# 🚀 HVMX Project Plan

## 📊 Resumo Executivo

**Total LOC**: ~5,000
**Duração**: 3 sprints (8-12 semanas)
**Status**: Ready to start

---

## 🗓️ Cronograma de Sprints

### Sprint 1: Core + MVP (4 semanas)

**Objetivo**: Runtime básico funcionando com Vulkan

| Arquivo | LOC | Status | Testes |
|---|---|---|---|
| `crates/hvmx-core/src/lib.rs` | 200 | 📝 Spec | ✅ 5 tests |
| `crates/hvmx-core/src/port.rs` | 150 | 📝 Spec | ✅ 8 tests |
| `crates/hvmx-core/src/net.rs` | 250 | 📝 Spec | ✅ 10 tests |
| `crates/hvmx-core/src/interact.rs` | 300 | 📝 Spec | ✅ 15 tests |
| `crates/hvmx-core/src/numb.rs` | 150 | 📝 Spec | ✅ 6 tests |
| `crates/hvmx-core/src/book.rs` | 150 | 📝 Spec | ✅ 4 tests |
| `crates/hvmx-jit/src/lib.rs` | 300 | 📝 Spec | ✅ 8 tests |
| `crates/hvmx-jit/src/ir.rs` | 400 | 📝 Spec | ✅ 12 tests |
| `crates/hvmx-jit/src/runtime.rs` | 300 | 📝 Spec | ✅ 10 tests |
| `crates/hvmx-jit/src/backend/vulkan.rs` | 600 | 📝 Spec | ✅ 20 tests |
| **SUBTOTAL Sprint 1** | **2,800** | **Ready** | **98 tests** |

**Deliverables**:

- ✅ Core runtime compilando
- ✅ Backend Vulkan funcional
- ✅ Suite de testes básica
- ✅ CI/CD pipeline

- ✅ Benchmarks preliminares

---

**Sprint 2: Backends + Memory (3 semanas)**

**Objetivo**: Multi-platform support + otimizações de memória

| Arquivo | LOC | Status | Testes |
|---|---|---|---|
| crates/hvmx-jit/src/backend/metal.rs | 500 | 📝 Spec | ✅ 18 tests |
| crates/hvmx-jit/src/backend/cuda.rs | 400 | 📝 Spec | ✅ 15 tests |
| crates/hvmx-jit/src/backend/cpu.rs | 200 | 📝 Spec | ✅ 8 tests |
| crates/hvmx-jit/src/optimizer.rs | 300 | 📝 Spec | ✅ 12 tests |
| crates/hvmx-memory/src/lib.rs | 150 | 📝 Spec | ✅ 5 tests |
| crates/hvmx-memory/src/unified.rs | 300 | 📝 Spec | ✅ 15 tests |
| crates/hvmx-memory/src/prefetch.rs | 200 | 📝 Spec | ✅ 8 tests |
| crates/hvmx-memory/src/tile.rs | 150 | 📝 Spec | ✅ 6 tests |
| **SUBTOTAL Sprint 2** | **2,200** | **Ready** | **87 tests** |

**Deliverables**:

- ✅ Metal backend (iOS/macOS)
- ✅ CUDA backend (NVIDIA)
- ✅ CPU fallback
- ✅ Unified memory allocator
- ✅ Prefetching strategies
- ✅ Cross-platform benchmarks

---

**Sprint 3: Scheduler + Production (3 semanas)**

**Objetivo**: Heterogeneous scheduling + hardening

| Arquivo | LOC | Status | Testes |
|---|---|---|---|
| `crates/hvmx-scheduler/src/lib.rs` | 150 | 📝 Spec | ✅ 5 tests |
| `crates/hvmx-scheduler/src/partition.rs` | 250 | 📝 Spec | ✅ 10 tests |
| `crates/hvmx-scheduler/src/adaptive.rs` | 200 | 📝 Spec | ✅ 8 tests |
| `crates/hvmx-cli/src/main.rs` | 150 | 📝 Spec | ✅ 6 tests |
| `crates/hvmx-cli/src/commands/*.rs` | 150 | 📝 Spec | ✅ 9 tests |
| `crates/hvmx-cli/src/ui.rs` | 100 | 📝 Spec | ✅ 4 tests |
| `examples/*.rs` | 300 | 📝 Spec | ✅ 8 tests |
| `benches/*.rs` | 400 | 📝 Spec | ✅ 12 tests |
| `tests/*.rs` | 300 | 📝 Spec | ✅ 16 tests |
| **SUBTOTAL Sprint 3** | **2,000** | **Ready** | **78 tests** |

**Deliverables**:

- ✅ Heterogeneous scheduler (CPU+GPU)
- ✅ Adaptive routing
- ✅ CLI completo
- ✅ 4 exemplos funcionais
- ✅ Documentação completa
- ✅ Release 1.0

---

## 📈 Métricas por Sprint

| Sprint | LOC Planejado | LOC Acumulado | Testes | Coverage |
|---|---|---|---|---|
| 1 | 2,800 | 2,800 | 98 | 85% |
| 2 | 2,200 | 5,000 | 185 | 87% |
| 3 | 2,000 | 7,000 | 263 | 90% |

---

## 🧪 Estratégia de Testes

### Unit Tests (60%)

- Core: Port, Net, Interact operations
- JIT: IR generation, optimization passes
- Memory: Allocation, prefetching, coherency
- Backends: Mock GPU, CPU fallback

### Integration Tests (30%)

- End-to-end reduction
- Multi-backend switching
- Memory management
- CLI workflows

### Benchmarks (10%)

- Latency: Small graphs (< 1ms)
- Throughput: Large graphs (> 100ms)
- Comparison: CPU vs Vulkan vs Metal vs CUDA
- Energy: Power efficiency metrics

---

## 📦 Estrutura de Entrega

### Cada Sprint Entrega:

```
sprint-N/
├── code/
│   ├── implemented/     # Código pronto
│   ├── tests/           # Testes passando
│   └── docs/            # Documentação atualizada
├── metrics/
│   ├── loc_report.md    # LOC por arquivo
│   ├── coverage.html    # Cobertura de testes
│   └── benchmarks.json  # Resultados de performance
└── artifacts/
    ├── CHANGELOG.md     # O que mudou
```

```
├── ROADMAP.md       # Próximos passos
└── release-notes/   # Notas de release
```

---

## 🎯 Critérios de Sucesso

### Sprint 1 ✅

- [ ] Core runtime compila sem warnings
- [ ] Backend Vulkan roda em Android/Linux
- [ ] Benchmarks mostram speedup > 2x vs CPU
- [ ] CI passa em 3 plataformas

### Sprint 2 ✅

- [ ] 3+ backends funcionando
- [ ] Zero-copy em SoCs mobile
- [ ] Benchmarks em 5+ dispositivos
- [ ] Documentação API completa

### Sprint 3 ✅

- [ ] Scheduler adaptativo funcional
- [ ] CLI user-friendly
- [ ] 4 demos funcionando
- [ ] Release candidate 1.0
- [ ] Site hvmx.dev no ar

---

## 🚀 Quick Start (Para Começar Agora)

bash

```bash
# 1. Criar estrutura
mkdir -p hvmx/{crates/{hvmx-core,hvmx-jit,hvmx-memory,hvmx-scheduler,hvmx-cli},benches,examples,tests,docs}

# 2. Inicializar workspace
cargo new --lib hvmx
cd hvmx

# 3. Criar Cargo.toml do workspace
cat > Cargo.toml << 'EOF'
[workspace]
members = [
    "crates/hvmx-core",
    "crates/hvmx-jit",
    "crates/hvmx-memory",
    "crates/hvmx-scheduler",
    "crates/hvmx-cli",
]
resolver = "2"

[workspace.package]
version = "1.0.0"
edition = "2021"
license = "MIT OR Apache-2.0"
EOF

# 4. Inicializar crates
for crate in hvmx-core hvmx-jit hvmx-memory hvmx-scheduler hvmx-cli; do
    cargo init --lib crates/$crate
done

# 5. Setup CI
mkdir -p .github/workflows
# (arquivo CI será criado no próximo passo)

# 6. Começar Sprint 1
# (arquivos serão criados incrementalmente)
```

---

## 📝 Próximos Passos Imediatos

1. **Criar repositório GitHub** → `https://github.com/scoobiii/hvmx`
2. **Implementar Sprint 1 - Semana 1** → `hvmx-core` básico
3. **Setup CI/CD** → GitHub Actions

4. **Primeiro benchmark** → CPU vs Vulkan

---

## 📊 Dashboard de Progresso

| Componente | LOC | Status | Testes | Coverage |
|---|---|---|---|---|
| hvmx-core | 1,200 | 🔴 0% | 0/48 | 0% |
| hvmx-jit | 2,000 | 🔴 0% | 0/50 | 0% |
| hvmx-memory | 800 | 🔴 0% | 0/34 | 0% |
| hvmx-scheduler | 600 | 🔴 0% | 0/23 | 0% |
| hvmx-cli | 400 | 🔴 0% | 0/19 | 0% |
| **TOTAL** | **5,000** | **0%** | **0/174** | **0%** |

🟢 = Completo | 🟡 = Em progresso | 🔴 = Não iniciado

---

**Ready to start Sprint 1? 🚀**