

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

PROF^a REBECA SCHROEDER FREITAS
DISCIPLINA DE ENGENHARIA DE SOFTWARE

LETÍCIA CAPITANI TRAPP SAMPAIO
GONÇALO FILIPE DOS SANTOS GARRIDO DE AZEVEDO COSTA

ANÁLISE NO CONTEXTO DE ENGENHARIA DE SOFTWARE DO DESENVOLVIMENTO
DE UM JOGO DE RPG QUE ENSINE PROGRAMAÇÃO

JOINVILLE

2023

SUMÁRIO

1	DESCRIÇÃO DO PROBLEMA	3
1.1	ESCOPO DO SOFTWARE.....	3
1.1.1	Justificativa do projeto	3
1.1.2	Finalidade do projeto	3
1.1.3	Objetivos do projeto.....	4
1.1.4	Descrição do produto	4
1.1.5	Stakeholders	5
1.1.6	Entregas do projeto.....	5
1.1.7	Critérios de aceitação	6
1.1.8	Restrições	6
1.1.9	Riscos.....	6
2	REQUISITOS DO SOFTWARE.....	7
2.1	HISTORIAS DE USUÁRIO	7
2.2	PROTÓTIPOS BÁSICOS DO SISTEMA	8
3	ESTIMATIVA DE DURAÇÃO DO PROJETO COMPLETO	11
3.1	CÁLCULO PFNA	12
3.2	CÁLCULO KLOCS	13
3.3	MÉTODO COCOMO	13
4	DIAGRAMA DE CLASSES DO PROJETO UML	15
5	TESTES UNITARIOS.....	16
	REFERÊNCIAS.....	17

1 DESCRIÇÃO DO PROBLEMA

1.1 ESCOPO DO SOFTWARE

1.1.1 Justificativa do projeto

O mercado de tecnologia cresce exponencialmente a cada ano, provando ser promissor para profissionais do nicho. No entanto, a formação de profissionais e especialistas não acompanhou tal evolução, causando um déficit de mão de obra capacitada na área. Analisando os motivos da escassez de mão de obra qualificada, é evidenciado que o ensino de pensamento lógico está defasado nas escolas brasileiras.

O ensino de programação e lógica para jovens oferece uma série de benefícios, tanto para o desenvolvimento cognitivo quanto para habilidades futuras, alguns deles são: Estímulo do pensamento lógico, fomentação da criatividade, aprimoramento das habilidades matemáticas, incentivo a colaboração e trabalho em equipe.

Porém, o primeiro contato com programação básica deve ser lúdico para que possa se experimentar a lógica de programação de forma envolvente e divertida. Desse modo, surge a necessidade de abordagens inovadoras e atrativas no ensino de programação, partindo do dessa análise, é proposto neste artigo o desenvolvimento de um software baseado em um jogo de RPG (Role-Playing Game) interativo, implementado em escolas e cursos de computação atuando na capacitação gradual de jovens.

Para contextualizar, tradicionalmente, um jogo de RPG fundamenta-se em assumir o papel de personagens com diferentes classes, cada uma com suas próprias características e habilidades únicas. O objetivo do jogo, geralmente, é atravessar uma série de dungeons, que são, na maioria das vezes, caminhos que simulam ambientes como cavernas ou masmorras. Nesses caminhos são apresentados uma série de desafios, como armadilhas e monstros que precisam ser enfrentados pelos personagens.

À medida que o personagem avança nessas dungeons é acumulado Pontos de Experiências (XP), que permite subir de nível e melhorar as habilidades e atributos das classes. Basicamente, é uma jornada de crescimento e aprimoramento à medida que supera desafios e ganha experiência ao longo do caminho.

1.1.2 Finalidade do projeto

Desenvolver um jogo educativo que facilite o aprendizado de programação e análise lógica, tornando o ensino de programação mais acessível e interessante para diferentes faixas etárias. Além disso, contribuir para a formação de habilidades tecnológicas desde a infância.

1.1.3 Objetivos do projeto

- a) Criar um ambiente lúdico e educacional para o ensino de conceitos de programação básica;
- b) Estímulo do pensamento lógico e resolução de programas;
- c) Proporcionar uma experiência interativa que motive os usuários a se envolverem ativamente no aprendizado;

1.1.4 Descrição do produto

O software estará disponível para download em desktop e mobile, adaptado para diferentes versões de dispositivos, e cada usuário deve conter um login e senha vinculados, criados a partir de um cadastro de usuário. O usuário poderá criar seu personagem decidindo sua classe entre as classes: Mago, Guerreiro e Druida. A mudança de classe afetará no número de Defesa e Força de cada personagem, estes valores serão gerados automaticamente e aumentados assim que o XP total for suficiente. Cada usuário terá uma trilha de aprendizagem personalizada para seu nível de conhecimento, sendo que, assim que instalado, o aplicativo propõe uma pequena prova para avaliar suas competências, caso esteja mais avançado irá pular as etapas básicas.

O jogo terá objetivos diários para serem realizados acumulando XP a cada conclusão, o sistema será baseado em um esquema de Ilhas que representam assuntos importantes e relevantes em computação, cada uma dessas Ilhas possuem uma série de Níveis contendo cada sub assunto. Cada Nível terá uma parte teórica, exercícios extras e uma dungeon interativa para testar os conhecimentos, em cada uma dessas dungeons o usuário deve acertar exercícios de programação para passar por obstáculos e derrotar monstros, níveis mais altos possuirão monstros mais fortes e dungeons com mais armadilhas, além de exercícios mais complexos. Na dungeon, caso o usuário erre um exercício de ataque, reduz sua Força naquele turno, caso erre um exercício de defesa, reduz sua Defesa, sendo que as reduções são proporcionais ao acerto e desempenho do código.

1.1.5 Stakeholders

Internos

- a) Equipe de Desenvolvimento: Desenvolvedores, programadores e designers responsáveis pela criação e implementação do jogo, além do gerente de projeto encarregado de coordenar as atividades da equipe;
- b) Equipe Educacional: Especialistas em pedagogia e computação envolvidos na criação de conteúdo educativo e na integração do jogo na educação;
- c) Equipe de Testes: Testadores internos que garantem a qualidade e a usabilidade do produto antes do lançamento;
- d) Gerência Executiva: Tomadores de decisão dentro da organização que garantem apoio financeiro e estratégico para o projeto;

Externos

- a) Educadores, Escolas e Universidades: Professores que utilizarão o jogo como ferramenta educacional em sala de aula e diretores escolares interessados em implementar inovações no ensino;
- b) Pais e Responsáveis: Stakeholders preocupados com a educação de seus filhos, que podem influenciar a aceitação e adoção do jogo;
- c) Usuários Finais (Estudantes e interessados em iniciar em programação): Jovens que utilizarão o jogo como meio de aprendizado de programação;
- d) Especialistas em Tecnologia e Educação: Profissionais externos que podem oferecer insights valiosos para aprimorar a eficácia educativa do jogo;
- e) Parceiros de Negócios e Sócios: Empresas ou organizações que podem fornecer apoio financeiro, recursos adicionais ou parcerias estratégicas para promover o jogo;
- e) Comunidade em Geral: A aceitação e apoio da comunidade em geral podem impactar o sucesso e a longevidade do projeto;

1.1.6 Entregas do projeto

- a) Análise do desempenho do usuário baseado nos acertos/erros;
- b) Material de apoio para educadores e estudantes;
- c) Versão funcional do jogo;

1.1.7 Critérios de aceitação

- a) Funcionalidade adequada em diferentes versões de dispositivos;
- b) Aprovação de educadores e especialistas em pedagogia;
- c) Feedback positivo dos usuários de teste;
- d) Progresso contínuo no aprendizado em programação por parte dos usuários;

1.1.8 Restrições

- a) Orçamento limitado;
- b) Prazos específicos para implementação;
- c) Restrições de hardware ou software para os usuários finais;

1.1.9 Riscos

- a) Mudança nas tendências educacionais que podem afetar a aceitação do produto;
- b) Desafios técnicos durante o desenvolvimento;

2 REQUISITOS DO SOFTWARE

2.1 HISTORUAS DE USUÁRIO

Como Usuário Estudante gostaria de...

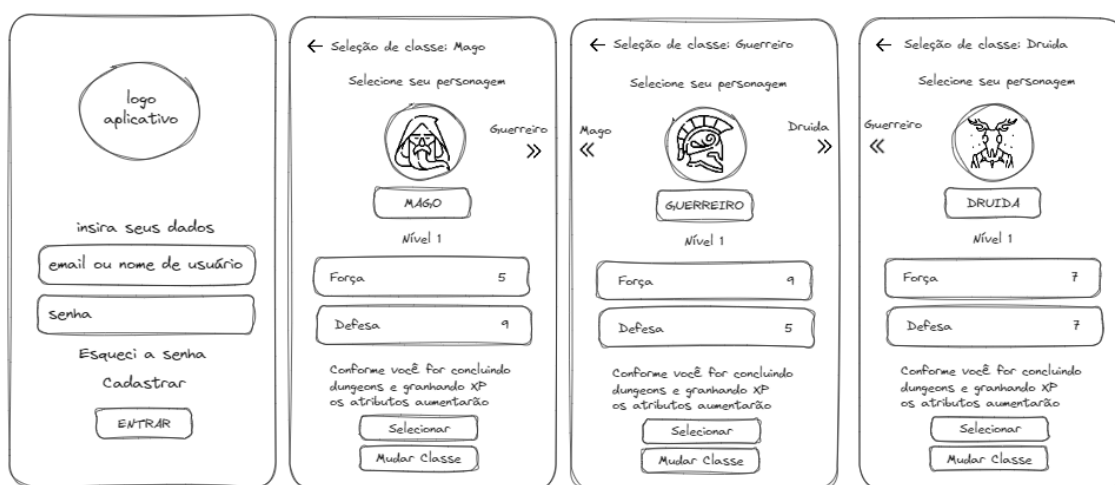
- a) Me cadastrar para poder ter uma conta no jogo;
- b) Fazer login para poder acessar o jogo;
- c) Criar meu perfil e personagem personalizado para facilitar a identificação e interatividade;
- d) Mudar a linguagem de programação usada para uma a qual tenho mais facilidade de entendimento;
- e) Buscar meus amigos através do nome de perfil para poder ver seus respectivos progressos;
- f) Ingressar nas Ilhas dos principais conteúdos de computação para ser introduzidas aos seus assuntos;
- g) Me vincular a Níveis de dificuldades da respectiva Ilha para estudar um assunto específico;
- h) Ter acesso a uma parte teórica no Nível para poder estudar antes de realizar a Dungeon Final;
- i) Ter acesso a exercícios resolvidos no Nível para entender, na prática, como funciona o código naquele assunto;
- j) Ter acesso a um fórum de dúvidas no Nível, para poder tirar minhas dúvidas específicas sobre o assunto;
- k) Acessar a Dungeon Final quando estiver pronto para terminar o Nível para colocar em prática todos meus conhecimentos aprendidos e testar minhas habilidades de programação;
- l) Ter acesso a dicas para que possa me ajudar durante a Dungeon Final;
- m) Ter acesso a feedbacks ao final da Dungeon Final para facilitar o entendimento dos meus erros e acertos, além da qualidade de desempenho do meu código;
- n) Ter acesso a uma aba, durante a Dungeon Final, que permita desistir e voltar para os exercícios extras, teoria e fórum de dúvidas para revisar o conteúdo;
- o) Ter acesso ao feedback geral da Ilha para ver o cálculo dos meus pontos de XP acumulados baseado no meu desempenho;
- p) Ter acesso a um ranking diário e geral comparado com o de outros usuários para que me motive a realizar as tarefas;

q) Ter acesso a tarefas diárias para me motivar a continuar progredindo no aplicativo;

2.2 PROTÓTIPOS BÁSICOS DO SISTEMA

O desenvolvimento do protótipo do sistema busca fornecer uma representação visual da lógica de funcionamento do software em uma fase avançada. Considerando que este protótipo é uma versão mais simples, a ênfase foi dada à exposição das telas principais do sistema.

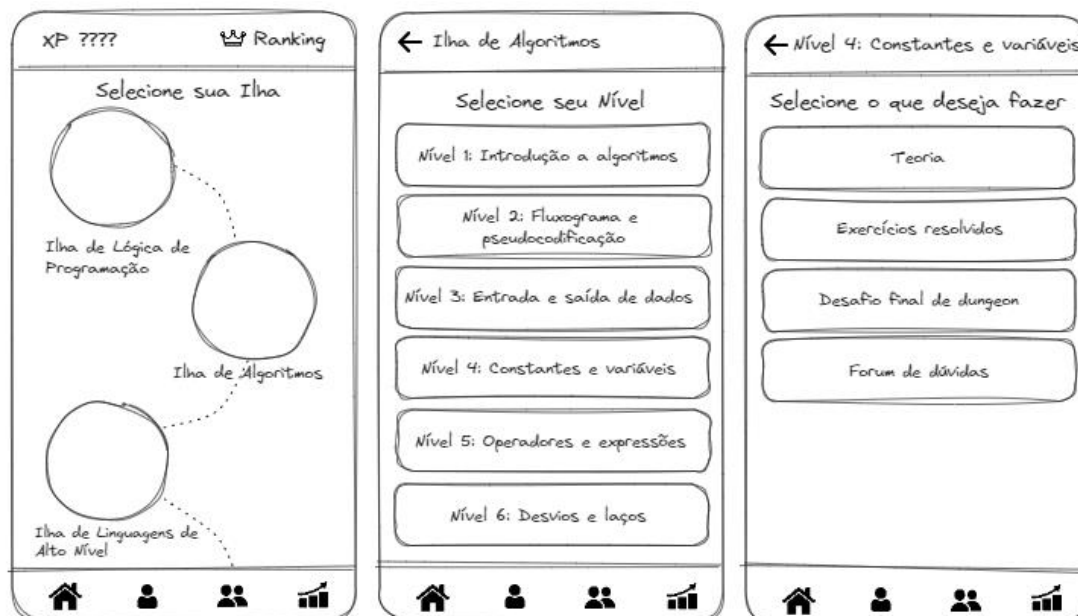
No início, é sugerida uma tela padrão destinada à execução do login. Na eventualidade de o usuário não possuir um login, é iniciado um processo de cadastro, destacando-se a inclusão da opção de escolha de classes. Neste contexto, o usuário tem a capacidade de optar entre três classes distintas: Mago, que apresenta uma Força inicial de 5 e uma Defesa inicial de 9; Guerreiro, com uma Força inicial de 9 e uma Defesa inicial de 5; e Druida, que mantém equilíbrio, possuindo uma Força e Defesa iniciais ambas avaliadas em 7.



Fonte: Desenvolvido pelos autores

Após o usuário realizar a escolha da classe desejada, é então apresentado ao sistema das Ilhas, as quais concentram os principais tópicos relacionados à computação. Nesse contexto, o usuário tem a oportunidade de selecionar uma Ilha específica, sendo, a partir desse ponto, introduzido a uma gama de sub-assuntos associados àquele conteúdo em particular. Ao optar por um sub-assunto, o usuário se depara com opções diversas, como explorar a

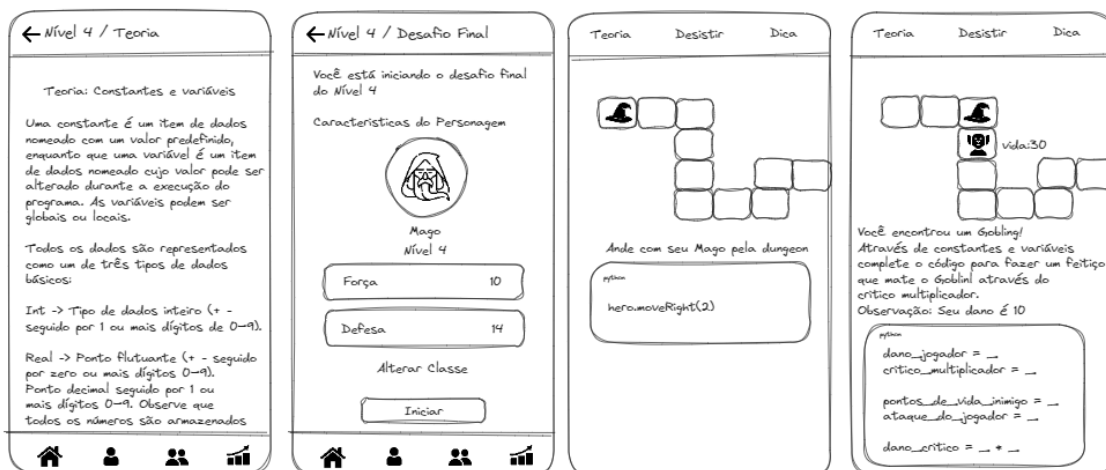
parte teórica, revisar exercícios resolvidos relacionados ao tema, avançar diretamente para o Desafio Final de Dungeon para completar o Nível, ou ainda, esclarecer suas dúvidas por meio do fórum disponibilizado.



Fonte: Desenvolvido pelos autores

Por último, são representadas neste protótipo básico duas das quatro funcionalidades do Nível: a seção teórica e a etapa final da Dungeon. A fase da Dungeon Final se inicia com uma revisão do estado atual do personagem, exibindo as quantidades de Força e Defesa vigentes, além de oferecer a opção de alternar para outro personagem antes de iniciar o jogo. Quando o usuário decidir começar, é imerso na dungeon, mantendo sempre a opção de acessar a teoria, desistir ou solicitar dicas.

Durante a exploração da Dungeon através do console, o usuário tem a capacidade de alterar a linguagem de programação conforme sua preferência. Ao se deparar com armadilhas ou confrontar monstros, é proposto um pequeno desafio associado ao tema específico do Nível. Essa abordagem proporciona uma experiência interativa e educativa, integrando os conceitos teóricos ao contexto prático da resolução de desafios.



Fonte: Desenvolvido pelos autores

3 ESTIMATIVA DE DURAÇÃO DO PROJETO COMPLETO

Para estimar a duração deste projeto, é empregado o Método Cocomo, fundamentado no conceito de análise de pontos por função. Tal abordagem visa quantificar cada um dos tipos de funções essenciais no processo de desenvolvimento, a saber: Entradas Externas (EE), Saídas Externas (SE), Consultas Externas (CE), Arquivos Lógicos Internos (ALI) e Arquivos de Interface Externa (AIE). Essa análise proporciona uma base sólida para avaliar o esforço e a complexidade associados a cada componente funcional do projeto, constituindo um elemento crucial na projeção temporal global.

a) Entradas Externas (EE)

Informações	Somatório
Códigos e comandos fornecidos pelo jogador para controlar o personagem ou realizar ações no jogo	1
Dados de usuário inseridos na tela de login	1
Cadastro para ingressar no jogo	1
Solicitações feitas pelo jogador: Escolha de classe, Ilhas, Níveis, teorias, Exercícios Extras e Dungeons	6
Somatório Total	9

b) Saídas Externas (SE)

Informações	Somatório
Interface do usuário interativa que muda conforme o erro/acerto do código digitado em tempo real	1
Relatório geral e diário de desempenho do usuário	2
Mensagem de erro caso o código digitado apresente erros que impeçam de ser executado	1
Dicas caso o usuário esteja enfrentando dificuldades em captar a lógica do enunciado	1
Atualização na interface dos pontos de XP do usuário ao final do Nível	1
Somatório Total	6

c) Consultas Externas (CE)

Informações	Somatório
O jogo pode fazer consultas a um servidor para salvar e carregar dados do jogador a partir de um banco de dados	1
Busca de usuários por perfil	1
Cálculo de XP acumulado em cada Ilha	1
Somatório Total	3

d) Arquivos Lógicos Internos (ALI)

Informações	Somatório
O jogo terá um motor de jogo interno que controla a lógica do jogo, ou seja: Movimentação do personagem/inimigo, informações de saúde e defesa dos jogadores	4
Banco de dados contendo as definições de Ilhas, Níveis, teorias, Exercícios Extras e Dungeons	5
Somatório Total	9

e) Arquivos de Interface Externa (AIE)

Dado que este software não está projetado para interagir com diversos sistemas ou componentes de software, torna-se desnecessária a inclusão de Arquivos de Interface Externa (AIE).

3.1 CÁLCULO PFNA

Analisando o quantitativo de cada função e considerando os parâmetros de medição de complexidade **MÉDIA**, é utilizado o seguinte processo para calcular os Pontos de Função Não-Ajustados (PFNA):

Função	Quantitativo x Complexidade Média	Resultado
EE	9 x 4	36
SE	6 x 5	30
CE	3 x 4	12
ALI	9 x 10	90
AIE	0 x 7	0

Através do somatório dos resultados, se obtém o valor total da contagem de PFNA como sendo 168.

Com base no resultado obtido para a quantidade de PFNA, é viável calcular a estimativa correspondente em KLOCs (milhares de linhas de código), tendo em mente que o projeto será desenvolvido na linguagem Python, adota-se a premissa de que 1 PFNA equivale a 53 linhas de código (LOCs):

27	Lisp	64.00	5.00		
28	Prolog	64.00	5.00		
29	SH (shell scripts)	64.00	5.00		
30	Quick Basic	60.95	5.25		
31	Zimbu	58.18	5.50		
32	C++	53.33	6.00		
33	Go	53.33	6.00		
34	Java	53.33	6.00		
35	PHP	53.33	6.00		
36	Python	53.33	6.00		
37	C#	51.20	6.25		
38	Xio	51.20	6.25		
39	Ada 95	49.23	6.50		

Fonte: Site Namcook Analytics

3.2 CÁLCULO KLOCs

COCOMO Básico				
Complexidade	α	β	ϵ	θ
Simples	2,4	1,05	2,5	0,38
Média	3,0	1,12	2,5	0,35
Alta	3,6	1,20	2,5	0,32

$$\text{Esforço (E)} = \alpha \cdot \text{KLOC}^\beta$$

$$\text{Tempo (T)} = \epsilon \cdot E^\theta$$

A formação do cálculo para KLOCs é delineada da seguinte maneira:
 $KLOC = (PFNA * LOCs)/1000$, efetuando a substituição pelos valores já conhecidos:

$$KLOC = \frac{163 * 53}{1000}$$

$$KLOC = 8.639$$

3.3 MÉTODO COCOMO

Com base nos KLOCs obtidos, é factível calcular o Esforço e a Duração do projeto empregando o método paramétrico COCOMO. Vale ressaltar que este projeto é classificado como de complexidade MÉDIA, além disso, para todos os

cálculos é utilizado uma precisão de duas casas decimais e arredondamento para cima quando a primeira casa decimal for igual a 5.

Fonte: Moodle da disciplina de Engenharia de Software da UDESC

Usando a formação do cálculo do Esforço delineado acima e efetuando a substituição pelos valores já conhecidos:

$$E = 3.0 * (8.639)^{1.12}$$

$$E = 33.57 \text{ pessoa mês}$$

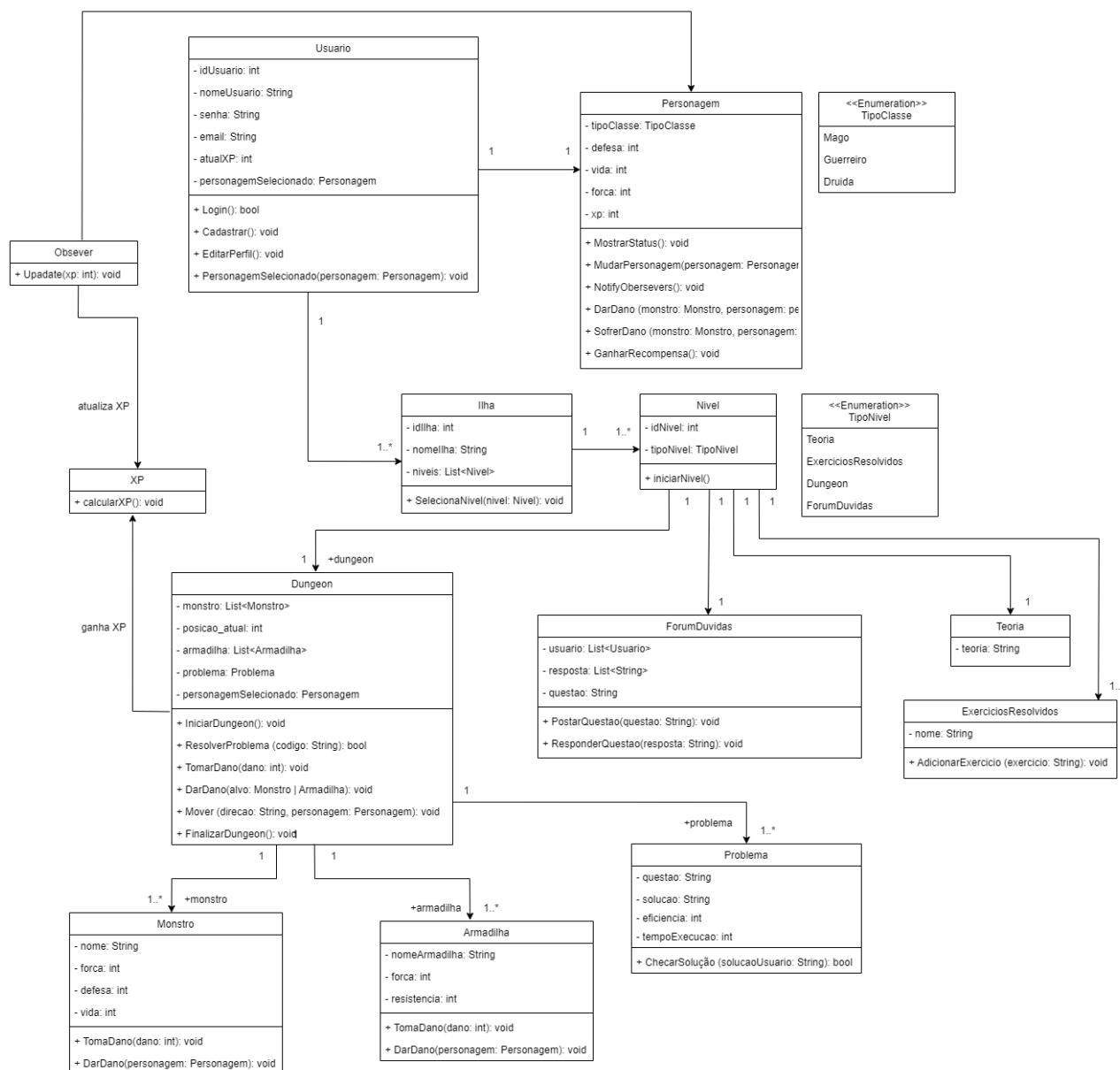
Por último, usando a formação do cálculo de Duração delineado acima e efetuando a substituição pelos valores já conhecidos:

$$D = 2.5 * (33.57)^{0.35}$$

$$D = 8.55 \text{ meses}$$

Desta forma, a duração do projeto será de aproximadamente 9 meses.

4 DIAGRAMA DE CLASSES DO PROJETO UML



Fonte: Desenvolvido pelos autores

O esquema acima engloba as principais categorias do projeto, destacando seus atributos e métodos, assim como as interações entre elas. A opção pelo emprego do padrão de projeto Observer justifica-se pela imperativa necessidade de informar o usuário acerca de quaisquer alterações nos cálculos relativos à experiência (XP). Tal necessidade surge em virtude das atualizações nos valores de defesa e força do personagem, as quais, por sua vez, são suscetíveis a modificações. Desta forma, é plausível inferir que o usuário demonstrará interesse nessas mudanças, visto que estas propiciam uma otimização na jogabilidade.

5 TESTES UNITARIOS

Os procedimentos referentes aos testes unitários foram conduzidos com base na análise da criticidade das classes, com especial atenção à implementação das classes mais suscetíveis a erros, nomeadamente a classe Dungeon, Personagem e Usuario. Foi utilizado o framework PyTest para executar esses testes.

Na classe Dungeon, foram introduzidos os conceitos de monstros, armadilhas e problemas diretamente na própria classe. Em uma abordagem mais aprofundada, essas implementações poderiam ser externalizadas para classes distintas. No entanto, a garantia do funcionamento na classe Dungeon estabelece uma base sólida. Um processo semelhante foi aplicado à classe Personagem, que incorpora os conceitos de força e defesa. Optou-se por não subdividir nas classes específicas de Mago, Guerreiro e Druida, visando uma implementação mais simplificada, dado que a funcionalidade da classe Personagem foi previamente assegurada.

As demais classes apresentam conceitos mais convencionais e não apresentam uma criticidade tão abrangente para o software em questão. Os testes unitários estão disponíveis para referência neste [link](#).

REFERÊNCIAS

MIDIA, F451. **Título:** O crescimento do mercado de TI e a crescente demanda por profissionais da área. Disponível em: <https://gizmodo.uol.com.br/o-crescimento-do-mercado-de-ti-e-a-crescente-demanda-por-profissionais-da-area/>. Acesso em: 23 nov. 2023.

PRADA, Charles. **Título:** Escopo do Projeto: passo a passo para montar o seu ainda hoje sem erros! Disponível em: <https://www.euax.com.br/2019/01/exemplo-de-escopo-de-projeto/>. Acesso em: 23 nov. 2023.

SOARES, Letícia; COSTA, Maria Janaína. **Título:** Game of Codes: Especificação de requisitos de um software pedagógico de introdução à lógica de programação. Disponível em: https://memoria.ifrn.edu.br/bitstream/handle/1044/2060/Games_of_Codes-Requisitos-Corrigido.pdf?sequence=1&isAllowed=y. Acesso em: 23 nov. 2023.

Título: Function Point Languages Table. Disponível em: <https://www.qsm.com/resources/function-point-languages-table>. Acesso em: 23 nov. 2023.

Título: Programação para crianças: importância, ferramentas e mais! Disponível em: <https://www.eniac.com.br/blog/programacao-para-criancas-importancia-ferramentas-e-mais#:~:text=A%20programação%20é%20considerada%20uma,para%20o%20futuro%20das%20crianças>. Acesso em: 23 nov. 2023.

Título: Observer em C++. Disponível em: <https://refactoring.guru/pt-br/design-patterns/observer/cpp/example>. Acesso em: 27 nov. 2023.

Título: Orientações básicas na elaboração de um diagrama de classes. Disponível em: <https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224>. Acesso em: 27 nov. 2023.

Título: Programming Languages. Disponível em: <https://namcook.com/how-srm-works/Case%20Example-6.pdf>. Acesso em: 30 nov. 2023.

Moodle UDESC disciplina de Engenharia de Software