



# Capacitação Tecnológica em Engenharia e Desenvolvimento de Software

**D10 - Projeto Final**  
*2º Oferecimento NOV/DEZ 2022*

**Professores:** Prof. Dr. João Batista  
Prof. Dr. Paulo Cugnasca  
**Monitores:** João Campos  
Marcos D. Nascimento

*joaocamargo@usp.br*  
*cugnasca@usp.br*  
*joaogcampos@usp.br*  
*marcos.devaner@usp.br*

**Novembro de 2022**



# Agenda da Apresentação

- Tabela Resumo
- Dinâmica da Disciplina
- Grupos
  - Apresentação dos Alunos
- Navegação no Zoom
- Dúvidas?



# Tabela Resumo

Grupos	Alunos	Tema
1	Lucas Saraceni Carvalho	Calculadora
1	Pedro Henrique Machado Almeida	
2	Antonio Oliveira Pedroso da Cruz	Plataforma para assistir filmes/séries
2	Thiago Santin Parussolo	
3	Bruno Maia de Oliveira Duarte	Aplicativo de Chat Online
3	Matheus de Paula Silva	
4	Hector Shin Emura	Aplicativo para Controle de Finanças Pessoais
4	Hideki Goecking Hanaoka	
5	Caique Novaes Tsurumaki Pereira	Sistema de apostas esportivas para os jogos da Copa do Mundo 2022
5	Thiago Perroni Meletti	
6	Gabriel Galdino Gomes	Controle de Finanças Pessoais
6	Marcos Paulo Pacífico	
7	Gustavo Costa de Andrade	Calculadora
7	Luiz Eduardo Tojal Ramos dos Santos	
8	Alexandre Jun Tsubaki Oide	Aplicação para ajudar em estudo baseado na técnica de Pomodoro
8	Vitor Luiz Bortoloti Pereira	
9	Gabriel Barberini Rodrigues Carvalho	Controle de finanças pessoais
9	João Pedro Albiero Cezar	
10	Giovani Tavares de Andrade	Aplicação Desktop para controle de gastos/finanças pessoais
10	Pedro Augusto Cardoso Campos	
11	Luccas araujo benjamin	App para de previsão do tempo
11	Mario Guilherme Batista	
12	Gabriel Di Vanna Camargo	Calculadora
12	Matheus Ribeiro Lira	
13	Henrique Araújo De Carvalho	Controle de Finanças Pessoais
13	Henrique Eduardo dos Santos de Souza	
13	Milton Katuyiro Marubayashi Junior	



# Dinâmica da Disciplina

- Início = 15 min
- Apresentação Inicial dos Grupos ( $13 * 1,5$  a  $2,0$  min) = total aproximado de 25 min
- Trabalho em Grupos:
  - Conversa dos professores com os grupos (1 ou 2 vezes por aula)
  - Trabalho prático dos grupos até 21h30
- Apresentação Final dos Grupos ( $13 * 1,5$ min) = total aproximado de 25 min (com slides)
- Encerramento = 5 min

Aulas: (1) 16/11, (2) 23/11, (3) 30/11, (4) 07/12, (5) 14/12 (Apresentação Final e Demonstração)

Entrega final do Projeto de Conclusão de Curso: 21/12 (até 23h59)



# Relatórios Semanais

## **Estrutura do Documento (cumulativo):**

- Grupo
- Título
- Objetivo
- Descrição do projeto
- Requisitos funcionais
- Requisitos não funcionais
- Cronograma completo (início ao fim, atualizado)
- O que deu certo. O que deu errado. (para cada aula)
- Aprendizado (para cada aula)
- Auto-Avaliação individual (para cada aula)

**Entregar PDF até 23h59 do dia seguinte da aula (será passado “template”).**



# Entregas Finais

**Manual do Usuário**

**Vídeo de Demonstração do Produto: 4 a 5 min, formato mp4, 1080p.**

**Documentação Final:**

**Relatório adaptado 5o relatório semanal, incluindo:**

**Acesso ao GitHub**

**Auto-Avaliação Final**

**Máximo de 20 páginas**

**Formato PDF**

**Todas as entregas finais até as 23h59m da 2a feira posterior a última aula.**



# Apresentação Final

## Na apresentação final, dentre outros aspectos:

- Deixar claro o objetivo do projeto;
- Elencar os requisitos funcionais e não funcionais;
- Apresentar diagrama com visão geral da arquitetura (podem incluir *design* de classes, estrutura da base de dados, etc.);
- Listar se foi utilizado padrão de arquitetura de projeto e/ou padrão de *design*;
- Informar as linguagens de programação utilizadas;
- Informar as bibliotecas utilizadas;
- Informar se foram implementados testes unitários;
- Informar se foi implementado algum item relativo à segurança;
- Apresentar vídeo com o funcionamento da aplicação.



# Avaliação

**Avaliação continuada em cada aula:**

**Apresentação Inicial**

**Interações**

**Apresentação Final**

**Relatório (dia seguinte)**

**Produto Final:**

**Software funcionando, apresentado para banca na 5a aula**

**Vídeo**

**Relatório Final / Manual do Usuário**

**Documentação (GitHub)**

**OBS: Toda a documentação poderá ser entregue até 21/12 (até 23h59)**





# Grupo 1

Alunos:

Lucas Saraceni Carvalho  
Pedro Henrique Machado Almeida

---

## Objetivos

---

Esse projeto tem por objetivo a criação de uma aplicação de Calculadora para Windows, utilizando C#, e os frameworks WPF ou UWP ou WinUI3.0 e XAML.

---

## Requisitos

---

- Preferencialmente fazer uso do banco de dados SQLite para persistência de dados, quando aplicável.
- Preferencialmente utilizar um padrão de arquitetura em camadas, por exemplo MVC ou MVVM.
- Implementar botões para operações básicas: soma, subtração, divisão e multiplicação.
- Implementar botão para operação de raiz quadrada e porcentagem
- Implementar um botão para limpar os resultados.
- Implementar a exibição pelo menos dos últimos 5 cálculos, no formato exemplificado:  $X + Y = Z$ .
- Implementar a persistência em banco de dados dos últimos 5 cálculos. Ao abrir a aplicação, pelo menos os últimos 5 cálculos devem ser exibidos no formato exemplificado:  $X + Y = Z$ .
- Permitir cópia e cola de números.
- Permitir utilização de parênteses para indicar precedência de operações.
- Opcional: Implementar operações matemáticas: mod, exp,  $x^y$ , log,  $\pi$ ,  $n!$
- Opcional: implementar testes de unidade.



# Grupo 2

Alunos:

Antonio Oliveira Pedroso da Cruz  
Thiago Santin Parussolo

Plataforma para assistir filmes/séries



# Grupo 3

Alunos:

Bruno Maia de Oliveira Duarte  
Matheus de Paula Silva

Aplicativo de Chat Online



# Grupo 4

Alunos:

Hector Shin Emura  
Hideki Goeking Hanaoka

---

## Objetivos

---

Esse projeto tem por objetivo a criação de uma aplicação de controle de finanças pessoais para Windows, utilizando C#, e os frameworks WPF ou UWP ou WinUI3.0 e XAML.

---

## Requisitos

---

## Requisitos

---

- Preferencialmente fazer uso do banco de dados SQLite para persistência de dados, quando aplicável.
- Preferencialmente utilizar um padrão de arquitetura em camadas, por exemplo MVC ou MVVM.
- O aplicativo deve permitir a criação de categorias para os gastos.
- O aplicativo deve permitir o lançamento de gastos com os seguintes dados (data, valor, descrição e categoria).
- O aplicativo deve permitir a visualização do total de gastos por mês.
- O aplicativo deve permitir a visualização do total de gastos por categoria do mês.
- O aplicativo deve permitir a edição das categorias cadastradas.
- O aplicativo deve permitir a edição ou exclusão dos gastos lançados.
- Opcional: Permitir cadastrar cor e ícone por categoria.
- Opcional: Desenvolver um gráfico no formato de pizza que permita a visualização dos gastos por categoria e mês.
- Opcional: Desenvolver um gráfico no formato de barras verticais para comparação dos gastos entre meses e categoria.
- Opcional: Desenvolver um gráfico de linha que permita a visualização dos gastos por dia.
- Opcional: Implementar testes de unidade.



# Grupo 5

Alunos:

Caique Novaes  
Thiago Meletti

Sistema de apostas esportivas para os jogos da Copa do Mundo 2022



# Grupo 6

Alunos:

Gabriel Galdino Gomes  
Marcos Paulo Pacífico

---

## Objetivos

---

Esse projeto tem por objetivo a criação de uma aplicação de controle de finanças pessoais para Windows, utilizando C#, e os frameworks WPF ou UWP ou WinUI3.0 e XAML.

---

## Requisitos

---

## Requisitos

---

- Preferencialmente fazer uso do banco de dados SQLite para persistência de dados, quando aplicável.
- Preferencialmente utilizar um padrão de arquitetura em camadas, por exemplo MVC ou MVVM.
- O aplicativo deve permitir a criação de categorias para os gastos.
- O aplicativo deve permitir o lançamento de gastos com os seguintes dados (data, valor, descrição e categoria).
- O aplicativo deve permitir a visualização do total de gastos por mês.
- O aplicativo deve permitir a visualização do total de gastos por categoria do mês.
- O aplicativo deve permitir a edição das categorias cadastradas.
- O aplicativo deve permitir a edição ou exclusão dos gastos lançados.
- Opcional: Permitir cadastrar cor e ícone por categoria.
- Opcional: Desenvolver um gráfico no formato de pizza que permita a visualização dos gastos por categoria e mês.
- Opcional: Desenvolver um gráfico no formato de barras verticais para comparação dos gastos entre meses e categoria.
- Opcional: Desenvolver um gráfico de linha que permita a visualização dos gastos por dia.
- Opcional: Implementar testes de unidade.



# Grupo 7

Alunos:

Gustavo Costa de Andrade  
Luiz Eduardo Tojal

---

## Objetivos

---

Esse projeto tem por objetivo a criação de uma aplicação de Calculadora para Windows, utilizando C#, e os frameworks WPF ou UWP ou WinUI3.0 e XAML.

---

## Requisitos

---

- Preferencialmente fazer uso do banco de dados SQLite para persistência de dados, quando aplicável.
- Preferencialmente utilizar um padrão de arquitetura em camadas, por exemplo MVC ou MVVM.
- Implementar botões para operações básicas: soma, subtração, divisão e multiplicação.
- Implementar botão para operação de raiz quadrada e porcentagem
- Implementar um botão para limpar os resultados.
- Implementar a exibição pelo menos dos últimos 5 cálculos, no formato exemplificado:  $X + Y = Z$ .
- Implementar a persistência em banco de dados dos últimos 5 cálculos. Ao abrir a aplicação, pelo menos os últimos 5 cálculos devem ser exibidos no formato exemplificado:  $X + Y = Z$ .
- Permitir cópia e cola de números.
- Permitir utilização de parênteses para indicar precedência de operações.
- Opcional: Implementar operações matemáticas: mod, exp,  $x^y$ , log,  $\pi$ ,  $n!$
- Opcional: implementar testes de unidade.



# Grupo 8

Alunos:

Alexandre Jun Tsubaki Oide  
Vitor Luiz Bortoloti Pereira

Aplicação para ajudar em estudo baseado na técnica de Pomodoro





# Grupo 9

Alunos:

Marcos Paulo Pacifico  
Gabriel Galdino Gomes

## Objetivos

Esse projeto tem por objetivo a criação de uma aplicação de controle de finanças pessoais para Windows, utilizando C#, e os frameworks WPF ou UWP ou WinUI3.0 e XAML.

## Requisitos

## Requisitos

- Preferencialmente fazer uso do banco de dados SQLite para persistência de dados, quando aplicável.
- Preferencialmente utilizar um padrão de arquitetura em camadas, por exemplo MVC ou MVVM.
- O aplicativo deve permitir a criação de categorias para os gastos.
- O aplicativo deve permitir o lançamento de gastos com os seguintes dados (data, valor, descrição e categoria).
- O aplicativo deve permitir a visualização do total de gastos por mês.
- O aplicativo deve permitir a visualização do total de gastos por categoria do mês.
- O aplicativo deve permitir a edição das categorias cadastradas.
- O aplicativo deve permitir a edição ou exclusão dos gastos lançados.
- Opcional: Permitir cadastrar cor e ícone por categoria.
- Opcional: Desenvolver um gráfico no formato de pizza que permita a visualização dos gastos por categoria e mês.
- Opcional: Desenvolver um gráfico no formato de barras verticais para comparação dos gastos entre meses e categoria.
- Opcional: Desenvolver um gráfico de linha que permita a visualização dos gastos por dia.
- Opcional: Implementar testes de unidade.



# Grupo 10

Alunos:

Gabriel Barberini Rodrigues Carvalho  
João Albiero

---

## Objetivos

---

Esse projeto tem por objetivo a criação de uma aplicação de controle de finanças pessoais para Windows, utilizando C#, e os frameworks WPF ou UWP ou WinUI3.0 e XAML.

---

## Requisitos

---

## Requisitos

---

- Preferencialmente fazer uso do banco de dados SQLite para persistência de dados, quando aplicável.
- Preferencialmente utilizar um padrão de arquitetura em camadas, por exemplo MVC ou MVVM.
- O aplicativo deve permitir a criação de categorias para os gastos.
- O aplicativo deve permitir o lançamento de gastos com os seguintes dados (data, valor, descrição e categoria).
- O aplicativo deve permitir a visualização do total de gastos por mês.
- O aplicativo deve permitir a visualização do total de gastos por categoria do mês.
- O aplicativo deve permitir a edição das categorias cadastradas.
- O aplicativo deve permitir a edição ou exclusão dos gastos lançados.
- Opcional: Permitir cadastrar cor e ícone por categoria.
- Opcional: Desenvolver um gráfico no formato de pizza que permita a visualização dos gastos por categoria e mês.
- Opcional: Desenvolver um gráfico no formato de barras verticais para comparação dos gastos entre meses e categoria.
- Opcional: Desenvolver um gráfico de linha que permita a visualização dos gastos por dia.
- Opcional: Implementar testes de unidade.



# Grupo 11

Alunos:

Giovani Tavares de Andrade  
Pedro Augusto Cardoso Campos

---

## Objetivos

---

Esse projeto tem por objetivo a criação de uma aplicação de controle de finanças pessoais para Windows, utilizando C#, e os frameworks WPF ou UWP ou WinUI3.0 e XAML.

---

## Requisitos

---

---

## Requisitos

---

- Preferencialmente fazer uso do banco de dados SQLite para persistência de dados, quando aplicável.
- Preferencialmente utilizar um padrão de arquitetura em camadas, por exemplo MVC ou MVVM.
- O aplicativo deve permitir a criação de categorias para os gastos.
- O aplicativo deve permitir o lançamento de gastos com os seguintes dados (data, valor, descrição e categoria).
- O aplicativo deve permitir a visualização do total de gastos por mês.
- O aplicativo deve permitir a visualização do total de gastos por categoria do mês.
- O aplicativo deve permitir a edição das categorias cadastradas.
- O aplicativo deve permitir a edição ou exclusão dos gastos lançados.
- Opcional: Permitir cadastrar cor e ícone por categoria.
- Opcional: Desenvolver um gráfico no formato de pizza que permita a visualização dos gastos por categoria e mês.
- Opcional: Desenvolver um gráfico no formato de barras verticais para comparação dos gastos entre meses e categoria.
- Opcional: Desenvolver um gráfico de linha que permita a visualização dos gastos por dia.
- Opcional: Implementar testes de unidade.



# Grupo 12

Alunos:

Luccas Benjamin  
Mario Batista

---

## Objetivos

---

Esse projeto tem por objetivo a criação de uma aplicação Windows para visualização da previsão do tempo, utilizando C#, e os frameworks WPF ou UWP ou WinUI3.0 e XAML.

---

## Requisitos

---

- Preferencialmente fazer uso do banco de dados SQLite para persistência de dados, quando aplicável.
- Preferencialmente utilizar um padrão de arquitetura em camadas, por exemplo MVC ou MVVM.
- Para este projeto, não é obrigatório utilizar dados reais de localidade e clima. Os dados exibidos na aplicação podem estar previamente embarcados na aplicação.
- O aplicativo deve exibir informações da localidade do usuário (cidade e estado).
- O aplicativo deve exibir as temperaturas mínima, máxima e atual do dia.
- O aplicativo deve exibir a probabilidade de chuva do dia.
- O aplicativo deve exibir a umidade relativa do ar no dia, velocidade do vento, pressão, índice UV.
- O aplicativo deve exibir um gráfico da temperatura ao longo do dia.
- O aplicativo deve exibir uma lista com a previsão do tempo ao longo da semana (temperatura máxima, mínima e probabilidade de chuva).
- Opcional: Utilizar a localidade real do usuário.
- Opcional: Utilizar uma API para exibir os dados reais da localidade do usuário.
- Opcional: Permitir o cadastro de localidades preferidas para visualização da previsão do tempo.
- Opcional: Usar imagens para representar a situação atual do clima (ensolarado, nublado, chuvoso, etc)
- Opcional: implementar testes de unidade.



# Grupo 13

Alunos:

Gabriel Di Vanna Camargo  
Matheus Ribeiro Lira

---

## Objetivos

---

Esse projeto tem por objetivo a criação de uma aplicação de Calculadora para Windows, utilizando C#, e os frameworks WPF ou UWP ou WinUI3.0 e XAML.

---

## Requisitos

---

- Preferencialmente fazer uso do banco de dados SQLite para persistência de dados, quando aplicável.
- Preferencialmente utilizar um padrão de arquitetura em camadas, por exemplo MVC ou MVVM.
- Implementar botões para operações básicas: soma, subtração, divisão e multiplicação.
- Implementar botão para operação de raiz quadrada e porcentagem
- Implementar um botão para limpar os resultados.
- Implementar a exibição pelo menos dos últimos 5 cálculos, no formato exemplificado:  $X + Y = Z$ .
- Implementar a persistência em banco de dados dos últimos 5 cálculos. Ao abrir a aplicação, pelo menos os últimos 5 cálculos devem ser exibidos no formato exemplificado:  $X + Y = Z$ .
- Permitir cópia e cola de números.
- Permitir utilização de parênteses para indicar precedência de operações.
- Opcional: Implementar operações matemáticas: mod, exp,  $x^y$ , log,  $\pi$ ,  $n!$
- Opcional: implementar testes de unidade.



# Dúvidas?

# Dúvidas?