



SÃO  
PAULO  
TECH  
SCHOOL

# Engenharia de Software

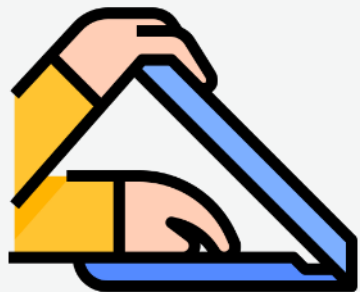
## Arquitetura de Software

Aula 10 – Correção da Atividade em Sala

**Fábio Figueredo**

fabio.figueredo@sptech.school

# Regras básicas da sala de aula



1. **Notebooks Fechados:** Aguarde a liberação do professor;
2. Celulares em modo **silencioso e guardado**, para não tirar sua atenção
  - Se, caso haja uma situação urgente e você precisar **atender ao celular**, peça licença para sair da sala e atenda fora da aula.



3. **Proibido usar Fones de ouvido:** São liberados apenas com autorização do professor.

4. **Foco total no aprendizado**, pois nosso tempo em sala de aula é precioso.

- Venham sempre com o **conteúdo da aula passada em mente** e as atividades realizadas.
- **Evitem faltas** e **procure ir além** daquilo que lhe foi proposto.
- **Capricho, apresentação e profundidade** no assunto serão observados.
- **“frequentar as aulas e demais atividades curriculares aplicando a máxima diligência no seu aproveitamento”** (Direitos e deveres dos membros do corpo discente - Manual do aluno, p. 31)



# Regras básicas da sala de aula



As aulas podem e devem ser divertidas! Mas:

- **Devemos respeitar uns aos outros** – cuidado com as brincadeiras.
  - “observar e cumprir o regime escolar e disciplinar e comportar-se, dentro e fora da Faculdade, **de acordo com princípios éticos condizentes**” (Direitos e deveres dos membros do corpo discente – Manual do aluno, p. 31)

# Boas práticas no Projeto

## COMPROMISSO



COM VOCÊ:  
**ARRISQUE**, NÃO  
TENHA MEDO DE  
ERRAR



COM OS  
PROFESSORES:  
ORGANIZE A **ROTINA**  
PARA OS ESTUDOS

COM OS COLEGAS:  
**PARTICIPAÇÃO**  
**ATIVA** E PRESENTE



COM O PROJETO:  
**RESPEITO** E  
FLEXIBILIDADE

  
*Respeito*

# Boas práticas no Projeto

Reações **defensivas** não levam  
ao envolvimento verdadeiro!

Transforme cada problema e  
cada dificuldade em uma  
**OPORTUNIDADE** de aprendizado  
e crescimento.

## EVITE:

- Justificativas e Desculpas
- Transferir a culpa
- Se conformar com o que sabe
- Se comparar com o outro

# Dica: **Como ter sucesso** (Maiores índices de aprovações)

## Comprometimento

- Não ter faltas e atrasos. Estar presente (*Não fazer 2 coisas ao mesmo tempo*)
- Fazer o combinado cumprindo os prazos

## Atitudes Esperadas:

- **Profissionalismo**: Entender que não é mais ensino médio (*Atitude, comportamento, etc.*)
- **Não estar aqui só pelo** estágio ou pelo diploma
- Não ficar escondido: precisa **experimentar**
- **Trabalhar** em grupo e **participar** na aula
- **Não ser superficial** ou “achar que sabe”
- **Não se enganar** utilizando de “cola”
- Assumir a responsabilidade: Não colocar a culpa em outra coisa. **Não se vitimizar.**

# Avaliações

**Socioemocional: Binária** (reprova ou aprova). *Feedbacks durante o semestre.*

**Pesquisa e Inovação: Binária** (reprova ou aprova). *Feedback no final de cada Sprint.*

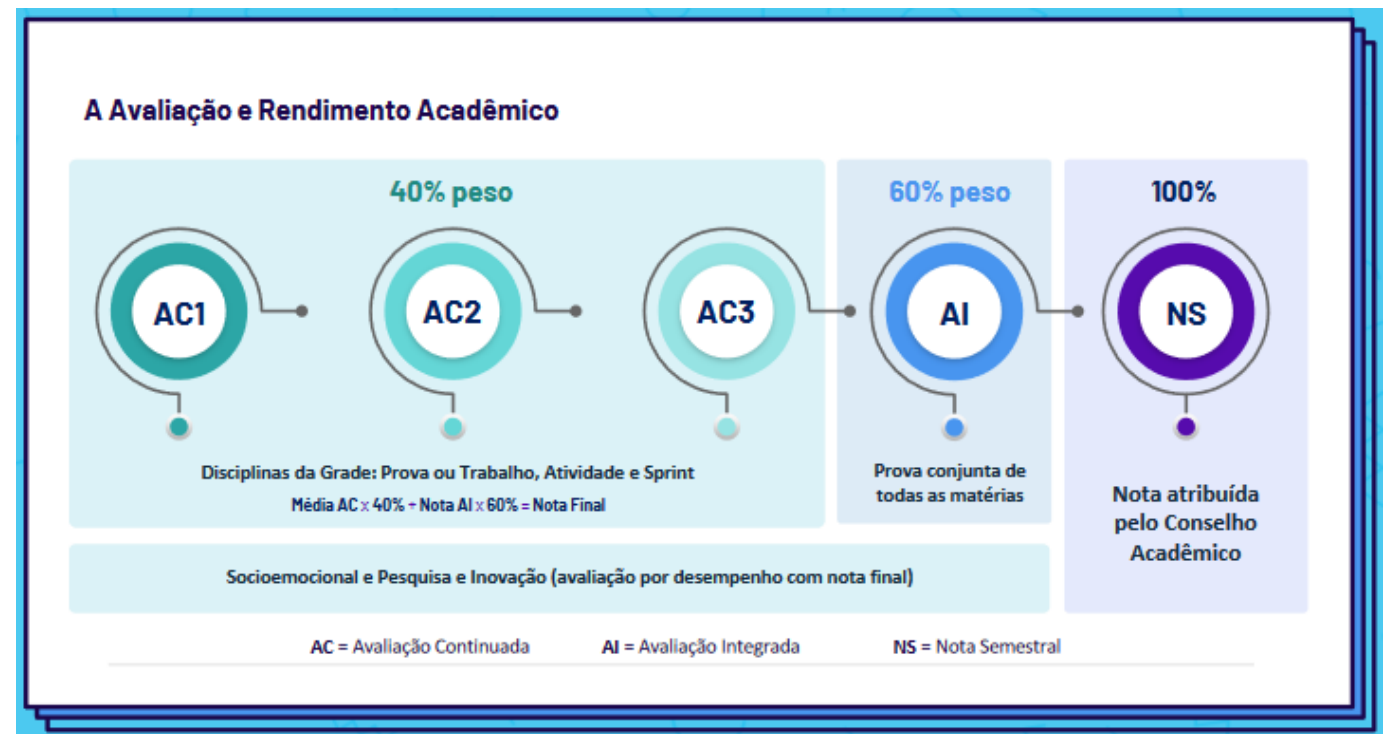
**Média = 6**

## 40% da Nota – Continuadas

*1 Continuada por Sprint com possibilidade de inspeção Individual (São 3 continuadas)*

## 60% da Nota

*1º Semestre = Projeto Individual ;  
Demais semestres: Avaliação Integrada*



Manual do Aluno





Nosso objetivo:

**Aprender/Ensinar processos,  
métodos e ferramentas para  
construção e manutenção de  
softwares profissionais.**



Palavra-chave dessa Sprint:

# PRAGMATISMO

## **prag·má·ti·co**

adjetivo

1. Relativo à pragmática ou ao pragmatismo.
2. Que tem motivações relacionadas com a .ação ou com a eficiência. = PRÁTICO

adjetivo e substantivo masculino

3. Que ou quem revela um sentido prático e sabe ou quer agir com eficácia.

# Nosso Caminho

Legenda: Conteúdo / **Entregável PI** / Onde Estamos



S3

## Final do Semestre

- Qualidade e Testes
- Processos de Software
- Apresentação PI
- Avaliação Integrada

- **Entregável Sprint 3**

**Entrega: 19/05/2022**

S2

- Design de Interfaces Web
- Projeto de Software
- Arquitetura de Software

- **Desenho de Arquitetura**
- **Protótipo em Alta Resolução**
- **Diagrama de Solução de Software - Container**
- **Planilha de Arquitetura**

**PROVA: 14/04/2022 | Entrega: 25/04/2022**

S1

- Apresentação
- UI/UX
- Fatores Humanos
- Design de Interação
- Design de Interfaces + Bootcamp

- **Jornada do Usuário**
- **Prototipação das Telas**

**Entrega: 03/03/2022**

The background is a dark, monochromatic abstract composition. It features a series of thin, light-colored wavy lines that flow across the frame, creating a sense of movement and depth. Interspersed among these lines are numerous small, out-of-focus light points, resembling bokeh or distant stars, which add a sparkling texture to the overall design.

**O QUE CAI NA PROVA!**

# Continuada 2 – 14/04

## O QUE CAI:

- UI e UX
- Fatores Humanos
- Design para Web
- Desenho de Arquitetura

The background is a dark, textured surface with intricate, wavy, light-colored lines that create a sense of depth and movement. Scattered throughout are numerous small, out-of-focus light spots, resembling bokeh or distant stars, which add to the ethereal and abstract quality of the image.

**Exercício para Praticar!**

# Passo a passo – Desenho de Arquitetura

1. Identificar os Objetivos da Arquitetura
2. Cenários Chave
  1. O que é crítico para o negócio?
  2. O que gera alto impacto?
3. Fazer a visão global (overview) da Aplicação
  1. Determinar o tipo da sua aplicação (WEB, Mobile, etc)
  2. Identificar as restrições no desenvolvimento (Rede, Segurança, Sistema Operacional)
  3. Identificar estilos importantes de arquitetura (Camadas, SOA) – Vamos ver mais a frente.
  4. Determinar as tecnologias relevantes (Spring, Node.JS)
4. Desenhar no quadro ou folha de papel
5. Identificar os assuntos chaves (Key Issues: Qualidade, Deploy, Execução, Usabilidade)
6. Cuidar dos itens Transversais (Caching, Comunicação, Autenticação, etc).

# Tópicos da Aula

- Correção exercício da aula passada
- Liberação da correção dos cases enviados para treino
- Apoio aos grupos



The background is a dark, monochromatic abstract composition. It features a series of thin, white, wavy lines that flow and swirl across the frame, creating a sense of movement and depth. Interspersed among these lines are numerous small, out-of-focus white circles, resembling bokeh or distant stars, which add to the ethereal and cosmic feel of the image. The overall effect is one of dynamic yet graceful motion.

**CORREÇÃO!**

# Exercício em Classe

## Migração de Sistema escrito em Visual Basic 6.0

Um Supermercado cresceu muito, saiu de 2 lojas para 30 lojas e contratou sua empresa para um projeto de reescrever o aplicativo existente em uma nova linguagem, utilizando linguagens e framework modernos.

A atual aplicação utiliza a arquitetura de Cliente -> Servidor, sendo que a aplicação são executáveis que rodam no Windows XP e a base de dados é em SQL Server 6.5. São 3 caixas por loja e mais 2 máquinas administrativas, o cliente tem como foco utilizar soluções nas estações que não tenham custo de licenciamento. Existem 4 executáveis na aplicação:

- **Frente de Caixa:** Faz interface com vários dispositivos como por exemplo: impressora fiscal, leitor de código de barras.
- **Módulo de Suprimentos:** Tem o cadastro dos produtos, das compras, dos recebimentos e tem um leitor de código de barra.
- **Módulo de Gerenciamento:** Contém os cadastros. módulos de controle de estoque e de exportação de dados para o sistema Financeiro.
- **Módulo de Relatórios:** Relatórios operacionais que podem ser exportados ou impressos.

O cliente já comprou os computadores novos que vão rodar nos caixas, todos vieram com Linux, ele também já comprou as impressoras fiscais que se comunicarão com os computadores via USB, assim como os leitores de códigos de barra. A empresa já tem um grupo de desenvolvedores e você precisará repassar conhecimento para o time, assim como tudo que será desenvolvido.

O cliente sonha em poder acessar o módulo de gerenciamento da casa dele e em fazer análises em Dashboards e também gostaria que o Site Institucional recebesse uma atenção!

# Exercício em Classe

**Grupo 1:** Front-end Frente de Caixa – Desempenho, Segurança, Usabilidade e Escalabilidade;

**Grupo 3:** Módulo de Suprimentos– Desempenho, Usabilidade e Segurança;

**Grupo 4:** Módulo de Relatórios –Integridade das informações

**Grupo 5:** Back-end – Compatibilidade, Segurança, Disponibilidade e Escalabilidade;

**Grupo 6:** Back-end – APIs de integração com Sistemas Legados – Segurança, Reusabilidade e Rastreabilidade;

**Grupo 7:** Banco de Dados – Compatibilidade, Portabilidade, Segurança e Disponibilidade;

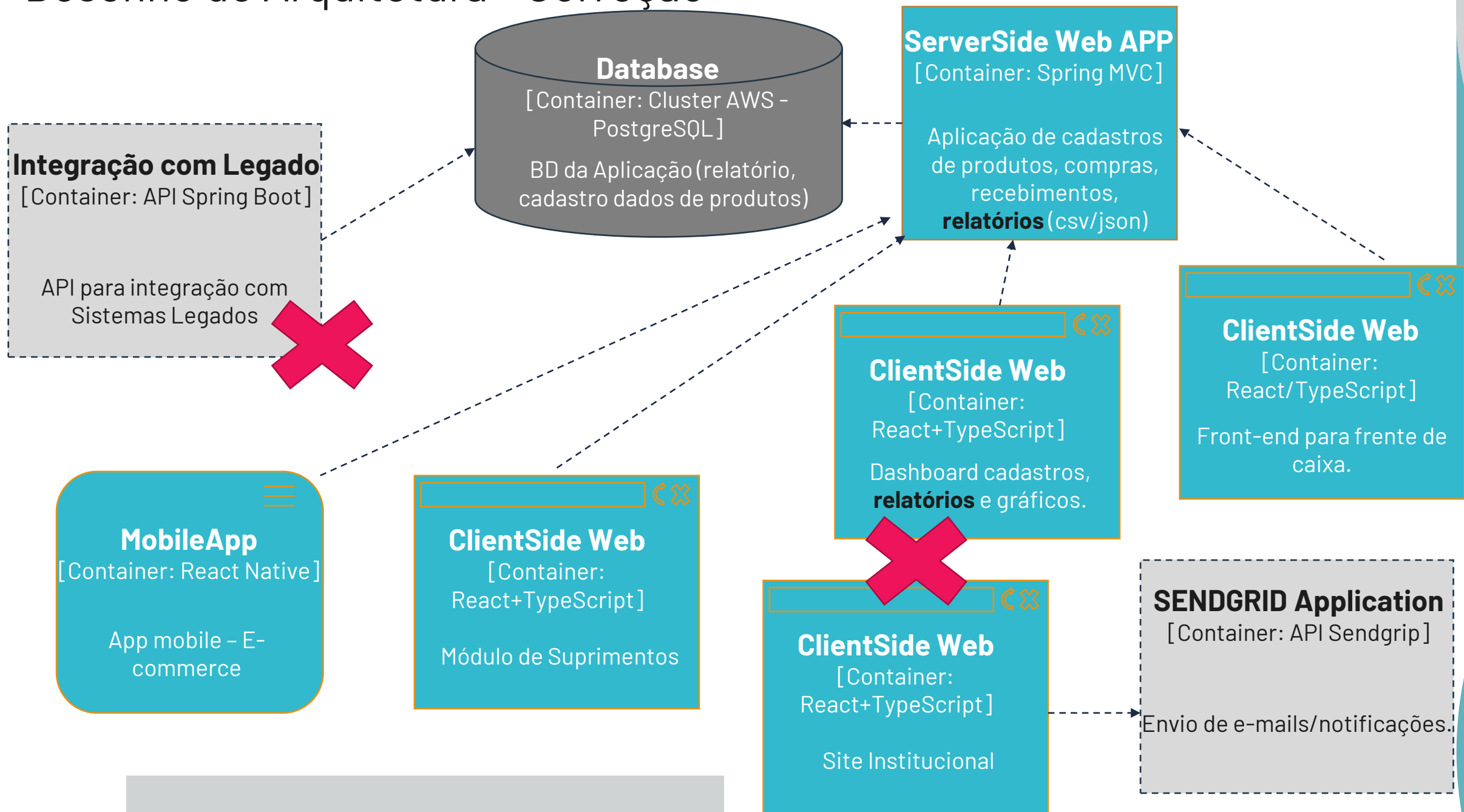
**Grupo 8:** Dashboard – Usabilidade, Segurança;

~~**Grupo 2:** Aplicativo das aplicações Desktops – Usabilidade, Confiabilidade e Escalabilidade.~~

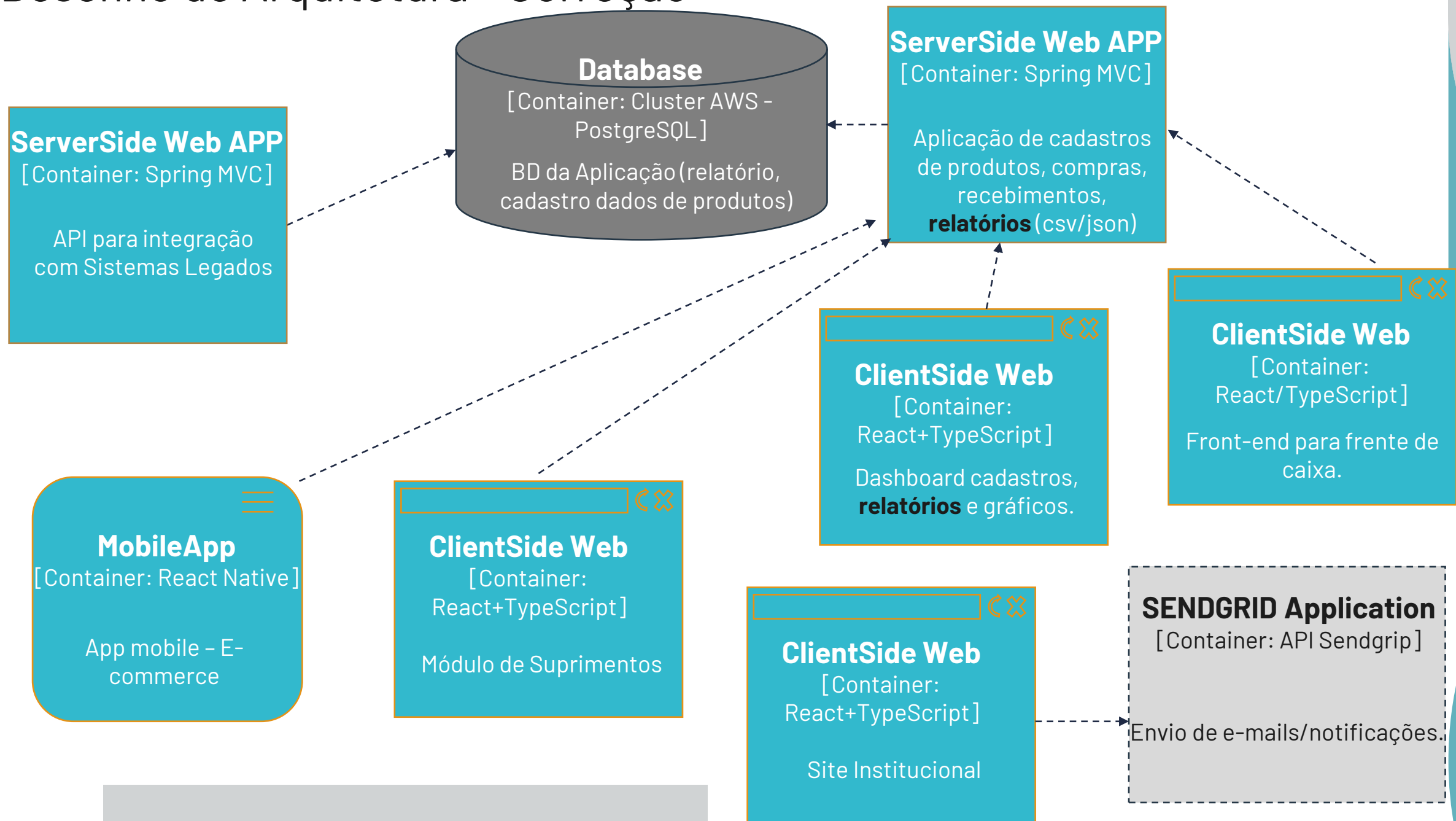
**Grupo 2:** APP Mobile – Usabilidade, Confiabilidade e Escalabilidade.

**Grupo 9:** Site Institucional – Usabilidade e Manutenibilidade.

# Desenho de Arquitetura - Correção



# Desenho de Arquitetura - Correção



# Exercício em Classe

## Migração de Sistema escrito em Visual Basic 6.0

Um **Supermercado** cresceu muito, **saiu de 2 lojas para 30 lojas** e contratou sua empresa para um projeto de reescrever o aplicativo existente em uma nova linguagem, utilizando linguagens e framework modernos.

A atual aplicação utiliza a arquitetura de **Cliente -> Servidor**, sendo que a aplicação são executáveis que **rodam no Windows XP** e a base de dados é em **SQL Server 6.5**. São 3 caixas por loja e mais 2 máquinas administrativas, **o cliente tem como foco utilizar soluções nas estações que não tenham custo de licenciamento**. Existem 4 executáveis na aplicação:

- **Frente de Caixa:** Faz interface com vários dispositivos como por exemplo: impressora fiscal, leitor de código de barras.
- **Módulo de Suprimentos:** Tem o cadastro dos produtos, das compras, dos recebimentos e tem um leitor de código de barra.
- **Módulo de Gerenciamento:** Contém os cadastros. módulos de controle de estoque e de exportação de dados para o sistema Financeiro.
- **Módulo de Relatórios:** Relatórios operacionais que podem ser exportados ou impressos.

O cliente já comprou os computadores novos que vão rodar nos caixas, todos vieram com Linux, ele também já comprou as impressoras fiscais que se comunicarão com os computadores via USB, assim como os leitores de códigos de barra.

**A empresa já tem um grupo de desenvolvedores e você precisará repassar conhecimento para o time, assim como tudo que será desenvolvido.**

O cliente sonha em poder acessar o módulo de gerenciamento da casa dele e em fazer análises em Dashboards e também gostaria que o Site Institucional recebesse uma atenção!

# Banco de Dados

**Grupo 7: Banco de Dados** – Compatibilidade, Portabilidade, Segurança e Disponibilidade;

**Trecho do case:** A atual aplicação utiliza a arquitetura de Cliente -> Servidor, sendo que a aplicação são executáveis que rodam no Windows XP e a base de dados é em SQL Server 6.5.



**Compatibilidade:** Neste contexto, uma opção SQL é mais compatível que uma opção Oracle. SQL Server é compatível com a tecnologia anterior.

**Portabilidade:** Possibilidade de mudar o SGBD quando necessário.

**Segurança:** Criptografia.

**Disponibilidade:** Redundância (pode ser em localidades diferentes) – Replicação ou Mirroring (Espelhamento).

# Back-end – Integração Legado

**Grupo 6: Back-end** – APIs de integração com Sistemas Legados – Segurança, Reusabilidade e Rastreabilidade;

**Trecho do case:** ...

## Microservice

[Container: .Net Core]

APIs de integração com sistemas legados

## Integração Log Application

[Container: API Log]

API para integração com ferramenta para armazenamento e análise de logs

**Segurança:** Autenticação (token, jwt, oauth, https).

**Reusabilidade:** API facilita o reuso, outros sistemas podem chamar.

**Rastreabilidade:** Logs ou ferramentas específicas para rastreamento – Splunk.



# Front-End Frente de Caixa

**Grupo 1: Front-end Frente de Caixa** – Desempenho, Segurança, Usabilidade e Escalabilidade;

**Trecho do case:** O cliente já comprou os computadores novos que vão rodar nos caixas, todos vieram com Linux, ele também já comprou as impressoras fiscais que se comunicarão com os computadores via USB, assim como os leitores de códigos de barra.

## Client Side Desktop

[Container: C# .NET]

**Frente de Caixa -**  
Sistema instalado nos  
caixas do supermercado

**Desempenho:** Máquinas com configuração apropriada. Como deve ser executado localmente, não depende da rede.

**Segurança:** Autenticação através do usuário logado na máquina, abertura da aplicação automaticamente no login.

**Usabilidade:** Java Swing. Dado o contexto, web não é uma boa opção.

**Escalabilidade:** Como a aplicação é instalada na máquina, conforme aumenta a quantidade de máquinas, basta instalar a aplicação. Porém, os demais containers precisam atender o aumento da demanda (BD, API, etc).

# Módulo de Suprimentos

**Grupo 3: Módulo de Suprimentos** – Desempenho, Usabilidade e Segurança;

**Trecho do case: ...**

## **Client Side Desktop**

[Container: C# .NET]

### **Módulo de Suprimentos -**

Cadastro de Compras e recebimentos através do leitor de códigos de barra

**Desempenho:** Máquinas com configuração apropriada. Como deve ser executado localmente, não depende da rede.

**Usabilidade:** Java Swing. Dado o contexto, web não é uma boa opção.

**Segurança:** Autenticação através do usuário logado na máquina, abertura da aplicação automaticamente no login.

# Módulo de Relatórios

**Grupo 4: Módulo de Relatórios** – Integridade das informações; – OPERACIONAL / DASHBOARD – Tático e estratégico

**Trecho do case: ...**

## MicroService/ Server Side Console APP

[Container: .Net Core/Phyton]

**Relatórios** – Extrair os dados tratados para os Relatórios.

**Integridade das Informações:** Check SUM – Validar se a soma dos valores inseridos “batem” .

Por exemplo, em importação de arquivos o HEADER pode informar quantas linhas existem no arquivo.

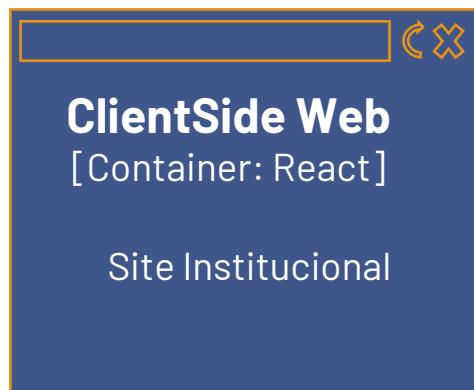
Avaliar periodicamente a integridade dos dados, pode ser uma equipe responsável por essa atividade.

Sanitizar, ou seja, verificar arquivos antes de processá-los.

# Site Institucional

**Grupo 9: Site Institucional** – Usabilidade e Manutenibilidade.

**Trecho do case:** O cliente sonha em poder acessar o módulo de gerenciamento da casa dele e em fazer análises em Dashboards e também gostaria que o Site Institucional recebesse uma atenção!



**Usabilidade:** Utilizar tecnologias que ofereçam funcionalidades e componentes necessários para uma boa experiência de usuário (componentes com funcionamento conhecido pelo usuário).

**Manutenibilidade:** Permitir que o próprio usuário administrativo, realize a atualização dos conteúdos do site. Por exemplo, utilizar um CMS.

# Back-end

**Grupo 5: Back-end** – Compatibilidade, Segurança, Disponibilidade e Escalabilidade;

**Trecho do case: ...**

## Server Side Console Application

[Container: .NET Core]

Integração entre app dos caixas com banco de dados.

## Database Local

[Container: SQL Server]

Armazena localmente os dados das transações

**Compatibilidade:** Selecionar linguagem compatível com a solução.

**Segurança:** Autenticação das APIs e criptografia dos dados trafegados

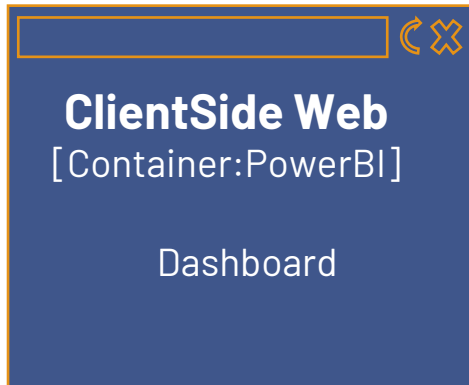
**Disponibilidade:** Armazenamento das transações local para o caso de indisponibilidade do ambiente. Banco de dados local ou controle de filas.

**Escalabilidade:** Como armazenamento local, o aumento das transações serão tratadas localmente.

# Dashboard

**Grupo 8: Dashboard** – Usabilidade, Segurança;

**Trecho do case:** O cliente sonha em poder acessar o módulo de gerenciamento da casa dele e em fazer análises em Dashboards e também gostaria que o Site Institucional recebesse uma atenção!



**Usabilidade:** PowerBI oferece uma interface amigável para consulta e análise de informações. É um software de mercado próprio para essa função. Pode-se utilizar alternativas opensource como: Redash, Metabase, Google Data Studio.

**Segurança:** Acesso restrito a usuários cadastrados. Acesso ao banco de dados somente pela aplicação.

# APP Mobile

**Grupo 2: APP Mobile** (Antes era Aplicativo das aplicações Desktops) – Usabilidade, Confiabilidade/Segurança e Escalabilidade.

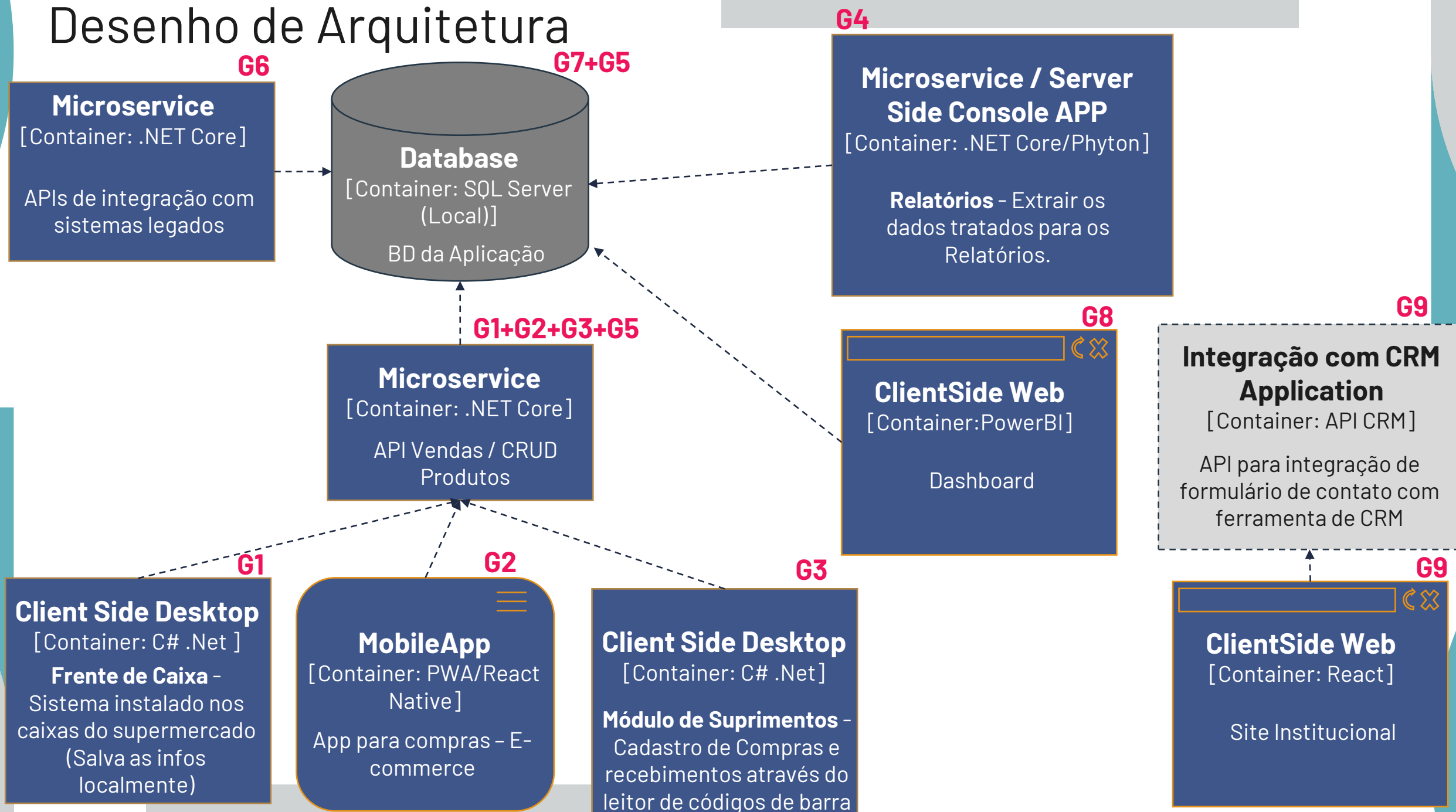


**Usabilidade:** Utilizar tecnologias que ofereçam funcionalidades e componentes necessários para uma boa experiência de usuário (componentes com funcionamento conhecido pelo usuário).

**Confiabilidade/Segurança:** Acesso restrito a usuários logados.

**Escalabilidade:** API deve ser escalável para suportar volumes de acessos nas sazonalidades.

# Desenho de Arquitetura





**Agradeço**  
a sua atenção!

**Fábio Figueredo**

fabio.figueredo@sptech.school

SÃO  
PAULO  
TECH  
SCHOOL