

Licenciatura en Producción de Simuladores y Videojuegos

“Diseño e implementación de un sistema automatizado de exportación de cartas para videojuegos y su impacto en la eficiencia productiva”

T E S I S

2026

Presenta:

Remeta Agustín Marcelo

Resumen

La presente investigación analiza el impacto de la automatización del *pipeline* de exportación de datos en la eficiencia productiva del desarrollo de videojuegos independientes, centrándose en el título *Deckbuilder* "*The Tavern at the End of the Road*". El estudio identifica una brecha de datos donde los flujos de trabajo manuales y fragmentados (que involucran múltiples conversiones de archivos y modificaciones estructurales manuales en *JSON*) generan dependencia técnica, aumentan el error humano y presentan riesgos de *burnout* en el equipo. Mediante un análisis comparativo de tres estados del sistema (Manual, Intermedio y Automatizado), se evaluó la implementación de una herramienta personalizada utilizando *Google Apps Script* y *Firebase Realtime Database*. Los resultados demuestran una reducción del 94.4% en el tiempo de integración, pasando de 180 segundos a solo 10 segundos por ciclo. Asimismo, la transición hacia una "fuente de verdad" centralizada eliminó los errores de sintaxis y otorgó autonomía e independencia al área de diseño, validando que la inversión en herramientas internas es fundamental para la escalabilidad del proyecto y la productividad del equipo.

UAI
Universidad Abierta
Interamericana

Abstract

This research analyzes the impact of data export pipeline automation on production efficiency within independent video game development, focusing on the Deckbuilder title "The Tavern at the End of the Road." The study identifies a data gap where manual and fragmented workflows—involving multiple file conversions and manual structural modifications in JSON—lead to technical dependency, increased human error, and risks of team burnout. Through a comparative analysis of three system states (Manual, Intermediate, and Automated), the implementation of a custom tool using Google Apps Script and Firebase Realtime Database was evaluated. The results demonstrate a 94.4% reduction in integration time, decreasing from 180 seconds to only 10 seconds per cycle. Furthermore, the transition toward a centralized single source of truth eliminated syntax errors and provided autonomy to the design department, validating that investment in internal tools is fundamental for project scalability and team productivity.

UAI
Universidad Abierta
Interamericana

Índice

Introducción **1**
- Planteamiento del Problema 2
- Justificación y Pregunta de Investigación 2
- Hipótesis 4

Estado de la cuestión **5**
- El Pipeline como Solución General y Estructurada 5
- El Factor Humano: Productividad y Error 5
- La Brecha de Datos en Roles no Técnicos 6
- Usabilidad de Herramientas Internas y "Guardrails" 6

Marco teórico **7**
- Asset Pipeline 7
- Automatización 7
- Gestión de datos en desarrollo de videojuegos 7
- Deckbuilder 8

Estrategia Metodológica **9**
- Diseño del Experimento 9
- Operacionalización de Variables 10
- Procedimiento de Ejecución 11
- Análisis de Resultados 11

Sistemas utilizados **12**
- Sistema manual 12
- Sistema intermedio 16
- Sistema Automatizado 18

Resultados y Métricas **21**

Análisis de Datos **22**
- Tiempo de implementación/ exportación 22
- Cantidad de pasos y herramientas utilizadas 22
- Flexibilidad del diseño 22

Conclusiones **24**
- Validación de la Hipótesis 24
- Impacto del Diseño de Herramientas en la Eficiencia del Equipo de Desarrollo 24
- Análisis de Rentabilidad y Escalabilidad 24
- Reflexión Final 25

Referencias **26**

UAI Universidad Abierta Interamericana

Introducción

El presente trabajo de investigación surge de la experiencia profesional y los desafíos técnicos enfrentados en el marco del desarrollo de videojuegos independientes. Desde mediados de 2024, me desempeñé como socio y desarrollador en Crestfallen Game Studio, un equipo multidisciplinario donde la naturaleza compacta del equipo demanda una participación activa no solo en la programación, mi rol principal, sino también en áreas estratégicas como el marketing y la gestión operativa.

Actualmente, el estudio se encuentra en la producción de The Tavern at the End of the Road, un título del género *Deckbuilder* con un fuerte enfoque narrativo. En este proyecto, la jugabilidad se sustenta en un sistema de combate táctico donde las decisiones del jugador y el progreso de la historia construyen dinámicamente su mazo de cartas.

La elección de este objeto de estudio no es casual; responde a una problemática recurrente observada en el flujo de trabajo diario: la fricción técnica que suele existir entre el área de programación y el resto del equipo creativo. Durante el desarrollo, se identificó una necesidad constante de optimizar las herramientas internas para que el programador actúe como un facilitador, permitiendo que diseñadores y artistas integren contenido de manera autónoma y eficiente. Por lo tanto, esta investigación se propone explorar una solución que disminuya este problema, transformando la asistencia técnica en un sistema de soporte robusto que potencie la agilidad del estudio independiente.

La relevancia del proyecto reside en su capacidad para conectar los conocimientos de la carrera. Encuentra su rigor científico en Metodología de la Investigación, mediante un análisis comparativo y cuantitativo de procesos. Asimismo, se consolida en el Taller de

UAI
Universidad Abierta
Interamericana

Trabajo Final como una respuesta profesional a desafíos reales de la industria. Sus beneficios se ven en diferentes sectores: optimiza la producción de videojuegos, aplica conocimientos de Programación Avanzada y Motores Gráficos II para la creación de herramientas de soporte, y finalmente, otorga autonomía al área de Guión y Narrativa para la iteración de contenido.

Planteamiento del Problema

Uno de los desafíos primordiales identificados durante las etapas tempranas del desarrollo fue la integración de datos en el motor de juego. Si bien el diseño de los atributos de las cartas (efectos, costos y clases) se realizaba de forma externa mediante hojas de cálculo, el traslado de esta información carecía de un flujo de trabajo optimizado. Este escenario derivó en un *pipeline* fragmentado y manual, donde la falta de integración técnica obligaba al equipo a realizar transformaciones de archivos y ajustes estructurales artesanales para asegurar la compatibilidad de los datos. Como consecuencia, el proceso no solo resultaba excesivamente lento y vulnerable al error humano, sino que consolidaba una dependencia técnica absoluta: el diseñador perdía la autonomía para iterar de forma ágil, transformando la ingesta de datos en un cuello de botella que requería la intervención constante del área de programación.

Justificación y Pregunta de Investigación

La necesidad de optimizar los tiempos de producción y fomentar la autonomía de los distintos miembros del equipo motiva el inicio de esta investigación. Es fundamental evaluar si la inversión en el desarrollo de herramientas internas puede mitigar el agotamiento del equipo y acelerar los ciclos de balanceo del juego.

A partir de este escenario, se plantean las siguientes preguntas de investigación:

UAI
Universidad Abierta Interamericana

¿Puede la automatización del flujo de datos entre herramientas de diseño y el motor de juego mejorar la eficiencia y autonomía de las áreas de diseño en el proceso de integración al videojuego?

¿Es realmente necesaria esta mejora?

A lo largo de este documento, se analizarán diferentes estados por los que pasó este sistema, desde lo manual hasta la automatización, para validar si es posible aliviar la carga de trabajo del diseño mediante la creación de herramientas personalizadas.

UAI
Universidad Abierta
Interamericana

Hipótesis

La implementación de un sistema automatizado de integración de datos entre herramientas externas y el motor de juego disminuye la dependencia técnica del programador y reduce el tiempo y la complejidad operativa en la creación e implementación de contenido por parte de las distintas áreas del desarrollo de videojuegos.

UAI
Universidad Abierta
Interamericana

Estado de la cuestión

El *Pipeline* como Solución General y Estructurada

La literatura reciente destaca que el establecimiento de un *pipeline* estructurado es una herramienta metodológica fundamental para asegurar la escalabilidad de un proyecto. Según Grufstedt (2022), la formalización de estos procesos permite que los diseñadores se enfoquen en los objetivos de diseño en lugar de perderse en la resolución de problemas técnicos, mitigando así los obstáculos operativos desde la raíz.

Lear (2021) advierte que existe un compromiso técnico (*trade-off*) importante: la creación de herramientas personalizadas para automatizar estos puentes constituye un "pasivo" técnico. Esto se debe a que su diseño e implementación demandan una alta experiencia en ingeniería de software y un mantenimiento constante frente a las actualizaciones de los motores.

El Factor Humano: Productividad y Error

Un argumento central para la automatización es el factor humano. Como se explora en *Art Is A Mess* (2020), la perfección es inalcanzable debido a factores como el cansancio o las distracciones, y la automatización está directamente vinculada a la felicidad y productividad de los empleados.

Esta visión se ve reforzada por antecedentes de la industria como el caso de Tapps Games (2024), donde se documentó que procesos manuales fragmentados (de hasta 17 pasos) no solo provocan errores frecuentes de sintaxis en archivos como *.json*, sino que generan un riesgo real de *burnout* [1] en el equipo. El estudio demuestra que un error de formato puede elevar el tiempo de integración de 3 minutos a más de 10, validando la necesidad de una

[1] Síndrome psicológico causado por un cúmulo de circunstancias, como el estrés crónico, la sobrecarga profesional, los horarios excesivos, el agotamiento emocional y una insoportable sensación de falta de realización personal.(Real Academia Nacional de Medicina de España, s. f.).

fuente de verdad centralizada que elimine la divergencia entre el diseño y el producto final.

La Brecha de Datos en Roles no Técnicos

La dificultad de integrar contenido no es exclusiva del diseño de juegos comerciales. Marichal Monsonís (2025) identifica una brecha de datos donde la complejidad de las estructuras técnicas (matrices, texto plano) resulta incomprensible para profesionales sin formación en programación.

Aunque el estudio de Marichal Monsonís se sitúa en videojuegos para rehabilitación, su análisis es perfectamente extrapolable al rol del diseñador en un *deckbuilder*. Si el diseñador posee el conocimiento del balance de las cartas pero carece de un sistema que "traduzca" sus hojas de cálculo al entorno 3D, gran parte de la información no llega a aprovecharse eficientemente, limitando la capacidad de iteración y la efectividad de la herramienta.

Usabilidad de Herramientas Internas y "Guardrails"

Finalmente, la efectividad de un sistema automatizado depende de su diseño de experiencia de usuario. Gayatri (2025) sostiene que las herramientas internas deben priorizar la reducción de fricción mediante principios como la divulgación progresiva [2] y la implementación de guardrails o barreras de seguridad.

Estos mecanismos de control, como validaciones en tiempo real y *logs* de actividad, son fundamentales para la autonomía del diseñador. La provisión de retroalimentación inmediata prescinde de la intervención directa del programador para el diagnóstico de fallos en la ingesta de datos. Esto permite evolucionar un *script* elemental hacia un sistema de procesamiento autónomo, garantizando la continuidad operativa y optimizando el uso de los recursos técnicos.

[2] Estrategia de diseño que consiste en mostrar información o funcionalidades solo a medida que el usuario las necesita. (Nielsen, 2006)

UAI Universidad Abierta Interamericana

Marco teórico

Asset *Pipeline*

El asset *pipeline* es una solución reutilizable para el problema crítico de convertir recursos de origen (en formatos propietarios y ricos en información) a formatos de ejecución ligeros y eficientes para el motor (Lear, 2021).

Automatización

La automatización es la aplicación de tecnología, programas, robótica o procesos para lograr resultados con un mínimo aporte humano (IBM, s. f.).

Gestión de datos en desarrollo de videojuegos

Data-driven design

Proceso mediante el cual cualquier decisión relacionada con el *UI* y *UX* de la web o aplicación (es decir: estructura, funcionalidad, etc.) está basada en los datos que se han obtenido mediante herramientas de analítica (Busquets, 2024)

Hojas de cálculo de *Google* (*Google Sheets*)

Hojas de cálculo de *Google* es una app en línea que te permite crear hojas de cálculo, aplicarles formato y trabajar con otras personas en simultáneo.

JSON

JSON es un formato para la especificación de datos usado de manera habitual en las aplicaciones y sitios web. Su principal utilidad se da en el intercambio de datos entre sistemas informáticos. (Zúñiga, 2023)

UAI
Universidad Abierta
Interamericana

CSV

Los archivos *CSV* son un formato simple y ampliamente compatible para almacenar y compartir datos. Son fáciles de usar en varias herramientas de software, lo que los hace ideales para organizar y gestionar información estructurada. Los archivos *CSV* se utilizan principalmente para un intercambio de datos simple en lugar de presentaciones elaboradas. (Adobe, s. f.).

Bases de datos

Una base de datos es un repositorio digital para almacenar, gestionar y proteger colecciones organizadas de datos. (Kosinski, s. f.).

Deckbuilder

Un juego de construcción de mazos o *deckbuilder* es un juego de naipes cuyo sistema se centra en la creación de un conjunto de cartas óptimo para alcanzar las condiciones de victoria. (Kobazu, s. f.).

UAI
Universidad Abierta
Interamericana

Estrategia Metodológica

El presente trabajo se define como una investigación aplicada de carácter comparativo. El experimento consiste en un análisis del flujo de datos (data *pipeline*), tomando como punto de partida la información ya consolidada en una hoja de cálculo y como punto de llegada su integración funcional dentro del motor de juego.

Diseño del Experimento

La unidad de análisis es el proceso de sincronización y exportación. El experimento no evalúa la creación de contenido individual en la hoja de cálculo sino la eficiencia de los mecanismos de transferencia. Se analizarán tres estados del sistema que representan la evolución técnica del proyecto:

1. Sistema Manual (Flujo inicial): Proceso fragmentado que requiere manipulación de archivos intermedios (*.csv, .json*) y edición manual de código o activos.
2. Sistemas Intermedios: Versiones de transición con automatizaciones parciales.
3. Sistema Automatizado (Versión optimizada): Flujo de trabajo directo donde la herramienta de diseño se comunica de forma integrada con el motor.

UAI
Universidad Abierta
Interamericana

Operacionalización de Variables

Para medir el impacto de la automatización, se han definido las siguientes variables:

Tabla 1

Matriz de variables, indicadores y técnicas de recolección de datos.

Variable	Tipo	Definición Operacional	Método de medición
Tipo de sistema de gestión	Independiente	Estado del sistema (Manual, Intermedio, Automatizado)	Categorización por fase de desarrollo.
Tiempo de implementación/ exportación	Dependiente	Tiempo total desde que los datos están listos en el *Excel* hasta que son funcionales en el juego.	Cronometría: Registro del proceso de exportación/ importación completo.
Cantidad de pasos operativos	Dependiente	Número de acciones manuales que el diseñador debe realizar para completar la integración.	Conteo de acciones diferenciando entre exportación, uso de software de terceros y edición manual de código
Herramientas utilizadas	Dependiente	Diversidad de software necesario para completar el flujo.	Mapeo técnico: Registro de herramientas intermedias e intervención técnica manual.
Flexibilidad de diseño	Dependiente	Capacidad de mantener el flujo de trabajo sin requerir cambios en el código.	Registro de Intervención: Necesidad de un programador para "destrabar" la integración.

UAI Universidad Abierta Interamericana

Procedimiento de Ejecución

El experimento se llevará a cabo mediante la ejecución de un ciclo completo de sincronización de la base de datos de cartas en cada uno de los tres sistemas. El procedimiento será el siguiente:

1. Preparación: Se parte de una hoja de cálculo con datos de cartas ya definidos.
2. Ejecución del Flujo: Se inicia el proceso de exportación/importación según las reglas de cada sistema (Manual, Intermedio o Automatizado).
3. Medición de Variables:
    - Se cronometra el tiempo total del flujo de punta a punta.
    - Se enumeran las herramientas utilizadas (ej. conversores web, editores de texto, *Unity*).
    - Se contabilizan los pasos operativos realizados por el usuario.
    - Se identifica si el proceso pudo completarse sin modificar el código o sin la intervención de un programador.

Análisis de Resultados

Los datos obtenidos permitirán comparar la funcionalidad de cada sistema. Se busca demostrar que el sistema optimizado reduce el tiempo de implementación y elimina la necesidad de programas intermedios, logrando que el área de diseño gane autonomía técnica frente a la de programación.

UAI
Universidad Abierta
Interamericana

Sistemas utilizados

Sistema manual

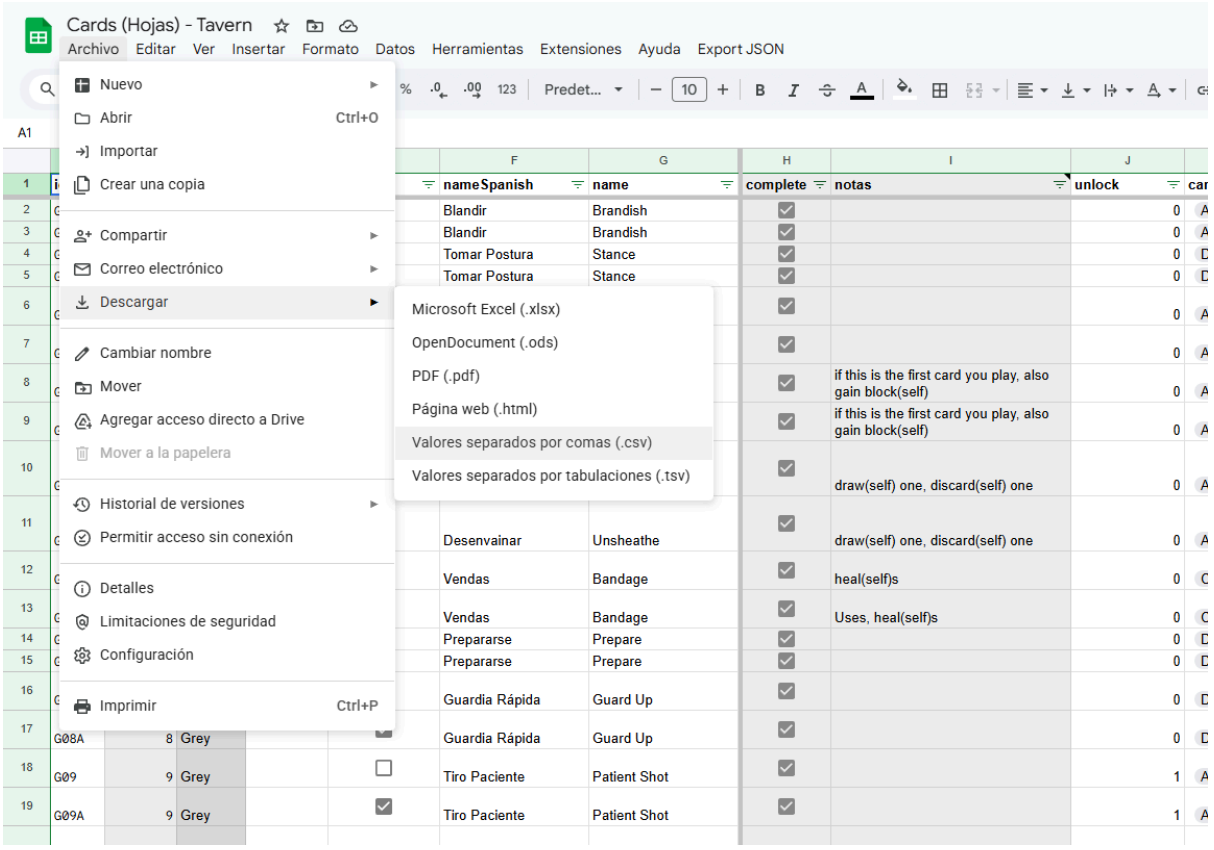
Pasos para exportar

Paso 1: Exportación a Formato *CSV*:

El usuario debe exportar la hoja de cálculo manualmente como un archivo de valores separados por comas (*.csv*).

Figura 1

Exportar como .csv en la hoja de cálculo.



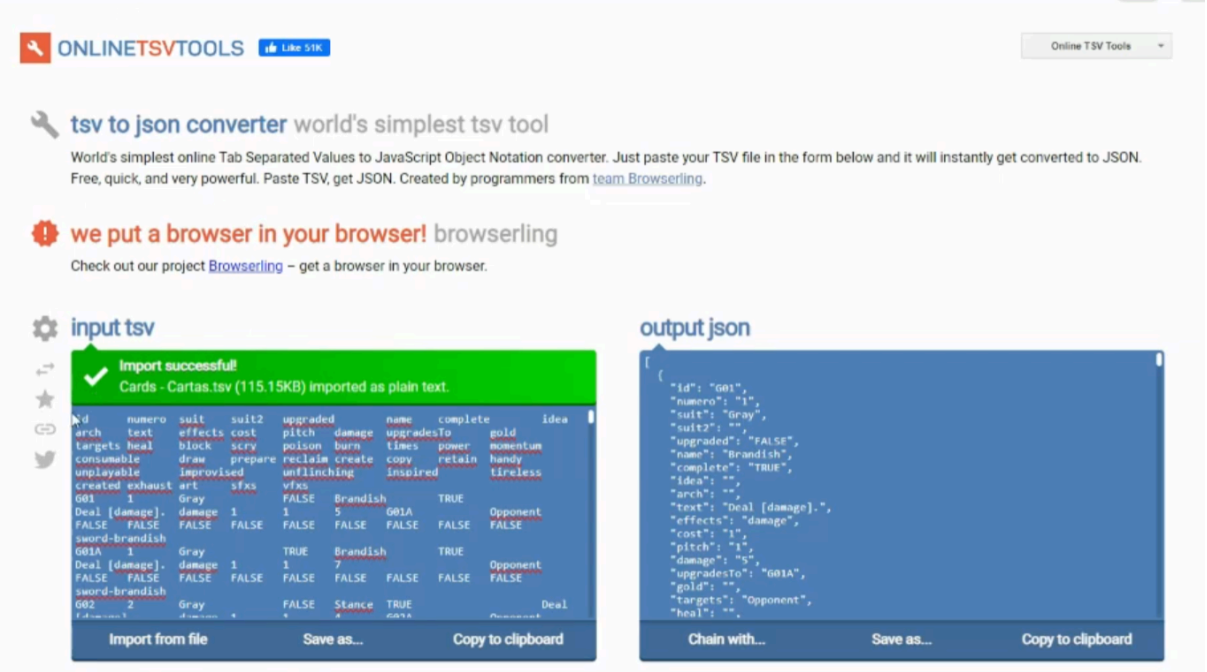
UAI Universidad Abierta Interamericana

Paso 2: Conversión mediante Herramienta Web Externa:

El usuario debe abrir un navegador web para utilizar un conversor de formatos. Se carga el archivo *CSV* para obtener una salida en formato JSON.

Figura 2

Conversor online.



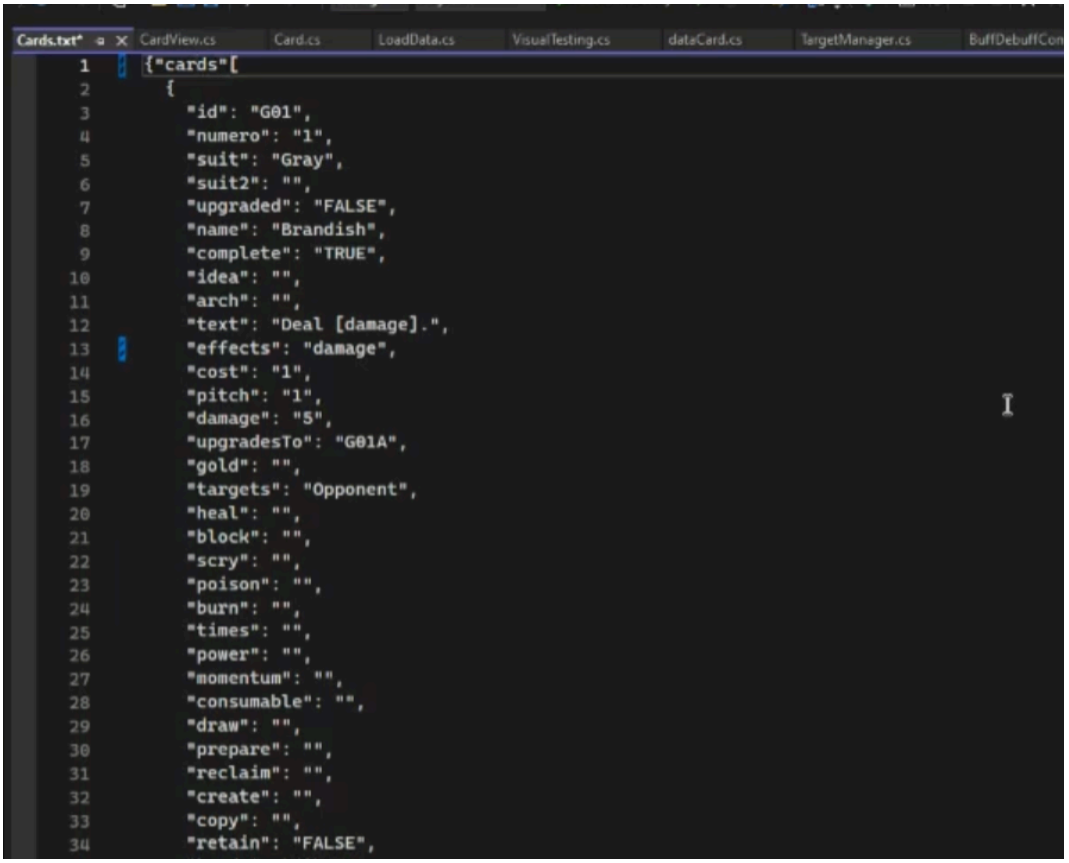
Paso 3: Tratamiento y Modificación Manual del *JSON*:

Una vez obtenido el código *JSON*, se copia y pega en un editor de texto. Aquí ocurre el paso más propenso al error: dado que el sistema de serialización de *Unity* requiere un objeto contenedor para procesar listas, el usuario debía envolver manualmente todo el contenido entre llaves y asignar una etiqueta raíz (ej. {"cards": [...]}).

UAI
Universidad Abierta
Interamericana

Figura 3

Editor de texto.



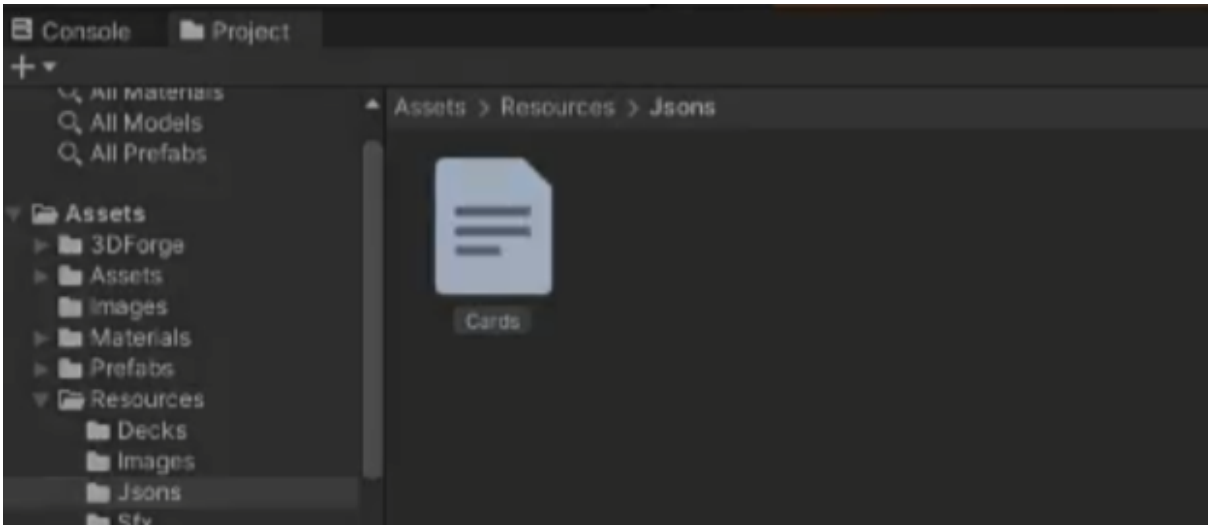
Paso 4: Integración en el Directorio del Motor (*Unity*):

El archivo *JSON* final debe guardarse manualmente dentro de la jerarquía de carpetas del proyecto (Assets/Resources/Cards), sobrescribiendo la versión anterior.

UAI
Universidad Abierta
Interamericana

Figura 4

Jerarquía de carpetas.



Paso 5: Validación Técnica en el Motor:

El diseñador debe volver a *Unity*, esperar a que el motor refresque los archivos y ejecutar el juego para comprobar si el proceso fue exitoso o si la consola arroja errores de sintaxis provocados por la manipulación manual en el paso 4.

Figura 5

Cartas dentro del juego.



UAI
Universidad Abierta
Interamericana

Sistema intermedio

En esta segunda etapa, se desarrolló un sistema integrado directamente en la hoja de cálculo. El objetivo fue transformar el flujo de trabajo fragmentado de la primera etapa en un proceso semi-automatizado, donde la inteligencia del sistema residiera en la propia hoja de cálculo para mitigar los errores humanos derivados de tareas repetitivas.

Implementación mediante *Google Apps Script*

Para lograr esta integración, se utilizó la extensión *Google Apps Script*, una plataforma basada en *JavaScript* que permite extender las funcionalidades de las hojas de cálculo de *Google*. Esta herramienta permitió crear un "puente técnico" que automatiza la extracción de datos sin que el diseñador deba abandonar su entorno de trabajo principal.

Funcionalidades del Sistema

El código desarrollado para esta fase se encargó de realizar las tareas que antes se hacían de forma manual y artesanal:

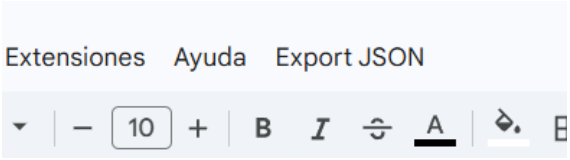
Interfaz de Usuario Personalizada:

El *script* genera un menú propio en la barra superior de la hoja de cálculo llamado "*Export JSON*" (ver Figura 1). Esto reduce drásticamente la cantidad de pasos operativos, ya que el flujo se inicia con un solo clic.

UAI
Universidad Abierta
Interamericana

Figura 6

Menú en hoja de cálculo.



Encapsulamiento y Formateo Estructural:

El código envuelve automáticamente todos los datos en una "clave raíz" y asegura que cada valor tenga el formato correcto (números, textos o booleanos) según lo requiere el motor de juego. Al automatizar este paso, se eliminan los errores de sintaxis comunes en la edición manual de archivos *JSON*.

Resultado de exportación

El resultado de la exportación se presenta directamente en una ventana emergente dentro del *Excel* (ver Figura 2). El diseñador solo debe copiar ese texto y pegarlo en el archivo correspondiente de *Unity*.

Figura 7

Resultado de exportación.



UAI
Universidad Abierta
Interamericana

Pasos para exportar

Paso 1: Tocar botón *Export JSON*

El usuario debe tocar el botón "*Export JSON*" (ver Figura 6).

Paso 2: Copiar y reemplazar el *JSON*:

El usuario debe copiar el resultado dado por el sistema (ver Figura 7). Una vez obtenido el código *JSON* se pega en un editor de texto modificando el *JSON* actual.

Paso 3: Validación Técnica en el Motor:

El diseñador debe volver a *Unity*, esperar a que el motor refresque los archivos y ejecutar el juego para comprobar si el proceso fue exitoso. (ver Figura 5)

Sistema Automatizado

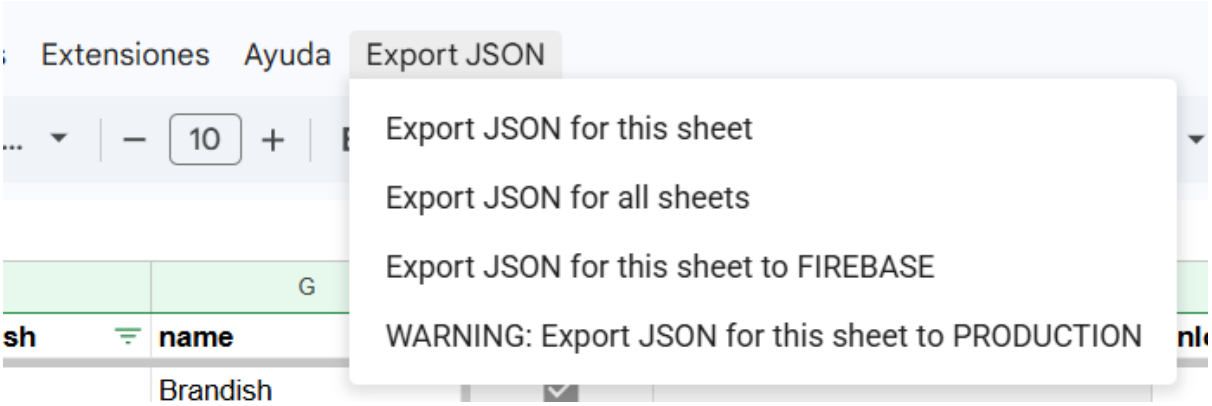
La fase final de la evolución del sistema consistió en la supresión total del intercambio manual de archivos mediante la integración de una base de datos *NoSQL* en la nube (*Firebase Realtime Database*).

A diferencia de las fases anteriores, donde el sistema generaba un archivo de texto que debía ser copiado, en esta etapa se dotó al *script* de *Google* de capacidades de comunicación web, haciendo que el sistema realice una carga directa de los datos procesados hacia el servidor. Complementariamente, se desarrolló en el motor *Unity* un sistema de sincronización automática que, al iniciar la ejecución del videojuego, realiza una petición de lectura para recuperar e integrar los datos actualizados de forma inmediata.

UAI
Universidad Abierta
Interamericana

Figura 8

Menú de hoja de cálculo con Firebase incluido.



Nota. Se agregó adicionalmente la opción de modificar las cartas en una build ya existente (Production).

Figura 9

Código de importación de cartas.

```
using System.Collections;
using UnityEngine;
using UnityEngine.Networking;

public class LoadJsons : MonoBehaviour
{
    static readonly string jsonPath = "assets/resources/jsons";

    public void GetRequestCoroutine()
    {
        StartCoroutine(getRequest("https://tu-proyecto.firebaseio.com/data.json" + LoadData.instance.version + ".json")); //url de ejemplo
    }
    protected IEnumerator getRequest(string uri)
    {

        UnityWebRequest uwr = UnityWebRequest.Get(uri);
        yield return uwr.SendWebRequest();

        if (uwr.isNetworkError)
        {
            Debug.Log("Error While Sending: " + uwr.error);
        }
        else
        {
            Debug.Log("Received: " + uwr.downloadHandler.text);
            System.IO.File.WriteAllText(jsonPath + "/Cards.json", uwr.downloadHandler.text);
            uwr.Dispose();
        }
    }
}
```

UAI
Universidad Abierta
Interamericana

Pasos para exportar

Paso 1: Tocar botón *Export JSON* to FIREBASE

El usuario debe tocar el botón "*Export JSON for this sheet to FIREBASE*" (ver Figura 8).

Paso 2 (opcional): Validación Técnica en el Motor:

El diseñador debe volver a *Unity*, y ejecutar el juego para comprobar si el proceso fue exitoso. Esta etapa se ha vuelto prescindible, dado que la automatización del sistema reduce significativamente el margen de error durante el proceso de exportación. (ver Figura 5).

UAI
Universidad Abierta
Interamericana

Resultados y Métricas

A continuación, se presentan los datos recolectados durante las pruebas de rendimiento de los diferentes sistemas. El siguiente cuadro comparativo detalla los tiempos de ejecución, la cantidad de pasos requeridos, las herramientas utilizadas y el nivel de flexibilidad alcanzado por el equipo de diseño en cada fase del *pipeline*:

Tabla 2

Resultados.

Tipo de sistema de gestión	Tiempo de implementación /exportación	Cantidad de pasos operativos	Herramientas utilizadas	Flexibilidad de diseño
Sistema Manual	180s	5	1. Hojas de cálculo 2. Conversor *Web* Externo 3. Editor de Texto 4. *Unity*	Baja
Sistema Intermedio	70s	3	1. Hojas de cálculo 2. Editor de Texto 3. *Unity*	Media
Sistema Automatizado	10s	1	1. Hojas de cálculo	Alta

UAI Universidad Abierta Interamericana

Análisis de Datos

Tiempo de implementación/exportación

El tiempo de implementación/exportación, pasó de 180 segundos en el sistema manual a solo 10 segundos en el sistema automatizado.

Esta mejora del 94.4% en la velocidad de integración valida la premisa de que la automatización está directamente vinculada a la productividad.

Al reducir el tiempo de una tarea repetitiva, se mitiga el impacto de factores como el cansancio o las distracciones, los cuales suelen derivar en errores humanos durante procesos prolongados.

Cantidad de pasos y herramientas utilizadas

La transición de un flujo de 5 pasos con 4 herramientas distintas hacia un sistema de un solo paso centralizado en la hoja de cálculo marca el éxito de la herramienta interna.

Como sostiene Gayatri (2025), una herramienta interna bien diseñada debe priorizar la reducción de la fricción en la ejecución de tareas críticas.

Al eliminar la necesidad de conversores web y editores de texto externos, el sistema pasa de ser un proceso fragmentado, similar a los casos de *burnout* documentados por Tapps Games (2024), a una "fuente de verdad" centralizada que garantiza la consistencia del contenido.

Flexibilidad del diseño

El salto en la flexibilidad de diseño de "Baja" a "Alta" representa el logro de la independencia del desarrollo dentro del equipo.

UAI
Universidad Abierta
Interamericana

En el sistema manual, la baja flexibilidad obligaba a una dependencia técnica constante del programador para validar la sintaxis del *JSON*.

El sistema automatizado actúa como el puente técnico necesario para cerrar la brecha de datos identificada por Marichal Monsonís (2025), permitiendo que el diseñador tenga autonomía operativa total.

UAI
Universidad Abierta
Interamericana

Conclusiones

Esta investigación permitió validar que la transición de un flujo de trabajo manual a uno automatizado transforma drásticamente la dinámica de producción en el desarrollo de videojuegos. A continuación, se detallan los hallazgos principales:

Validación de la Hipótesis

Se confirma la hipótesis planteada: la implementación de un sistema automatizado disminuye la dependencia técnica del programador y reduce significativamente la complejidad operativa. La integración de herramientas como *Google Apps Script* y *Firebase* permitió que el área de diseño ganara una autonomía total, eliminando la necesidad de que un programador intervenga para validar o corregir la sintaxis de los datos.

Impacto del Diseño de Herramientas en la Eficiencia del Equipo de Desarrollo

El sistema no solo mejoró los tiempos, sino que cerró la brecha de datos entre perfiles técnicos y no técnicos. Al aplicar principios de diseño de herramientas internas, como la reducción de fricción y el uso de menús personalizados, se transformó una tarea técnica tediosa en una acción intuitiva para el diseñador. Esto permite que el equipo se enfoque en el balanceo y la creatividad en lugar de en la resolución de problemas técnicos operativos.

Análisis de Rentabilidad y Escalabilidad

La implementación de sistemas automatizados plantea un balance necesario entre la inversión técnica inicial y el retorno de eficiencia. En el marco de esta investigación, se identificó que la rentabilidad de estas herramientas está intrínsecamente ligada a la escala del proyecto.

UAI
Universidad Abierta
Interamericana

En títulos con un alto volumen de datos o ciclos de balanceo constantes (como un *deckbuilder*), la automatización es altamente rentable, ya que el ahorro del 94.4% en el tiempo de integración compensa rápidamente las horas de desarrollo del sistema.

Para proyectos de corta duración o con una carga mínima de datos, la inversión de tiempo requerida para programar el "puente técnico" puede no estar justificada, ya que el tiempo de desarrollo del sistema superaría al tiempo total que se ahorraría en las exportaciones manuales.

Reflexión Final

El desarrollo de The Tavern at the End of the Road se vio beneficiado por una infraestructura que, aunque representó un desafío técnico inicial, permitió establecer un ritmo de trabajo constante y sistemático. La automatización debe entenderse como una herramienta estratégica: esencial para garantizar la escalabilidad y la salud del equipo en proyectos complejos, pero que requiere una evaluación previa de la relación costo-beneficio según las dimensiones del videojuego a desarrollar.

UAI
Universidad Abierta
Interamericana

Referencias

Adobe. (s. f.). *What is a CSV file?* Recuperado el 1 de marzo de 2026. https://www.adobe.com/acrobat/resources/document-files/what-is-a-csv-file.html

Fernán García de Zúñiga. (2023, 22 de septiembre). Formato *JSON*: Qué es y para qué sirve. Arsys Blog. https://www.arsys.es/blog/formato-json-que-es-y-para-que-sirve

Gayatri. (2025, 16 de abril). *Internal tools UX design best practices. DronaHQ Blog*. https://uxdesign.cc/ux-design-for-internal-tools-3509cbf30fd

Grufstedt, Y., Kauppinen, T., & Kultima, A. (2022). Desarrollo de un proceso de producción de videojuegos para docentes universitarios. En J. Domenech (Ed.), *8th International Conference on Higher Education Advances* (HEAd'22) (pp. 1153–1160). Editorial de la Universitat Politècnica de València. https://acris.aalto.fi/ws/portalfiles/portal/85053384/Developing_a_Game_Production_Pipeline_for_University_Educators.pdf

Marichal Monsonís, D. (2025). Estudio de usabilidad y procesamiento de datos en videojuegos para rehabilitación. https://riull.ull.es/xmlui/bitstream/handle/915/42731/Estudio%20de%20usabilidad%20y%20procesamiento%20de%20datos%20en%20videojuegos%20para%20rehabilitacion.pdf?sequence=1

Kosinski, M. (s. f.). ¿Qué es una base de datos? Recuperado el 1 de marzo de 2026. https://www.ibm.com/es-es/think/topics/database

Microsoft. (s. f.). *Software* de hojas de cálculo de *Microsoft Excel*. Recuperado el 1 de marzo de 2026. https://www.microsoft.com/es-ar/microsoft-365/excel/spreadsheets

UAI Universidad Abierta Interamericana

IBM. (s. f.). *What is automation?* Recuperado el 25 de febrero de 2026. https://www.ibm.com/think/topics/automation

James Lear (2021). *The video game asset pipeline: A pattern approach to visualization. University of the West of England.* https://uwe-repository.worktribe.com/output/6844257/the-video-game-asset-pipeline-a-pattern-approach-to-visualization

Kobazu. (s. f.). Juegos de cartas y *deck builders.* Recuperado el 1 de marzo de 2026, https://kobazu.com/juegos-de-cartas-y-deck-builders/

Nielsen, J. (2006, 4 de junio). *Progressive disclosure. Nielsen Norman Group.* https://www.nngroup.com/articles/progressive-disclosure/

Real Academia Nacional de Medicina de España. (s. f.). Síndrome de desgaste profesional. En Diccionario de términos médicos. Recuperado el 20 de febrero de 2026. https://dtme.ranm.es/terminos/Burnout.html?id=196

Veräjänkorva, A. (2020). *Art is a mess: Developing a content pipeline for a video game [Tesis de maestría, Metropolia University of Applied Sciences]. Theseus.* https://www.theseus.fi/handle/10024/344983

UAI Universidad Abierta Interamericana