1. **Objetivo do projeto**

Nosso objetivo é proporcionar aos usuários um site de fácil acesso e listas de tarefas com grande diversidade de recursos. Disponibilizamos 3 (três) modelos de listas, sendo eles pessoais, empresariais e acadêmicos.

O site da Univers possuí temática de universo, sendo dividido em 7(sete) partes: home, sobre, ferramentas, planos, produtos, login e cadastro, buscando facilitar o acesso e o uso das listas, assim podendo proporcionar ao usuário a chance de conhecer a empresa, seus métodos e produtos.

Produtos:

* Fenix list- Pessoal

Esse modelo é uma lista para uso pessoal destinada a facilitar a organização da sua rotina, oferecendo recursos para classificar suas tarefas entre prioridades e prazos de entrega, evitando assim o atraso ou a não conclusão dela.

* Pegasus list- Acadêmico

Esse modelo possuí funções dedicadas a uso acadêmico, disponibilizando a oportunidade de organizar seus trabalhos, provas e tarefas de acordo com seu plano acadêmico (mensal, semestral ou trimestral). É possível também compartilhar a lista com sua equipe (até 6 pessoas), porém apenas aqueles que possuírem a senha poderão acessar a lista, e apenas o autor poderá alterá-la.

* Draco list- Empresarial

Esse modelo mantém os padrões da lista acadêmica, mas conta com uma possibilidade de compartilhamento com até 30(trinta) pessoas.

1. **Arquitetura**

A arquitetura utilizada é a Client-server(cliente-servidor). Essa arquitetura busca estruturar o sistema como um conjunto de processos cooperativos, uma aplicação distribuída, dessa maneira existem os fornecedores de recursos e serviços a rede(servidores), e os requerentes desses recursos os serviços, chamados de clientes ou usuário.

1. **Acesso e execução do código fonte**

* Nome: Univers. Esse nome é ligado a temática do site;
* Descrição: Serviço de Frontend com APIs (Interface de programação de aplicações) no contexto de listas de tarefas;
* Acesso: Ferramenta de versionamento utilizada: GitHub Desktop, HTML 5 e CSS. O acesso é feito por membros da equipe, através de um endereço https;

1. **Alterações, testes e validações**

Todos os testes foram feitos de maneira unitária, posteriormente compartilhados com o time, seguindo os pontos abaixo:

* Toda alteração no código deve ser realizada respeitando o processo de junção do GitHub, informando os outros membros da equipe;
* Para a validação da alteração é preciso que todo o time esteja ciente da junção ou compartilhamento de mudanças, evitando assim desentendimentos e problemas na estrutura do site;
* Antes de qualquer alteração é preciso realizar teste de validação, afirmando que as alterações são seguras, funcionais e necessárias;

1. **Atualização e monitoramento**

Os códigos das alterações são realizados pelo Visual Studio Code, utilizando o GitHub como ferramenta de versionamento, sendo esse utilizado para as atualizações e os monitoramentos;