA">https://youtu.be/vg67sXewPeA</a>

# Planejamento

O processo ágil Scrum foi utilizado para gerir o planejamento do projeto. A tabela abaixo representa o **Backlog do Produto** e contém as funcionalidades a serem desenvolvidas no software, descritas em forma de **histórias de usuário**.

Backlog do Produto	
ID	História de Usuário
#1	Como usuário, eu preciso cadastrar um cliente
#2	Como usuário, eu preciso cadastrar um fornecedor
#3	Como usuário, eu preciso cadastrar uma festa
#4	Como usuário, eu preciso calcular o valor total da festa
#5	Como usuário, eu preciso atualizar o status de pagamento de uma festa
#6	Como usuário, eu preciso buscar um cliente ou fornecedor pelo ID ou pelo nome
#7	Como usuário, eu preciso buscar todas as festas de um cliente
#8	Como usuário, eu preciso buscar todas as festas de uma data

As funcionalidades e demais tarefas foram distribuídas em três sprints, como descrito na tabela abaixo.

Sprints			
ID	Início	Término	Backlog
#1	29/05/2023	31/05/2023	<ul style="list-style-type: none"><li>• Funcionalidade #1</li><li>• Funcionalidade #2</li><li>• Funcionalidade #3</li><li>• Funcionalidade #4</li></ul>
#2	01/06/2023	03/06/2023	<ul style="list-style-type: none"><li>• Funcionalidade #5</li><li>• Funcionalidade #6</li><li>• Funcionalidade #7</li><li>• Funcionalidade #8</li></ul>
#3	05/06/2023	09/06/2023	<ul style="list-style-type: none"><li>• Realização de testes caixa preta</li><li>• Implementação de testes caixa branca</li></ul>

# Estrutura de Arquivos

Os principais diretórios e arquivos do projeto estão descritos abaixo:

Estrutura de Arquivos	
/bin	Binários de execução do software e dos testes unitários.
/data	Diretório contendo os arquivos ".dat" responsável por armazenar os dados cadastrados pelo software.
/data/customer.dat	Arquivo binário responsável por armazenar os clientes cadastrados.
/data/party.dat	Arquivo binário responsável por armazenar as festas cadastradas.
/data/supplier.dat	Arquivo binário responsável por armazenar os fornecedores cadastrados.
/obj	Diretório de arquivos objeto (".o").
/src	Diretório contendo os arquivos do código-fonte do software.
makefile	Arquivo contendo um conjunto de diretivas usadas pela ferramenta de compilação para o software principal.
makefile-tests	Arquivo contendo um conjunto de diretivas usadas pela ferramenta de compilação para os testes unitários.
clear-data.sh	Shell script utilitário responsável por excluir os arquivos ".dat" do diretório /data.
run-tests.sh	Shell script que compila e executa os testes unitários.
run.sh	Shell script que compila e executa o software principal.

# Funcionalidades

Quando executado, o software exibe um menu com nove opções de funcionalidades:

Software de Gestão do Salão de Festas

1. Cadastrar cliente
2. Cadastrar fornecedor
3. Cadastrar festa
4. Calcular valor da festa
5. Alterar status do pagamento de uma festa
6. Pesquisar cliente ou fornecedor pelo ID
7. Relatório de festas por cliente
8. Relatório de festas por data
9. Sair

Informe uma opção [1 - 9]:

A tabela abaixo contém a descrição de cada uma das funcionalidades do software:

Funcionalidades		
Opção	Funcionalidade	Descrição
1	Cadastrar cliente	Solicita as informações necessárias para cadastro de um cliente e insere um novo cliente no arquivo <b>client.dat</b> .
2	Cadastrar fornecedor	Solicita as informações necessárias para cadastro de uma festa e insere um novo fornecedor no arquivo <b>supplier.dat</b> .
3	Cadastrar festa	Solicita as informações necessárias para cadastro de uma festa e insere uma nova festa no arquivo <b>party.dat</b> .
4	Calcular valor da festa	Solicita o ID de uma festa previamente criada e imprime o valor da festa, com base nas <b>tabelas 1 e 2</b> estabelecidas nos requisitos.
5	Alterar status do pagamento de uma festa	Solicita o ID de uma festa previamente criada e o novo status de pagamento e realiza a alteração no arquivo <b>party.dat</b> .
6	Pesquisar cliente ou fornecedor pelo ID ou nome	Solicita o tipo de entidade a ser pesquisada (cliente ou fornecedor) e a forma como deve ser pesquisada (ID ou nome). Por fim, imprime todos os dados de todas as entidades encontradas.
7	Relatório de festas por cliente	Solicita o ID de um cliente previamente cadastrado e imprime todas as informações das festas encontradas.
8	Relatório de festas por data	Solicita uma data e imprime todas as informações de todas as festas agendadas para esta data.
9	Sair	Encerra a execução do software.

# Casos de Teste

Os testes caixa branca foram implementados no arquivo "**src/tests/tests.c**", utilizando a biblioteca "munit". A tabela abaixo contém as informações de **entrada** e **saída esperada** de cada caso de teste:

Casos de Teste				
ID	Escopo	Entrada	Saída esperada	Resultado
#1	Criação de cliente	<b>Nome:</b> Luca Ferrari Azalim <b>Endereço:</b> Rua das Flores, 123 <b>Telefone:</b> 31912345678	1 (Sucesso)	Aprovado
#2	Criação de fornecedor	<b>Nome:</b> Casa do Sol <b>Telefone:</b> 31923456789	1 (Sucesso)	Aprovado
#3	Criação de festa	<b>ID do cliente:</b> <ID do cliente cadastrado pelo teste #1> <b>ID do fornecedor:</b> <ID do fornecedor cadastrado pelo teste #2> <b>Nº de convidados:</b> 50 <b>Data:</b> 27/02/2024 ( <i>Terça-feira</i> ) <b>Hora de início:</b> 19 <b>Tema:</b> Fundo do Mar <b>Parcelas:</b> 2	1 (Sucesso)	Aprovado
#4	Criação de festa com cliente inexistente	<b>ID do cliente:</b> 1000 ...	<b>0 (Erro)</b> → O cliente correspondente ao ID informado não existe.	Aprovado
#5	Cliente de festa com fornecedor inexistente	... <b>ID do fornecedor:</b> 1000 ...	<b>0 (Erro)</b> → O fornecedor correspondente ao ID informado não existe.	Aprovado

#6	Criação de festa em horários não permitidos aos sábados	... <b>Data:</b> 27/01/2024 ( <i>Sábado</i> ) <b>Hora de início:</b> 19 ...	<b>0 (Erro)</b> → Aos sábados, o horário de início das festas deve ser às 12h ou às 18h.	Aprovado
#7	Criação de festa em data e horário conflitante	... <b>Data:</b> 27/02/2024 ( <i>Terça-feira</i> ) <b>Hora de início:</b> 20 ...	<b>0 (Erro)</b> → Já existe uma festa marcada para essa data e horário.	Aprovado
#8	Cálculo do valor final da festa com pagamento à vista	<b>Dia da semana:</b> 6 ( <i>Sábado</i> ) <b>Convidados:</b> 50 <b>Parcelas:</b> 1	2069.1	Aprovado
#9	Cálculo do valor final da festa com pagamento em duas parcelas	<b>Dia da semana:</b> 0 ( <i>Domingo</i> ) <b>Convidados:</b> 100 <b>Parcelas:</b> 2	3799.05	Aprovado
#10	Busca de cliente pelo ID	<b>ID do cliente:</b> <ID do cliente cadastrado pelo teste #1>	Cliente não nulo	Aprovado