Directory Structure:

```
README.md
css/
  base.css
  buttons.css
  character-cards.css
  character-creation.css
  companion.css
  footer.css
  forms.css
  header.css
  main.css
  modals.css
  reset.css
  responsive.css
  variables.css
generate_story.py
index.html
js/
  app.js
  companion.js
  jquery-companion.js
  storage.js
package.json
```

File: css\reset.css

==========

```
/* http://meyerweb.com/eric/tools/css/reset/
   v2.0 | 20110126
   License: none (public domain)
*/

html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
margin: 0;
padding: 0;
border: 0;
font-size: 100%;
font: inherit;
vertical-align: baseline;
}
/* HTML5 display-role reset for older browsers */
article, aside, details, figcaption, figure,
footer, header, hgroup, menu, nav, section {
display: block;
}
body {
line-height: 1;
}
ol, ul {
list-style: none;
}
blockquote, q {
quotes: none;
}
blockquote:before, blockquote:after,
q:before, q:after {
content: '';
content: none;
}
table {
border-collapse: collapse;
border-spacing: 0;
}
```

File: js\app.js

==========

```
// Elementos DOM
const characterForm = document.getElementById('characterForm');
const rollAttributesBtn = document.getElementById('rollAttributes');
const generateLoreBtn = document.getElementById('generateLore');
const saveCharacterBtn = document.getElementById('saveCharacter');
const clearFormBtn = document.getElementById('clearForm');
const savedCharactersList = document.getElementById('savedCharactersList');
const characterModal = document.getElementById('characterModal');
const loadingModal = document.getElementById('loadingModal');
const closeModalBtn = document.getElementById('closeModal');
const deleteCharacterBtn = document.getElementById('deleteCharacter');
const editCharacterBtn = document.getElementById('editCharacter');
const characterDetails = document.getElementById('characterDetails');
const loreText = document.getElementById('loreText');

// Dados de estado
let characters = [];
let currentCharacterId = null;

// Classe para ícones de cada classe de personagem
const classIcons = {
    'Arcanista': 'fa-hat-wizard',
    'Bárbaro': 'fa-gavel',
    'Bardo': 'fa-guitar',
    'Bucaneiro': 'fa-sailboat',
    'Caçador': 'fa-shoe-prints',
    'Cavaleiro': 'fa-horse-head',
    'Clérigo': 'fa-pray',
    'Druida': 'fa-leaf',
    'Guerreiro': 'fa-jedi',
    'Inventor': 'fa-tools',
    'Ladino': 'fa-mask',
    'Lutador': 'fa-fist-raised',
    'Nobre': 'fa-crown',
    'Paladino': 'fa-shield-alt'
};

// Referência ao storage
const storage = characterStorage;

// Inicialização
document.addEventListener('DOMContentLoaded', function() {
    setupEventListeners();
    renderCharactersList();
    if (typeof magoCompanion !== 'undefined') {
        magoCompanion.init();
    } else {
        console.error('magoCompanion não está definido. Verifique companion.js');
    }
```

```
});

// Configuração de listeners de eventos
function setupEventListeners() {
    saveCharacterBtn.addEventListener('click', saveCharacter);
    clearFormBtn.addEventListener('click', clearForm);
    rollAttributesBtn.addEventListener('click', rollAttributes);
    generateLoreBtn.addEventListener('click', generateCharacterLore);
    document.querySelectorAll('.modal .delete, #closeModal').forEach(closeBtn => {
        closeBtn.addEventListener('click', () => closeModal(characterModal));
    });
    deleteCharacterBtn.addEventListener('click', deleteCurrentCharacter);
    editCharacterBtn.addEventListener('click', editCurrentCharacter);
}

// Carregar personagens do localStorage
function loadCharacters() {
    const savedCharacters = localStorage.getItem('rpg_characters');
    if (savedCharacters) {
        characters = JSON.parse(savedCharacters);
        renderCharactersList();
    }
}

// Manipuladores de eventos
function saveCharacter(e) {
    e.preventDefault();
    const name = document.getElementById('charName').value.trim();
    if (!name) {
        showMessage('Por favor, dê um nome ao seu personagem.', 'is-danger');
        return;
    }

    const character = {
        id: currentCharacterId || null,
        name: name,
        race: document.getElementById('charRace').value,
        class: document.getElementById('charClass').value,
        alignment: document.getElementById('charAlignment').value,
        attributes: {
            strength: parseInt(document.getElementById('attrStr').value),
            dexterity: parseInt(document.getElementById('attrDex').value),
            constitution: parseInt(document.getElementById('attrCon').value),
            intelligence: parseInt(document.getElementById('attrInt').value),
            wisdom: parseInt(document.getElementById('attrWis').value),
            charisma: parseInt(document.getElementById('attrCha').value)
        },
        background: document.getElementById('charBackground').value
    };

    const savedCharacter = storage.saveCharacter(character);
    renderCharactersList();
```

```
    // Resetar o formulário e o botão de salvar
    clearForm();
    saveCharacterBtn.innerHTML = '<i class="fas fa-save"></i>  Salvar Personagem';
    saveCharacterBtn.classList.remove('is-warning');
    saveCharacterBtn.style.color = '';

    // Feedback baseado em se foi uma criação ou atualização
    const isUpdate = currentCharacterId !== null;
    currentCharacterId = null;

    if (isUpdate) {
        showMessage(`Personagem ${savedCharacter.name} atualizado com sucesso!`, 'is-success');
        magoCompanion.speak(`${savedCharacter.name} foi atualizado em seu grimório!`, 4000);
    } else {
        showMessage(`Personagem ${savedCharacter.name} salvo com sucesso!`, 'is-success');
        magoCompanion.speak(`${savedCharacter.name} foi adicionado ao seu grimório de heróis!`, 4000);
    }

    document.querySelector('.saved-characters-section').scrollIntoView({ behavior: 'smooth' });
}

function rollAttributes() {
    function rollStat() {
        const rolls = Array.from({length: 4}, () => Math.floor(Math.random() * 6) + 1);
        rolls.sort((a, b) => a - b);
        return rolls.slice(1).reduce((sum, roll) => sum + roll, 0);
    }
    document.getElementById('attrStr').value = rollStat();
    document.getElementById('attrDex').value = rollStat();
    document.getElementById('attrCon').value = rollStat();
    document.getElementById('attrInt').value = rollStat();
    document.getElementById('attrWis').value = rollStat();
    document.getElementById('attrCha').value = rollStat();
}

function renderCharactersList() {
    const characters = storage.getAllCharacters();
    console.log('Personagens carregados:', characters); // Debug
    if (characters.length === 0) {
        savedCharactersList.innerHTML = '<p class="empty-list-message">Nenhum herói criado ainda. Comece a forjar sua lenda!
        return;
    }


    savedCharactersList.innerHTML = '';
    characters.forEach(character => {
        const characterCard = document.createElement('div');
        characterCard.className = 'character-card';
        characterCard.dataset.id = character.id;
        const classIcon = classIcons[character.class] || 'fa-user';
        characterCard.innerHTML = `
            <div class="has-text-centered mb-3">
                <div class="character-avatar">
```

```
                <i class="fas ${classIcon}"></i>
            </div>
        </div>
        <div class="has-text-centered">
            <p class="title is-5 medieval-title">${character.name}</p>
            <p class="subtitle is-6">${character.race} ${character.class}</p>
            <p class="is-size-7">${formatDate(character.createdAt)}</p>
        </div>
    `;
    characterCard.addEventListener('click', () => openCharacterModal(character.id));
    savedCharactersList.appendChild(characterCard);
  });
}

function formatDate(dateString) {
  const date = new Date(dateString);
  return date.toLocaleDateString('pt-BR');
}

function openCharacterModal(characterId) {
  const character = storage.getCharacterById(characterId);
  if (!character) {
    console.error('Personagem não encontrado:', characterId);
    return;
  }

  currentCharacterId = characterId;
  characterDetails.innerHTML = `
    <div class="columns is-multiline">
      <div class="column is-8">
        <h3 class="title is-3 medieval-title">${character.name}</h3>
        <p class="subtitle is-5">${character.race} ${character.class} (${character.alignment})</p>
        <div class="content mt-4">
          <h4 class="title is-5 medieval-title">Atributos</h4>
          <div class="columns is-multiline">
            <div class="column is-6">
              <p><strong>Força:</strong> ${character.attributes.strength}</p>
              <p><strong>Destreza:</strong> ${character.attributes.dexterity}</p>
              <p><strong>Constituição:</strong> ${character.attributes.constitution}</p>
            </div>
            <div class="column is-6">
              <p><strong>Inteligência:</strong> ${character.attributes.intelligence}</p>
              <p><strong>Sabedoria:</strong> ${character.attributes.wisdom}</p>
              <p><strong>Carisma:</strong> ${character.attributes.charisma}</p>
            </div>
          </div>
        </div>
      </div>
      <div class="column is-4 has-text-centered">
        <div class="character-avatar" style="width: 120px; height: 120px; margin: 0 auto;">
          <i class="fas ${classIcons[character.class] || 'fa-user'}" style="font-size: 4rem;"></i>
        </div>
        <p class="is-size-7 mt-3">Criado em ${formatDate(character.createdAt)}</p>
```

```
            </div>
            ${character.background ? `
                <div class="column is-12">
                    <div class="content mt-4">
                        <h4 class="title is-5 medieval-title">História de Fundo</h4>
                        <p>${character.background}</p>
                    </div>
                </div>
            ` : ''}
        </div>
    `;
    document.getElementById('loreContent').style.display = 'none';

    // Configurar os botões do modal
    const editBtn = document.getElementById('editCharacter');
    const deleteBtn = document.getElementById('deleteCharacter');
    const closeBtn = document.getElementById('closeModal');

    // Remover event listeners antigos para evitar duplicação
    editBtn.replaceWith(editBtn.cloneNode(true));
    deleteBtn.replaceWith(deleteBtn.cloneNode(true));
    closeBtn.replaceWith(closeBtn.cloneNode(true));

    // Adicionar novos event listeners
    document.getElementById('editCharacter').addEventListener('click', editCurrentCharacter);
    document.getElementById('deleteCharacter').addEventListener('click', deleteCurrentCharacter);
    document.getElementById('closeModal').addEventListener('click', () => closeModal(characterModal));

    openModal(characterModal);
}

function editCurrentCharacter() {
    if (!currentCharacterId) return;

    const character = storage.getCharacterById(currentCharacterId);
    if (!character) {
        showMessage('Personagem não encontrado.', 'is-danger');
        return;
    }

    // Preencher o formulário com os dados do personagem
    document.getElementById('charName').value = character.name;
    document.getElementById('charRace').value = character.race;
    document.getElementById('charClass').value = character.class;
    document.getElementById('charAlignment').value = character.alignment;
    document.getElementById('attrStr').value = character.attributes.strength;
    document.getElementById('attrDex').value = character.attributes.dexterity;
    document.getElementById('attrCon').value = character.attributes.constitution;
    document.getElementById('attrInt').value = character.attributes.intelligence;
    document.getElementById('attrWis').value = character.attributes.wisdom;
    document.getElementById('attrCha').value = character.attributes.charisma;
    document.getElementById('charBackground').value = character.background || '';
```

```
    // Atualizar o botão de salvar para indicar que é uma edição
    saveCharacterBtn.innerHTML = '<i class="fas fa-save"></i>  Atualizar Personagem';
    saveCharacterBtn.classList.add('is-warning');

    // Garantir que o texto permaneça branco
    saveCharacterBtn.style.color = 'white';

    // Fechar o modal e rolar para o formulário
    closeModal(characterModal);
    characterForm.scrollIntoView({ behavior: 'smooth' });

    // Feedback visual
    showMessage('Editando personagem: ' + character.name, 'is-info');
    if (typeof magoCompanion !== 'undefined') {
        magoCompanion.speak(`Editando ${character.name}. Faça suas alterações!`, 3000);
    }
}

function deleteCurrentCharacter() {
    if (!currentCharacterId) return;

    // Confirmar exclusão
    if (confirm('Tem certeza que deseja excluir este personagem? Esta ação não pode ser desfeita.')) {
        const character = storage.getCharacterById(currentCharacterId);
        if (!character) {
            showMessage('Personagem não encontrado.', 'is-danger');
            return;
        }

        // Excluir o personagem
        const success = storage.deleteCharacter(currentCharacterId);

        if (success) {
            // Atualizar a lista de personagens
            renderCharactersList();

            // Fechar o modal
            closeModal(characterModal);

            // Feedback visual
            showMessage(`Personagem ${character.name} excluído com sucesso!`, 'is-warning');
            if (typeof magoCompanion !== 'undefined') {
                magoCompanion.speak(`${character.name} foi removido do seu grimório de heróis!`, 3000);
            }

            // Resetar o ID atual
            currentCharacterId = null;
        } else {
            showMessage('Erro ao excluir personagem.', 'is-danger');
        }
    }
}
```

```
// Função para abrir o modal de carregamento, centralizado na viewport
function openLoadingModal() {
    if (!loadingModal) {
        console.error('Modal de carregamento não encontrado');
        return;
    }

    loadingModal.classList.add('is-active');

    const modalContent = loadingModal.querySelector('.modal-content');
    if (modalContent) {
        const viewportHeight = window.innerHeight;
        const viewportWidth = window.innerWidth;
        const modalHeight = modalContent.offsetHeight;
        const modalWidth = modalContent.offsetWidth;

        const topPosition = (viewportHeight - modalHeight) / 2 + window.scrollY;
        const leftPosition = (viewportWidth - modalWidth) / 2 + window.scrollX;

        modalContent.style.top = `${topPosition}px`;
        modalContent.style.left = `${leftPosition}px`;
        modalContent.style.transform = 'none';
        modalContent.style.opacity = '0';

        setTimeout(() => {
            modalContent.style.opacity = '1';
        }, 50);
    }
}

// Função para gerar o prompt com contexto de raças e classes
function generatePrompt(characterData) {
    const raceSummaries = {
        'Humano': 'Versátil e adaptável, sem bônus ou penalidades especiais. Representam a maioria em Arton, com grande prese
        'Anão': 'Robustos e resilientes, com bônus em Constituição e penalidade em Carisma. Habitam montanhas como Doherimn
        'Dahllan': 'Semelhantes a meio-elfos, ligados à natureza, com bônus em Sabedoria e Destreza, penalidade em Inteligência.
        'Elfo': 'Graciosos e longevos, com bônus em Destreza e Inteligência, penalidade em Constituição. Vivem em florestas como
        'Goblin': 'Pequenos e astutos, com bônus em Destreza, penalidades em Força ou Carisma. Especializados em furtividade e
        'Lefou': 'Meio-demônios afetados pela Tormenta, com bônus em dois atributos (exceto Carisma) e penalidade em Carisma.
        'Minotauro': 'Fortes e ferozes, com bônus em Força e Constituição, penalidades em Inteligência ou Carisma. Excelentes em
        'Qareen': 'Meio-gênios com habilidades mágicas, bônus em Carisma e Inteligência, penalidade em Sabedoria. Podem conc
        'Golem': 'Seres artificiais, com alta resistência e força, sem necessidade de sono ou comida. Podem ter imunidades a certo
        'Hynne': 'Pequenas criaturas mágicas, com bônus em Destreza e Carisma, penalidade em Força. Têm habilidades de sorte
        'Kliren': 'Semelhantes a gnomos, com bônus em Inteligência, penalidade em Força. Têm habilidades lógicas, como usar Int
        'Medusa': 'Inspiradas na mitologia, com possíveis habilidades de petrificação, bônus em Carisma ou Destreza. Têm aparên
        'Osteon': 'Esqueletos animados, possivelmente não mortos, com imunidades a efeitos como sono, bônus em Constituição.
        'Sereia/Tritão': 'Seres aquáticos, com bônus em Destreza e Constituição, habilidades de natação e sobrevivência na água,
        'Sílfide': 'Relacionadas ao ar e vento, com bônus em Destreza e Inteligência, habilidades mágicas ligadas ao elemento, leve
        'Suraggel': 'Descendentes de anjos ou demônios, com bônus em Sabedoria ou Carisma, habilidades baseadas em sua her
        'Trong': 'Provavelmente trolls ou criaturas similares, com bônus em Força e Constituição, habilidades de regeneração, resis
    };

    const classSummaries = {
```

```
        'Arcanista': 'Especialista em magias arcanas, lança feitiços ofensivos e utilitários, como estudante da Academia Arcana de
        'Bárbaro': 'Guerreiro focado em força e fúria, ganha bônus em dano e resistência ao entrar em raiva, ideal para combates b
        'Bardo': 'Carismático, usa música e performance para inspirar aliados e manipular inimigos, lança magias leves, central em
        'Bucaneiro': 'Espadachim e navegador, com habilidades de acrobacia e pirataria, proficiente em espadas e navegação, com
        'Caçador': 'Rastreador e arqueiro, com habilidades de sobrevivência e manejo de animais, protege fronteiras, como nas flo
        'Cavaleiro': 'Cavaleiro montado, focado em combate com armadura pesada e lança, segue código de cavalaria, central em
        'Clérigo': 'Sacerdote devoto, cura aliados e combate com magias divinas, ligado a deuses como Khalmyr, essencial em apo
        'Druida': 'Conectado à natureza, lança magias baseadas em elementos, pode se transformar, protege florestas, como em L
        'Guerreiro': 'Combatente geral, proficiente em armas e armaduras, equilibrado sem especializações, comum em batalhas d
        'Inventor': 'Tecnólogo, cria gadgets e máquinas, usa dispositivos em combate e exploração, inovador, central em estratégia
        'Ladino': 'Ladrão furtivo, especialista em furtividade, armadilhas e ataques sorrateiros, comum em ruas de Malpetrim, centra
        'Lutador': 'Combatente corpo a corpo, focado em artes marciais, proficiente em lutas desarmadas, ágil em combate próximo
        'Nobre': 'Classe de status social, com liderança e influência, pode ter treinamento em combate, central em política e diplom
        'Paladino': 'Guerreiro sagrado, segue código de honra, lança magias divinas, combate o mal, como em Yuden, central em
    };

    const raceSummary = raceSummaries[characterData.race] || raceSummaries['Humano'];
    const classSummary = classSummaries[characterData.class] || classSummaries['Guerreiro'];

    return `
        Gere uma história curta (100-200 palavras) para um personagem de RPG no mundo de Arton, do sistema Tormenta 20. Ac
        - Nome: ${characterData.name}
        - Raça: ${characterData.race} (${raceSummary})
        - Classe: ${characterData.class} (${classSummary})
        - Alinhamento: ${characterData.alignment}
        - Atributos: Força ${characterData.attributes.strength}, Destreza ${characterData.attributes.dexterity}, Constituição ${charac
        A história deve ser escrita em português, em um estilo de fantasia épica, refletindo o lore de Arton e as características da r
    `;
}

// Função para chamar o servidor local Python
async function fetchBackstoryFromLocal(prompt) {
    try {
        const response = await fetch('http://127.0.0.1:5000/generate', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json'
            },
            body: JSON.stringify({ prompt: prompt })
        });

        if (!response.ok) {
            throw new Error('Erro na requisição ao servidor local: ' + response.statusText);
        }

        const data = await response.json();
        console.log('Resposta do servidor local:', data);

        if (data.error) {
            throw new Error(data.error);
        }

        return data.backstory || null;
```

```
    } catch (error) {
        console.error('Erro ao gerar história localmente:', error);
        return null;
    }
}

// Função antiga como fallback
function generateSimpleLore(character) {
    const raceTemplates = {
        'Humano': [
            `Nascido na cidade de Valkaria, ${character.name} cresceu sob a proteção da Deusa da Humanidade.`,
            `Originário de Nova Malpetrim, ${character.name} enfrentou os perigos da Tormenta desde jovem.`,
            `Vindo de Tamu-ra, ${character.name} deixou sua vila para buscar um destino maior em Arton.`
        ],
        'Anão': [
            `${character.name} foi forjado nas minas de Doherimm, o reino anão sob as montanhas.`,
            `Nascido em Khalifor, ${character.name} carrega o orgulho de sua linhagem de ferreiros.`,
            `Criado entre martelos e bigornas, ${character.name} saiu de uma fortaleza anã para explorar Arton.`
        ],
        'Dahllan': [
            `${character.name} nasceu em um bosque sagrado de Lenórienn, conectada às forças da natureza.`,
            `Criada entre flores e espíritos selvagens, ${character.name} protege os segredos de Allihanna.`,
            `Vinda de florestas intocadas, ${character.name} dança com as plantas em sua jornada por Arton.`
        ],
        'Elfo': [
            `${character.name} nasceu nas florestas de Lenórienn, lar ancestral dos elfos de Arton.`,
            `Vindo da corte de Allania, ${character.name} carrega as tradições mágicas de seu povo.`,
            `${character.name} cresceu entre as árvores de Gilfien, mestre dos arcos e feitiços.`
        ],
        'Goblin': [
            `Nascido em uma toca em Tollon, ${character.name} sobreviveu com astúcia e malícia.`,
            `${character.name} cresceu entre os goblins de Zakharov, roubando para provar seu valor.`,
            `Vindo das ruínas de Arton, ${character.name} é um goblin esperto fugindo da Tormenta.`
        ],
        'Lefou': [
            `${character.name} emergiu de uma área rubra em Lamnor, marcado pela Tormenta.`,
            `Criado entre sombras e pesadelos, ${character.name} carrega o fardo de sua origem demoníaca.`,
            `Vindo das terras corruptas, ${character.name} busca redenção ou poder em Arton.`
        ],
        'Minotauro': [
            `${character.name} nasceu nas estepes de Galrasia, criado entre guerreiros selvagens.`,
            `Com chifres imponentes, ${character.name} deixou sua tribo em busca de glória em Lamnor.`,
            `${character.name} cresceu em cavernas de Trebuck, temido por sua força bruta.`
        ],
        'Qareen': [
            `${character.name} nasceu em um oásis de Wynlla, herdeiro de desejos mágicos.`,
            `Vindo de um portal em Vectora, ${character.name} encanta com seu sangue de gênio.`,
            `${character.name} cresceu em mercados de Yuden, negociando feitiços e promessas.`
        ],
        'Golem': [
            `${character.name} foi ativado em um laboratório esquecido de Deheon, feito de pedra e magia.`,
            `Sem memórias de criação, ${character.name} vaga por Arton em busca de propósito.`,
            `Construído em Norm, ${character.name} é uma relíquia viva de eras passadas.`
```

```
    ],
    'Hynne': [
        `${character.name} nasceu em Tapista, entre halflings e suas festas alegres.`,
        `Criado em Fortuna, ${character.name} usa sua sorte para escapar de perigos.`,
        `Vindo de Pequeno Povo, ${character.name} é pequeno mas cheio de bravura.`
    ],
    'Kliren': [
        `${character.name} nasceu em Triunfo, obcecado por enigmas e invenções.`,
        `Criado em Vectora, ${character.name} resolve problemas com sua mente afiada.`,
        `Vindo de um vilarejo em Deheon, ${character.name} é um mestre da lógica.`
    ],
    'Medusa': [
        `${character.name} surgiu em cavernas de Sckharshantallas, com serpentes como companhia.`,
        `Vinda de ruínas em Lamnor, ${character.name} encanta e petrifica com seu olhar.`,
        `${character.name} cresceu isolada em Trebuck, temida por sua aparência exótica.`
    ],
    'Osteon': [
        `${character.name} foi animado em um ritual sombrio em Galrasia, ossos sem carne.`,
        `Vindo de cemitérios de Malpetrim, ${character.name} busca entender sua existência.`,
        `Criado por necromancia em Vectora, ${character.name} é um eco da morte em Arton.`
    ],
    'Sereia/Tritão': [
        `${character.name} emergiu das águas de Portsmonth, guardião dos segredos do mar.`,
        `Nascido em recifes de Tamu-ra, ${character.name} canta com as ondas.`,
        `Vindo do oceano de Arton, ${character.name} explora a terra com curiosidade.`
    ],
    'Sílfide': [
        `${character.name} nasceu nas nuvens de Wynlla, leve como o vento.`,
        `Criada entre brisas de Lenórienn, ${character.name} dança com os elementos.`,
        `Vinda de picos de Yuden, ${character.name} voa livremente por Arton.`
    ],
    'Suraggel': [
        `${character.name} descende de celestiais em Thyatis, tocado por luz divina.`,
        `Nascido em sombras de Lamnor, ${character.name} carrega um legado infernal.`,
        `Vindo de um plano espiritual, ${character.name} busca equilíbrio em Arton.`
    ],
    'Trong': [
        `${character.name} nasceu em pântanos de Sckharshantallas, regenerando-se a cada ferida.`,
        `Criado em cavernas de Galrasia, ${character.name} é uma força da natureza.`,
        `Vindo de florestas selvagens, ${character.name} ruge contra os inimigos de Arton.`
    ]
};

const classTemplates = {
    'Arcanista': [
        `Estudante da Academia Arcana de Yuden, ${character.name} domina os segredos da magia.`,
        `Após um tomo proibido em Vectora, ${character.name} busca poder arcano perdido.`,
        `${character.name} explora Arton, lançando feitiços contra a Tormenta.`
    ],
    'Bárbaro': [
        `Criado nas estepes de Lamnor, ${character.name} entra em fúria contra seus inimigos.`,
        `Vindo de Galrasia, ${character.name} luta com a força de um titã selvagem.`,
        `${character.name} protege sua tribo em Trebuck com gritos de guerra.`
```

```
],
'Bardo': [
    `${character.name} canta lendas em tavernas de Norm, encantando multidões.`,
    `Formado em Triunfo, ${character.name} inspira aliados com suas canções.`,
    `Viajante de Deheon, ${character.name} coleta histórias de Arton.`
],
'Bucaneiro': [
    `${character.name} navegou os mares de Portsmonth, espada na mão.`,
    `Criado entre piratas de Tamu-ra, ${character.name} dança com lâminas.`,
    `${character.name} busca tesouros em Vectora, mestre da agilidade.`
],
'Caçador': [
    `${character.name} rastreia presas em Sckharshantallas, arco em punho.`,
    `Treinado em Bielefeld, ${character.name} protege as fronteiras de Arton.`,
    `${character.name} caça criaturas da Tormenta com instintos afiados.`
],
'Cavaleiro': [
    `${character.name} cavalga em Vectora, seguindo um código de honra.`,
    `Nascido em Norm, ${character.name} luta com lança e armadura pesada.`,
    `${character.name} defende o Reinado com sua montaria leal.`
],
'Clérigo': [
    `${character.name} serve o Pantheon em Valkaria, canalizando poder divino.`,
    `Criado em Thyatis, ${character.name} cura aliados com fé inabalável.`,
    `${character.name} combate o mal em Trebuck com bênçãos sagradas.`
],
'Druida': [
    `${character.name} foi iniciado em Lenórienn, guardião da natureza.`,
    `Vindo de Tamu-ra, ${character.name} fala com os espíritos da floresta.`,
    `${character.name} transforma-se para proteger Arton do desequilíbrio.`
],
'Guerreiro': [
    `Treinado em Norm, ${character.name} é um mestre das armas de Arton.`,
    `Veterano da Tormenta, ${character.name} busca glória em combate.`,
    `${character.name} protege os fracos com espada e escudo em Deheon.`
],
'Inventor': [
    `${character.name} criou máquinas em Vectora, gênio da tecnologia.`,
    `Nascido em Triunfo, ${character.name} usa gadgets contra a Tormenta.`,
    `${character.name} explora Arton com invenções revolucionárias.`
],
'Ladino': [
    `${character.name} cresceu em Malpetrim, mestre do furtos e sombras.`,
    `Membro dos Ladrões de Deheon, ${character.name} vive na clandestinidade.`,
    `${character.name} engana inimigos em Portsmonth com astúcia.`
],
'Lutador': [
    `${character.name} lutou em arenas de Yuden, punhos como armas.`,
    `Criado em Norm, ${character.name} domina artes marciais.`,
    `${character.name} enfrenta desafios em Arton com força bruta.`
],
'Nobre': [
    `${character.name} nasceu em Valkaria, líder por direito de sangue.`,
```

```
        `Criado em Vectora, ${character.name} comanda com carisma e espada.`,
        `${character.name} negocia paz em Deheon, herdeiro de um legado.`
    ],
    'Paladino': [
        `${character.name} jurou em Yuden combater a Tormenta com justiça.`,
        `Escolhido por Khalmyr, ${character.name} brilha como farol em Arton.`,
        `${character.name} carrega uma armadura sagrada em Norm.`
    ]
};

const alignmentTemplates = {
    'Leal e Bom': [
        `Guiado por honra, ${character.name} protege os inocentes de Arton.`,
        `Com justiça no coração, ${character.name} enfrenta o mal sem hesitar.`
    ],
    'Neutro e Bom': [
        `${character.name} ajuda quem precisa, sem se prender a regras rígidas.`,
        `Seguindo seu coração, ${character.name} faz o bem em Arton.`
    ],
    'Caótico e Bom': [
        `${character.name} age por bondade, desafiando leis injustas.`,
        `Livre e generoso, ${character.name} busca o bem à sua maneira.`
    ],
    'Leal e Neutro': [
        `A ordem é essencial para ${character.name}, mesmo em escolhas difíceis.`,
        `${character.name} mantém a estrutura do Reinado com devoção.`
    ],
    'Neutro': [
        `${character.name} busca equilíbrio, julgando cada caso em Arton.`,
        `${character.name} vive conforme a situação exige, sem extremos.`
    ],
    'Caótico e Neutro': [
        `${character.name} valoriza sua liberdade acima de tudo em Arton.`,
        `${character.name} segue seus instintos, rejeitando restrições.`
    ],
    'Leal e Mau': [
        `${character.name} segue um código egoísta, servindo seus interesses.`,
        `Calculista, ${character.name} usa a ordem para seu ganho em Arton.`
    ],
    'Neutro e Mau': [
        `${character.name} busca seus objetivos sem se importar com outros.`,
        `Pragmático, ${character.name} prioriza o poder em Arton.`
    ],
    'Caótico e Mau': [
        `${character.name} espalha caos e destruição por onde passa.`,
        `${character.name} rejeita autoridade, vivendo para seus desejos sombrios.`
    ]
};

const raceOptions = raceTemplates[character.race] || raceTemplates['Humano'];
const classOptions = classTemplates[character.class] || classTemplates['Guerreiro'];
const alignmentOptions = alignmentTemplates[character.alignment] || alignmentTemplates['Neutro'];
```

```
    const raceText = raceOptions[Math.floor(Math.random() * raceOptions.length)];
    const classText = classOptions[Math.floor(Math.random() * classOptions.length)];
    const alignmentText = alignmentOptions[Math.floor(Math.random() * alignmentOptions.length)];

    let attributeTexts = [];
    if (character.attributes.strength >= 14) {
        attributeTexts.push(`A força de ${character.name} impressiona até os guerreiros de Lamnor.`);
    } else if (character.attributes.strength <= 8) {
        attributeTexts.push(`${character.name} compensa sua fraqueza com outras habilidades.`);
    }
    if (character.attributes.intelligence >= 14) {
        attributeTexts.push(`A mente de ${character.name} rivaliza com os sábios de Vectora.`);
    } else if (character.attributes.intelligence <= 8) {
        attributeTexts.push(`${character.name} prefere ações práticas a pensamentos complexos.`);
    }
    if (character.attributes.charisma >= 14) {
        attributeTexts.push(`O carisma de ${character.name} atrai seguidores em todo o Reinado.`);
    } else if (character.attributes.charisma <= 8) {
        attributeTexts.push(`${character.name} evita multidões, preferindo a solidão.`);
    }
    const attributeText = attributeTexts.length > 0 ? attributeTexts[Math.floor(Math.random() * attributeTexts.length)] : '';

    const adventureHooks = [
        `${character.name} busca enfrentar os horrores da Tormenta em Arton.`,
        `Uma profecia em Triunfo aponta ${character.name} como chave para o Reinado.`,
        `Um artefato em Lamnor chama ${character.name} em sonhos sombrios.`,
        `${character.name} explora Deheon em busca de respostas sobre a Tormenta.`,
        `Após perder tudo para a Tormenta, ${character.name} jurou vingança.`,
        `Rumores de um tesouro em Vectora levam ${character.name} a uma jornada.`
    ];
    const adventureText = adventureHooks[Math.floor(Math.random() * adventureHooks.length)];

    return `${raceText} ${classText} ${alignmentText} ${attributeText} ${adventureText}`;
}

// Função atualizada para gerar a história com o servidor local e corrigir o textArea
async function generateCharacterLore() {
    let characterData;
    if (currentCharacterId) {
        characterData = storage.getCharacterById(currentCharacterId);
    } else {
        characterData = {
            name: document.getElementById('charName').value,
            race: document.getElementById('charRace').value,
            class: document.getElementById('charClass').value,
            alignment: document.getElementById('charAlignment').value,
            attributes: {
                strength: parseInt(document.getElementById('attrStr').value),
                dexterity: parseInt(document.getElementById('attrDex').value),
                constitution: parseInt(document.getElementById('attrCon').value),
                intelligence: parseInt(document.getElementById('attrInt').value),
                wisdom: parseInt(document.getElementById('attrWis').value),
                charisma: parseInt(document.getElementById('attrCha').value)
```

```
            },
            background: document.getElementById('charBackground').value
        };
    }

    if (!characterData.name) {
        showMessage('Por favor, dê um nome ao seu personagem antes de gerar sua história.', 'is-danger');
        return;
    }

    openLoadingModal();
    if (typeof magoCompanion !== 'undefined') {
        magoCompanion.speak("Deixe-me consultar os pergaminhos antigos...", 3000);
    }

    const prompt = generatePrompt(characterData);
    console.log('Prompt enviado:', prompt);
    const backstoryFromLocal = await fetchBackstoryFromLocal(prompt);

    let finalBackstory;
    if (backstoryFromLocal) {
        finalBackstory = backstoryFromLocal;
        if (typeof magoCompanion !== 'undefined') {
            magoCompanion.speak("Uma história digna das tavernas de Arton!", 3000);
        }
    } else {
        console.log('Servidor local falhou, usando fallback...');
        finalBackstory = generateSimpleLore(characterData);
        if (typeof magoCompanion !== 'undefined') {
            magoCompanion.speak("Minha magia falhou, mas os velhos contos ainda servem!", 3000);
        }
        showMessage('O servidor local falhou, mas geramos uma história alternativa!', 'is-warning');
    }

    const backgroundTextArea = document.getElementById('charBackground');
    backgroundTextArea.value = finalBackstory;

    if (currentCharacterId) {
        const character = storage.getCharacterById(currentCharacterId);
        if (character) {
            character.background = finalBackstory;
            storage.saveCharacter(character);
        }
    }

    closeModal(loadingModal);
    showMessage('História gerada com sucesso!', 'is-success');
    backgroundTextArea.classList.add('highlight');
    setTimeout(() => backgroundTextArea.classList.remove('highlight'), 1000);
}

function openModal(modal) {
    if (!modal) {
```

```
        console.error('Modal não encontrado');
        return;
    }
    modal.classList.add('is-active');
    const modalCard = modal.querySelector('.modal-card');
    if (modalCard) {
        modalCard.style.opacity = '0';
        setTimeout(() => modalCard.style.opacity = '1', 50);
    }
}

function closeModal(modal) {
    modal.classList.remove('is-active');
}

function showMessage(message, type = 'is-info') {
    const messageEl = document.createElement('div');
    messageEl.className = `notification ${type} is-light`;
    messageEl.style.position = 'fixed';
    messageEl.style.top = '1rem';
    messageEl.style.right = '1rem';
    messageEl.style.zIndex = '9999';
    messageEl.style.maxWidth = '300px';
    messageEl.style.boxShadow = '0 0 10px rgba(0, 0, 0, 0.2)';

    const closeBtn = document.createElement('button');
    closeBtn.className = 'delete';
    closeBtn.addEventListener('click', () => document.body.removeChild(messageEl));

    messageEl.appendChild(closeBtn);
    messageEl.appendChild(document.createTextNode(message));
    document.body.appendChild(messageEl);
    setTimeout(() => {
        if (document.body.contains(messageEl)) document.body.removeChild(messageEl);
    }, 5000);
}

function clearForm() {
    characterForm.reset();
    currentCharacterId = null;
    saveCharacterBtn.innerHTML = '<i class="fas fa-save"></i>  Salvar Personagem';
    saveCharacterBtn.classList.remove('is-warning');
    saveCharacterBtn.style.color = '';
}
```

File: css\base.css

==========

```
/* Estilos gerais da página */
body {
    background-image: url('../assets/img/background-compressed.png');
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
    color: var(--text-color);
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    min-height: 100vh;
    position: relative;
    background-position: 50% 0;
    transform: translateZ(0);
    perspective: 1000px;
    will-change: transform;
}



.container.is-fluid.main-container {
    padding: 0px;
}

/* Overlay de névoa */
.fog-overlay {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-image: url('../assets/img/fog-overlay.webp');
    background-size: cover;
    opacity: 0.1;
    pointer-events: none;
    z-index: 10;
}

/* Container principal */
.main-container {
    display: flex;
    min-height: 100vh;
    padding: 2rem;
    position: relative;
}

.initial-section {
    width: 100%;
    height: 100%;
    display: flex;
    justify-content: space-between;
```

```
}

/* Seção esquerda (Header + Character Creation) */
.left-section {
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: flex-start;
    width: 50%;
    padding-top: 1rem;
}

/* Títulos em estilo medieval */
.medieval-title {
    font-family: 'MedievalSharp', cursive;
    color: var(--accent-color);
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.7);
}

/* Painéis */
.box {
    background-color: var(--panel-bg);
    border: 2px solid var(--panel-border);
    box-shadow: 0 0 15px rgba(0, 0, 0, 0.5);
    color: var(--text-color);
    border-radius: 8px;
}

/* Utilitários gerais */
.hidden {
    opacity: 0;
    pointer-events: none;
}
.fade-in {
    opacity: 1;
}


@keyframes fadeIn {
    from { opacity: 0; }
    to { opacity: 1; }
}

/* Adicione esta classe para o container principal */
.parallax-wrapper {
    position: relative;
    z-index: 1;
    transform-style: preserve-3d;
    transform: translateZ(0);
}
```

File: css\main.css

==========

```
/* Arquivo principal que importa todos os outros */
@import 'variables.css';
@import 'base.css';
@import 'header.css';
@import 'forms.css';
@import 'buttons.css';
@import 'character-creation.css';
@import 'character-cards.css';
@import 'companion.css';
@import 'modals.css';
@import 'footer.css';
@import 'responsive.css';
```

File: generate_story.py

==========

```
from transformers import pipeline
from flask import Flask, request, jsonify
from flask_cors import CORS
import logging

app = Flask(__name__)
CORS(app)  # Habilita CORS para todas as rotas

# Configura logging pra ver erros e resultados no terminal
logging.basicConfig(level=logging.DEBUG)
logger = logging.getLogger(__name__)

try:
    # Carrega o modelo uma vez no início
    logger.info("Carregando o modelo distilgpt2...")
    generator = pipeline('text-generation', model='distilgpt2')
    logger.info("Modelo carregado com sucesso!")
except Exception as e:
    logger.error(f"Erro ao carregar o modelo: {str(e)}")
    generator = None

@app.route('/generate', methods=['POST'])
def generate():
    if generator is None:
        logger.error("Modelo não carregado!")
        return jsonify({'error': 'Modelo de linguagem não disponível'}), 500

    try:
        data = request.get_json()
        prompt = data.get('prompt', '')
        logger.info(f"Recebido prompt: {prompt}")

        # Gera a história com mais criatividade e tokens
        result = generator(prompt, max_new_tokens=300, temperature=0.8, top_p=0.95, truncation=True)
        full_text = result[0]['generated_text']
        logger.info(f"Texto gerado completo: {full_text}")

        # Remove o prompt e garante que o texto restante seja válido
        backstory = full_text.replace(prompt, '').strip()
        logger.info(f"História extraída: {backstory}")

        # Verifica se o backstory é válido (não vazio e tem pelo menos 50 palavras)
        if not backstory or len(backstory.split()) < 50:
            backstory = "Orlak, um goblin astuto, rastreia as florestas de Arton com seu arco em punho."
            logger.warning("História gerada inválida ou curta, usando fallback interno.")

        return jsonify({'backstory': backstory})
    except Exception as e:
        logger.error(f"Erro ao gerar história: {str(e)}")
```

```
        return jsonify({'error': str(e)}), 500

if __name__ == '__main__':
    app.run(host='127.0.0.1', port=5000)
```

File: css\modals.css

==========

```
/* Modal */
.modal-card-head, .modal-card-foot {
    background-color: var(--bg-color);
    border-color: var(--panel-border);
}

.modal-card-body {
    background-color: var(--panel-bg);
    color: var(--text-color);
}

#characterDetails p, #characterDetails p strong, #characterDetails title {
    color: var(--text-color);
}

#characterDetails .medieval-title {
    color: var(--accent-color);
}

/* Modal de Carregamento - ajustado para posição relativa ao scroll */
#loadingModal {
    z-index: 9999;
    display: none; /* Garantimos que o modal comece oculto */
}

#loadingModal.is-active {
    display: flex; /* Flex para centralizar corretamente quando ativo */
}

#loadingModal .modal-background {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.5);
}

#loadingModal .modal-content {
    position: fixed; /* Sempre fixo em relação à viewport */
    background-color: var(--panel-bg);
    border: 3px solid var(--panel-border);
    border-radius: 8px;
    padding: 1rem;
    width: auto;
    max-width: 90%;
    text-align: center;
    box-shadow: 0 0 20px rgba(0, 0, 0, 0.5);
    transition: opacity 0.3s ease; /* Transição suave ao aparecer */
```

```
}

#loadingModal .modal-content p {
    font-size: 1.2rem;
    color: var(--text-color);
    margin-top: 1rem;
}

/* Ajuste da animação do progress bar */
@keyframes indeterminateProgress {
    0% { width: 30%; left: -30%; }
    100% { width: 30%; left: 100%; }
}

#loadingModal .progress {
    background-image: linear-gradient(to right, var(--panel-bg) 30%, var(--accent-color) 30%);
    animation: indeterminateProgress 1.5s infinite linear;
}

/* Estilos para o modal de personagem */
#characterModal {
    z-index: 9999;
    overflow-y: auto;
}

#characterModal .modal-background {
    position: fixed;
}

#characterModal .modal-card {
    max-width: 90%;
    width: 800px;
    margin: 2rem auto;
    background-color: var(--panel-bg);
    border: 3px solid var(--panel-border);
    position: fixed;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    max-height: 90vh;
    overflow-y: auto;
}

#characterModal .modal-card-head,
#characterModal .modal-card-foot {
    background-color: var(--secondary-color);
    border-color: var(--panel-border);
    position: sticky;
}

#characterModal .modal-card-head {
    top: 0;
    z-index: 1;
```

```
}

#characterModal .modal-card-foot {
    bottom: 0;
    z-index: 1;
}

#characterModal .modal-card-body {
    background-color: var(--panel-bg);
    color: var(--text-color);
    padding: 1.5rem;
}

/* Ajuste para a história de fundo */
#characterModal .content.mt-4 {
    width: 100%;
    grid-column: 1 / -1;
    margin-top: 2rem !important;
}

#characterModal .content.mt-4 h4 {
    margin-bottom: 1rem;
}

#characterModal .content.mt-4 p {
    text-align: justify;
    line-height: 1.6;
    padding: 1rem;
    background: rgba(0, 0, 0, 0.2);
    border-radius: 8px;
    border: 1px solid var(--panel-border);
}
```

File: css\header.css

==========

```
/* Header */
.hero {
    background-color: transparent;
    width: 100%;
    text-align: center;
    margin: 1.5rem 0;
}

.hero-body {
    padding: 0;
    margin-top: 0;
}

.hero .title {
    font-size: 3.5rem;
    letter-spacing: 2px;
    margin-bottom: 0.5rem;
}

.hero .subtitle {
    color: var(--text-color);
    font-size: 1.5rem;
}
```

File: index.html

==========

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
    <title>Forjador de Lendas | Gerador de Personagens RPG - Tormenta 20</title>
    <!-- Reset CSS -->
    <link rel="stylesheet" href="css/reset.css">
    <!-- Bulma CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.4/css/bulma.min.css">
    <!-- Font Awesome -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
    <!-- Google Fonts -->
    <link href="https://fonts.googleapis.com/css2?family=MedievalSharp&display=swap" rel="stylesheet">
    <!-- CSS personalizado -->
    <link rel="stylesheet" href="css/main.css">
</head>

<body>
    <!-- Overlay de névoa -->
    <div class="fog-overlay"></div>

    <!-- Container principal -->
    <div class="container is-fluid main-container">
        <div class="initial-section">
            <!-- Seção esquerda -->
            <div class="left-section">
                <!-- Header -->
                <header class="hero">
                    <div class="hero-body">
                        <h1 class="title is-1 has-text-centered medieval-title">Forjador de Lendas</h1>
                        <h2 class="subtitle has-text-centered medieval-title">Crie heróis épicos para Arton</h2>
                    </div>
                </header>

                <!-- Seção de criação de personagem -->
                <div class="character-creation-section">
                    <div class="character-panel-container">
                        <div class="column is-12">
                            <div class="box character-panel">
                                <h3 class="title is-3 has-text-centered medieval-title">Crie seu Herói</h3>

                                <!-- Formulário de criação -->
                                <form id="characterForm">
                                    <!-- Coluna da esquerda - Dados básicos -->
                                    <div class="basic-info-column">
                                        <div class="field">
                                            <label class="label">Nome</label>
```

```
    <div class="control has-icons-left">
      <input class="input" type="text" id="charName"
        placeholder="Nome do personagem" required>
      <span class="icon is-small is-left">
        <i class="fas fa-user"></i>
      </span>
    </div>
  </div>

  <div class="field">
    <label class="label">Raça</label>
    <div class="control has-icons-left">
      <div class="select is-fullwidth">
        <select id="charRace">
          <option value="Humano">Humano</option>
          <option value="Anão">Anão</option>
          <option value="Dahllan">Dahllan</option>
          <option value="Elfo">Elfo</option>
          <option value="Goblin">Goblin</option>
          <option value="Lefou">Lefou</option>
          <option value="Minotauro">Minotauro</option>
          <option value="Qareen">Qareen</option>
          <option value="Golem">Golem</option>
          <option value="Hynne">Hynne</option>
          <option value="Kliren">Kliren</option>
          <option value="Medusa">Medusa</option>
          <option value="Osteon">Osteon</option>
          <option value="Sereia/Tritão">Sereia/Tritão</option>
          <option value="Sílfide">Sílfide</option>
          <option value="Suraggel">Suraggel</option>
          <option value="Trong">Trong</option>
        </select>
      </div>
      <span class="icon is-small is-left">
        <i class="fas fa-people-group"></i>
      </span>
    </div>
  </div>

  <div class="field">
    <label class="label">Classe</label>
    <div class="control has-icons-left">
      <div class="select is-fullwidth">
        <select id="charClass">
          <option value="Arcanista">Arcanista</option>
          <option value="Bárbaro">Bárbaro</option>
          <option value="Bardo">Bardo</option>
          <option value="Bucaneiro">Bucaneiro</option>
          <option value="Caçador">Caçador</option>
          <option value="Cavaleiro">Cavaleiro</option>
          <option value="Clérigo">Clérigo</option>
          <option value="Druida">Druida</option>
          <option value="Guerreiro">Guerreiro</option>
```

```
                <option value="Inventor">Inventor</option>
                <option value="Ladino">Ladino</option>
                <option value="Lutador">Lutador</option>
                <option value="Nobre">Nobre</option>
                <option value="Paladino">Paladino</option>
              </select>
            </div>
            <span class="icon is-small is-left">
              <i class="fas fa-hat-wizard"></i>
            </span>
          </div>
        </div>

        <div class="field">
          <label class="label">Alinhamento</label>
          <div class="control has-icons-left">
            <div class="select is-fullwidth">
              <select id="charAlignment">
                <option value="Leal e Bom">Leal e Bom</option>
                <option value="Neutro e Bom">Neutro e Bom</option>
                <option value="Caótico e Bom">Caótico e Bom</option>
                <option value="Leal e Neutro">Leal e Neutro</option>
                <option value="Neutro">Neutro</option>
                <option value="Caótico e Neutro">Caótico e Neutro</option>
                <option value="Leal e Mau">Leal e Mau</option>
                <option value="Neutro e Mau">Neutro e Mau</option>
                <option value="Caótico e Mau">Caótico e Mau</option>
              </select>
            </div>
            <span class="icon is-small is-left">
              <i class="fas fa-balance-scale"></i>
            </span>
          </div>
        </div>
      </div>

      <!-- Coluna da direita - Atributos -->
      <div class="attributes-column">
        <p class="title is-5 label medieval-title attributes-title">Atributos</p>
        <div class="attributes-container">
          <div class="field">
            <label class="label">Força</label>
            <div class="control">
              <input class="input" type="number" id="attrStr" min="3" max="18"
                value="10">
            </div>
          </div>
          <div class="field">
            <label class="label">Destreza</label>
            <div class="control">
              <input class="input" type="number" id="attrDex" min="3" max="18"
                value="10">
            </div>
```

```
            </div>
            <div class="field">
              <label class="label">Constituição</label>
              <div class="control">
                <input class="input" type="number" id="attrCon" min="3" max="18"
                  value="10">
              </div>
            </div>
            <div class="field">
              <label class="label">Inteligência</label>
              <div class="control">
                <input class="input" type="number" id="attrInt" min="3" max="18"
                  value="10">
              </div>
            </div>
            <div class="field">
              <label class="label">Sabedoria</label>
              <div class="control">
                <input class="input" type="number" id="attrWis" min="3" max="18"
                  value="10">
              </div>
            </div>
            <div class="field">
              <label class="label">Carisma</label>
              <div class="control">
                <input class="input" type="number" id="attrCha" min="3" max="18"
                  value="10">
              </div>
            </div>
            <button type="button" id="rollAttributes"
              class="button is-info is-fullwidth">
              <span class="icon">
                <i class="fas fa-dice"></i>
              </span>
              <span>Rolar Atributos</span>
            </button>
          </div>
        </div>
      </form>

      <!-- Descrição/Backstory -->
      <div class="column is-12">
        <div class="field">
          <label class="label">História de Fundo</label>
          <div class="control">
            <textarea class="textarea" id="charBackground"
              placeholder="Descreva brevemente a história do seu personagem (opcional)"></textarea>
          </div>
        </div>
      </div>

      <!-- Botões -->
      <div class="column is-12">
```

```
                        <div class="field is-grouped is-grouped-centered">
                            <div class="control">
                                <button type="button" id="saveCharacter" class="button is-primary">
                                    <span class="icon">
                                        <i class="fas fa-save"></i>
                                    </span>
                                    <span>Salvar Personagem</span>
                                </button>
                            </div>
                            <div class="control">
                                <button type="button" id="generateLore" class="button is-info">
                                    <span class="icon">
                                        <i class="fas fa-book-open"></i>
                                    </span>
                                    <span>Gerar História</span>
                                </button>
                            </div>
                            <div class="control">
                                <button type="button" id="clearForm" class="button is-light">
                                    <span class="icon">
                                        <i class="fas fa-undo"></i>
                                    </span>
                                    <span>Limpar</span>
                                </button>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <!-- Mago Companion -->
    <div class="companion-container">
        <div class="companion-speech-bubble hidden">
            <p id="companionText">Olá, aventureiro! Estou aqui para ajudá-lo a criar sua lenda em Arton!</p>
        </div>
        <img src="assets/img/mago-avatar-compressed.png" loading="lazy" alt="Mago Companion"
            class="companion-avatar">
    </div>
</div>

<!-- Seção de personagens salvos -->
<div class="saved-characters-section">
    <div class="container">
        <div class="column is-8-desktop is-offset-2-desktop is-12-tablet">
            <div class="box saved-characters">
                <h3 class="title is-4 has-text-centered medieval-title">Seus Heróis</h3>
                <div id="savedCharactersList" class="is-flex is-flex-wrap-wrap is-justify-content-center">
                    <!-- Personagens salvos serão exibidos aqui via JavaScript -->
                    <p class="empty-list-message">Nenhum herói criado ainda. Comece a forjar sua lenda!</p>
```

```
            </div>
          </div>
        </div>
    </div>
</div>

<!-- Modal para exibir o personagem -->
<div id="characterModal" class="modal">
    <div class="modal-background"></div>
    <div class="modal-card">
        <header class="modal-card-head">
            <p class="modal-card-title medieval-title">Detalhes do Herói</p>
            <button class="delete" aria-label="close"></button>
        </header>
        <section class="modal-card-body">
            <div id="characterDetails">
                <!-- Detalhes do personagem serão exibidos aqui -->
            </div>
            <div id="loreContent" class="mt-4">
                <h4 class="title is-5 medieval-title">História</h4>
                <div id="loreText" class="content">
                    <!-- História gerada será exibida aqui -->
                </div>
            </div>
        </section>
        <footer class="modal-card-foot">
            <button class="button is-success" id="editCharacter">Editar</button>
            <button class="button is-danger" id="deleteCharacter">Excluir</button>
            <button class="button" id="closeModal">Fechar</button>
        </footer>
    </div>
</div>

<!-- Modal de Carregamento para geração IA -->
<div id="loadingModal" class="modal">
    <div class="modal-background"></div>
    <div class="modal-content">
        <div class="box has-text-centered">
            <p class="title is-4 medieval-title">Invocando as Forças Arcanas</p>
            <p>Nosso mago está trabalhando em sua história...</p>
            <progress class="progress is-primary mt-3" max="100"></progress>
        </div>
    </div>
</div>

<!-- Footer -->
<footer class="footer">
    <div class="content has-text-centered">
        <p id="footer-text">
            <strong>Forjador de Lendas</strong> | Criado por Marques
            <span class="icon">
                <i class="fas fa-dice-d20"></i>
            </span>
```

```
            2025
          </p>
        </div>
    </footer>
    </div>

    <!-- Scripts -->
    <script src="js/storage.js"></script>
    <script src="js/companion.js"></script>
    <script src="js/app.js"></script>
</body>

</html>
```

File: README.md

==========

Forjador de Lendas

Bem-vindo ao **Forjador de Lendas**, um gerador de personagens para o sistema de RPG **Tormenta 20**, ambientado no rico

Funcionalidades
- **Seleção de Raças e Classes:** Escolha entre as 17 raças e 14 classes oficiais de Tormenta 20.
- **Geração de Atributos:** Role os dados (4d6, descartando o menor) para criar atributos únicos.
- **Histórias Automáticas:** Gere histórias de fundo baseadas nas escolhas do usuário, com referências ao lore de Arton.
- **Mago Companion:** Um assistente virtual que reage às escolhas do jogador com falas temáticas.
- **Interface Medieval:** Design inspirado em fantasia, com névoa, painéis rústicos e ícones personalizados.
- **Salvamento Local:** Armazene seus personagens no `localStorage` do navegador.
- **Modal Dinâmico:** Visualize detalhes dos personagens e edite ou exclua com facilidade.

Tecnologias Utilizadas
- **HTML5:** Estrutura da página.
- **CSS3:** Estilização com Bulma CSS, Font Awesome e animações personalizadas.
- **JavaScript:** Lógica interativa, incluindo geração de histórias e manipulação do DOM.
- **LocalStorage:** Persistência de dados no navegador.

Pré-requisitos
- Um navegador moderno (Chrome, Firefox, Edge, etc.).
- Nenhum servidor é necessário; o projeto roda localmente no cliente.

Como Executar
1. Clone o repositório:
   ```bash
   git clone https://github.com/seu-usuario/forjador-de-lendas.git
   ```
2. Abra o arquivo `index.html` em um navegador:
   - No Windows: clique com o botão direito no arquivo e selecione "Abrir com" > seu navegador.
   - Ou arraste o arquivo para uma aba do navegador.
3. Comece a criar seus heróis!

Estrutura do Projeto
```
forjador-de-lendas/

 assets/
    img/              # Imagens do projeto (fundo, mago companion, etc.)
    ...               # Outros assets
 css/
    reset.css         # Reset de estilos
    style.css         # Estilos personalizados
 js/
    app.js            # Lógica principal do gerador
    companion.js      # Lógica do mago companion
    storage.js        # Gerenciamento de armazenamento local
 index.html           # Página principal
 README.md            # Este arquivo
```

```

## Exemplos de Uso
- Escolha "Elfo" como raça e "Arcanista" como classe, clique em "Rolar Atributos" e depois em "Gerar História". Veja o mago rea
- Salve seu personagem e visualize-o na seção "Seus Heróis".

## Contribuições
Contribuições são bem-vindas! Sinta-se à vontade para abrir issues ou pull requests com melhorias, como:
- Adicionar mais raças ou classes customizadas.
- Integrar uma API de IA para histórias mais complexas.
- Melhorar a responsividade para dispositivos móveis.

1. Fork o repositório.
2. Crie uma branch para sua feature (`git checkout -b feature/nova-ideia`).
3. Commit suas mudanças (`git commit -m 'Adiciona nova feature'`).
4. Push para o branch (`git push origin feature/nova-ideia`).
5. Abra um Pull Request.

## Licença
Este projeto é licenciado sob a [MIT License](LICENSE). Sinta-se livre para usar, modificar e distribuir!

## Créditos
- **Desenvolvido por:** Marques
- **Inspiração:** Universo de Tormenta 20, criado pela Jambô Editora.
- **Apoio:** Comunidade de RPG brasileira.

## Contato
Dúvidas ou sugestões? Entre em contato comigo em [marquesviniciusmelomartins@gmail.com](mailto:marquesviniciusmeloma

---

**Forje sua lenda e que os deuses de Arton guiem seus passos!**
```

File: css\responsive.css

==========

```
/* Responsividade */
@media screen and (max-width: 1023px) {
    .main-container {
      flex-direction: column;
      padding: 1rem;
    }

    .left-section {
      width: 100%;
      padding-top: 1rem;
    }

    .character-creation-section {
      width: 100%;
    }
  }

  /* Ajustes gerais para mobile */
  @media screen and (max-width: 768px) {
    body {
      overflow-x: hidden;
      width: 100%;
      position: relative;
    }

    .main-container {
      padding: 1rem;
      width: 100%;
      overflow-x: hidden;
      margin-bottom: 80px; /* Espaço para o companion flutuante */
    }

    #characterForm {
      flex-direction: column;
      padding: 0.5rem;
    }

    .basic-info-column {
      flex: 1;
      border-right: none;
      border-bottom: 1px solid var(--panel-border);
      padding-right: 0;
      padding-bottom: 1rem;
      margin-bottom: 1rem;
    }

    .attributes-column {
      padding-left: 0;
    }
```

```
.attributes-container {
  grid-template-columns: repeat(2, 1fr);
  gap: 0.5rem;
}

.attributes-container .input {
  font-size: 1rem;
  height: 2rem;
  padding: 0.25rem;
}

/* Ajuste dos cartões de personagem */
.character-card {
  width: calc(50% - 1rem);
  margin: 0.5rem;
}

/* Ajuste dos botões em mobile */
.field.is-grouped.is-grouped-centered {
  flex-direction: column;
  width: 100%;
}

.field.is-grouped.is-grouped-centered .control {
  width: 100%;
  margin: 0.25rem 0;
}

.field.is-grouped.is-grouped-centered .control .button {
  width: 100%;
  justify-content: center;
  height: 3rem; /* Altura maior para melhor toque */
}

/* Ajuste dos modais para mobile */
#characterModal .modal-card {
  width: 95%;
  margin: 0.5rem;
  max-height: 95vh;
}

/* Ajuste da lista de personagens para mobile */
.saved-characters-section {
  padding: 1rem 0;
}

/* Ajuste do header */
.hero .title {
  font-size: 2.5rem;
}

.hero .subtitle {
```

```
    font-size: 1.2rem;
  }
}

/* Ajustes para telas muito pequenas */
@media screen and (max-width: 480px) {
  .character-card {
    width: 100%;
    margin: 0.5rem 0;
  }

  .hero .title {
    font-size: 2rem;
  }

  /* Ajuste para o formulário em telas muito pequenas */
  .attributes-container {
    grid-template-columns: repeat(2, 1fr);
    gap: 0.5rem;
  }

  /* Garantir que os campos de atributos tenham larguras iguais */
  .attributes-container .field {
    width: 100%;
    max-width: none;
  }

  .attributes-container .input {
    width: 100% !important;
    max-width: none !important;
  }
}
```

File: js\companion.js

==========

```
class MagoCompanion {
    constructor() {
        this.container = document.querySelector(".companion-container");
        this.speechBubble = document.querySelector(".companion-speech-bubble");
        this.companionText = document.getElementById("companionText");
        this.companionAvatar = document.querySelector(".companion-avatar");
        this.lastMessage = "";
        this.isDragging = false;
        this.dragOffset = { y: 0 };
        this.isMobile = window.innerWidth <= 1023;

        // Aguardar o DOM estar completamente carregado
        if (document.readyState === "loading") {
            document.addEventListener("DOMContentLoaded", () => this.init());
        } else {
            this.init();
        }
    }

    init() {
        console.log("Mago Companion inicializado");
        this.setupEventListeners();

        // Configurar arrasto apenas em dispositivos móveis
        if (this.isMobile) {
            this.setupDraggable();
            this.loadPosition();
        }

        this.greet();
    }

    setupEventListeners() {
        console.log("Configurando event listeners do Mago Companion");

        // Conecta os eventos de input/change com os métodos correspondentes
        const charName = document.getElementById("charName");
        const charRace = document.getElementById("charRace");
        const charClass = document.getElementById("charClass");
        const charAlignment = document.getElementById("charAlignment");
        const rollAttributesBtn = document.getElementById("rollAttributes");

        // Adicionar event listeners diretamente (sem tentar remover os antigos)
        if (charName) {
            charName.addEventListener("input", (e) => this.onNameInput(e));
            console.log("Event listener adicionado para charName");
        }

        if (charRace) {
```

```
    charRace.addEventListener("change", (e) => this.onRaceChange(e));
    console.log("Event listener adicionado para charRace");
}

if (charClass) {
    charClass.addEventListener("change", (e) => this.onClassChange(e));
    console.log("Event listener adicionado para charClass");
}

if (charAlignment) {
    charAlignment.addEventListener("change", (e) => this.onAlignmentChange(e));
    console.log("Event listener adicionado para charAlignment");
}

// Adicionar event listener para o botão de rolar atributos
if (rollAttributesBtn) {
    rollAttributesBtn.addEventListener("click", () => {
        console.log("Botão de rolar atributos clicado");
        this.onRollAttributes();
    });
    console.log("Event listener adicionado para rollAttributes");
}

// Eventos para outros botões
const buttonElements = ["generateLore", "saveCharacter", "clearForm"];

buttonElements.forEach((elementId) => {
    const el = document.getElementById(elementId);
    if (el) {
        el.addEventListener("click", () => {
            if (elementId === "generateLore") {
                this.speak("Deixe-me consultar os pergaminhos antigos...", 3000);
            } else if (elementId === "saveCharacter") {
                this.speak("Seu herói foi registrado em meu grimório!", 3000);
            } else {
                this.hideSpeechBubble();
            }
        });
    }
});

// Adiciona evento de clique no avatar para mostrar/esconder o balão
this.companionAvatar.addEventListener("click", (e) => {
    // Só mostra/esconde o balão se não estiver arrastando
    if (!this.isDragging) {
        this.toggleSpeechBubble();
    }
});

// Esconde o balão quando clicar fora
document.addEventListener("click", (e) => {
    if (!this.companionAvatar.contains(e.target) && !this.speechBubble.contains(e.target)) {
        this.hideSpeechBubble();
```

```
        }
    });

    // Atualiza o estado mobile quando a janela é redimensionada
    window.addEventListener("resize", () => {
        const wasMobile = this.isMobile;
        this.isMobile = window.innerWidth <= 1023;

        // Se mudou de desktop para mobile ou vice-versa
        if (wasMobile !== this.isMobile) {
            if (this.isMobile) {
                // Mudou para mobile
                this.setupDraggable();
                this.loadPosition();
                this.companionAvatar.style.cursor = "grab";
            } else {
                // Mudou para desktop
                this.removeDraggable();
                this.resetDesktopPosition();
                this.companionAvatar.style.cursor = "default";
            }
        }
    });

    // Salvar posição quando a janela for fechada (apenas em mobile)
    window.addEventListener("beforeunload", () => {
        if (this.isMobile) {
            this.savePosition();
        }
    });
}

setupDraggable() {
    // Remover listeners existentes para evitar duplicação
    this.removeDraggable();

    // Mouse events (desktop)
    this.companionAvatar.addEventListener("mousedown", this.handleMouseDown);
    document.addEventListener("mousemove", this.handleMouseMove);
    document.addEventListener("mouseup", this.handleMouseUp);

    // Touch events (mobile)
    this.companionAvatar.addEventListener("touchstart", this.handleTouchStart);
    document.addEventListener("touchmove", this.handleTouchMove);
    document.addEventListener("touchend", this.handleTouchEnd);

    // Prevenir comportamento padrão de arrastar imagem
    this.companionAvatar.addEventListener("dragstart", this.handleDragStart);
}

removeDraggable() {
    // Remover todos os event listeners de arrasto
    this.companionAvatar.removeEventListener("mousedown", this.handleMouseDown);
```

```
    document.removeEventListener("mousemove", this.handleMouseMove);
    document.removeEventListener("mouseup", this.handleMouseUp);

    this.companionAvatar.removeEventListener("touchstart", this.handleTouchStart);
    document.removeEventListener("touchmove", this.handleTouchMove);
    document.removeEventListener("touchend", this.handleTouchEnd);

    this.companionAvatar.removeEventListener("dragstart", this.handleDragStart);
}

// Usando arrow functions para manter o contexto 'this'
handleMouseDown = (e) => this.startDrag(e);
handleMouseMove = (e) => this.drag(e);
handleMouseUp = () => this.endDrag();
handleTouchStart = (e) => this.startDrag(e);
handleTouchMove = (e) => this.drag(e);
handleTouchEnd = () => this.endDrag();
handleDragStart = (e) => e.preventDefault();

startDrag(e) {
    if (!this.isMobile) return; // Só permite arrasto em mobile

    this.isDragging = true;
    this.companionAvatar.style.cursor = "grabbing";

    // Adicionar classe visual para feedback
    this.companionAvatar.classList.add("dragging");

    // Capturar a posição inicial do mouse/toque
    const clientY = e.clientY || (e.touches && e.touches[0].clientY);

    // Calcular o offset entre o ponto de clique e o topo do elemento
    const rect = this.container.getBoundingClientRect();
    this.dragOffset.y = clientY - rect.top;

    // Prevenir comportamento padrão para evitar problemas em dispositivos móveis
    if (e.type === "touchstart") {
        e.preventDefault();
    }
}

drag(e) {
    if (!this.isDragging || !this.isMobile) return;

    // Obter a posição atual do mouse/toque
    const clientY = e.clientY || (e.touches && e.touches[0].clientY);

    if (clientY === undefined) return;

    // Calcular a nova posição considerando o offset (apenas vertical)
    const newTop = clientY - this.dragOffset.y;

    // Limitar a posição para não sair da tela
```

```
        const maxY = window.innerHeight - this.container.offsetHeight;
        const boundedTop = Math.max(0, Math.min(newTop, maxY));

        // Aplicar a nova posição (apenas vertical)
        // Usamos valores percentuais para manter a posição relativa à viewport
        const topPercent = (boundedTop / window.innerHeight) * 100;

        // Aplicar a posição com transformação para melhor desempenho
        this.container.style.transform = `translateY(${boundedTop}px)`;
        this.container.style.top = "auto";
        this.container.style.bottom = "auto";

        // Manter fixo no lado direito
        this.container.style.right = "0";
        this.container.style.left = "auto";

        // Prevenir comportamento padrão para evitar rolagem da página
        if (e.type === "touchmove") {
            e.preventDefault();
        }
    }

    endDrag() {
        if (this.isDragging && this.isMobile) {
            this.isDragging = false;
            this.companionAvatar.style.cursor = "grab";

            // Remover classe visual
            this.companionAvatar.classList.remove("dragging");

            // Converter a transformação em posição top para salvar
            const transform = this.container.style.transform;
            const match = transform.match(/translateY$$([^)]+)px$$/);
            if (match && match[1]) {
                const topValue = parseFloat(match[1]);
                const topPercent = (topValue / window.innerHeight) * 100;
                this.container.style.top = `${topPercent}%`;
                this.container.style.transform = "none";
            }

            this.savePosition();
        }
    }

    savePosition() {
        if (!this.isMobile) return;

        const position = {
            top: this.container.style.top,
            bottom: this.container.style.bottom,
        };
        localStorage.setItem("magoCompanionPosition", JSON.stringify(position));
    }
```

```
loadPosition() {
    if (!this.isMobile) return;

    try {
        const savedPosition = localStorage.getItem("magoCompanionPosition");
        if (savedPosition) {
            const position = JSON.parse(savedPosition);
            this.container.style.top = position.top || "auto";
            this.container.style.bottom = position.bottom || "auto";

            // Se não tiver posição definida, usar a posição padrão
            if (position.top === "auto" && position.bottom === "auto") {
                this.resetMobilePosition();
            }
        } else {
            this.resetMobilePosition();
        }

        // Garantir que esteja fixo no lado direito
        this.container.style.right = "0";
        this.container.style.left = "auto";
        this.container.style.transform = "none";
    } catch (error) {
        console.error("Erro ao carregar posição do companion:", error);
        this.resetMobilePosition();
    }
}

resetMobilePosition() {
    // Posição padrão no canto inferior direito para mobile
    this.container.style.top = "auto";
    this.container.style.right = "0";
    this.container.style.bottom = "20px";
    this.container.style.left = "auto";
    this.container.style.transform = "none";
}

resetDesktopPosition() {
    // Restaurar posição original para desktop
    this.container.style.top = "-1%";
    this.container.style.right = "-12%";
    this.container.style.bottom = "auto";
    this.container.style.left = "auto";
    this.container.style.transform = "none";
}

toggleSpeechBubble() {
    if (this.speechBubble.classList.contains("hidden")) {
        this.showLastMessage();
    } else {
        this.hideSpeechBubble();
    }
```

```
}

speak(text, duration = null) {
    console.log("Mago diz:", text);
    this.lastMessage = text;
    this.speechBubble.classList.remove("hidden");
    this.speechBubble.classList.add("fade-in");
    this.companionText.textContent = text;

    // Efeito visual no avatar quando fala
    this.companionAvatar.classList.add("pulse");
    setTimeout(() => {
        this.companionAvatar.classList.remove("pulse");
    }, 1000);

    if (duration) {
        setTimeout(() => this.hideSpeechBubble(), duration);
    }
}

hideSpeechBubble() {
    this.speechBubble.classList.add("hidden");
    this.speechBubble.classList.remove("fade-in");
}

showLastMessage() {
    if (this.lastMessage) {
        this.speak(this.lastMessage);
    } else {
        this.greet();
    }
}

greet() {
    const greetings = [
        "Bem-vindo ao Forjador de Lendas! Sou Merlin, seu guia em Arton!",
        "Um novo herói para Arton! Vamos criar uma lenda épica juntos!",
        "Por minhas barbas mágicas! Que tal forjar um destino no Reinado?",
    ];
    this.speak(greetings[Math.floor(Math.random() * greetings.length)]);
}

onNameInput(e) {
    const name = e.target.value.trim();
    if (name.length > 2) {
        const responses = [
            `${name}? Um nome que ecoará em Arton! Escolha sua raça!`,
            `Ah, ${name}! Vejo um futuro lendário no Reinado para você!`,
            `${name}... *acaricia a barba* Um herói contra a Tormenta, talvez?`,
        ];
        this.speak(responses[Math.floor(Math.random() * responses.length)]);
    }
}
```

```
onRaceChange(e) {
    console.log("Raça alterada:", e.target.value);
    const raceResponses = {
        Humano: "Humanos! Tão versáteis quanto os ventos de Arton!",
        Anão: "Um anão de Doherimm? Só não me peça para cavar contigo!",
        Dahllan: "Uma dahllan! Allihanna deve estar orgulhosa de você!",
        Elfo: "Um elfo de Lenórienn? Elegante, mas não se perca em séculos!",
        Goblin: "Um goblin? *checa os bolsos* Cuidado com Tollon em você!",
        Lefou: "Um lefou? *recua* Espero que a Tormenta não te siga!",
        Minotauro: "Um minotauro! Só não quebre minha torre com esses chifres!",
        Qareen: "Um qareen! Cuidado com os desejos que conceder por aí!",
        Golem: "Um golem? *observa* Quem te trouxe à vida, hein?",
        Hynne: "Um hynne! Pequeno, mas cheio de sorte, aposto!",
        Kliren: "Um kliren! Sua inteligência vai resolver muitos enigmas!",
        Medusa: "Uma medusa? *evita o olhar* Não me transforme em pedra!",
        Osteon: "Um osteon? *treme* Espero que não seja meu esqueleto animado!",
        "Sereia/Tritão": "Um sereia ou tritão! Não molhe minha túnica, por favor!",
        Sílfide: "Uma sílfide! Você flutua como os ventos de Wynlla!",
        Suraggel: "Um suraggel! Divino ou infernal, escolha com cuidado!",
        Trong: "Um trong! *se afasta* Não me coma, eu sou só um mago velho!",
    };
    this.speak(raceResponses[e.target.value] || "Uma raça intrigante para Arton!");
}

onClassChange(e) {
    console.log("Classe alterada:", e.target.value);
    const classResponses = {
        Arcanista: "Um arcanista! *limpa uma lágrima* Um irmão das artes mágicas!",
        Bárbaro: "Um bárbaro! Só não quebre minhas coisas na sua fúria!",
        Bardo: "Um bardo! Cante minhas glórias... quer dizer, as suas!",
        Bucaneiro: "Um bucaneiro! Espero que não roube meu cajado no mar!",
        Caçador: "Um caçador! Perfeito para rastrear horrores da Tormenta!",
        Cavaleiro: "Um cavaleiro! Sua honra brilha mais que minha magia!",
        Clérigo: "Um clérigo! Que os deuses do Pantheon te abençoem!",
        Druida: "Um druida! Allihanna aprova, mas sem lobos na minha torre!",
        Guerreiro: "Um guerreiro! Pronto para as batalhas de Arton!",
        Inventor: "Um inventor! *se anima* Mostre-me seus gadgets!",
        Ladino: "Um ladino! *esconde o bolso* Cuidado com os Ladrões de Deheon!",
        Lutador: "Um lutador! Seus punhos vão impressionar em Arton!",
        Nobre: "Um nobre! Sua presença é digna de Valkaria!",
        Paladino: "Um paladino! A luz de Khalmyr guia seus passos!",
    };
    this.speak(classResponses[e.target.value] || "Uma classe fascinante para Arton!");
}

onAlignmentChange(e) {
    console.log("Alinhamento alterado:", e.target.value);
    const alignment = e.target.value;
    if (alignment.includes("Mau")) {
        this.speak("Hmmm... *suspeita* Não traga a Tormenta para minha torre!");
    } else if (alignment.includes("Bom")) {
        this.speak("Um coração nobre! Arton precisa de você contra o mal!");
```

```
        } else {
            this.speak("Neutro? Equilíbrio é sábio, mas não seja indeciso!");
        }
    }

    onRollAttributes() {
        console.log("Rolando atributos...");
        const responses = [
            "*Agita as mãos* Que os dados de Nimb decidam seu destino!",
            "Rolando! Que Tanna-Toh revele sua força interior!",
            "*Sopra os dados* Um toque de Wynna para sua sorte!",
        ];
        const randomResponse = responses[Math.floor(Math.random() * responses.length)];

        // Forçar a exibição do balão de fala
        this.speak(randomResponse, 4000);
    }
}

// Criar uma instância global do MagoCompanion
let magoCompanionInstance = null;

// Função para inicializar o Mago Companion
function initMagoCompanion() {
    console.log("Inicializando Mago Companion");
    if (!magoCompanionInstance) {
        magoCompanionInstance = new MagoCompanion();
        window.magoCompanion = magoCompanionInstance;
    }
}

// Garantir que o Mago Companion seja inicializado quando o DOM estiver pronto
if (document.readyState === "loading") {
    document.addEventListener("DOMContentLoaded", initMagoCompanion);
} else {
    initMagoCompanion();
}
```

File: css\forms.css

==========

```
/* Estilos de formulários */
.label {
    color: var(--accent-color);
    font-weight: bold;
}

.input, .textarea, .select select {
    background-color: rgba(40, 30, 20, 0.7);
    border-color: var(--panel-border);
    color: var(--text-color);
    transition: all 0.3s;
}

.input:focus, .textarea:focus, .select select:focus {
    border-color: var(--accent-color);
    box-shadow: 0 0 0 0.125em rgba(212, 175, 55, 0.25);
}

#charName::placeholder {
    color: var(--text-color);
    opacity: 0.5;
}

#charBackground::placeholder {
    color: var(--text-color);
    opacity: 0.5;
}

/* Efeito de highlight para a textArea */
@keyframes highlightBackground {
    0% { background-color: rgba(35, 25, 15, 0.7); }
    50% { background-color: rgba(212, 175, 55, 0.2); }
    100% { background-color: rgba(35, 25, 15, 0.7); }
}

.highlight {
    animation: highlightBackground 1s ease;
}
```

File: js\storage.js

==========

```
/**
 * Sistema de armazenamento local para personagens
 */
class CharacterStorage {
    constructor() {
        this.storageKey = 'rpg_characters';
        this.characters = this.loadCharacters();
    }

    // Carrega personagens do localStorage
    loadCharacters() {
        const stored = localStorage.getItem(this.storageKey);
        return stored ? JSON.parse(stored) : [];
    }

    // Salva personagens no localStorage
    saveCharacters() {
        localStorage.setItem(this.storageKey, JSON.stringify(this.characters));
    }

    // Adiciona ou atualiza um personagem
    saveCharacter(character) {
        // Se não tiver ID, cria um novo
        if (!character.id) {
            character.id = this.generateId();
            character.createdAt = new Date().toISOString();
            this.characters.push(character);
        } else {
            // Atualiza um existente
            const index = this.characters.findIndex(c => c.id === character.id);
            if (index !== -1) {
                character.updatedAt = new Date().toISOString();
                this.characters[index] = character;
            } else {
                // ID não encontrado, cria novo
                character.id = this.generateId();
                character.createdAt = new Date().toISOString();
                this.characters.push(character);
            }
        }

        this.saveCharacters();
        return character;
    }

    // Obtém todos os personagens
    getAllCharacters() {
        return [...this.characters];
    }
```

```
    // Obtém um personagem pelo ID
    getCharacterById(id) {
        return this.characters.find(c => c.id === id);
    }

    // Remove um personagem
    deleteCharacter(id) {
        const index = this.characters.findIndex(c => c.id === id);
        if (index !== -1) {
            this.characters.splice(index, 1);
            this.saveCharacters();
            return true;
        }
        return false;
    }

    // Gera um ID único
    generateId() {
        return Date.now().toString(36) + Math.random().toString(36).substr(2, 5);
    }
}

// Exporta a instância
const characterStorage = new CharacterStorage();
```

File: css\character-creation.css

==========

```
/* Seção de criação de personagem */
.character-creation-section {
    display: flex;
    position: relative;
    width: 90%;
    min-height: 600px;
    border-radius: 15px;
    align-self: center;
  }

  .character-panel-container {
    width: 100%;
  }

  .character-panel {
    max-width: 800px;
    margin: 0 auto;
  }

  /* Ajustes no layout do formulário de criação */
  #characterForm {
    display: flex;
    gap: 2rem;
    padding: 1rem;
  }

  /* Coluna da esquerda (dados básicos) */
  .basic-info-column {
    flex: 0 0 58%;
    border-right: 2px solid var(--panel-border);
    padding-right: 2rem;
  }

  /* Coluna da direita (atributos) */
  .attributes-column {
    flex: 1;
    padding-left: 1rem;
  }

  .attributes-container {
    display: grid;
    grid-template-columns: repeat(2, 1fr);
    gap: 1rem;
    margin-top: 1rem;
  }

  .attributes-container .field {
    margin-bottom: 0.5rem;
    width: 100%;
```

```
}

.attributes-container .input {
  text-align: center;
  font-size: 1.2rem;
  height: 2.5rem;
  width: 100%; /* Garante que todos os inputs tenham a mesma largura */
  box-sizing: border-box; /* Inclui padding e borda na largura */
}

/* Título dos atributos */
.attributes-title {
  grid-column: span 2;
  margin-bottom: 1rem !important;
}

/* Botão de rolar atributos */
#rollAttributes {
  grid-column: span 2;
  margin-top: 1rem;
}

/* Responsividade para o formulário */
@media screen and (max-width: 768px) {
  .character-creation-section {
    min-height: auto;
  }

  #characterForm {
    flex-direction: column;
    gap: 1rem;
  }

  .basic-info-column {
    border-right: none;
    border-bottom: 1px solid var(--panel-border);
    padding-right: 0;
    padding-bottom: 1rem;
    margin-bottom: 1rem;
  }

  .attributes-column {
    padding-left: 0;
  }

  .attributes-container {
    display: grid;
    grid-template-columns: repeat(2, 1fr);
    gap: 0.75rem;
  }
}

/* Correção para telas muito pequenas */
```

```
@media screen and (max-width: 480px) {
  .attributes-container {
    display: grid;
    grid-template-columns: repeat(2, minmax(0, 1fr)); /* Força colunas iguais */
    gap: 0.5rem;
    width: 100%;
  }

  .attributes-container .field {
    width: 100%;
    margin: 0 0 0.5rem 0;
  }

  /* Garante que todos os inputs tenham a mesma largura */
  .attributes-container .input {
    width: 100%;
    font-size: 1rem;
    height: 2rem;
    padding: 0.25rem;
    box-sizing: border-box;
    max-width: none;
  }

  /* Remover qualquer estilo que possa estar causando o problema */
  .attributes-container .field:nth-child(odd),
  .attributes-container .field:nth-child(even) {
    width: 100%;
    max-width: none;
  }

  /* Forçar largura igual para todos os campos */
  #attrStr,
  #attrDex,
  #attrCon,
  #attrInt,
  #attrWis,
  #attrCha {
    width: 100% !important;
    max-width: none !important;
    min-width: 0 !important;
  }
}
```

File: js\jquery-companion.js

==========

```
/**
 * Mago Companion - Implementação com jQuery
 * Uma versão simplificada e confiável do assistente flutuante
 */

$(document).ready(() => {
    // Adicionar CSS necessário
    $("<style>")
      .prop("type", "text/css")
      .html(`
            @keyframes pulse {
                0% { transform: scale(1); }
                50% { transform: scale(1.05); }
                100% { transform: scale(1); }
            }
            .mago-pulse {
                animation: pulse 1s infinite;
            }
            .mago-fade-in {
                animation: fadeIn 0.3s forwards;
            }
            @keyframes fadeIn {
                from { opacity: 0; transform: translateY(10px); }
                to { opacity: 1; transform: translateY(0); }
            }
            #mago-companion {
                transition: opacity 0.3s ease;
            }
            #mago-speech-bubble {
                transition: opacity 0.3s ease, transform 0.3s ease;
                box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);
            }
        `)
      .appendTo("head")

    // Criar e adicionar o Mago Companion ao body
    $("body").append(`
        <div id="mago-companion" style="position: fixed; bottom: 20px; right: 20px; z-index: 9999; width: 80px; height: 80px;">
            <div id="mago-speech-bubble" style="display: none; position: absolute; bottom: 0; right: 90px; width: 250px; max-width:
                <p id="mago-text">Olá, aventureiro! Estou aqui para ajudá-lo a criar sua lenda em Arton!</p>
                <div style="position: absolute; top: 50%; right: -10px; width: 0; height: 0; margin-top: -10px; border-top: 10px solid tran
            </div>
            <img id="mago-avatar" src="assets/img/mago-avatar-compressed.png" alt="Mago Companion" style="width: 80px; heig
        </div>
      `)

    // Variáveis para controle do estado
    let lastMessage = ""
    let speechTimeout = null
```

```
// Função para mostrar o balão de fala
function showSpeechBubble(text, duration = null) {
  lastMessage = text
  $("#mago-text").text(text)
  $("#mago-speech-bubble").fadeIn(200).addClass("mago-fade-in")

  // Adicionar efeito de pulso ao avatar
  $("#mago-avatar").addClass("mago-pulse")
  setTimeout(() => {
    $("#mago-avatar").removeClass("mago-pulse")
  }, 1000)

  // Se houver um timeout anterior, limpe-o
  if (speechTimeout) {
    clearTimeout(speechTimeout)
    speechTimeout = null
  }

  // Se uma duração for especificada, esconda o balão após esse tempo
  if (duration) {
    speechTimeout = setTimeout(() => {
      hideSpeechBubble()
    }, duration)
  }
}

// Função para esconder o balão de fala
function hideSpeechBubble() {
  $("#mago-speech-bubble").fadeOut(200).removeClass("mago-fade-in")
}

// Função para alternar a visibilidade do balão de fala
function toggleSpeechBubble() {
  if ($("#mago-speech-bubble").is(":visible")) {
    hideSpeechBubble()
  } else {
    if (lastMessage) {
      showSpeechBubble(lastMessage)
    } else {
      greet()
    }
  }
}

// Função para saudação inicial
function greet() {
  const greetings = [
    "Bem-vindo ao Forjador de Lendas! Sou Merlin, seu guia em Arton!",
    "Um novo herói para Arton! Vamos criar uma lenda épica juntos!",
    "Por minhas barbas mágicas! Que tal forjar um destino no Reinado?",
  ]
  showSpeechBubble(greetings[Math.floor(Math.random() * greetings.length)])
```

```
}

// Resposta para mudança de nome
function onNameInput(e) {
  const name = $(e.target).val().trim()
  if (name.length > 2) {
    const responses = [
      `${name}? Um nome que ecoará em Arton! Escolha sua raça!`,
      `Ah, ${name}! Vejo um futuro lendário no Reinado para você!`,
      `${name}... *acaricia a barba* Um herói contra a Tormenta, talvez?`,
    ]
    showSpeechBubble(responses[Math.floor(Math.random() * responses.length)])
  }
}

// Resposta para mudança de raça
function onRaceChange(e) {
  const race = $(e.target).val()
  const raceResponses = {
    Humano: "Humanos! Tão versáteis quanto os ventos de Arton!",
    Anão: "Um anão de Doherimm? Só não me peça para cavar contigo!",
    Dahllan: "Uma dahllan! Allihanna deve estar orgulhosa de você!",
    Elfo: "Um elfo de Lenórienn? Elegante, mas não se perca em séculos!",
    Goblin: "Um goblin? *checa os bolsos* Cuidado com Tollon em você!",
    Lefou: "Um lefou? *recua* Espero que a Tormenta não te siga!",
    Minotauro: "Um minotauro! Só não quebre minha torre com esses chifres!",
    Qareen: "Um qareen! Cuidado com os desejos que conceder por aí!",
    Golem: "Um golem? *observa* Quem te trouxe à vida, hein?",
    Hynne: "Um hynne! Pequeno, mas cheio de sorte, aposto!",
    Kliren: "Um kliren! Sua inteligência vai resolver muitos enigmas!",
    Medusa: "Uma medusa? *evita o olhar* Não me transforme em pedra!",
    Osteon: "Um osteon? *treme* Espero que não seja meu esqueleto animado!",
    "Sereia/Tritão": "Um sereia ou tritão! Não molhe minha túnica, por favor!",
    Sílfide: "Uma sílfide! Você flutua como os ventos de Wynlla!",
    Suraggel: "Um suraggel! Divino ou infernal, escolha com cuidado!",
    Trong: "Um trong! *se afasta* Não me coma, eu sou só um mago velho!",
  }
  showSpeechBubble(raceResponses[race] || "Uma raça intrigante para Arton!")
}

// Resposta para mudança de classe
function onClassChange(e) {
  const characterClass = $(e.target).val()
  const classResponses = {
    Arcanista: "Um arcanista! *limpa uma lágrima* Um irmão das artes mágicas!",
    Bárbaro: "Um bárbaro! Só não quebre minhas coisas na sua fúria!",
    Bardo: "Um bardo! Cante minhas glórias... quer dizer, as suas!",
    Bucaneiro: "Um bucaneiro! Espero que não roube meu cajado no mar!",
    Caçador: "Um caçador! Perfeito para rastrear horrores da Tormenta!",
    Cavaleiro: "Um cavaleiro! Sua honra brilha mais que minha magia!",
    Clérigo: "Um clérigo! Que os deuses do Pantheon te abençoem!",
    Druida: "Um druida! Allihanna aprova, mas sem lobos na minha torre!",
    Guerreiro: "Um guerreiro! Pronto para as batalhas de Arton!",
```

```
    Inventor: "Um inventor! *se anima* Mostre-me seus gadgets!",
    Ladino: "Um ladino! *esconde o bolso* Cuidado com os Ladrões de Deheon!",
    Lutador: "Um lutador! Seus punhos vão impressionar em Arton!",
    Nobre: "Um nobre! Sua presença é digna de Valkaria!",
    Paladino: "Um paladino! A luz de Khalmyr guia seus passos!",
  }
  showSpeechBubble(classResponses[characterClass] || "Uma classe fascinante para Arton!")
}

// Resposta para mudança de alinhamento
function onAlignmentChange(e) {
  const alignment = $(e.target).val()
  if (alignment.includes("Mau")) {
    showSpeechBubble("Hmmm... *suspeita* Não traga a Tormenta para minha torre!")
  } else if (alignment.includes("Bom")) {
    showSpeechBubble("Um coração nobre! Arton precisa de você contra o mal!")
  } else {
    showSpeechBubble("Neutro? Equilíbrio é sábio, mas não seja indeciso!")
  }
}

// Resposta para rolar atributos
function onRollAttributes() {
  const responses = [
    "*Agita as mãos* Que os dados de Nimb decidam seu destino!",
    "Rolando! Que Tanna-Toh revele sua força interior!",
    "*Sopra os dados* Um toque de Wynna para sua sorte!",
  ]
  showSpeechBubble(responses[Math.floor(Math.random() * responses.length)], 4000)
}

// Configurar event listeners
$("#mago-avatar").click(toggleSpeechBubble)

// Esconder o balão quando clicar fora
$(document).on("click", (e) => {
  if (!$("#mago-avatar").is(e.target) && !$("#mago-speech-bubble").has(e.target).length) {
    hideSpeechBubble()
  }
})

// Adicionar event listeners para os campos do formulário
$("#charName").on("input", onNameInput)
$("#charRace").on("change", onRaceChange)
$("#charClass").on("change", onClassChange)
$("#charAlignment").on("change", onAlignmentChange)
$("#rollAttributes").on("click", onRollAttributes)

// Eventos para outros botões
$("#generateLore").on("click", () => {
  showSpeechBubble("Deixe-me consultar os pergaminhos antigos...", 3000)
})
```

```
  $("#saveCharacter").on("click", () => {
    showSpeechBubble("Seu herói foi registrado em meu grimório!", 3000)
  })

  // Exibir saudação inicial após um breve delay
  setTimeout(greet, 1000)

  // Expor a API do Mago Companion para o resto da aplicação
  window.magoCompanion = {
    speak: (text, duration) => {
      showSpeechBubble(text, duration)
    },
    onRaceChange: (e) => {
      onRaceChange(e)
    },
    onClassChange: (e) => {
      onClassChange(e)
    },
    onAlignmentChange: (e) => {
      onAlignmentChange(e)
    },
    onRollAttributes: () => {
      onRollAttributes()
    },
    hideSpeechBubble: () => {
      hideSpeechBubble()
    },
  }

  console.log("Mago Companion jQuery inicializado com sucesso!")
})
```

File: css\character-cards.css

==========

```
/* Cartões de personagens */
.character-card {
    background-color: rgba(35, 25, 15, 0.9);
    border: 2px solid var(--panel-border);
    border-radius: 8px;
    padding: 1rem;
    margin: 0.5rem;
    width: 200px;
    cursor: pointer;
    transition: all 0.3s;
}

.character-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 5px 15px rgba(0, 0, 0, 0.5);
    border-color: var(--accent-color);
}

.character-card .card-header {
    background-color: rgba(30, 20, 10, 0.8);
    border-radius: 5px 5px 0 0;
    padding: 0.5rem;
}

.character-card .card-content {
    padding: 0.5rem;
}

.character-card .subtitle {
    color: var(--text-color);
}

.character-avatar {
    background-color: var(--bg-color);
    border: 2px solid var(--panel-border);
    border-radius: 50%;
    width: 80px;
    height: 80px;
    margin: 0 auto;
    display: flex;
    align-items: center;
    justify-content: center;
}

.character-avatar i {
    font-size: 2.5rem;
    color: var(--accent-color);
}
```

```
.empty-list-message {
    color: rgba(232, 219, 190, 0.6);
    font-style: italic;
    text-align: center;
    padding: 2rem;
}

.saved-characters-section {
    background-color: rgba(35, 25, 15, 1);
    width: 100vw;
    padding: 2rem 0;
    margin-top: 2rem;
    position: relative;
}
```

File: css\footer.css

==========

```
/* Footer */
.footer {
    background-color: rgba(30, 20, 10, 1);
    border-top: 1px solid var(--panel-border);
    padding: 1.5rem;
}

#footer-text strong {
    color: var(--text-color);
}
```

File: css\buttons.css

==========

```
/* Botões */
.button {
    font-family: 'MedievalSharp', cursive;
    text-transform: uppercase;
    letter-spacing: 1px;
    transition: all 0.3s ease;
}

.button:hover {
    transform: translateY(-2px);
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
}

.button.is-primary {
    background: linear-gradient(145deg, var(--primary-color), var(--secondary-color));
}

.button.is-primary:hover {
    background: linear-gradient(145deg, var(--secondary-color), var(--primary-color));
}

.button.is-info {
    background-color: #4a6fa5;
}

.button.is-info:hover {
    background-color: #3a5a8c;
}
```

File: css\companion.css

==========

```
/* Companion - estilos base */
.companion-container {
    transition: transform 0.1s ease, top 0.3s ease, bottom 0.3s ease;
    will-change: transform; /* Melhora o desempenho de animações */
  }

  .companion-avatar {
    transition: transform 0.3s ease, box-shadow 0.3s ease;
  }

  .companion-avatar.dragging {
    transform: scale(1.05);
    box-shadow: 0 0 15px var(--accent-color);
  }

  .companion-speech-bubble {
    transition: opacity 0.3s ease, transform 0.3s ease;
    box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);
  }

  .companion-speech-bubble.hidden {
    opacity: 0;
    transform: scale(0.8);
    pointer-events: none;
  }

  .companion-speech-bubble p {
    word-wrap: break-word;
  }

  /* Estilos para desktop - layout original */
  @media screen and (min-width: 1024px) {
    .companion-container {
      width: 67%;
      position: fixed;
      top: -1%;
      right: -12%;
      display: block;
      padding: 1rem;
      z-index: -1;
      /* Remover propriedades de arrasto para desktop */
      touch-action: auto;
    }

    .companion-avatar {
      height: 100%;
      width: 100%;
      object-fit: cover;
      position: relative;
```

```
    cursor: default; /* Não é arrastável em desktop */
    border-radius: 0; /* Restaurar forma original */
    border: none;
    box-shadow: none;
  }

  .companion-speech-bubble {
    background: rgba(255, 255, 255, 0.9);
    border: 2px solid var(--panel-border);
    border-radius: 15px;
    padding: 15px;
    width: 280px;
    position: absolute;
    top: 13%;
    left: 20%;
    color: var(--secondary-color);
    font-family: "MedievalSharp", cursive;
    z-index: 5;
  }

  .companion-speech-bubble::after {
    content: "";
    position: absolute;
    bottom: -10px;
    right: 30px;
    width: 0;
    height: 0;
    border-left: 10px solid transparent;
    border-right: 10px solid transparent;
    border-top: 10px solid var(--panel-border);
  }
}

/* Estilos para mobile - ícone flutuante */
@media screen and (max-width: 1023px) {
  .companion-container {
    position: fixed; /* Garante que o companion fique fixo na viewport */
    right: 0; /* Fixado no lado direito */
    width: 80px;
    height: 80px;
    z-index: 1000;
    touch-action: none; /* Previne comportamento padrão de toque */
  }

  .companion-avatar {
    width: 80px;
    height: 80px;
    border-radius: 50%;
    border: 2px solid var(--accent-color);
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.3);
    object-fit: cover;
    background-color: var(--panel-bg);
    cursor: grab;
```

```
    -webkit-tap-highlight-color: transparent; /* Remove highlight ao tocar em dispositivos móveis */
  }

  .companion-avatar:active {
    cursor: grabbing;
  }

  .companion-speech-bubble {
    position: absolute;
    bottom: 90px;
    right: 0;
    width: 250px;
    max-width: 80vw;
    background: var(--panel-bg);
    border: 2px solid var(--accent-color);
    border-radius: 15px;
    padding: 15px;
    z-index: 1001;
    transform-origin: bottom right;
  }

  .companion-speech-bubble::after {
    content: "";
    position: absolute;
    bottom: -10px;
    right: 30px;
    width: 0;
    height: 0;
    border-left: 10px solid transparent;
    border-right: 10px solid transparent;
    border-top: 10px solid var(--accent-color);
  }
}

/* Animações para o companion */
@keyframes pulse {
  0% {
    transform: scale(1);
  }
  50% {
    transform: scale(1.05);
  }
  100% {
    transform: scale(1);
  }
}

@keyframes fadeInUp {
  from {
    opacity: 0;
    transform: translateY(20px);
  }
  to {
```

```
    opacity: 1;
    transform: translateY(0);
  }
}

.companion-avatar.pulse {
  animation: pulse 1s infinite;
}

.companion-speech-bubble.fade-in {
  animation: fadeInUp 0.3s forwards;
}
```

File: css\variables.css

==========

```
/* Estilo Base */
:root {
    --primary-color: #8B5A2B;
    --secondary-color: #4A2511;
    --accent-color: #D4AF37;
    --bg-color: #2C1B0F;
    --text-color: #E8DBBE;
    --panel-bg: rgba(35, 25, 15, 0.85);
    --panel-border: #8B5A2B;
}
```

File: package.json

==========


```
{
  "name": "rpgcharactermaker2",
  "version": "1.0.0",
  "description": "Bem-vindo ao **Forjador de Lendas**, um gerador de personagens para o sistema de RPG **Tormenta 20**, am
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```