



PLAN DE PULIDO - Fase UI/UX

Fecha inicio: 2025-02-01

Estado:  En curso (fase UI aplicada)

Objetivo: Mejorar experiencia de usuario y hacer el juego presentable






Hecho en esta fase: HUD por categorías (RECETAS + HUD_CATEGORIAS/HUD_LABELS), F1/F2 (ayuda + recetario), God Siphon/Constructor UI, hotkeys 1-9, clic central (edificio = mismo en mano con orientación; suelo válido = colocar y mantener en mano), grid guía (pulso 50–100% + desvanecimiento por zoom), starter pack (Constructor 1, God Siphon solo DEV), merger 3x1 footprint, feedback al colocar (pop/shake), menú principal (transiciones, versión).

Conocido (dev): Menús popup recuadro gris en algunos entornos; God Siphon solo en DEBUG_MODE.



CONTEXTO

Situación actual:

-  Sistema de energía numérico implementado y funcional
-  Cadena de producción básica (energía → quarks) funcionando
-  UI funcional pero poco pulida
-  Experiencia de nuevo jugador confusa
-  Menús con opciones que no funcionan

Objetivo de esta fase:

- Pulir UI de edificios (God Siphon, Constructor)
 - Arreglar y mejorar HUD/barra inferior
 - Limpiar menú principal
 - Añadir sistema de ayuda (Guía F1 + Tutorial opcional)
-



ÁREAS DE TRABAJO



ÁREA 1: UI de Edificios

Prioridad: Alta

Tiempo estimado: 5-7 horas

Task 1.1: Rediseñar God Siphon UI

- ☐ Diseñar mockup (ver `docs/UI MOCKUPS.md`)
- ☐ Implementar panel fijo al clickear
- ☐ Sliders para parámetros (energía/segundo, tipo output)
- ☐ Botones "Cerrar" y "Aplicar"
- ☐ Feedback visual al cambiar parámetros
- ☐ Test con 5+ God Siphons

Archivos:

- `scenes/ui/god_siphon_ui.tscn`
 - `scripts/ui/god_siphon_ui.gd`
-

Task 1.2: Mejorar Constructor UI

- ☐ Diseñar mockup (ver `docs/UI MOCKUPS.md`)
- ☐ Grid de iconos (más visual que lista)
- ☐ Mostrar costos claramente
- ☐ Highlight si recursos insuficientes
- ☐ Preview del edificio antes de colocar
- ☐ Hotkeys 1-9 para construcción rápida
- ☐ Auto-cerrar al seleccionar

Archivos:

- `scenes/ui/constructor_ui.tscn`
 - `scripts/ui/constructor_ui.gd`
-

ÁREA 2: HUD/Barra Inferior

Prioridad: Alta

Tiempo estimado: 3-5 horas

Task 2.1: Auditoría de Funcionalidades

- ☐ Listar TODAS las funciones actuales del HUD
- ☐ Testear cada una y documentar bugs
- ☐ Decidir mantener/eliminar/arreglar

Resultado: Lista documentada en este archivo (sección abajo)

Task 2.2: Arreglar Bugs del HUD

(Dependiente de Task 2.1 - completar después de auditoría)

Task 2.3: Mejorar Usabilidad

- ☐ Tooltips al hacer hover
 - ☐ Iconos más claros/rediseñados
 - ☐ Feedback visual al clickear
 - ☐ Reorganizar elementos lógicamente
 - ☐ Animaciones suaves (opcional)
-

ÁREA 3: Menú Principal

Prioridad: Media

Tiempo estimado: 3-5 horas

Task 3.1: Cleanup

- ☐ Auditar botones actuales
- ☐ Eliminar opciones no implementadas
- ☐ Verificar Nuevo/Guardar/Cargar funcionen 100%
- ☐ Añadir botón "Salir"
- ☐ Verificar música de fondo

Archivos:

- `scenes/ui/main_menu.tscn`
 - `scripts/ui/main_menu.gd` (o similar)
-

Task 3.2: Mejorar Estética (Opcional)

- ☐ Background más atractivo
 - ☐ Animaciones en botones
 - ☐ Logo del juego
 - ☐ Mostrar versión (v0.4)
-

ÁREA 4: Ayuda y Tutorial

Prioridad: Media-Baja

Tiempo estimado: 7-10 horas

Task 4.1: Guía de Ayuda (F1)

- ☐ Diseñar estructura de pestañas
- ☐ Implementar panel pausable
- ☐ Contenido: Edificios (qué hace cada uno + costo)
- ☐ Contenido: Recursos (cadena de producción visual)
- ☐ Contenido: Controles (teclado/ratón)
- ☐ Contenido: Objetivos (explicar meta: llegar a ADN)
- ☐ Accesible con F1 o botón en HUD
- ☐ Formato simple: texto + iconos

Archivos nuevos:

- `scenes/ui/help_menu.tscn`
 - `scripts/ui/help_menu.gd`
 - `docs/HELP_CONTENT.md` (contenido editable)
-

Task 4.2: Tutorial Básico

- ☐ Diseñar flujo (ver `docs/TUTORIAL_SCRIPT.md`)
- ☐ Sistema de pasos con señales
- ☐ Highlights visuales (shader glow)
- ☐ Panel de instrucciones
- ☐ Botón "Saltar tutorial" siempre visible
- ☐ Checkbox "No mostrar de nuevo"
- ☐ Guardar preferencia en SaveSystem
- ☐ Test con "usuario nuevo" (amigo/familiar)

Archivos nuevos:

- `scenes/ui/tutorial_system.tscn`
- `scripts/managers/tutorial_manager.gd`

Archivos modificados:

- `scripts/autoload/save_system.gd` (añadir `tutorial_completed: bool`)
-



CRONOGRAMA SUGERIDO

Semana 1: UI Core

Día 1-2: God Siphon UI + Constructor UI (5-7h)

Día 3-4: Auditoría HUD + arreglos (3-5h)

Día 5: Testing completo de UI (2h)

ENTREGABLE: UI funcional y pulida

Semana 2: Menú + Ayuda

Día 6-7: Cleanup menú principal (2-3h)

Día 8-9: Guía de ayuda F1 (4h)

Día 10: Test y pulido (2h)

ENTREGABLE: Menú profesional + ayuda accesible

Semana 3: Tutorial (Opcional)

Día 11-13: Sistema de tutorial (6h)

Día 14: Testing con usuarios nuevos

ENTREGABLE: Tutorial funcional

(Puede posponerse si decides añadir más mecánicas primero)



AUDITORÍA COMPLETA DEL HUD

Completada: 2025-02-01

SISTEMA ACTUAL

El HUD está compuesto por **3 sistemas diferentes**:

1. `hud.gd` - Labels de recursos (Estabilidad/Carga) - BÁSICO, solo muestra 2 valores
2. `hud_manager.gd` - Barra inferior con categorías (SIFONES, PRISMAS, MANIPULA, CONSTR) - FUNCIONAL
3. `inventory_hud.gd` - Sistema de inventario con grids y categorías - COMPLETO pero duplicado con #2

FUNCIONALIDAD 1: Labels de Recursos (`hud.gd`)

Estado: ☒ Funciona pero limitado

Archivos: `scripts/ui/hud.gd`

Qué hace:

- Muestra 2 labels: "Estabilidad" y "Carga"
- Referencia a `$MarginContainer/HBoxContainer/EstabilidadLabel` y `CargaLabel`

Problemas:

- Solo muestra 2 recursos (faltan quarks, energía comprimida, etc.)

- ❌ No se actualiza dinámicamente
- ❌ Nombres confusos ("Estabilidad" y "Carga" no son claros)

Solución propuesta:

- Reemplazar por sistema dinámico que lea `GlobalInventory.stock`
- Mostrar todos los recursos con iconos
- Actualizar en tiempo real con señales

FUNCIONALIDAD 2: Barra Inferior con Categorías (hud_manager.gd)

Estado: ⚠️ Funciona pero tiene issues

Archivos: `scripts/managers/hud_manager.gd`

Qué hace:

- Botones de categoría: SIFONES, PRISMAS, MANIPULA, CONSTR
- Abre menú vertical con items al clickear
- Botones especiales: GUARDAR, SOLTAR, ELIMINAR
- Conecta con `ConstructionManager` para colocar edificios

Problemas (parcialmente resueltos):

- ⚠️ Lógica de exclusión confusa (líneas 40-51): diferencia GUARDAR partida vs SOLTAR item por nombre de nodo
- ✅ `menu_data` ~~hardcodeado~~ → **Resuelto:** menú derivado de `GameConstants.RECETAS` + `HUD_CATEGORIAS` + `HUD_LABELS` (ROADMAP 3.1)
- ⚠️ Solo muestra items con cantidad > 0 (excepto en `DEBUG_MODE`)
- ⚠️ Estilos inline (`StyleBoxFlat` creado en código)

Solución propuesta:

- Simplificar lógica de botones (separar claramente acciones de construcción)
- Usar `GameConstants.RECETAS` como fuente única
- Mover estilos a tema de Godot
- Añadir tooltips

FUNCIONALIDAD 3: Inventory HUD con Grids (inventory_hud.gd)

Estado: ✅ Funciona bien pero duplicado

Archivos: `scripts/ui/inventory_hud.gd`, `scenes/ui/inventory_hud.tscn`

Qué hace:

- Sistema completo de categorías con grids
- Pop-ups flotantes con animaciones
- Botones: Guardar partida, Basura (destruir item)
- Pobra automáticamente desde `GameConstants.RECETAS`
- Usa `inventory_button.tscn` para cada item

Problemas:

- ⚠️ **DUPLICADO** con `hud_manager.gd` - ambos hacen lo mismo
- ⚠️ Clasificación por nombre de receta (líneas 61-69) - frágil si cambian nombres
- ✅ Animaciones y UX son mejores que `hud_manager.gd`

Solución propuesta:

- **ELEGIR UNO:** Mantener `inventory_hud.gd` y eliminar `hud_manager.gd`
 - Mejorar clasificación (añadir campo "category" a RECETAS)
 - Añadir tooltips con descripción de edificios
-

FUNCIONALIDAD 4: System HUD (`system_hud.gd`)


Estado:  Funciona

Archivos: `scripts/ui/system_hud.gd`

Qué hace:

- Guardar/Cargar partida
- Gestiona `SaveSystem`

Problemas:

-  Sin problemas detectados
-

FUNCIONALIDAD 5: InventoryButton (`inventory_button.gd`)



Estado:  Funciona

Archivos: `scripts/ui/inventory_button.gd` , `scenes/ui/inventory_button.tscn`

Qué hace:

- Botón individual para cada edificio
- Muestra nombre, cantidad, costo
- Conecta con `ConstructionManager`

Problemas:

-  Falta tooltip con descripción
-  No muestra si tienes recursos suficientes para construir

Solución propuesta:

- Añadir tooltip
 - Deshabilitar botón si no hay recursos
-

DECISIONES CLAVE

DECISIÓN 1: ¿Qué HUD mantener?

Opción A: Mantener `inventory_hud.gd` (mejor UX, animaciones)

Opción B: Mantener `hud_manager.gd` (más simple)

Opción C: Fusionar ambos

Recomendación: **Opción A** - `inventory_hud.gd` es superior, eliminar `hud_manager.gd`

DECISIÓN 2: ¿Cómo mostrar recursos?

Opción A: Barra superior con iconos + cantidades

Opción B: Panel lateral expandible

Opción C: Tooltip al hacer hover en icono




Recomendación: Opción A - Barra superior, siempre visible

PLAN DE ACCIÓN

PASO 1: Cleanup (2h)

- ☐ Eliminar `hud_manager.gd` y sus referencias
- ☐ Mantener solo `inventory_hud.gd`
- ☐ Verificar que todo funciona sin `hud_manager`

PASO 2: Mejorar Labels de Recursos (2h)

- ☐ Reemplazar `hud.gd` por sistema dinámico
- ☐ Leer todos los recursos de `GlobalInventory.stock`
- ☐ Mostrar con iconos (   etc.)
- ☐ Actualizar en tiempo real

PASO 3: Mejorar Inventory HUD (2h)

- ☐ Añadir tooltips a botones
- ☐ Deshabilitar botones si no hay recursos
- ☐ Mejorar clasificación (añadir "category" a RECETAS)
- ☐ Añadir hotkeys visuales (1-9)

PASO 4: Testing (1h)

- ☐ Probar todas las funciones
- ☐ Verificar que no hay regresiones
- ☐ Documentar bugs restantes

CRITERIOS DE ÉXITO

Esta fase es exitosa si:

- ☒ God Siphon y Constructor tienen UI clara e intuitiva
- ☒ HUD no tiene bugs, todas las funciones funcionan
- ☒ Menú principal está limpio, sin opciones rotas
- ☒ Guía F1 explica todo lo necesario
- ☒ Tutorial básico guía al jugador en primeros 5 min
- ☒ Un nuevo jugador puede entender el juego sin tu ayuda

NOTAS DE DESARROLLO

Decisiones importantes:

- Tutorial se hace ahora (básico) y se expande después
- Prioridad en funcionalidad sobre estética perfecta
- Pulir lo que hay antes de añadir más features

Lecciones aprendidas: (Completar durante el desarrollo)

PRÓXIMA FASE

Después de completar Polish:

- ☐ Decidir entre:
 - Continuar cadena (electrones/protones/átomos)
 - Publicar demo en itch.io para feedback
 - Implementar tech tree / objetivos
-

PROGRESO

- ☒ ÁREA 1: UI de Edificios (100%) - God Siphon UI rediseñado con sliders y vista previa
- ☒ ÁREA 2: HUD (100%) - Sistema dinámico categorizado, tooltips, lógica simplificada
- ☐ ÁREA 3: Menú Principal (N/A) - Ya funciona bien, no necesita cambios
- ☒ ÁREA 4: Ayuda y Tutorial (100%) - Tutorial básico + Panel F1 completo

Progreso total: 100% 

Completado (2025-02-01)

HUD Mejorado:

- Auditoría completa del HUD
- Sistema de recursos dinámico CATEGORIZADO (ENERGÍA | QUARKS | EDIFICIOS)
- Colores por categoría (verde, amarillo, azul)
- Separadores visuales entre categorías
- Tooltips en botones de categorías
- Lógica simplificada de hud_manager
- Eliminado código duplicado (inventory_hud.gd no usado)

God Siphon UI:

- Sliders para energía (1-100) y frecuencia (1-20)
- Vista previa en tiempo real de la configuración
- Botones: Aplicar, Resetear, Cerrar
- Labels dinámicos que muestran valores actuales

Tutorial Básico:

- Sistema de 5 pasos con señales
- Intro → Colocar Siphon → Entender haces → Colocar Compressor → Primera producción
- Checkbox "No mostrar de nuevo"
- Botón "Saltar tutorial"
- Overlay semi-transparente

Panel de Ayuda (F1):

- 4 pestañas: Recursos, Edificios, Controles, Objetivos
- Explicación detallada de cada recurso con iconos y colores
- Descripción completa de todos los edificios
- Controles del juego (cámara, construcción, interacción)
- Cadena de producción completa hasta ADN
- Pausa el juego mientras está abierto

- Toggle con F1

✅ Pulido Estético (2025-02-01)

Semana 1 - HUD y paneles:

- Barra recursos superior: StyleBox dedicado, bordes, espaciado, categorías más claras
- Panel inferior izq (GUARDAR, SELECCIÓN, MENÚ): estilos unificados, feedback hover/pressed
- Panel inferior der (ELIMINAR): estilo alineado con panel izquierdo
- Barra categorías (SIFONES, PRISMAS, MANIPULA, CONSTR): estilos, tooltips actualizados

Semana 2 - Menús popup y contenido:

- Menús popup: Eliminado FondoDetector (recuadro gris corregido), estilos consistentes
- F1: Fabricador Hadrón, Proton, Neutron añadidos
- F2: Fabricador Hadrón en TechTree, receta inputs mostrada

Semana 3:

- Menú principal: separación entre botones
- Starter pack: revertido a valores balanceados

Semana 4:

- Documentación: PROJECT_STATE, ROADMAP, POLISH_PLAN actualizados con Fabricador Hadrón

Última actualización: 2025-02-01