



**RELATÓRIO DE PROPOSTA
PROJETO “RED WHISKERS”**

LICENCIATURA EM ENGENHARIA INFORMÁTICA

Coordenador de Curso: André Sabino

Professor da UTC: José Lopes

André Custódio 20220112

João Coelho 20220753

Ricardo Dias 20220494

05/10/2024

O relatório encontra-se em condições para ser apresentado

Ciclo de Formação 2022/2025
Ano Letivo 2024/2025

Índice

Capítulo I – Concepção do Projeto	3
Tema	3
Pesquisa.....	3
Conceito	3
Objetivo	3
Publico Alvo	4
Capítulo II – Especificações Técnicas	5
Requisitos	5
Guiões de Teste:	5
Use Cases.....	6
Use Cases - Diagrama	7
Tecnologias	8
Integração com as diferentes Cadeiras.....	8
Inteligencia Artificial	9
Segurança Informática:.....	9
Arquitetura:	10
Modelo de Desenvolvimento	11
Wireframes:.....	11
Wireframes:.....	12
Paleta de Cores:.....	13

Capítulo I - Concepção do Projeto

Tema

- De forma a tornar, tanto o projeto em si, como o processo de desenvolvimento ao longo do semestre, em algo didático e menos maçante decidimos fazer um jogo web.
- O tema do jogo é a volta de um Panda Vermelho, a nossa mascote, que vai fugir de um jardim zoológico em direção ao seu *habitat* natural. Ao longo desta fuga o Panda, o jogador, terá de se desviar de vários obstáculos e percalços que apareceram no seu caminho, obter pontos, de forma a melhorar as suas habilidades e distinguir-se dos outros pandas, e competir entre si para ver quem chega mais perto do habitat.

Pesquisa

- A personagem principal do jogo, o pando vermelho (*Ailurus fulgen*), é uma espécie em ameaça de extinção. Este mamífero é caracterizado pelo seu pelo vermelho e porte pequeno, características que o tornam "fofo" ao olho do público. A sua aparição mais recente na cultura pop é no filme "Turning Red", um filme que fala de trauma geracional de uma forma leve e apropriada para crianças, através dos seus visuais coloridos e simpáticos.
- Jogos Web, no estilo de SideScroller, são bastante comuns. Os exemplos mais conhecidos são "Subway Surfers", mais jogado em mobile, e alguns mini-jogos do "Club Penguin", como o "Hydro Hopper" e o "Sled Racing". Estes jogos são conhecidos por serem extremamente viciantes, por serem jogos simples com várias recompensas. Esta atitude de constantemente estar a recompensar o jogador cria uma necessidade de continuar a jogar.
- Como tal, juntando os dois temas, apresentamos um jogo extremamente viciante, com visuais simples, cativantes e "fofos". Garantindo assim o uso e crescimento da aplicação.

Conceito

- O jogo será um sidescroller infinito, onde o jogador terá que se desviar de obstáculos, como referido no Tema, terá também um highscore (máxima distância percorrida), cosméticos e habilidades, comprados ou evoluídos numa loja, onde o mesmo pagará com os seus pontos. O multiplayer é real-time onde os jogadores jogam no mesmo mapa. O jogo acaba quando todos os jogadores morrerem.

Objetivo

- Tendo em conta os pontos anteriores, o nosso objetivo é criar um jogo simples e cativante, sensibilizando assim à possível extinção do panda vermelho.
- Como tal, a aplicação tem que ser fácil de usar e não pode "roubar" ao utilizador muito do seu tempo. Isto implica usar simplificar as interfaces e os controlos do jogador.
- Para o jogo ser cativante, os visuais têm que ser apelativos e memoráveis.

Publico Alvo

- Sendo o nosso tema um panda vermelho que se enquadra muita na cultura atual de sensibilização e preocupação da Natureza, o nosso público alvo são crianças na pré-adolescência. Este ponto também é reforçado pelo design do jogo que se baseia em gráficos cativantes, muito apelativos a esta demográfica.

Capítulo II - Especificações Técnicas

Requisitos

A lista de requisitos engloba todas as necessidades da aplicação. Cada ponto representa uma funcionalidade:

- Sistema de Autenticação encriptado (Login, SignUp, etc...);
- Sistema de missões diárias e semanais;
- Personalização da personagem
- Inventário do jogador
- Sistema de compra de Cosméticos
- Jogo Singleplayer:
 - Geração de obstáculos e PowerUps
 - Leitura dos Resultados
- Jogo Multiplayer:
 - Criação de Lobbies
 - Conexão via Web Sockets

Guiões de Teste:

Guiões de Teste ajudam-nos a melhor compreender a experiência do nosso utilizador ao usar o nosso produto. Sendo a nossa aplicação um jogo web, a nossa intenção é que o utilizador tenha uma experiência simples e rápida:

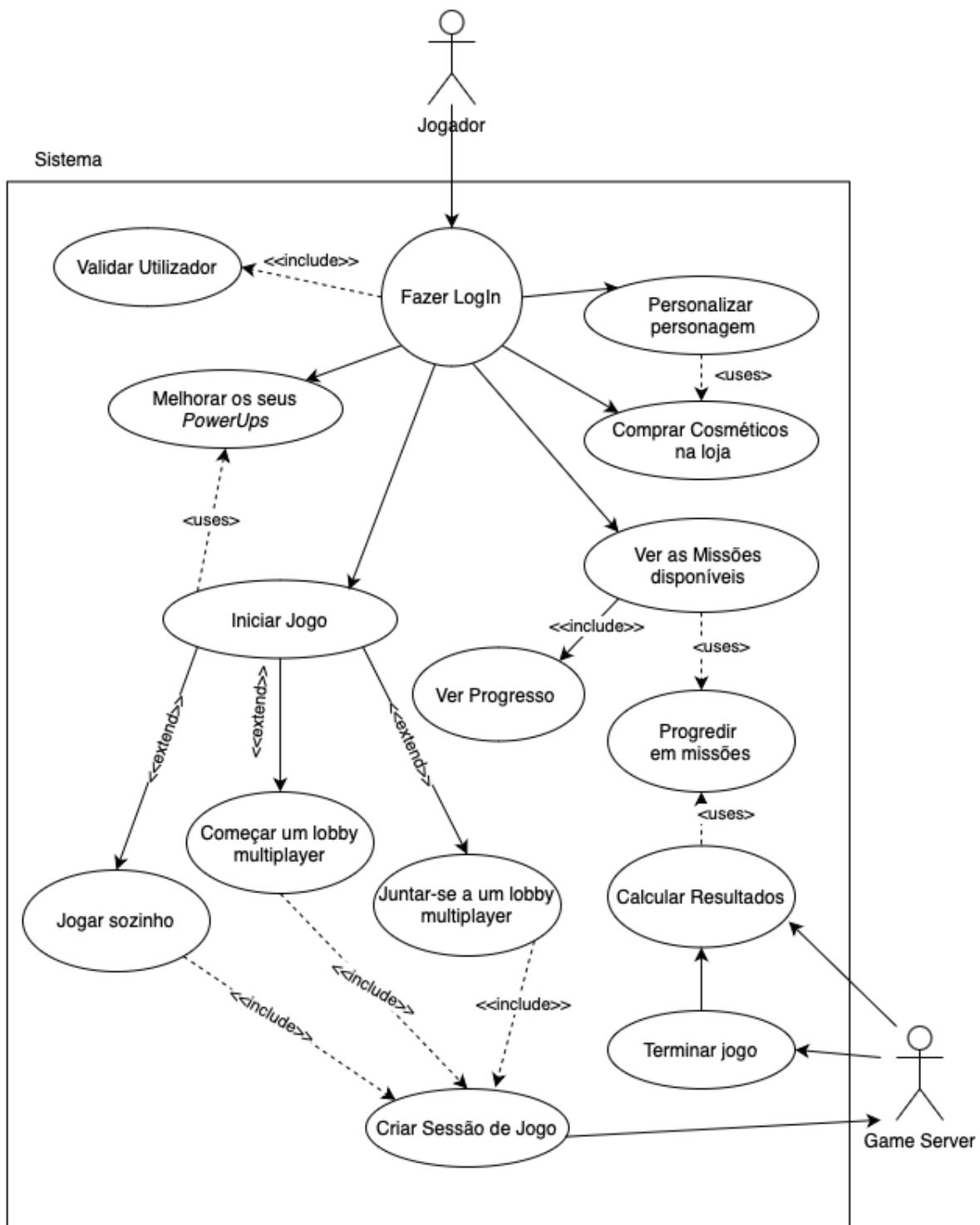
- 1º Guião (jogo singleplayer): Neste caso o objetivo do utilizador é jogar uma partida:
 - Começamos pelo utilizador ter que fazer login, permitindo verificar a identidade do utilizador.
 - De seguida o utilizador vê a página principal, onde consegue observar as suas missões e o seu panda vermelho.
 - Ao clicar no fundo, o jogo começa, tendo que se desviar de obstáculos e apanhar *PowerUps*.
 - Quando, finalmente, o jogador bate em algum obstáculo, é lhe mostrado a distância que ele percorrer e volta para a página principal.
- 2º Guião (jogo multiplayer): Neste caso o objetivo é igual ao anterior, mas com a adição de mais jogadores:

- Após o utilizador ter feito login ele clica no botão de multiplayer, na pagina principal.
 - De seguida é lhe perguntado se quer juntar-se a um jogo já existente, tendo que inserir um código, ou criar o seu próprio jogo, onde lhe será dado um código para partilhar.
 - Supondo que o utilizador quer criar o seu próprio jogo, ficará visível uma janela com as pessoas no seu "lobby" e a oportunidade de adicionar bots ao seu jogo.
 - Assim que todos os jogadores estiverem prontos, o botão para começar o jogo estará disponível.
 - Após todos os jogadores terem chocado contra um obstáculo, será mostrado o vencedor e a distancia percorrida por todos. Acabando na pagina principal.
-
- 3º Guião (comprar cosméticos e equipa-los):
 - Após o utilizador ter feito login, ele carrega no botão da loja.
 - Na pagina da loja, estão todos os cosméticos disponíveis para comprar.
 - Ao selecionar um cosmético, aparecerá uma previsualização do panda com o cosmético equipado. Após o segundo clique, o cosmético será comprado e guardado no inventário do jogador.
 - Ao carregar no botão do inventario, aparecem todos os cosméticos adquiridos pelo jogador. Finalmente, ao carregar num dos cosméticos, o mesmo é equipado pelo panda.

Use Cases

- No caso deste projeto o agente principal é o Jogador, o mesmo pode comprar cosméticos, personalizar o seu panda, melhorar os seus PowerUps e começar uma partida. Todas estas ações requerem autenticação por parte do jogador, de forma a conseguir validade a sua identidade.
- Outro agente com importância no sistema é o *Game Server*, ele gera todas as sessões de jogo, avisando a API quando um jogo termina e quais os seus resultados.

Use Cases - Diagrama



Tecnologias

As Tecnologias usadas no desenvolvimento do produto não só refletem os requisitos do mesmo mas também os nossos conhecimentos dos mesmos. De forma a garantir um desenvolvimento rápida e sem muito tempo de aprendizagem, decidimos usar tecnologias familiares, que já usamos em outros projetos. Segue a lista de tecnologias a usar:

- **Web App (FrontEnd):** React.js, de forma a podermos garantir um UI mais interativo.
- **API (BackEnd):** Node.js, sendo uma tecnologia que nós sentimos confortáveis a usar e que preenche os requisitos do projeto.
- **Base de Dados:** MySQL, aplica-se a mesma razão que a do ponto em cima.
- **Game Server:** Node.js, usando Web Sockets de forma a habilitar a implementação de sessões de jogo multijogador.
- **Inteligência Artificial:** Python, linguagem aprendida em aula.

Integração com as diferentes Cadeiras

Tendo em conta que o produto se enquadra no contexto de um projeto de semestre, temos que ter em conta as exigências das diferentes cadeiras que estão envolvidas.

Como tal segue a lista de como é que cada cadeira se encaixara no nosso projeto, seguida de uma análise mais extensiva nos próximos capítulos:

- **Inteligência Artificial:** Bots que replicarão o comportamento de um jogador normal, de forma a criar partidas competitivas, mesmo quando não há jogadores suficientes para o mesmo.
- **Segurança Informática:** Encriptação de dados partilhados entre as diferentes layers (Web App, API, BD, etc...), Security By Design, Cheating Detection, Encriptação da Base de Dados.
- **Sistemas Distribuídos:** Tolerância a falha, gestão de game sessions entre as várias replicas.
- **Engenharia Informática:** Planeamento, Unit Testing, Quality Testing.

Inteligencia Artificial

Justificação da necessidade de uma componente de IA

- No desenvolvimento da ideia geral do projeto, decidimos que seria melhor interesse do jogo se houvesse uma possibilidade de um jogador poder defrontar outro. Sendo assim criámos uma secção do jogo que é multiplayer, mas compreendemos que nem sempre existem jogadores disponíveis. Daí iremos implementar um conjunto de Bots onde o jogador pode escolher dentro deles qual ou quais defrontar, podendo serem ou não acrescentados num lobby multiplayer.

Definição dos algoritmos de IA a serem utilizados

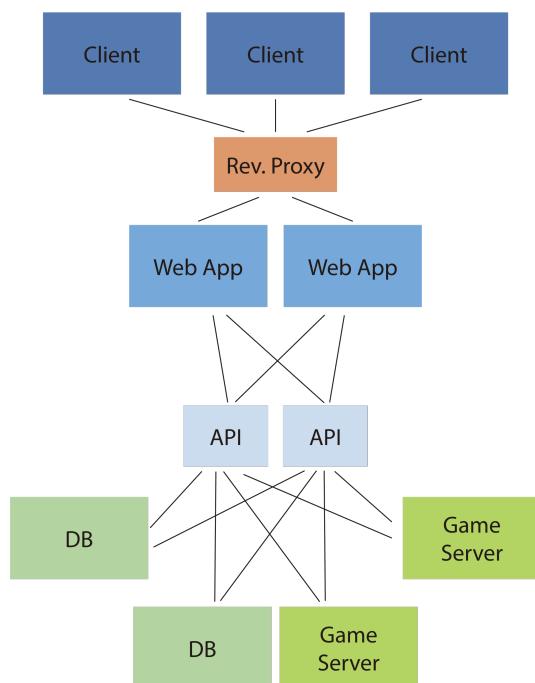
- Após investigarmos quais seriam as melhores abordagens para o nosso problema atual, chegámos à conclusão de que iremos ter Diferentes Bots com diferentes níveis de dificuldade.
- Na dificuldade Fácil o Bot utilizará um sistema de Waypoint para pathfinding e para decisões será um State Machine.
- Na dificuldade Intermédia os botes utilizarão uma versão mais refinada dos utilizados no modo mais fácil.
- Na dificuldade Difícil utilizaremos então o algoritmo de pathfinding A em combinação com uma Decision Tree, que dará ao Bot uma capacidade de navegação entre obstáculos consideravelmente maior comparativamente com as suas versões mais fáceis.

Segurança Informática:

- Para além dos problemas deseguranças comuns de uma aplicação, como a proteção de chaves de acesso, leakage de dados e vulnerabilidade do funcionamento do sistema, um jogo adiciona o problema dá inconsistência de dados ou "batota".
- Como tal, em adição á encriptação de informação partilhada entre o cliente e a aplicação, adicionamos um sistema de verificação de batota. Este sistema verifica as ações do jogador, determinando se as mesmas entram nos padrões esperados e possíveis, e os resultados das partidas, segundo a informação gerida pelo game server, como o layout da partida, os PowerUps gerados, etc...

Arquitetura:

- De forma a conseguirmos garantir que providenciamos os clientes com uma aplicação confiável e funcional, necessitamos garantir que a mesma é tolerante a falhas.
- Como tal, a arquitectura das diferentes interfaces da nosso sistema é extremamente importante.
- É necessário garantir que cada interface não está a ser sobrecarregada, isto implica dividir a nossa aplicação em diferentes layers, cada uma responsável pelo sua função. Neste caso, decidimos dividir o sistema nas seguintes partes: o Cliente, a Aplicação Web, a API, o Game Server e a Base de Dados.
 - Cliente: O Utilizador da aplicação.
 - Aplicação Web: A interface apresentada ao utilizador, responsável pela experiência do mesmo.
 - API: Que gera os pedidos do utilizador, comunicando com a Base de Dados e o Game Server.
 - Game Server: Gere as sessões de jogo dos utilizadores, sendo aqui onde está concentrada a game logic
 - Base de Dados: Onde se encontram grande parte dos dados, como a informação dos jogadores, cosméticos disponíveis, higscores, etc...
- Para garantir que, no caso de alguma das layers mencionadas for abaixo, a aplicação continua funcional, é necessário replica-las. De forma a garantir que os pedidos do cliente estão a ir para a API funcional e que não há sobrelocação de pedidos numa API, adicionamos um Reverse Proxy entre as duas layers.



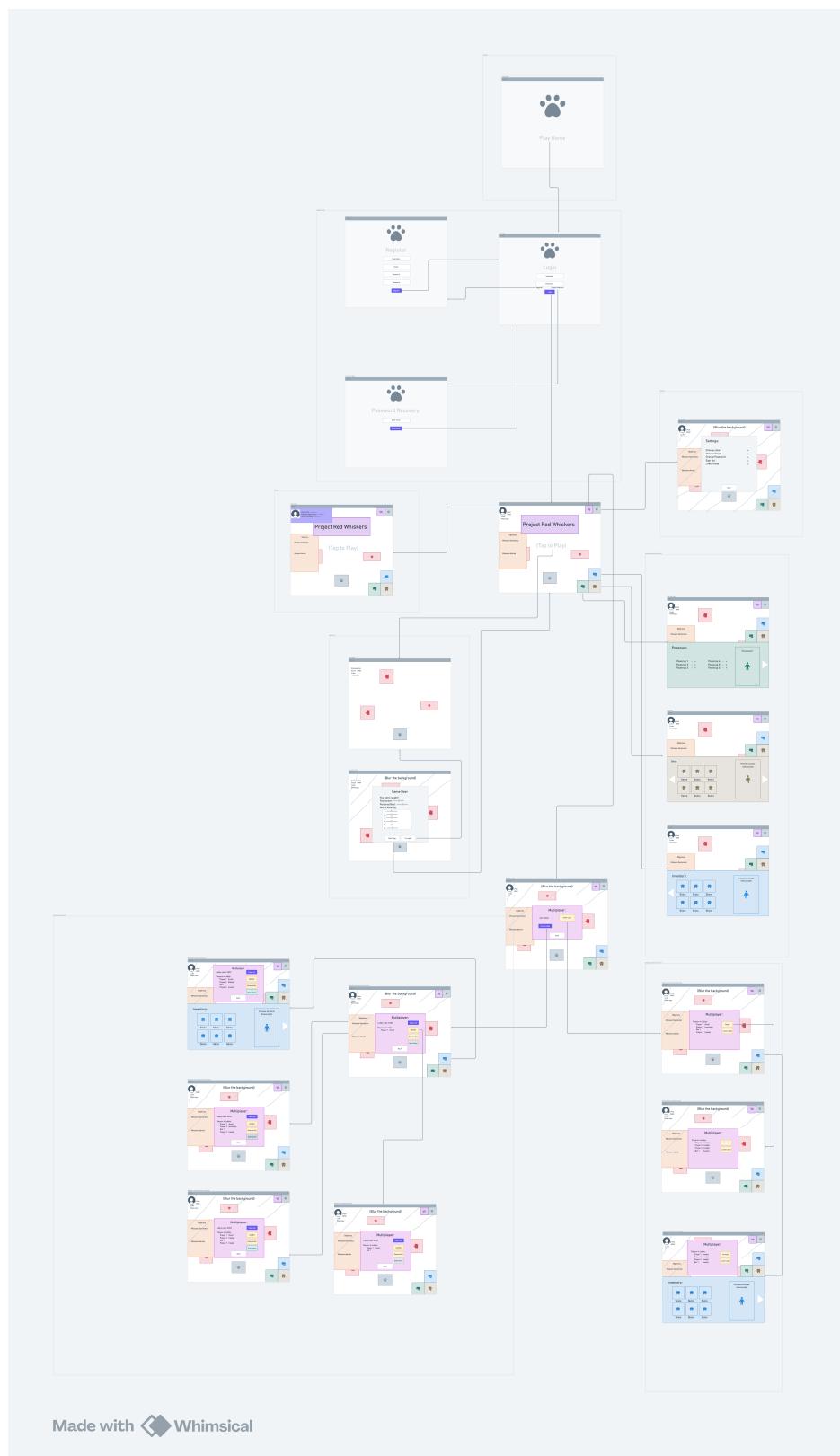
Modelo de Desenvolvimento

- A abordagem de desenvolvimento é extremamente importante num contexto como o que o projeto se encontra, tendo em conta a escassez de tempo e falta de conhecimento. Como tal o modelo de desenvolvimento que iremos usar reflete esses problemas.
- Devido a obrigatoriedade de relatórios semanais por varias cadeiras, um modelo ágil é o que preenche mais requisitos. Apesar disto, a prioridade de entregar um produto finalizado não nos permite estar constantemente a fazer alterações ao projeto.
- Como tal misturar o conceito ágil com o modelo incremental, onde continuamos a realizar o projeto por incrementos de uma semana, mas prioritizando a criação de uma aplicação básica finalizada onde podemos adicionar complexidade.

Wireframes:

- Jogos web são conhecidos por serem jogos simples, de recompensa imediata, ou seja, não requerem que o jogador se comprometa demasiado ao jogo. Como tal deve existir o minimo de navegação entre paginas e deixar tudo na palma do jogador, de forma a não aborrecer-lo.

Wireframes:



Made with  Whimsical

Paleta de Cores:

- O branding de um jogo é extremamente importante, definindo a sua memorabilidade no vasto mercado de jogos existentes. Como tal, precisamos de uma paleta de cores que defina a marca visual do produto, relacionando-se aos assuntos que ele trata.
- Desta forma criamos uma paleta de cores semelhante as cores do pelo de um panda vermelho, cores quentes e suaves, e uma cor de contraste semelhante ao alimento do pando, relvas e plantas.

