Go Deep

**VISÃO DO PROJETO**

Versão [1.2]

**Histórico de Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Sumário**

[***1***](#_heading=h.gjdgxs) ***INTRODUÇÃO 4***

[**1.1**](#_heading=h.4i7ojhp) **Problema 4**

[**1.2**](#_heading=h.30j0zll) **Declaração do Problema 4**

[**1.3**](#_heading=h.1fob9te) **Objetivos do Projeto 4**

[***2***](#_heading=h.3znysh7) ***STAKEHOLDERS 4***

[***3***](#_heading=h.2et92p0) ***VISÃO GERAL DO PRODUTO 4***

[**3.1**](#_heading=h.tyjcwt) **Declaração de Posição do Produto 4**

[**3.2**](#_heading=h.3dy6vkm) **Mínimo Produto Viável (MVP) 4**

[***4***](#_heading=h.1t3h5sf) ***VISÃO GERAL DO PROJETO 4***

[**4.1**](#_heading=h.4d34og8) **Organização do Projeto 5**

[***5***](#_heading=h.2s8eyo1) ***FERRAMENTAS, AMBIENTE E INFRAESTRUTURA 5***

[**5.1**](#_heading=h.17dp8vu) **Hardware 5**

[**5.2**](#_heading=h.3rdcrjn) **Software 5**

[***6***](#_heading=h.26in1rg) ***PROCESSO DE GERÊNCIA DE PROJETO 5***

[**6.1**](#_heading=h.lnxbz9) **Planejamento das Fases e Iterações do Projeto 5**

[**6.2**](#_heading=h.35nkun2) **Processo de Desenvolvimento e Mensuração 5**

[**6.3**](#_heading=h.2xcytpi) **Milestones e Objetivos do Projeto 5**

[**6.4**](#_heading=h.1ksv4uv) **Matriz de Comunicação 6**

[**6.5**](#_heading=h.44sinio) **Escalabilidade do Projeto 6**

[**6.6**](#_heading=h.2jxsxqh) **Gerenciamento de Riscos 6**

[**6.7**](#_heading=h.z337ya) **Critérios de Replanejamento 6**

[***7***](#_heading=h.3j2qqm3) ***LIÇÕES APRENDIDAS 6***

[***8***](#_heading=h.1y810tw) ***REFERÊNCIAS 7***

**VISÃO DO PROJETO**

# INTRODUÇÃO

## Declaração do Problema

|  |  |
| --- | --- |
| **O problema** | A necessidade mercadológica real de se criar mais ferramentas na linguagem Golang para aprendizado de máquina e tratamento de dados. |
| **Afeta** | Ambiente de aprendizado de máquinas. Análise e tratamento de dados. |
| **Cujo impacto é** | Demanda um maior trabalho no desenvolvimento de sistemas que utilizam dessas ferramentas |
| **Uma solução de sucesso seria** | Criar métodos de leitura e tratamento de dados de forma a adquiri-los de maneira adequada para a aplicação de algoritmos de aprendizagem, prover suporte para os dados de cada aplicação, fornecer diversos métodos de aprendizagem para extrair informações e estudar os dados adquiridos e tratados. |

## Objetivos do Projeto

O projeto Go Deep visa a criação de uma biblioteca de I.A (Inteligência artificial, aprendizado de máquina e aprendizado profundo) em Golang, com suporte para manejo, tratamento e análise de dados.

# STAKEHOLDERS

|  |  |  |
| --- | --- | --- |
| ***Nome*** | ***Descrição*** | ***Responsabilidades*** |
| Alunos da UnB | Estudantes da área de tecnologia da informação | Garante que haverá uma demanda de mercado para as características do produto.  Garante suporte a curto prazo caso haja demanda da comunidade.  Garante suporte a médio e longo prazo, caso a biblioteca tenha grande uso da comunidade (com código aberto e manutenção voluntária dos usuários). |
| Comunidade Golang | Programadores e desenvolvedores da linguagem | Garante a disseminação do produto na comunidade |

# VISÃO GERAL DO PRODUTO

## Declaração de Posição do Produto

|  |  |
| --- | --- |
| Para | Comunidade de Golang, I.A e data science. |
| Quem | Atender a comunidade de golang que desenvolve I.A |
| O (nome do produto) | GoDeep, a golang machine learning library |
| Que | Ferramentas convenientes e poderosas para se lidar com machine learning |
| Ao contrário | Golearn biblioteca), Python e C++ |
| Nosso produto | Tratamento de dados e algoritmos de IA que podem ser aplicados em qualquer contexto tratado pela biblioteca, de forma fácil e eficiente. |

## Mínimo Produto Viável (MVP)

Vemos possibilidades de aplicações em Estatísticas, Visão Computacional / Processamento de Imagens, Genética e Ciência de Dados em geral. Durante a palestra **Deep Learning with Go** o pesquisador Chris Benson comenta sobre o potencial da linguagem Go, suas possibilidades e vantagens, em seguida fala sobre a necessidade da criação de mais recursos na linguagem para promover o crescimento da comunidade.

Um exemplo clássico de aplicação da nossa biblioteca, é um sistema de pouso de drone a partir da classificação de imagens em 3 categorias: Seguro, emergência e perigo. A última referência (26) contém um trabalho da disciplina de Introdução a Processamento de Imagens, que explora o problema. Estamos trabalhando nesse uso da biblioteca para apresenta-lo o quanto antes.

# VISÃO GERAL DO PROJETO

## Organização do Projeto

|  |  |  |  |
| --- | --- | --- | --- |
| ***Papel*** | ***Atribuições*** | ***Responsável*** | ***Participantes*** |
| *Desenvolvedor , Gerente* | *Padrões de projeto e algoritmos de I.A e P.I & C.V* | *Ariel Serafim* | *Ariel e Jackes* |
| *Desenvolvedor , Gerente* | *Estatísticas e algoritmos de análise de dados.* | Jackes Fonseca | *Jackes, Pedro, Eduardo e Ariel* |
| *Desenvolvedor , Gerente* | *Tratamento de dados e algoritmos genética* | Pedro Henrique | *Ariel e Pedro* |
| *Desenvolvedor , Gerente* | *Testes e algoritmos de jogos* | Eduardo | *Eduardo e Pedro* |

# FERRAMENTAS, AMBIENTE E INFRAESTRUTURA

## Hardware

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Perfil*** | ***Tipo de Hardware*** | ***Configurações*** | ***Qtd. Planejada*** | ***Prazo Estimado*** | ***Observação*** |
| *Desenvolvedor/Usuário* | *Processador* | *Intel Celeron ou superior* | *01* |  |  |
| *Desenvolvedor/Usuário* | *Memória RAM* | *Mínimo 2GB – Recomendo 4GB* | *01* |  |  |
| *Desenvolvedor/Usuário* | *Hard Disk* | *Mínimo 250GB – Recomendado 500GB ou superior* | *01* |  |  |
| *Desenvolvedor/Usuário* | *Resolução de Tela* | *Recomendado 1280 x 1024* | *01* |  |  |

## Software

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ***Perfil*** | ***Tipo de Software*** | ***Nome da Ferramenta*** | ***Versão*** | ***Qtd Licenças Planejada*** | ***Prazo Estimado*** | ***Observação*** |
| *Imagens* | *biblioteca* | *gocv* | *0.26.0* |  |  |  |
| *Controle de versão* | *site a aplicativo* | *git e github* | *2.30.1* |  |  |  |

# PROCESSO DE GERÊNCIA DE PROJETO

## Planejamento das Fases e Iterações do Projeto

* ***Etapa 1; planejamento de estrutura:*** *Aqui planejaremos os padrões de projeto de cada seção da biblioteca e o padrão de projeto que interliga todas as partes. Essa etapa finaliza 2 semanas após a aprovação desse documento, porém queremos ter partes encapsuladas definidas 1 semana antes, para podermos iniciar a codificação.*
* ***Etapa 2; implementação de 1 tratamento completo de dados e um processo de aprendizagem:*** *Aqui temos por objetivo implementar um ciclo completo de leitura, filtragem, generalização e aprendizagem a partir de um dataset dentro de uma das 4 aplicações definidas como escopo.*
* ***Etapa 3; Expansão de aplicações:*** *Nessa etapa nossa missão será implementar o tratamento de dados das demais aplicações definidas no escopo.*
* ***Etapa 4; Expansão geral:*** *Por fim, trataremos de expandir os casos cobertos por cada aplicação e também a quantidade de algoritmos de aprendizagem, com foco nesse segundo.*

## Processo de Desenvolvimento e Mensuração

As atividades básicas do processo de desenvolvimento de software se darão em sua maior parte por meio do uso do Scrum visando seus princípios básicos tal como transparência, inspeção e adaptação. Este framework estará presente durante todo ciclo de vida de desenvolvimento por meio de processos como o planejamento, revisão e retrospectiva da sprint. Além disso, será feito o uso de conceitos presentes no PMBOK v6 que consiste em uma padronização que identifica e conceitua os processos, áreas de conhecimento, ferramentas e técnicas de gesto do projeto e no Management 3.0 que tem como base o pensamento complexo, se baseando em seus princípios de experimentações, melhoria contínua e auto-organização.

## Matriz de Comunicação

|  |  |  |  |
| --- | --- | --- | --- |
| ***Descrição*** | ***Área/ Envolvidos*** | ***Periodicidade*** | ***Produtos Gerados*** |
| *- Acompanhamento das Atividades em Andamento*  *- Acompanhamento dos Riscos, Compromissos, Ações Pendentes, Indicadores* | *- Equipe do Projeto* | *- Semanal* | *- Ata de reunião*  *- Relatório de situação do projeto*  *- Próximas tarefas semanais* |
|  | *- Quinzenal* |
| *- Comunicar situação do projeto* | *- Equipe*  *- Professor* | *- Semanal* | *- Ata de reunião*  *- Relatório de Situação do Projeto para o professor* |
| *relatório diário* | *-Equipe do projeto* | *-Diário* | *-Relatório de andamento diário.*  *-Necessidade de ajuda.*  *-Empecilhos ao desenvolvimento.* |

## Escalabilidade do Projeto

*Ao analisarmos a natureza da biblioteca, vemos que podemos separar as diferentes funcionalidades em 2 categorias.*

1. ***Algoritmos de Inteligência artificial:*** *O objetivo é que toda a parte de I.A seja genérica, isto é, uma vez que os dados tenham sido tratados em instâncias anteriores, a natureza dos dados indefere para os algoritmos de I.A. Assim, esperamos que adicionar novos algoritmos dessa natureza não envolve revisar as aplicações, mas sim desenvolver a partir de um ponto fixo, cujas condições são conhecidas.*
2. ***Tratamento de dados para aplicações específicas:*** *Para cada aplicação (Processamento de imagens, estatística, jogos e genética) haverá um tratamento de dados que tem por objetivo filtrar ruídos e tornar genérica a informação, ou seja, uma informação válida e em um formato conhecido pelos algoritmos de I.A.*

*A partir dessa construção de algoritmos e tratamentos de dados, esperamos atingir os princípios "Classes should be open for extension, but closed for modification." e "Strive for loosely coupled designs between objects that interact.".*

*Assim atingiremos um modelo de biblioteca cujo custo de crescimento não inclua modificação.*

## Gerenciamento de Riscos

* 1. ***Riscos da fase de desenho de Uml e planejamento de estrutura:*** *Planejamento de estrutura que leve a recodificação.*
  2. ***Riscos da fase de filtragem e******generalização de dados:*** *Perder informação relevante e/ou comprometer a integridade dos dados.*
  3. ***Riscos da fase de implementação dos algoritmos de IA****: implementar algoritmos de forma que as informações obtidas não sejam válidas.*
* *Estratégia para risco 6.6: implementar toda a biblioteca dentro de padrões de projeto com alto nível de confiabilidade.*
* *Estratégia para risco 6.7: estipular critérios de verificação humana e computacional de integridade de dados.*
* *Estratégia para risco 6.8: estipular critérios de verificação humana e computacional de integridade de resultados;*
* *Estratégia geral: manter um documento com o controle de riscos, qualidade e integridade do software e dados.*

## Critérios de Replanejamento

*Ainda baseado em padrões de projetos, o replanejamento ocorrerá quando identificarmos que os padrões de projetos utilizados podem ser substituídos por outros que possuem vantagens em relações aos atuais e/ou resolvam problemas atuais de forma mais eficiente.*

# LIÇÕES APRENDIDAS

Com o projeto, importantes habilidades e aprendizados foram adquiridos para o desenvolvimento de software. Dentro disso, pode-se citar que o grupo aprendeu de maneira significativa conceitos relacionados a ciclo de vida, desenvolvimento ágil (Scrum, XP, Management 3.0, PMBOK v6), padrões e gerência de projetos, levantamento de requisitos, construção e teste de software aplicados no ambiente de configuração e aprendizado de máquinas. Além disso, foram adquiridos conceitos da linguagem Golang e conteúdos relacionados às quatro aplicações utilizadas: visão computacional, genética, jogos e estatística.

# REFERÊNCIAS

|  |  |
| --- | --- |
| **1** | <https://www.kaggle.com/> - O Kaggle oferece um ambiente de Jupyter Notebooks sem configuração e personalizável. |
| **2** | <https://github.com/sjwhitworth/golearn> - Biblioteca com objetivos semelhantes. |
| **3** | <https://gocv.io/> - Biblioteca de processamento de imagens e visão computacional que será utilizada na parte de imagens do projeto. |
| **4** | [*https://www.youtube.com/watch?v=P5QObw\_kqbc&t=2463s*](https://www.youtube.com/watch?v=P5QObw_kqbc&t=2463s) *–- Vídeo sobre golang e I.A, do pesquisador que faz um chamado à comunidade de golang.* |
| ***5*** | <https://www.pdfdrive.com/head-first-design-patterns-e183973112.html> – Livro head first, Design Patterns, usado para montar a estrutura do projeto. |
| ***6*** | [https://pt.wikipedia.org/wiki/Intelig%C3%AAncia\_artificial](https://pt.wikipedia.org/wiki/Inteligência_artificial) |
| ***7*** | <https://golang.org/> - Site oficial golang |
| ***8*** | <https://en.wikipedia.org/wiki/Statistics> – Estatisticas |
| ***9*** | <https://en.wikipedia.org/wiki/Computer_vision> – Visão computacional |
| ***10*** | <https://en.wikipedia.org/wiki/Digital_image_processing> – Processamento de imagens |
| ***11*** | <https://en.wikipedia.org/wiki/Data_science> – Ciencia de dados |
| ***12*** | <https://chrisbenson.com/> - Pesquisador em I.A Chris Benson |
| ***13*** | <https://blog.golang.org/open-source> - Comunidade Golang |
| ***14*** | <https://www.tensorflow.org/> - Tensor flow |
| ***15*** | <https://pytorch.org/tutorials/> - Pytorch |
| ***16*** | <https://www.python.org/> - Python |
| ***17*** | <https://www.cplusplus.com/> - C++ |
| ***18*** | <https://chainer.org/> - Chainer |
| ***19*** | <https://trends.google.com/trends/explore?date=2009-10-11%202021-02-05&q=golang&hl=en-US> – Pesquisa uso golang |
| ***20*** | <https://trends.google.com/trends/explore?date=2020-01-06%202021-02-05&q=golang,Covid-19&hl=en-US> – Pesquisa golang |
| ***21*** | <https://blog.golang.org/8years> – Pesquisa golang antiga |
| ***22*** | <https://machinebox.io/> - Machine box |
| ***23*** | <https://www.youtube.com/watch?v=NZlIYr1slAk> – I.A jogando o jogo dino do google |
| ***24*** | [https://answers.opencv.org/question/179923/laptop-minimum-system-requirement-for-opencv/?answer=179952#:~:text=opencv%2D3.4.,system%20to%20generate%20object%20code](https://answers.opencv.org/question/179923/laptop-minimum-system-requirement-for-opencv/?answer=179952#:~:text=opencv-3.4.,system to generate object code). - Requisitos de hardware mínimos |
| ***25*** | <https://github.com/hybridgroup/gocv> - Versão atual gocv |
| ***26*** | <https://github.com/zaghetto/ImageProcessing/blob/master/Assignments/Assignment%203/Assignment3.pdf> – Aplicação Prática. |
| ***27*** |  |
| ***28*** |  |
| ***29*** |  |
| ***30*** |  |