CASO DE ESTUDIO 1

874535 Marcos Vivas
874927 Daniel SanAgustin

ÍNDICE

INTRODUCCIÓN

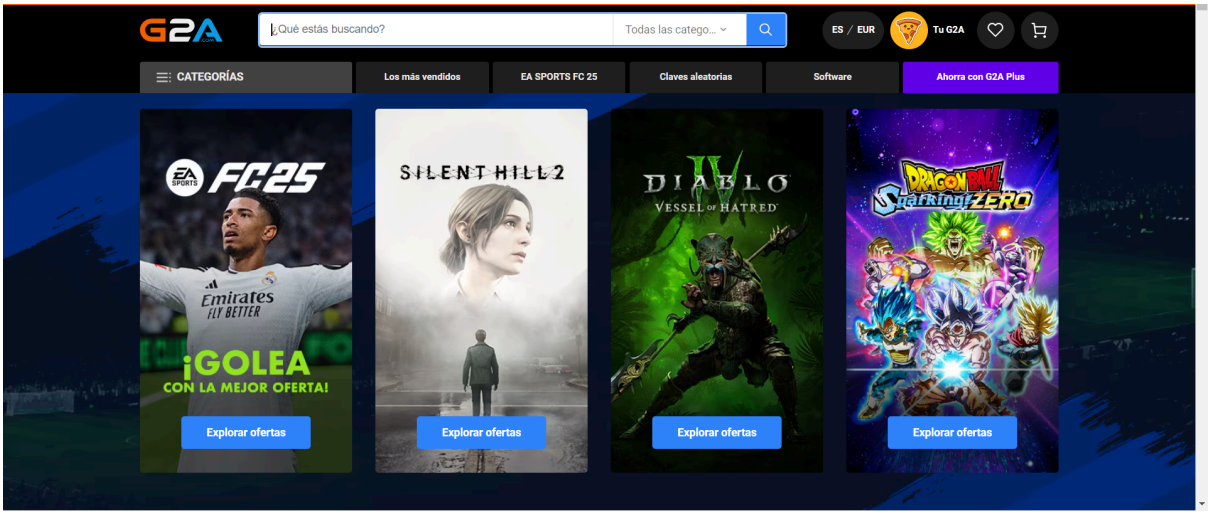
Vamos a continuar con la misma idea de una página de compra online de videojuegos. Como ya vimos en el caso de estudio 0, g2a cumple perfectamente con esa idea, ya que no tiene muchas más funciones que la compra de videojuegos o keys, por lo que la vamos a tomar como ejemplo para desarrollar nuestra página web.

OBJETIVOS

FUNCIONALIDADES:

-Navegación por el catálogo de videojuegos
-Acceder a información y opiniones de un videojuego
-Opción de registro para usuarios no registrados
-Ver nuestro perfil (información personal y/o compras realizadas)
-Sección de “cesta” donde tendremos videojuegos guardados
-Compra de videojuego (solo poder añadirlo a la cesta)

INTERFAZ DE EJEMPLO



MODELO DE NEGOCIO

1. E-commerce tradicional

En este modelo, la tienda es la propietaria de los videojuegos o claves que se venden. El sitio web actúa como un e-commerce tradicional

2. Gestión de catálogo

El sistema de información debe gestionar el inventario de claves de los juegos, permitiendo que el usuario vea en tiempo real la disponibilidad de productos.

3. Procesamiento de pagos y seguridad

Es fundamental contar con un sistema de procesamiento de pagos integrado y seguro que acepte diversas formas de pago (tarjetas, PayPal, criptomonedas, etc.). El sistema de información debe garantizar que las transacciones se realicen de forma segura, protegiendo los datos sensibles de los usuarios mediante técnicas como el cifrado y asegurando la conformidad con normativas de protección de datos, como el GDPR.

4. Entrega automática de productos

Una vez que el pago ha sido confirmado, el sistema de información debe automatizar la entrega de las claves de los videojuegos de manera instantánea.

5. Marketing y promociones

El sistema de información también debe estar vinculado a estrategias de marketing, permitiendo a la tienda crear campañas de descuento o promociones especiales.

6. Atención al cliente y gestión de incidencias

Un componente importante es la atención al cliente y la gestión de incidencias. El sistema de información debe estar preparado para manejar devoluciones o problemas con las claves entregadas, garantizando un servicio post-venta eficiente que incremente la satisfacción y fidelización de los clientes.

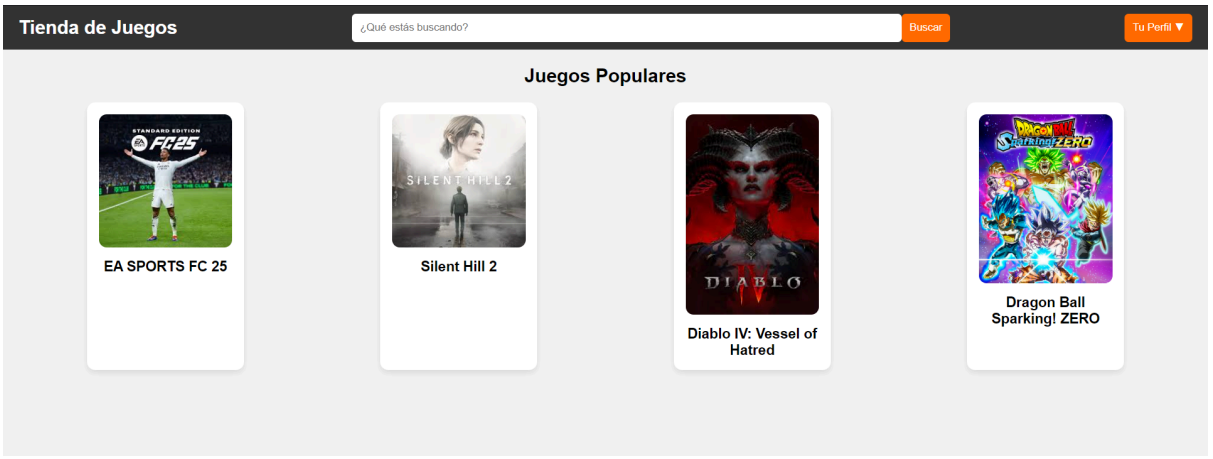
8. Modelo de ingresos

En este modelo, los ingresos provienen directamente de la venta de productos digitales (videojuegos).

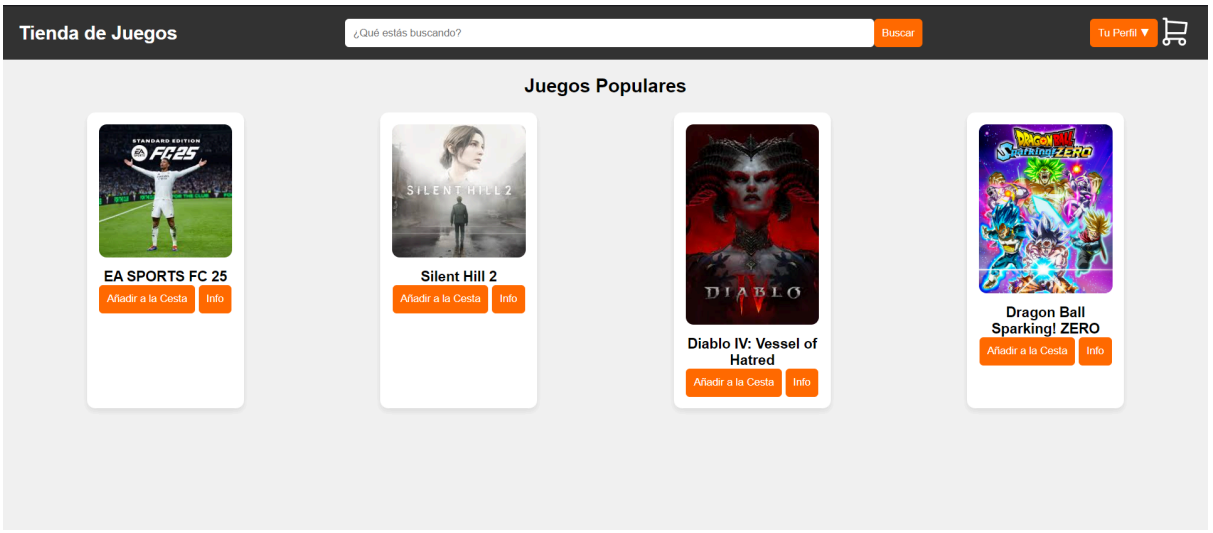
9. Desafíos legales y licencias

Es fundamental que el sistema de información gestione correctamente las licencias de los videojuegos que se venden asegurando que se cumplan las normativas.

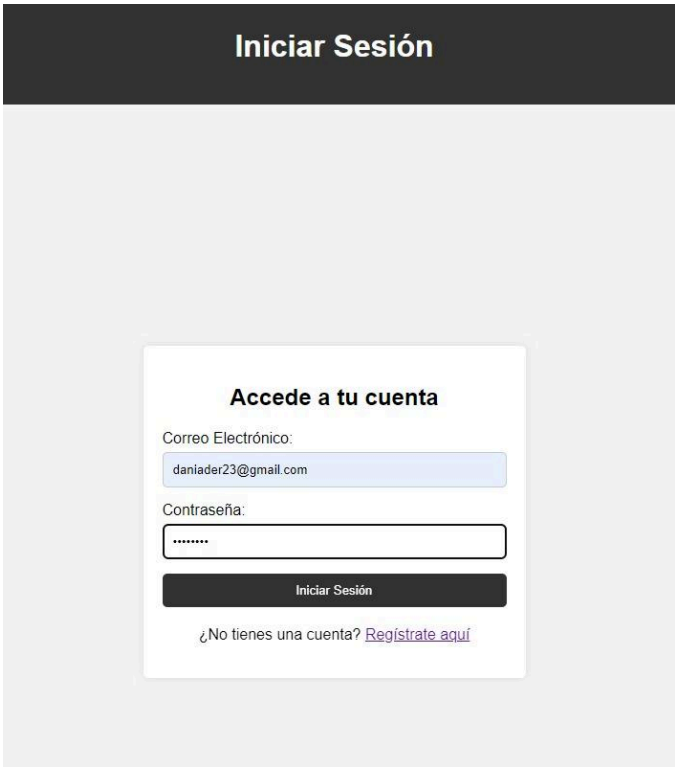
PRIMER PROTOTIPO DE LA VISTA DE LA WEB



PROTOTIPO FINAL PARA ESTA ENTREGA



APARTADO DE REGISTRO EN LA VISTA DE LA WEB



CONEXIÓN CON EL SERVIDOR

app > JS index.js > ...

```
1   import express from "express";
2   import path from "path";
3   import { fileURLToPath } from "url";
4
5   // Fix para __dirname
6   const __dirname = path.dirname(fileURLToPath(import.meta.url));
7
8   // Configuración del servidor
9   const app = express();
10  app.set("port", 4000);
11
12  // Middleware para servir archivos estáticos
13  app.use(express.static(path.join(__dirname, "../public")));
14  app.use(express.static(path.join(__dirname, "../")));
15
16
17  // Rutas
18  //Esto lo podemos quitar ya que especificamos rutas directas a directorios que se pueden omitir con
    el use, estan para recordar por si las necesitamos mas adelante
19  app.get("/", (req, res) => res.sendFile(path.join(__dirname, "pages", "login.html")));
20  app.get("/register", (req, res) => res.sendFile(path.join(__dirname, "pages", "register.html")));
21  app.get("/casoEstudio1", (req, res) => res.sendFile(path.join(__dirname, "../" ,"casoEstudio1.
    html")));
22  app.get(/style.css/, (req, res) => res.sendFile(path.join(__dirname, "public", "style.css")));
23
24  // Iniciar el servidor
25  app.listen(app.get("port"), () => {
26    console.log("Server corriendo en puerto", app.get("port"));
27  });
28
```

METODOLOGÍA DE TRABAJO

Para desarrollar esta práctica, hemos trabajado de manera colaborativa durante una semana, dividiendo las tareas según nuestras habilidades.

- **Herramientas**: Hemos utilizado Visual Studio Code como entorno de desarrollo para escribir el código, y YouTube como fuente de tutoriales y referencias para resolver dudas específicas.

- **Distribución del trabajo**:
    - Marcos se encargó de la creación de la página web utilizando HTML y CSS. Diseñó las secciones principales como la página de productos, el carrito de compras y la página de inicio.
    - Daniel gestionó toda la parte del servidor y cargar la web.

- **Dificultades encontradas**:
    - La principal dificultad fue el poder pre-visualizar la web, hasta que encontramos una extensión de visual studio se nos hacía pesado.
    - Cargar la web en el servidor de primeras nos costó, pero gracias a tutoriales hemos podido hacerlo.
    - Hacer varias pantallas y tener que tener varios “.html” también, porque nunca habíamos trabajado con eso.