

# Guia de Execução – Desktop São Paulo

## Dicionário de Dados (90 pontos)

Para gerar o dicionário de dados vamos usar o seguinte script (o código abaixo está no git em dia 1):

```
SELECT
  case
    when lower(C.name) = 'id' then 'PK'
    when lower(C.name) like 'id%' then 'FK'
    else ''
  end AS 'CHAVE',
  C.name AS 'NOME COLUNA',
  TY.name AS 'TIPO',
  C.max_length AS 'Tamanho',
  C.is_nullable AS 'ACEITA NULL',
  C.is_identity AS 'IS IDENTITY'
FROM sys.columns C
INNER JOIN sys.tables T ON T.OBJECT_ID = C.OBJECT_ID
INNER JOIN sys.types TY ON TY.user_type_id = C.user_type_id
WHERE T.name = 'demo'
```

No lugar de demo pode ser colocado o nome da tabela desejada. Se você executar no SQL Server e copiar + colar no template obterá o resultado desejado. Você precisa disso para todas as tabelas.

Para isso você precisará rodar o código SQL que foi disponibilizado nos Data Files.

## Regras de Negócio (125 pontos)

Você precisa fazer uma Trigger complexa, os deletes da trigger dão 50 pontos e o resto concede 70 pontos. A entrega concede 5 pontos. Entregar o trigger.sql e um arquivo de instruções da trigger dizendo que o histórico está em:

"C:\Users\Public\Documents\historico.txt".

O código abaixo mostra a implementação completa da Trigger e está disponível na pasta "códigos dia 1" no github para consulta.

---

```
CREATE TRIGGER limpaServer ON Servidores
AFTER DELETE AS
BEGIN
    DECLARE @AnexCount INT;
    DECLARE @MesaCount INT;
    DECLARE @PartCount INT;
    DECLARE @Name VARCHAR(200);
    DECLARE @Cmd VARCHAR(2000);

    DELETE FROM MensagemAnexo
    WHERE IdMensagem IN (
        SELECT Id
        FROM Mensagem
        WHERE IdServidor IN (SELECT Id FROM deleted)
    );
    SET @AnexCount = @@ROWCOUNT;

    DELETE FROM Mensagens WHERE IdServidor IN (SELECT Id FROM deleted);
    SET @MesaCount = @@ROWCOUNT;

    DELETE FROM ParticipantesServidor WHERE IdServidor = old.Id;
    SET @PartCount = @@ROWCOUNT;

    SELECT TOP 1 @Name = Nome
    FROM deleted

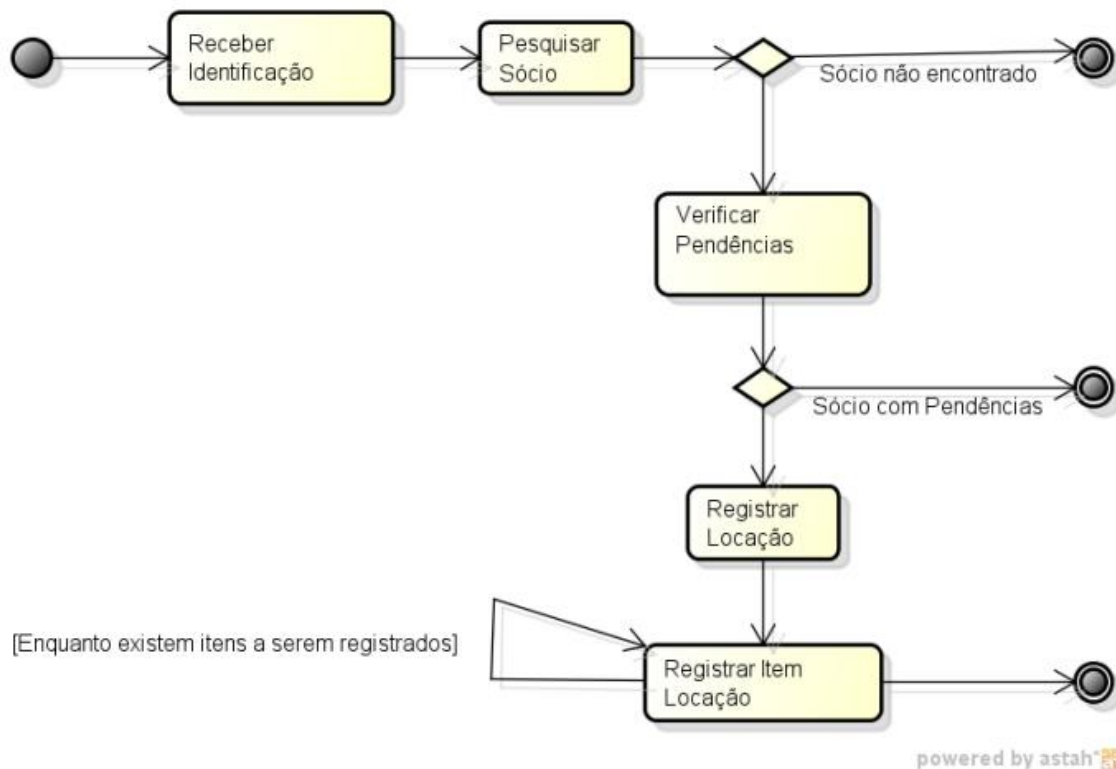
    SET @Cmd = 'Echo "' +
        CONVERT(VARCHAR(200), GETDATE()) + " - " +
        @Name + " - " +
        @PartCount + " - " +
        @MesaCount + " - " +
        @AnexCount +
        '" >> C:\Users\Public\Documents\historico.txt';

    EXEC xp_cmdshell @Cmd;
```

## Diagrama de Atividade (85 pontos)

Tarefa importante e não muito custosa. Não vale a pena fazer todos os casos de validação. Vamos focar obter nota 2 de julgamento, ou seja, além de atender os padrões UML: "O diagrama contém atividades, para preenchimento dos dados, inclusão de membros e envio do convite."

Abaixo um exemplo de diagrama de atividades. Ele tem todos os recursos que você precisará implementar. Separe em blocos as atividades. Use os círculos pretos e pretos com círculo para início e fim. Faça cada paço em quadrados e "ifs" ou seja, condicionais com losangos.



Faça o Setup do Banco e da Aplicação para começar a ganhar pontos de desenvolvimento

Faça uma tela principal de qualquer modo. Seguir o layout da poucos pontos por muito esforço.

Nossa jogada principal será: Vamos implementar o discord sem mensagem alguma. Criar servidores, sistema de convites, membros, login, pesquisar usuários, ver servidores e membros. Mas vamos ignorar completamente o sistema de mensagem (a não ser que sobre tempo). É possível obter até 700 pontos com essa técnica. (se sobrar tempo para fazer coisas de layouts e usar o guia de estilo, podemos ganhar mais 100 pontos).

Cadastro de Servidores (50 pontos)

Pode iniciar pelo cadastro de servidores. Não precisa de login necessariamente, mas depois precisaremos arrumar o Id do Proprietário (o banco de dados começa com alguns dados, pode usar id 1 sem problemas. Talvez até pra sempre).

Todos os usuários poderão criar um servidor, a validação para aluno vale 5 pontos.

Os 50 pontos são sem precisar do sistema de convites.

### Sistema de Login (60 pontos)

### Vendo servidores e usuários (125 pontos)

Ordenações bobas e algumas queries legais. São pontos bons para conseguir. Basta mostrar os dados dos servidores e mensagens diretas como pedido no projeto teste.

### Sistema de Convite (80 pontos)

Improvise o sistema de convites sem ser na por mensagem direta, podemos ganhar 80 pontos sem fazer sistema de mensagens.

Instalador que cria o banco de dados (40 pontos) [IMPORANTE: VAMOS EVITAR PROBLEMAS COM OS AVALIADORES E FAZER ISSO DIREITO]

Vamos fazer um fake instaler. A ideia é muito simples, ao ser executado o programa irá apresentar uma mensagem de "Programa instalado com sucesso" criando um atalho dele mesmo na área de trabalho, caso já exista, o programa executa normalmente.

1. Vamos nos basear no seguinte trecho do roteiro de teste: **Gerou o instalador do projeto. Rodar o instalador e verificar se funcionou (ou abriu o projeto automaticamente, ou ao clicar no link da área de trabalho ou do menu, abriu a aplicação)** Assim é impossível diferenciar, basta dizermos que realmente instalamos a aplicação (o instalador precisará estar com todas as dependências na mesma pasta). Abaixo o código do fake installer que estará também no git.

```
using System.IO;
using System.Reflection;
using System.Diagnostics;

// Pega a pasta atual
var assembly = Assembly.GetEntryAssembly();
string crrPath = Path.GetDirectoryName(assembly.Location);

if (crrPath == "C:\\Users\\Public\\AppData")
{
    // Roda aplicação normalmente
    Console.WriteLine("Minha aplicação incrível");
    Console.ReadKey(true);
    return;
}

// Faz uma operação qualquer criando o banco de dados

var dir = new DirectoryInfo(crrPath);
foreach (FileInfo file in dir.GetFiles())
{
    file.CopyTo(Path.Combine("C:\\Users\\Public\\AppData", file.Name), true);
}

Console.WriteLine("Aplicação instalada");
Process.Start("C:\\Users\\Public\\AppData\\fakeinstaller.exe");
```

Servidores e Membros em tempo Real (20 pontos)

Pesquisar Usuários (60 pontos)