**Historias de Usuario para Pet2Pet V1**

**Consideraciones para la Selección:**

1. **Equipo Trainee/Junior:**
   * Priorizar funcionalidades con complejidad técnica manejable
   * Evitar implementaciones que requieran arquitecturas complejas
   * Favorecer soluciones estándar sobre customizaciones complejas
2. **Objetivos V1:**
   * Crear una base sólida y funcional
   * Entregar valor inmediato a usuarios
   * Permitir crecimiento futuro
   * Mantener la calidad del código

**Épicas e Historias Seleccionadas para V1**

**ÉPICA 1: Gestión de Usuarios (Core)**

**Frontend**

1. **[FE-GU-01]** Registro con email
   * Formulario básico de registro
   * Validaciones esenciales
   * Mensajes de error claros
2. **[FE-GU-03]** Login
   * Formulario de login simple
   * Recuperación de contraseña básica
   * Manejo de sesión
3. **[FE-GU-04]** Gestión de perfil básico
   * Edición de datos personales
   * Upload de foto de perfil
   * Información básica
4. **[FE-GU-05]** Gestión básica de mascotas
   * Agregar mascota
   * Información básica de mascota
   * Foto de mascota

**Backend**

1. **[BE-GU-01]** Autenticación básica
   * Registro de usuarios
   * Login
   * JWT simple
   * Validaciones básicas
2. **[BE-GU-02]** Gestión de perfiles
   * CRUD de perfiles
   * Almacenamiento de fotos
   * Validaciones básicas

**ÉPICA 2: Interacción Social (Básica)**

**Frontend**

1. **[FE-IS-01]** Creación de publicaciones
   * Texto
   * Una imagen
   * Privacidad básica (público/privado)
2. **[FE-IS-02]** Interacciones básicas
   * Me gusta
   * Comentarios simples
   * Feed básico

**Backend**

1. **[BE-IS-01]** Gestión de publicaciones
   * CRUD de publicaciones
   * Almacenamiento de imágenes
   * Feed simple

**ÉPICA 3: Búsqueda Simple**

**Frontend**

1. **[FE-BR-01]** Búsqueda básica
   * Búsqueda por nombre
   * Filtros básicos
   * Resultados paginados

**Backend**

1. **[BE-BR-01]** Motor de búsqueda simple
   * Búsqueda por campos básicos
   * Filtrado simple
   * Paginación

**ÉPICA 4: Mascota Virtual (Versión Básica)**

**Frontend**

1. **[FE-MV-01]** Creación de mascota virtual
   * Personalización básica
   * Nombre
   * Tipo de mascota
2. **[FE-MV-02]** Interacciones básicas
   * Alimentar
   * Jugar
   * Ver estado

**Backend**

1. **[BE-MV-01]** Gestión básica de mascota virtual
   * CRUD de mascota virtual
   * Estados básicos
   * Cálculos simples de felicidad

**Stack Tecnológico Recomendado para V1**

**Frontend:**

* React 18
* React Router
* Axios
* React Hook Form
* Tailwind CSS
* Context API (en lugar de Redux para empezar)

**Backend:**

* FastAPI
* SQLAlchemy
* JWT para autenticación
* PostgreSQL
* Almacenamiento simple de archivos

**Infraestructura:**

* Despliegue simple (puede ser Render, Railway o similar)
* Base de datos hosteada (ElephantSQL o similar)
* Almacenamiento S3 básico

**Recomendaciones para el Desarrollo**

**1. Prácticas de Código:**

* Usar tipos estáticos en Python
* Implementar validaciones básicas
* Mantener el código limpio y documentado
* Crear pruebas unitarias básicas

**2. Arquitectura:**

* Mantener la estructura simple pero escalable
* Usar patrones básicos pero efectivos
* Implementar manejo de errores básico
* Seguir principios SOLID básicos

**3. Base de Datos:**

* Diseño de tablas simple y directo
* Índices básicos
* Relaciones simples
* Evitar consultas complejas

**4. UI/UX:**

* Diseño limpio y funcional
* Componentes reutilizables básicos
* Feedback claro al usuario
* Responsive design básico

**5. Seguridad:**

* Autenticación básica pero segura
* Validación de inputs
* Sanitización de datos
* Protección básica de rutas

**Roadmap de Implementación Sugerido**

1. **Semana 1-2:**
   * Setup del proyecto
   * Autenticación básica
   * Estructura de base de datos
2. **Semana 3-4:**
   * Gestión de perfiles
   * Upload de imágenes básico
   * CRUD de mascotas
3. **Semana 5-6:**
   * Publicaciones básicas
   * Feed simple
   * Interacciones básicas
4. **Semana 7-8:**
   * Mascota virtual básica
   * Búsqueda simple
   * Testing y correcciones

Esta versión 1 proporciona:

* Funcionalidad core completa
* Base sólida para futuras características
* Complejidad manejable para el equipo
* Valor real para usuarios desde el inicio