```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MeuPet IA - Protótipo</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        body {
            font-family: 'Comic Sans MS', cursive;
            background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
            min-height: 100vh;
            display: flex;
            justify-content: center;
            align-items: center;
        }

        .phone-container {
            width: 320px;
            height: 600px;
            background: #fff;
            border-radius: 25px;
            box-shadow: 0 20px 40px rgba(0,0,0,0.3);
            overflow: hidden;
            position: relative;
        }

        .status-bar {
            height: 30px;
            background: #f8f9fa;
            display: flex;
            justify-content: space-between;
            align-items: center;
            padding: 0 15px;
            font-size: 12px;
            color: #666;
        }

        .game-area {
```

```
    height: 400px;
    background: linear-gradient(180deg, #87CEEB 0%, #98FB98 100%);
    position: relative;
    overflow: hidden;
}

.clouds {
    position: absolute;
    top: 20px;
    left: 0;
    right: 0;
}

.cloud {
    background: rgba(255,255,255,0.8);
    border-radius: 50px;
    position: absolute;
    animation: float 6s ease-in-out infinite;
}

.cloud1 {
    width: 60px;
    height: 30px;
    left: 20px;
    animation-delay: 0s;
}

.cloud2 {
    width: 80px;
    height: 35px;
    right: 30px;
    animation-delay: 2s;
}

@keyframes float {
    0%, 100% { transform: translateY(0); }
    50% { transform: translateY(-10px); }
}

.pet-container {
    position: absolute;
    bottom: 120px;
    left: 50%;
    transform: translateX(-50%);
```

```
    text-align: center;
}

.pet {
    width: 120px;
    height: 120px;
    border-radius: 50%;
    margin: 0 auto 10px;
    position: relative;
    transition: all 0.8s ease;
    cursor: pointer;
    animation: bounce 3s ease-in-out infinite;
}

@keyframes bounce {
    0%, 100% { transform: translateY(0); }
    50% { transform: translateY(-5px); }
}

.pet.sad {
    background: linear-gradient(145deg, #8B7D8B, #696969);
    box-shadow: inset 0 5px 15px rgba(0,0,0,0.3);
}

.pet.happy {
    background: linear-gradient(145deg, #FFB6C1, #FF69B4);
    box-shadow: 0 10px 20px rgba(255,105,180,0.4);
    transform: scale(1.1);
}

.pet.evolved {
    background: linear-gradient(145deg, #FFD700, #FFA500);
    box-shadow: 0 15px 30px rgba(255,215,0,0.5);
    transform: scale(1.2);
}

.eyes {
    position: absolute;
    top: 35px;
    left: 50%;
    transform: translateX(-50%);
}

.eye {
```

```
    width: 18px;
    height: 18px;
    background: #000;
    border-radius: 50%;
    display: inline-block;
    margin: 0 8px;
    position: relative;
    transition: all 0.5s ease;
}

.eye.sad {
    transform: rotate(-15deg);
    opacity: 0.7;
}

.eye.happy {
    background: #2E8B57;
    transform: scale(1.2);
}

.eye.evolved {
    background: radial-gradient(circle, #00FFFF 30%, #0000FF 70%);
    transform: scale(1.3);
    box-shadow: 0 0 10px rgba(0,255,255,0.5);
}

.mouth {
    position: absolute;
    top: 70px;
    left: 50%;
    transform: translateX(-50%);
    transition: all 0.5s ease;
}

.mouth.sad {
    width: 30px;
    height: 15px;
    border: 3px solid #333;
    border-top: none;
    border-radius: 0 0 30px 30px;
    transform: translateX(-50%) rotate(180deg);
}

.mouth.happy {
```

```
    width: 40px;
    height: 20px;
    border: 4px solid #2E8B57;
    border-top: none;
    border-radius: 0 0 40px 40px;
}

.mouth.evolved {
    width: 45px;
    height: 22px;
    border: 4px solid #FFD700;
    border-top: none;
    border-radius: 0 0 45px 45px;
    box-shadow: 0 0 8px rgba(255,215,0,0.3);
}

.pet-name {
    font-size: 14px;
    color: #333;
    margin-top: 5px;
    transition: all 0.5s ease;
}

.stats {
    height: 80px;
    background: #f8f9fa;
    padding: 10px;
    border-top: 1px solid #dee2e6;
}

.stat-bar {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 8px;
}

.stat-label {
    font-size: 12px;
    color: #666;
    width: 60px;
}

.progress-bar {
```

```
    flex: 1;
    height: 12px;
    background: #e9ecef;
    border-radius: 6px;
    margin: 0 10px;
    overflow: hidden;
}

.progress-fill {
    height: 100%;
    border-radius: 6px;
    transition: width 0.5s ease;
}

.happiness .progress-fill { background: linear-gradient(90deg, #FF6B6B, #FF8E85); }
.health .progress-fill { background: linear-gradient(90deg, #4ECDC4, #44A08D); }
.evolution .progress-fill { background: linear-gradient(90deg, #FFD93D, #FF6B6B); }

.actions {
    height: 90px;
    padding: 15px;
    display: flex;
    justify-content: space-around;
    align-items: center;
    background: #fff;
}

.action-btn {
    width: 60px;
    height: 60px;
    border: none;
    border-radius: 50%;
    font-size: 24px;
    cursor: pointer;
    transition: all 0.3s ease;
    box-shadow: 0 4px 8px rgba(0,0,0,0.2);
}

.feed { background: linear-gradient(145deg, #FF9A9E, #FECFEF); }
.play { background: linear-gradient(145deg, #A8EDEA, #FED6E3); }
.clean { background: linear-gradient(145deg, #FFECD2, #FCB69F); }
.talk { background: linear-gradient(145deg, #A8E6CF, #DCEDC1); }

.action-btn:hover {
```

```
    transform: translateY(-3px);
    box-shadow: 0 6px 12px rgba(0,0,0,0.3);
}

.chat-bubble {
    position: absolute;
    top: 50px;
    left: 20px;
    right: 20px;
    background: rgba(255,255,255,0.95);
    padding: 15px;
    border-radius: 15px;
    box-shadow: 0 5px 15px rgba(0,0,0,0.2);
    opacity: 0;
    transform: translateY(-20px);
    transition: all 0.5s ease;
    font-size: 14px;
    text-align: center;
}

.chat-bubble.show {
    opacity: 1;
    transform: translateY(0);
}

.evolution-effect {
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    pointer-events: none;
    opacity: 0;
}

.sparkle {
    position: absolute;
    width: 4px;
    height: 4px;
    background: #FFD700;
    border-radius: 50%;
    animation: sparkle 1s ease-out forwards;
}
```

```
        @keyframes sparkle {
            0% {
                opacity: 1;
                transform: scale(0);
            }
            50% {
                opacity: 1;
                transform: scale(1);
            }
            100% {
                opacity: 0;
                transform: scale(0) translateY(-50px);
            }
        }
    </style>
</head>
<body>
    <div class="phone-container">
        <div class="status-bar">
            <span>🔋 98%</span>
            <span><strong>MeuPet IA</strong></span>
            <span>📶 ●●●●</span>
        </div>

        <div class="game-area">
            <div class="clouds">
                <div class="cloud cloud1"></div>
                <div class="cloud cloud2"></div>
            </div>

            <div class="chat-bubble" id="chatBubble">
                Oi... eu sou Lumina... você pode cuidar de mim? 🥺
            </div>

            <div class="pet-container">
                <div class="pet sad" id="pet">
                    <div class="eyes">
                        <div class="eye sad"></div>
                        <div class="eye sad"></div>
                    </div>
                    <div class="mouth sad"></div>
                </div>
                <div class="pet-name" id="petName">Lumina - Filhote Triste</div>
            </div>
```

```
        <div class="evolution-effect" id="evolutionEffect"></div>
    </div>

    <div class="stats">
        <div class="stat-bar happiness">
            <span class="stat-label">😊 Amor</span>
            <div class="progress-bar">
                <div class="progress-fill" id="happinessBar" style="width: 20%"></div>
            </div>
            <span id="happinessValue">20%</span>
        </div>

        <div class="stat-bar health">
            <span class="stat-label">💚 Saúde</span>
            <div class="progress-bar">
                <div class="progress-fill" id="healthBar" style="width: 30%"></div>
            </div>
            <span id="healthValue">30%</span>
        </div>

        <div class="stat-bar evolution">
            <span class="stat-label">✨ Evolução</span>
            <div class="progress-bar">
                <div class="progress-fill" id="evolutionBar" style="width: 10%"></div>
            </div>
            <span id="evolutionValue">10%</span>
        </div>
    </div>

    <div class="actions">
        <button class="action-btn feed" onclick="feedPet()">🍎</button>
        <button class="action-btn play" onclick="playWithPet()">🎾</button>
        <button class="action-btn clean" onclick="cleanPet()">🛁</button>
        <button class="action-btn talk" onclick="talkToPet()">💬</button>
    </div>
</div>

<script>
    let petStats = {
        happiness: 20,
        health: 30,
        evolution: 10,
        stage: 'sad' // sad, happy, evolved
```

```
};

const messages = {
    sad: [
        "Obrigada por cuidar de mim... 🥺",
        "Eu me sinto um pouco melhor...",
        "Você é gentil comigo... isso é bom 💙",
        "Será que um dia vou ser bonita?",
        "Sua presença me aquece o coração..."
    ],
    happy: [
        "Você me faz muito feliz! 😊",
        "Estou me sentindo mais confiante!",
        "Obrigada por acreditar em mim! 💖",
        "Sinto que estou mudando por dentro...",
        "Você viu? Minhas cores estão mais vivas!"
    ],
    evolved: [
        "Olhe como eu me tornei! ✨",
        "Você me ajudou a descobrir minha verdadeira beleza!",
        "Agora posso ajudar outras criaturas como você me ajudou! 🌟",
        "Obrigada por nunca desistir de mim! 💫",
        "Juntos, somos invencíveis!"
    ]
};

function updateStats() {
    document.getElementById('happinessBar').style.width = petStats.happiness + '%';
    document.getElementById('healthBar').style.width = petStats.health + '%';
    document.getElementById('evolutionBar').style.width = petStats.evolution + '%';

    document.getElementById('happinessValue').textContent = petStats.happiness + '%';
    document.getElementById('healthValue').textContent = petStats.health + '%';
    document.getElementById('evolutionValue').textContent = petStats.evolution + '%';

    checkEvolution();
}

function checkEvolution() {
    const pet = document.getElementById('pet');
    const eyes = pet.querySelectorAll('.eye');
    const mouth = pet.querySelector('.mouth');
    const petName = document.getElementById('petName');
```

```
    if (petStats.evolution >= 80 && petStats.stage !== 'evolved') {
        // Evolução final
        petStats.stage = 'evolved';
        pet.className = 'pet evolved';
        eyes.forEach(eye => eye.className = 'eye evolved');
        mouth.className = 'mouth evolved';
        petName.textContent = 'Lumina - Criatura Celestial ✨';
        createEvolutionEffect();
        showMessage("Incrível! Eu evoluí para minha forma final! ✨🌟");

    } else if (petStats.evolution >= 40 && petStats.stage === 'sad') {
        // Primeira evolução
        petStats.stage = 'happy';
        pet.className = 'pet happy';
        eyes.forEach(eye => eye.className = 'eye happy');
        mouth.className = 'mouth happy';
        petName.textContent = 'Lumina - Criatura Alegre 💖';
        createEvolutionEffect();
        showMessage("Olha! Estou ficando mais bonita! 😊💖");
    }
}

function createEvolutionEffect() {
    const effect = document.getElementById('evolutionEffect');
    effect.style.opacity = '1';

    for (let i = 0; i < 15; i++) {
        setTimeout(() => {
            const sparkle = document.createElement('div');
            sparkle.className = 'sparkle';
            sparkle.style.left = Math.random() * 100 + '%';
            sparkle.style.top = Math.random() * 100 + '%';
            effect.appendChild(sparkle);

            setTimeout(() => {
                if (sparkle.parentNode) {
                    sparkle.parentNode.removeChild(sparkle);
                }
            }, 1000);
        }, i * 100);
    }

    setTimeout(() => {
        effect.style.opacity = '0';
```

```
        }, 2000);
    }

    function showMessage(message) {
        const bubble = document.getElementById('chatBubble');
        bubble.textContent = message;
        bubble.classList.add('show');

        setTimeout(() => {
            bubble.classList.remove('show');
        }, 3000);
    }

    function feedPet() {
        petStats.health = Math.min(100, petStats.health + 15);
        petStats.happiness = Math.min(100, petStats.happiness + 10);
        petStats.evolution = Math.min(100, petStats.evolution + 5);
        updateStats();

        const messages_feed = {
            sad: ["Hmm... isso está gostoso... obrigada 🥺", "Minha barriguinha está menos
vazia..."],
            happy: ["Delicioso! Você conhece meus gostos! 😊", "Isso me dá energia para
brincar!"],
            evolved: ["Que refeição divina! ✨", "Posso sentir a energia cósmica nesta comida!
🌟"]
        };

        const messageArray = messages_feed[petStats.stage];
        showMessage(messageArray[Math.floor(Math.random() * messageArray.length)]);
    }

    function playWithPet() {
        petStats.happiness = Math.min(100, petStats.happiness + 20);
        petStats.health = Math.min(100, petStats.health + 5);
        petStats.evolution = Math.min(100, petStats.evolution + 8);
        updateStats();

        const messages_play = {
            sad: ["Isso... isso é divertido! 🥺✨", "Obrigada por brincar comigo..."],
            happy: ["Adorei brincar! Vamos de novo? 😊", "Você me faz rir! Haha! 💖"],
            evolved: ["Que jogo fantástico! ✨", "Brincar com você é mágico! 🌟"]
        };
```

```
        const messageArray = messages_play[petStats.stage];
        showMessage(messageArray[Math.floor(Math.random() * messageArray.length)]);
    }

    function cleanPet() {
        petStats.health = Math.min(100, petStats.health + 20);
        petStats.happiness = Math.min(100, petStats.happiness + 15);
        petStats.evolution = Math.min(100, petStats.evolution + 6);
        updateStats();

        const messages_clean = {
            sad: ["Obrigada por me limpar... me sinto um pouco melhor", "Você é tão cuidadosa
comigo... 💙"],
            happy: ["Agora estou limpinha e cheirosa! 😊", "Você cuida tão bem de mim! 💖"],
            evolved: ["Brilhando como uma estrela! ✨", "Pureza celestial restaurada! 🌟"]
        };

        const messageArray = messages_clean[petStats.stage];
        showMessage(messageArray[Math.floor(Math.random() * messageArray.length)]);
    }

    function talkToPet() {
        petStats.happiness = Math.min(100, petStats.happiness + 25);
        petStats.evolution = Math.min(100, petStats.evolution + 10);
        updateStats();

        const messageArray = messages[petStats.stage];
        showMessage(messageArray[Math.floor(Math.random() * messageArray.length)]);
    }

    // Clique no pet para interação especial
    document.getElementById('pet').addEventListener('click', function() {
        petStats.happiness = Math.min(100, petStats.happiness + 15);
        petStats.evolution = Math.min(100, petStats.evolution + 5);
        updateStats();

        const messages_pet = {
            sad: ["Seu toque é reconfortante... 🥺💙", "Você realmente se importa comigo..."],
            happy: ["Adoro seus carinhos! 😊💖", "Isso me faz sentir especial!"],
            evolved: ["Sua energia se conecta com a minha! ✨", "Somos uma equipe perfeita!
🌟"]
        };

        const messageArray = messages_pet[petStats.stage];
```

```
        showMessage(messageArray[Math.floor(Math.random() * messageArray.length)]);
    });

    // Inicialização
    updateStats();

    // Simulação de degradação lenta das stats (realismo)
    setInterval(() => {
        if (petStats.happiness > 0) petStats.happiness = Math.max(0, petStats.happiness - 0.5);
        if (petStats.health > 0) petStats.health = Math.max(0, petStats.health - 0.3);
        updateStats();
    }, 10000); // A cada 10 segundos
  </script>
</body>
</html>
```