**GERENCIAMENTO DE PROJETO – SCRUM**

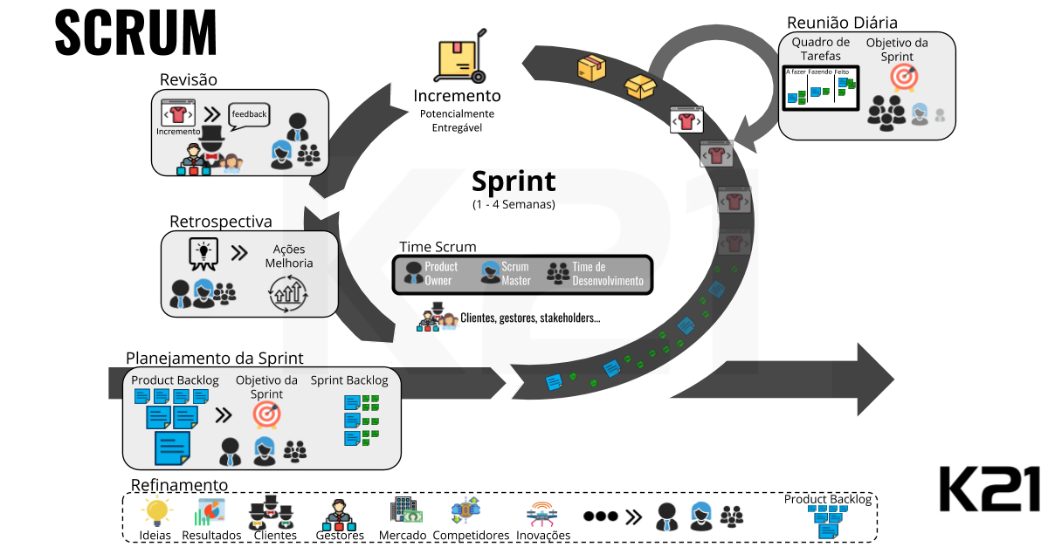
Scrum é uma metodologia ágil para gestão e planejamento de projetos de software.

No Scrum, os projetos são devidos em ciclos (tipicamente mensais) chamados de Sprints. O Sprint representa um Time Box dentro do qual um conjunto de atividades deve ser executado. Metodologias ágeis de desenvolvimento de software são iterativas, ou seja, o trabalho é dividido em iterações, que são chamadas de Sprints no caso do Scrum.

As funcionalidades a serem implementadas em um projeto são mantidas em uma lista que é conhecida como Product Backlog. No início de cada Sprint, faz-se um Sprint Planning Meeting, ou seja, uma reunião de planejamento na qual o Product Owner prioriza os itens do Product Backlog e a equipe seleciona as atividades que ela será capaz de implementar durante o Sprint que se inicia. As tarefas alocadas em um Sprint são transferidas do Product Backlog para o Sprint Backlog.

A cada dia de uma Sprint, a equipe faz uma breve reunião (normalmente de manhã), chamada Daily Scrum. O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia.

Ao final de um Sprint, a equipe apresenta as funcionalidades implementadas em uma Sprint Review Meeting. Finalmente, faz-se uma Sprint Retrospective e a equipe parte para o planejamento do próximo Sprint. Assim reinicia-se o ciclo.

Veja abaixo o funcionamento do Scrum:

**Figura 01** – SCRUM,

fonte: *https://knowledge21.com.br/, acesso em 24 de abril de 2020*

**PROCESSO DE DESENVOLVIMENTO DE SOFTWARE**

O processo que será usado no desenvolvimento do aplicativo será o modelo cascata. Trata-se de um modelo de desenvolvimento de software sequencial no qual o processo é visto como um fluir constante para frente (como uma cascata) através das fases:

* Análise de requisitos;
* Projeto;
* Implementação;
* Testes (validação);
* Manutenção de software;

**MODELAGEM DO SISTEMA - UML**

Para modelagem do sistema usaremos a UML (Unified Modeling Language) é uma linguagem-padrão para a elaboração da estrutura de projetos de software.

Veremos a seguir os diagramas **estáticos** (estrutural) e **dinâmicos** (comportamentais), eles serão de grande ajuda no processo de desenvolvimento do sistema, pois nos ajudam a identificar os requisitos. Os diagramas estáticos vão auxiliar na criação dos requisitos não-funcionais e os diagramas dinâmicos na criação dos requisitos funcionais.

Mapa colorido com texto preto sobre fundo branco

Descrição gerada automaticamente**DIAGRAMA DE USE CASES** – usado para se comunicar com os stakeholders, por sua clareza e facilidade de mostrar situações e comportamentos dos personagens do sistema.

*Diagrama de caso de uso.* ***Fonte****: autor próprio.*

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente**DIAGRAMA DE CLASSES** – usado para mostrar a visualização das classes que compõe o sistema, seus métodos e atributos.

*Diagrama de classes.* ***Fonte****: autor próprio.*

**DIAGRAMA DE MÁQUINA DE ESTADO** – mostra o comportamento de um elemento por meio de um conjunto finito de transições de estado.

Uma imagem contendo mapa, screenshot

Descrição gerada automaticamente

*Diagrama de máquina de estado.* ***Fonte****: autor próprio.*

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente**DIAGRAMA DE DIAGRAMA DE ATIVIDADE** – descreve os passos a serem percorridos para a conclusão de uma atividade específica.

*Diagrama de atividade.* ***Fonte****: autor próprio.*

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente**DIAGRAMA DE SEQUÊNCIA** – mostra a ordem temporal em que mensagens são trocadas entre objetos envolvidos em um determinado processo com passos finitos.

*Diagrama de sequência.* ***Fonte****: autor próprio.*

## **CONFIGURAÇÃO DO SERVIDOR**

### **REQUISITOS DE HARDWARE:**

- Marca: HP

- Modelo:P06782-S01

**Especificações:**

**Processador:**

- Quad-Core Intel Xeon E-2124

- Clock do Processador: 3.3 GHz

- Cache de Processador: 8MB (L3)

- Processador Suportado: 1

**Memória:**

- Memória: 16GB (1x16GB) Dual Rank x8 DDR4-2666 Unbuffered

- Máxima Suportada: 64GB (4 x 16GB UDIMM @2600MHz)

- Slots Disponíveis: 3

- Slots Totais: 4 (4 DIMMs slots por processador, 2 Canais, 2 DIMMs Por Canal)

**Armazenamento:**

- Disco Padrão: 1TB SATA 6G 7.2K 3.5

- Baia para discos: Até 4 discos LFF Non Hot Plug (com kit opcional de habilitação de 2LFF adicionais)

- Armazenamento Máximo: 24TB (4x 4TB + 2x 4TB (com kit opcional de habilitação de 2LFF adicionais)

- Controladora de disco:  (1) HPE Smart Array S100i SR Gen10 SW RAID NOTA: APENAS SATA. SUPORTE APENAS A WINDOWS

**Geral:**

- Gerenciamento Padrão: HP iLO (Firmware iLO 5)

- Níveis de RAID Suportados: 0, 1, 10, 5

- Unidade Óptica: Nenhum padrão - Opcional

- Placa de rede: (1) HPE 1Gb 2-port 332i network Adapter

- Porta USB: frontais: 1xUSB 3.0, 1xUSB iLO Service Port. Traseira: 2xUSB 3.0. Interna: 1x USB 3.0

- Slot PCI-E: 2 slots PCIe 3.0

- Fonte: HPE 350W E-star 1.0 Power Supply FIO Kit

- Cabo padrao: C13-NBR14136 (Padrão BR) 10A 1.83m

- Formato: Torre (4U)

### **REQUISITOS DE SOFTWARE:**

-Sistema Operacional: Linux Debian;

-Banco de dados: Mysql;

-Proteção: KasperySky Server Edition;

-Servidor Web: Apache;

-Versão PHP: 7.4.2;

-Hospedagem: HOSTNET;

-Outros: Certificado SSL, acesso via FTP usando softwares livres.

Total de custos com hardware: R$ 4.599,90 (fonte: Kabum.com).

Total de custos com Software: R$ 850,00 (fonte: autor próprio).

**LINGUAGEM DE PROGRAMAÇÃO ORIENTADA A OBJETOS**

Para a implementação do sistema na sua versão final optemos por escolher a linguagem JAVA, Java é uma linguagem de programação orientada a objetos desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Em 2008 o Java foi adquirido pela empresa Oracle Corporation.

abaixo listaremos suas principais vantagens:

* Orientada a objetos;
* O código gerado (bytecode) pode ser compilado para rodar em qualquer plataforma, seja ela:
  + Windows;
  + Linux;
  + Mac-OS;
* Comunidade ativa;
* Grande gama de tutoriais disponível na internet;
* Grande número de profissionais habilitados com o uso dessa linguagem;

**DESIGN PATTERN**

Orientação a objetos resolve as grandes dores de cabeça que tínhamos na programação procedural, restringindo e centralizando responsabilidades. Mas alguns problemas não podemos resolver simplesmente com orientação a objetos, pois não existe palavra chave para uma funcionalidade tão específica.

Alguns desses pequenos problemas aparecem com tanta frequência que as pessoas desenvolvem uma solução "padrão" para ele. Com isso, ao nos defrontarmos com um desses problemas clássicos, podemos rapidamente implementar essa solução genérica com uma ou outra modificação, de acordo com nossa necessidade. Essas soluções padrões tem o nome de Design Patterns (Padrões de Projeto).

**FACTORY METHOD**

O padrão Factory Method é um padrão de criação, ele possui a seguinte intenção:

* Definir uma interface para criar um objeto, mas deixar as subclasses decidirem que classe instanciar.

O Factory Method permite adiar a instanciação para subclasses. Ou seja, ao invés de criar objetos diretamente em uma classe concreta, nós definimos uma interface de criação de objetos e cada subclasse fica responsável por criar seus objetos.

**Frameworks**

Frameworks de software são mini-arquiteturas reutilizáveis que provêm a estrutura genérica e comportamento para famílias de abstrações de software, junto com um contexto de metáforas que especificam sua aplicação e uso dentro de um dado domínio. Em termos de software orientado a objetos, frameworks correspondem a um conjunto de classes cooperantes que permitem uma reutilização de design para uma classe de software específica. Assim, um framework provê uma orientação arquitetural, definindo quais as responsabilidades e colaborações entre objetos são necessárias para implementar uma determinada funcionalidade padrão.

Frameworks escolhidos:

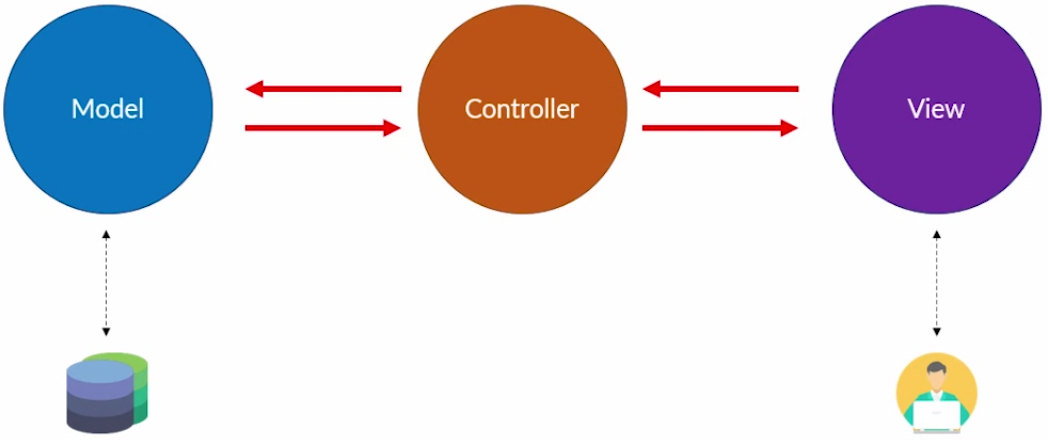
* Bootstrap;
* React;

O Bootstrap é uma estrutura de front-end gratuita para desenvolvimento da Web mais rápido e fácil. O Bootstrap inclui modelos de design baseados em HTML e CSS para tipografia, formulários, botões, tabelas, navegação, modais, carrosséis de imagens e muitos outros, além de plug-ins JavaScript opcionais.

O Bootstrap também oferece a capacidade de criar facilmente designs responsivos

O React é uma biblioteca JavaScript de código aberto com foco em criar interfaces de usuário em páginas web. É mantido pelo Facebook, Instagram, outras empresas e uma comunidade de desenvolvedores individuais. É utilizado nos sites da Netflix, Imgur, Feedly, Airbnb, SeatGeek, HelloSign, Walmart e outros.

**PADRÃO DE ARQUITETURA ESCOLHIDO - MVC**

Para a implementação do nosso sistema, iremos usar a arquitetura MVC, pois acreditamos que irá possibilitar um melhor desempenho da aplicação e tornará a manutenção do sistema menos complexa, podendo os módulos do sistema serem substituídos sem a necessidade de mexer nos demais módulos .

A imagem abaixo descreve o funcionamento do padrão MVC:

**Figura 1**. Diagrama do padrão MVC