

Evidencia de Aprendizaje – Módulos de software codificados y probados

GA7-220501096-AA2-EV02

Fase – 3 Ejecución – Área técnica

Por:

Santiago Arango Rodriguez

Centro de la Tecnología del Diseño y la Productividad Empresarial

Regional Cundinamarca – SENA - Girardot

Análisis y Desarrollo de Software

Ficha 2977481

Instructor: Milton Ivan Barbosa

19 de Septiembre del 2025

Contenido

1. Introducción	3
3. Objetivos.....	4
3.1 Objetivo general	4
3.2 Objetivos específicos.....	4
4. Presentación de la documentación y explicación de la fase actual de desarrollo	5
4.1 Presentación de la documentación.....	7
4.1.2 Diagrama de Clases (Sistema completo)	7
4.1.2 Diagramas de Casos de Uso (Fase I)	8
4.1.3 Historias de Usuario (Fase I)	11
4.1.4 Prototipos de Interfaz (Fase I)	12
4.1.5 Informe Técnico, Plan de Trabajo	17
5. Conclusiones.....	19
6. Referencias.....	19

1. Introducción

Mi proyecto Gaming Network busca ser una solución completa para la administración de una sala de internet gaming. La idea general es que el sistema tenga diferentes módulos que permitan manejar usuarios, controlar las estaciones, gestionar reservas, llevar cobros y facturación, tener una tienda de productos y servicios, generar reportes y hasta contar con un sistema de fidelización.

En este documento estoy recopilando la parte de análisis y diseño que me sirve como base para el desarrollo. Aquí incluyo los diagramas de casos de uso, las historias de usuario, los prototipos de interfaz y el diagrama de clases, que en conjunto muestran cómo está pensado el sistema y cómo va a responder a las necesidades de los usuarios y de los administradores.

2. Justificación

Las salas de internet gaming necesitan una herramienta tecnológica que le permita administrar de forma eficiente sus operaciones diarias, ya que actualmente este tipo de procesos suelen manejarse de manera manual o con soluciones limitadas. Gaming Network surge como respuesta a esta necesidad, ofreciendo un sistema integral que centralice la gestión de usuarios, equipos, reservas, facturación, tienda y reportes en una sola plataforma.

El desarrollo de este software permitirá optimizar recursos, mejorar el control operativo, brindar una mejor experiencia a los clientes y facilitar la toma de decisiones a partir de estadísticas confiables. Además, el proyecto representa una oportunidad académica y profesional para aplicar los conocimientos en análisis, diseño e

implementación de software bajo metodologías ágiles, lo que asegura un aprendizaje práctico alineado con los estándares de la industria.

3. Objetivos

3.1 Objetivo general

- Desarrollar el software Gaming Network como una solución integral para la administración de salas de internet gaming, utilizando tecnologías modernas y la metodología ágil Scrum con el fin de garantizar eficiencia, escalabilidad y facilidad de uso.
-

3.2 Objetivos específicos

1. Diseñar la arquitectura del sistema mediante diagramas, prototipos y documentación técnica que sirvan como base para el desarrollo.
2. Implementar los diferentes módulos del sistema (usuarios, equipos, reservas, facturación, tienda, reportes y fidelización) siguiendo buenas prácticas de programación y arquitectura en capas.
3. Realizar pruebas de calidad y desplegar el sistema en un entorno productivo, entregando también la capacitación necesaria para el uso adecuado de la herramienta.

4. Presentación de la documentación y explicación de la fase actual de desarrollo

El proyecto Gaming Network fue concebido como una plataforma integral diseñada para la gestión de una sala de internet gaming. La visión completa de nuestro sistema incluye múltiples módulos: manejo de usuarios, control de estaciones, reservas, tienda de productos y servicios, reportes de ingresos, programas de fidelización, entre otros.

Actualmente nos encontramos en la Fase I: Implementación del Núcleo Funcional (MVP).

En esta etapa se priorizaron los módulos más esenciales para garantizar una primera versión funcional del sistema:

- Usuarios, para permitir el registro, inicio de sesión y administración de roles.
- Estaciones, para mostrar el estado de cada equipo (disponible, en uso, en mantenimiento).
- Reservas, para que los clientes puedan apartar estaciones en horarios específicos.

Dentro de esta fase ya se alcanzaron los primeros avances significativos: el módulo de Usuarios fue desarrollado en Java, con funcionalidades completas de registro, inicio de sesión y gestión de usuarios con acceso a base de datos. Este progreso asegura la validación inicial del sistema y constituye la base sobre la cual se integrarán los demás módulos.

La elección de estos componentes responde al objetivo de construir un Producto Mínimo Viable (MVP) que permita validar la plataforma en la práctica y sentar las bases para añadir los módulos restantes en fases futuras.

Por este motivo, los diagramas de casos de uso, historias de usuario y prototipos corresponden principalmente a los tres módulos priorizados en esta fase. Asimismo, los diagramas de clases y el diseño general muestran la visión completa del sistema, destacando qué parte corresponde a esta primera implementación.

4.1.2 Diagrama de Clases (Sistema completo)

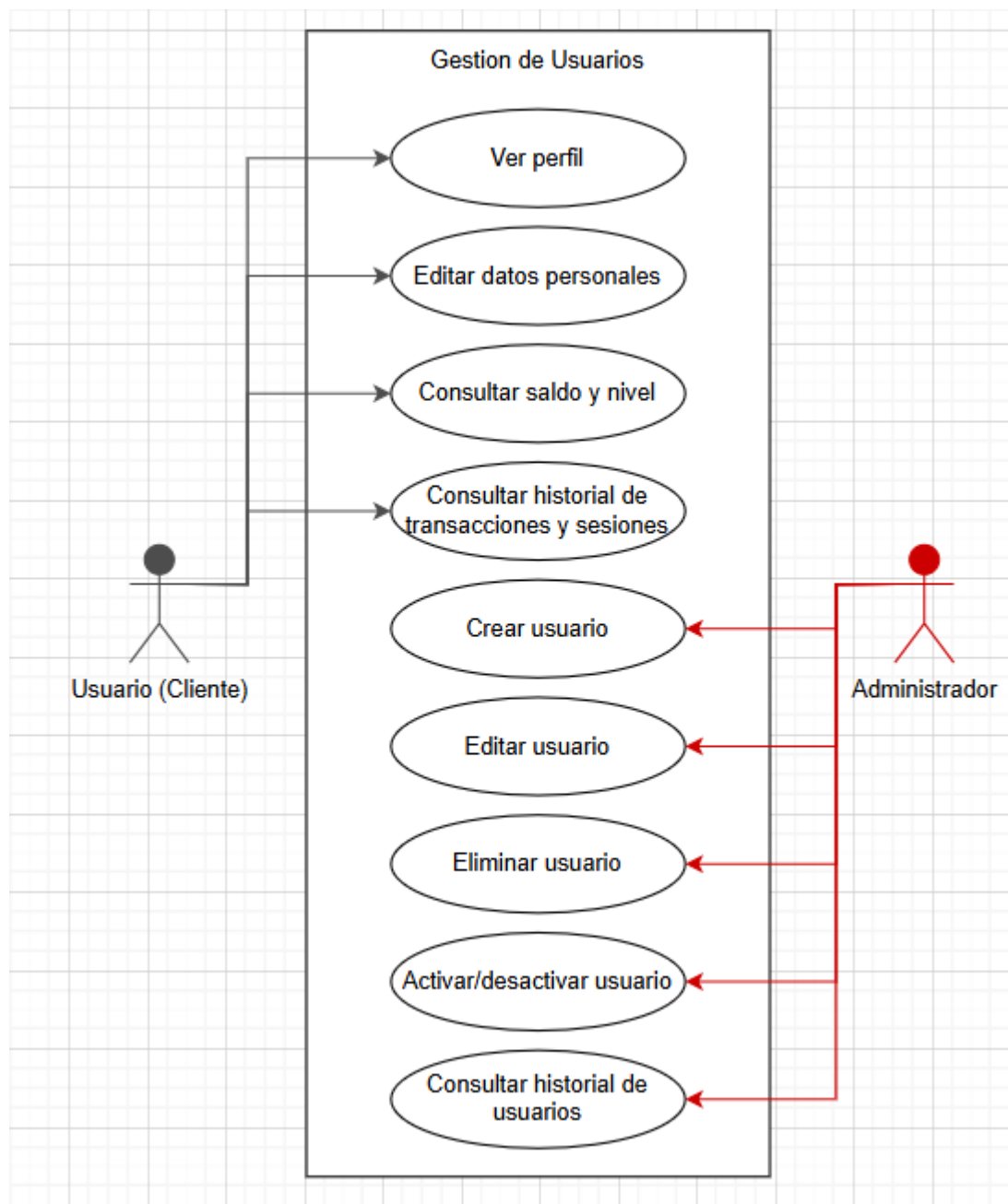
```

classDiagram
    class Sistema {
        +nombre: string
        +version: string
        +monitorearUsuario(): void
        +cerrarSesionRemotamente(usuario: Usuario): void
        +generarReporte(): void
        +ControlarAccesoComputador(usuario: Usuario, computador: Computador): boolean
    }
    class Administrador {
        +agregarUsuario(usuario: Usuario): void
        +editarUsuario(id: int, usuario: Usuario): void
        +eliminarUsuario(id: int): void
        +bloquearUsuario(id: int): void
        +buscarUsuario(nombre: string): Usuario[]
        +definirTarifas(tarifa: Tarifa): void
    }
    class Usuario {
        +id: int
        +nombre: string
        +email: string
        +contraseña: string
        +fechaRegistro: Date
        +estado: EstadoUsuario (enum: ACTIVO, INACTIVO, BLOQUEADO)
        +registrar(): void
        +autenticar(usuario: string, contraseña: string): boolean
        +recuperarContraseña(email: string): void
        +actualizarPerfil(nombre: string, email: string, contraseña: string): void
    }
    class Computador {
        +idComputador: int
        +tipo: string
        +estado: EstadoComputador (enum: DISPONIBLE, EN_USO, RESERVADO)
        +ubicacion: string
        +actualizarEstado(nuevoEstado: EstadoComputador): void
        +mostrarDisponibilidad(): boolean
    }
    class Reserva {
        +idReserva: int
        +usuario: Usuario
        +computador: Computador
        +fechaHora: DateTime
        +estado: EstadoReserva (enum: PENDIENTE, CONFIRMADA, CANCELADA)
        +realizarReserva(usuario: Usuario, computador: Computador, fechaHora: DateTime): void
    }
    class Sesión {
        +idSesión: int
        +usuario: Usuario
        +Computador: Computador
        +horaInicio: DateTime
        +horaFin: DateTime
        +iniciarSesion(usuario: Usuario, computador: Computador): void
        +finalizarSesion(): void
        +calcularTiempoUso(): double
        +mostrarTiempoRestante(): double
    }
    class Tarifa {
        +idTarifa: int
        +tipoComputador: string
        +precioPorHora: double
        +fechaVigencia: Date
        +actualizarPrecio(nuevoPrecio: double): void
    }
    class Cobro {
        +idCobro: int
        +usuario: Usuario
        +sesion: Sesión
        +tarifa: Tarifa
        +monto: double
        +fechaCobro: DateTime
        +calcularCosto(): double
        +registrarPago(monto: double): void
        +generarFactura(): Factura
    }
    class Historial {
        +idHistorial: int
        +sesiones: List<Sesión>
        +transacciones: List<Cobro>
        +registrarsesion(sesion: Sesión): void
        +registrarTransaccion8cobro(cobro: Cobro): void
        +consultaHistorial(): void
        +exportarDatosCSV(): void
    }
    class Factura {
        +idFactura: int
        +usuario: Usuario
        +fechaEmision: DateTime
        +monto: double
        +detalle: string
        +exportarPDF(): void
    }
    Sistema "1" -- "1..*" Administrador
    Sistema "1" -- "1" Computador
    Sistema "1..*" -- "1..*" Reserva
    Administrador "1" -- "1..*" Usuario
    Usuario "1" -- "1..*" Sesión
    Usuario "1" -- "1" Tarifa
    Sesión "1" -- "1..*" Historial
    Sesión "1" -- "1" Cobro
    Sesión "1" -- "1..*" Factura
    Cobro "1" -- "1" Factura
    Cobro "1" -- "1..*" Historial
    Cobro "1" -- "1" Factura
    
```

Nota: Diagrama elaborado por Santiago Arango Rodriguez

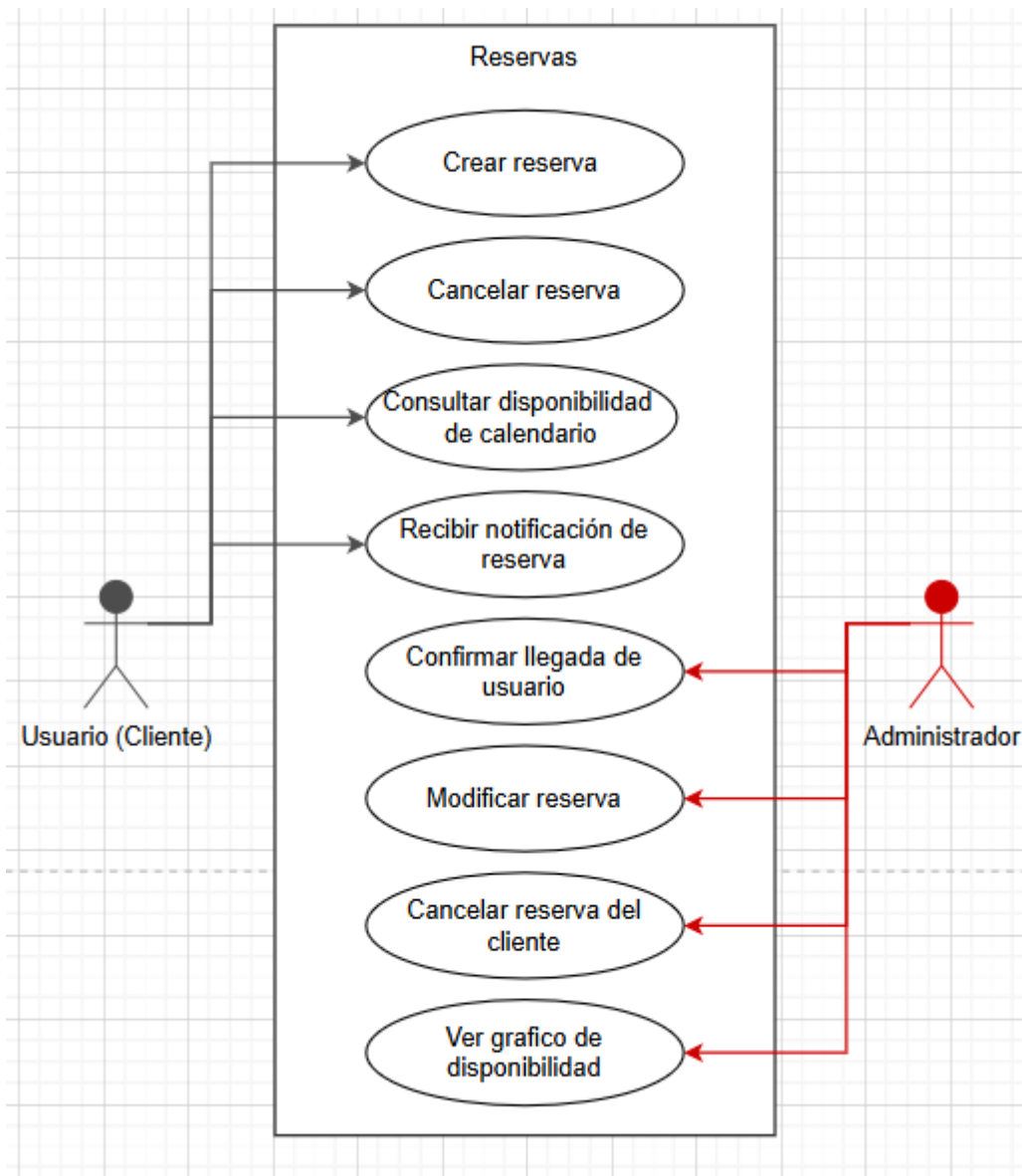
4.1.2 Diagramas de Casos de Uso (Fase I)

Ilustración 2 Diagrama de CU Usuarios



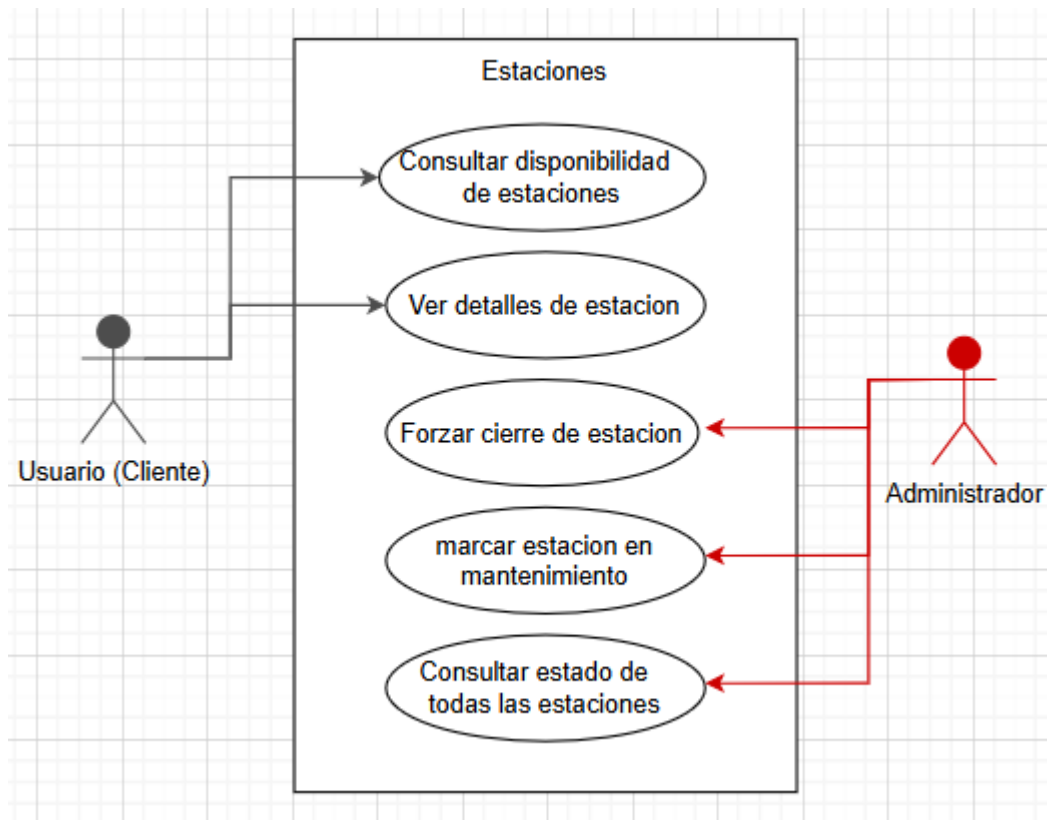
Nota: Diagrama elaborado por Santiago Arango Rodriguez

Ilustración 3 Diagrama de CU Reservas



Nota: Diagrama elaborado por Santiago Arango Rodriguez

Ilustración 4 Diagrama de CU Estaciones



Nota: Diagrama elaborado por Santiago Arango Rodriguez

4.1.3 Historias de Usuario (Fase I)

Tabla 1 Historia de Usuario - Gestion de Usuarios

Gestión de Usuarios	
ID	Historia de Usuario
HU1	Como usuario , quiero ver y actualizar mis datos personales para mantener mi información correcta en el sistema .
HU2	Como usuario , quiero consultar mi saldo y nivel de uso para saber si tengo descuentos o beneficios disponibles .
HU3	Como usuario , quiero acceder a mi historial de transacciones y sesiones para tener control sobre mis gastos y el tiempo de uso .

Tabla 2 Historia de Usuario - Reservas

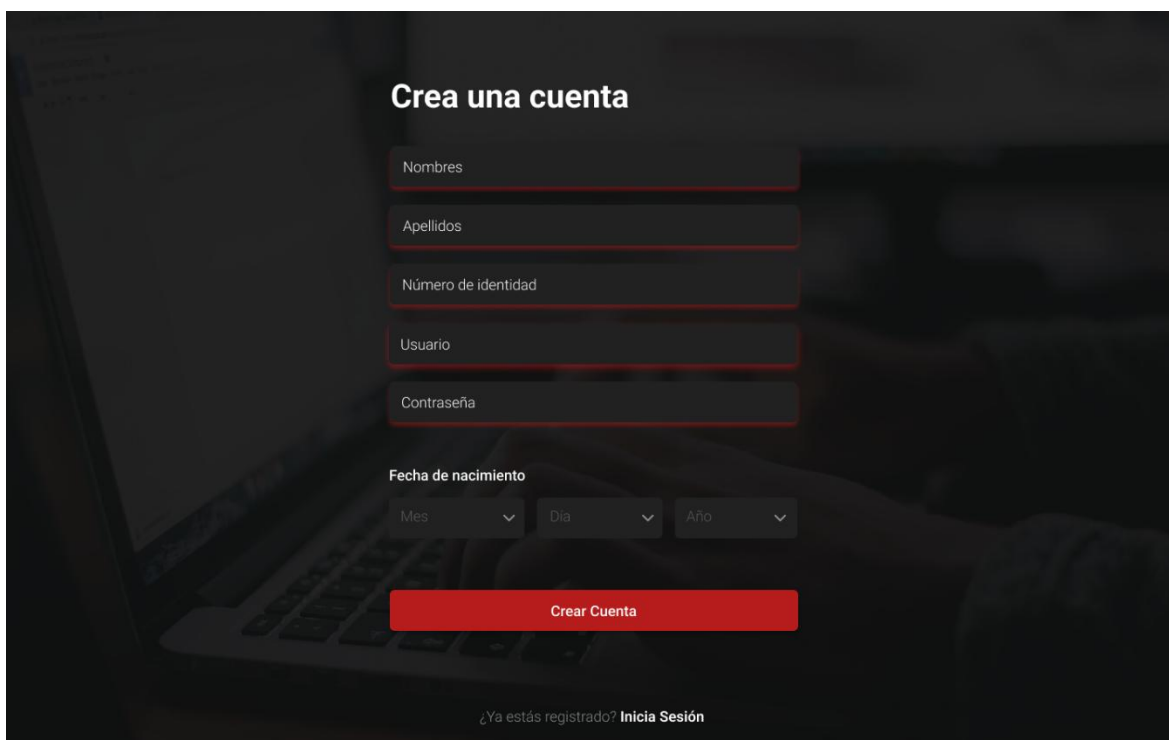
Reservas	
ID	Historia de Usuario
HU4	Como usuario , quiero reservar una estación en una fecha y hora específicas para asegurarme de tener disponibilidad cuando llegue .
HU5	Como usuario , quiero cancelar una reserva para liberar el espacio si no voy a asistir .
HU6	Como usuario , quiero recibir notificaciones previas a mi reserva para no olvidar mi turno y llegar a tiempo .
HU7	Como usuario , quiero consultar la disponibilidad de estaciones en un calendario para planear mejor mi visita .

Tabla 3 Historia de usuario - Estaciones

Estaciones	
ID	Historia de Usuario
HU8	Como usuario , quiero ver qué estaciones están disponibles, ocupadas o en mantenimiento para elegir rápidamente dónde jugar .
HU9	Como usuario , quiero consultar los detalles de una estación (usuario actual, tiempo restante) para decidir si me conviene esperar o reservar otra .

4.1.4 Prototipos de Interfaz (Fase I)

Imagen 1 Prototipo Registro

El prototipo de interfaz de registro se muestra sobre un fondo oscuro con una imagen desenfocada de un teclado y una mano. El título "Crea una cuenta" está en blanco. Los campos de entrada son rectángulos oscuros con bordes rojos: "Nombres", "Apellidos", "Número de identidad", "Usuario" y "Contraseña". La sección "Fecha de nacimiento" incluye tres selectores para "Mes", "Día" y "Año". Un botón rojo "Crear Cuenta" está centrado. En la parte inferior, el texto "¿Ya estás registrado? Inicia Sesión" incluye un enlace en blanco.

Crea una cuenta

Nombres

Apellidos

Número de identidad

Usuario

Contraseña

Fecha de nacimiento

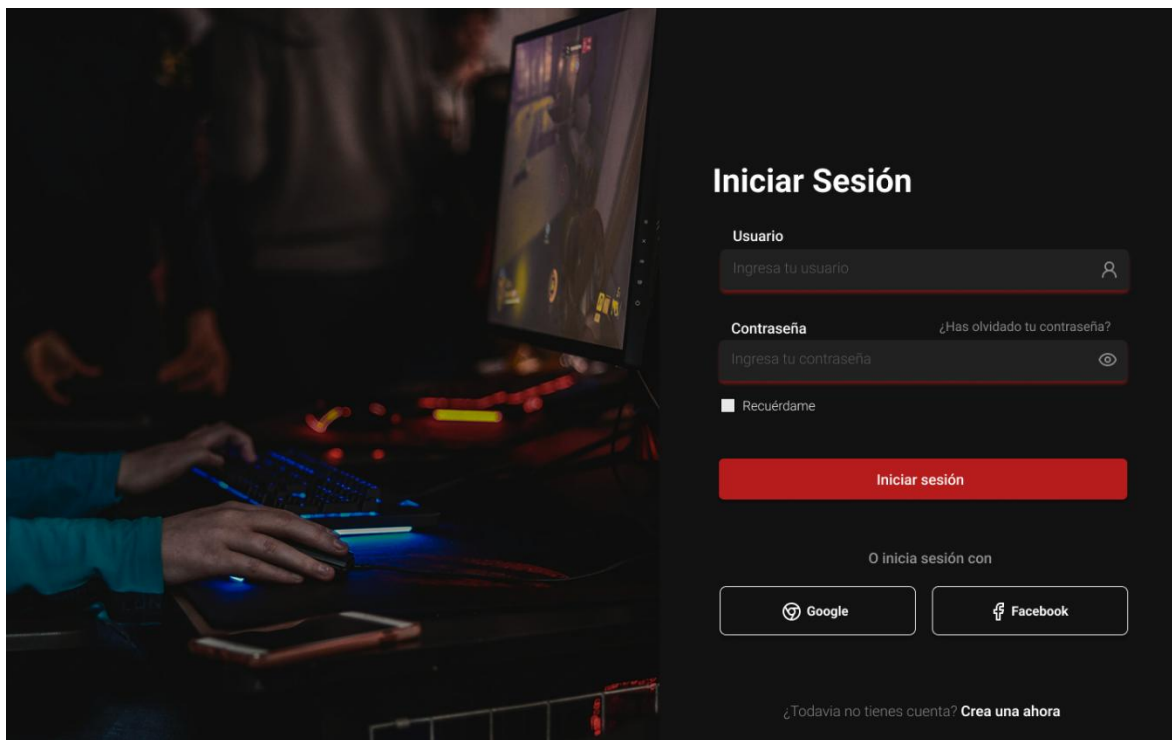
Mes ▼ Día ▼ Año ▼

Crear Cuenta

¿Ya estás registrado? [Inicia Sesión](#)

Nota: Prototipo elaborado por Santiago Arango Rodriguez

Imagen 2 Prototipo Inicio de sesión



The image shows a dark-themed login form overlay on a background image of a person gaming. The form is titled "Iniciar Sesión" and includes fields for "Usuario" and "Contraseña". It also features a "Recuérdame" checkbox, a red "Iniciar sesión" button, and social login options for Google and Facebook. A link to "Crea una cuenta" is at the bottom.

Iniciar Sesión

Usuario

Ingresa tu usuario


Contraseña [¿Has olvidado tu contraseña?](#)


Ingresa tu contraseña

☐ Recuérdame

Iniciar sesión

O inicia sesión con

 Google

 Facebook

[¿Todavía no tienes cuenta? Crea una ahora](#)

Nota: Prototipo elaborado por Santiago Arango Rodriguez

Imagen 3 Prototipo Gestion de usuarios

GAMING
NETWORK

16:25 Sabado, 22 de octubre de 2025

>

Usuarios

Todos los usuariosHistorial de sesionesHistorial de de saldoHistorial de de ticketsRoles y permisos

Todos los usuariosTodos los usuariosTodos los usuariosBuscar por login

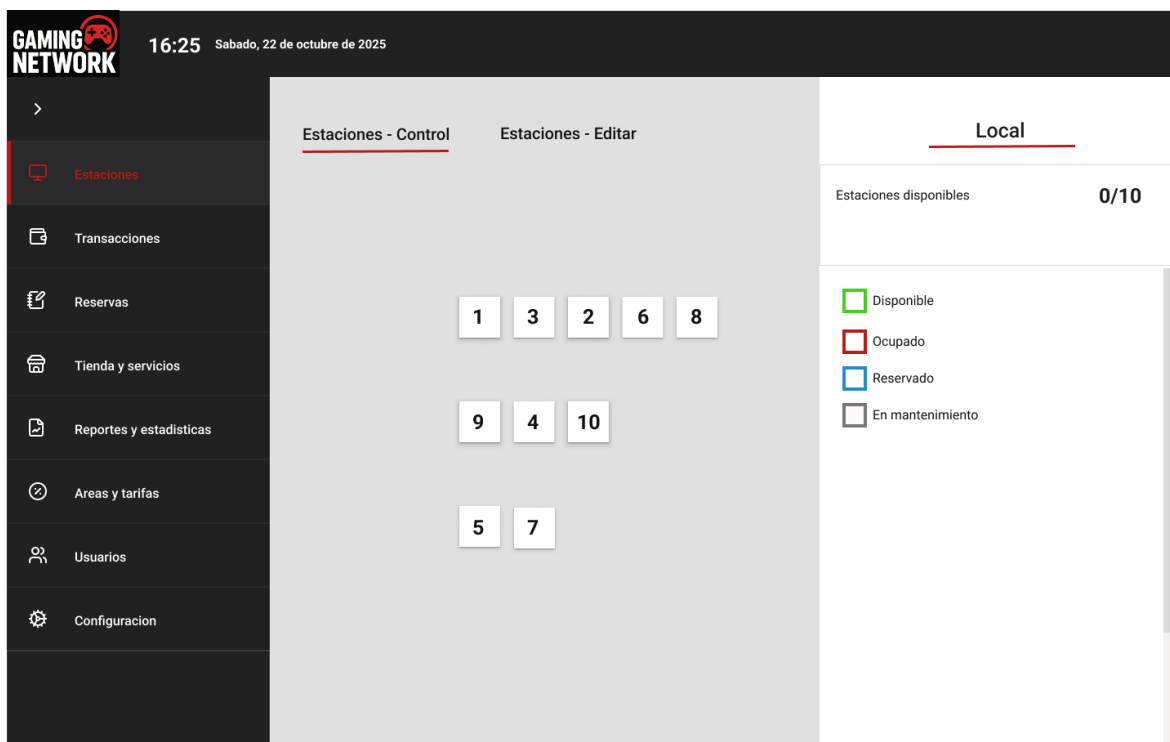
Todo el tiempoAgregar usuario

Login, usuario, tipo...

Usuario	Correo electronico	Tipo	Estado	Nivel (Max 30)	Descuento	Saldo USD	Registro
gamerpro21	gamerpro21mail.com	Estandar	Activo	12 (40%)	10%	\$350	2024-11-10
darklegend88	darklegend88@gmail.com	Estandar	Activo	15 (50%)	25%	\$500	2023-07-04
pixelgirl99	pixelgirl89@gmail.com	Estandar	Activo	8 (26.7%)	0%	\$18	2024-06-18
shadowhunterx	shadowhunterx@gmail.com	Estandar	Activo	6 (20%)	0%	\$120	2025-1-25
netking07	netking07@gmail.com	Estandar	Activo	13 (43.3%)	15%	\$89	2023-12-30
turbomaster	turbomaster@gmail.com	Estandar	Activo	9 (30%)	0%	\$32	2024-03-07
cyberfox27	cyberfox27@gmail.com	Estandar	Activo	5 (16.7%)	0%	\$210	2022-09-12
truckman04	truckman04@gmail.com	Estandar	Activo	10 (33.3%)	0%	\$52	2024-07-28

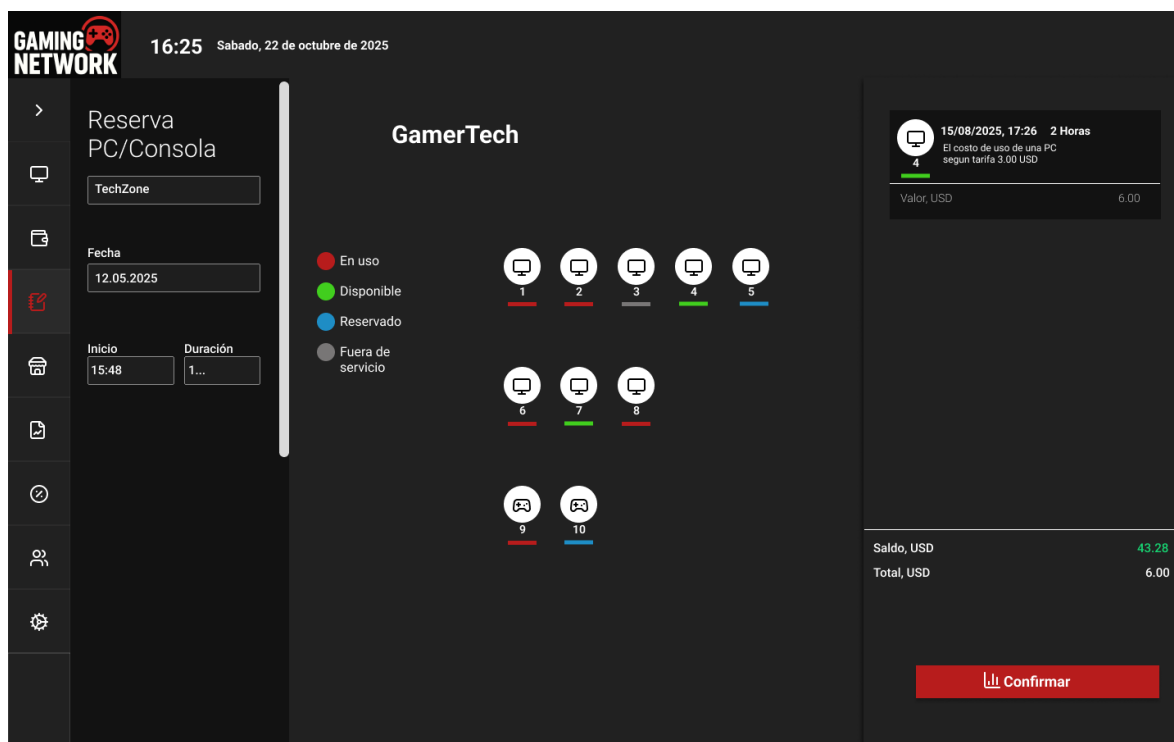
Nota: prototipo elaborado por Santiago Arango Rodriguez

Imagen 4 Prototipo Estaciones



Nota: Prototipo elaborado por Santiago Arango Rodriguez

Imagen 5 Prototipo Reservas



Nota: Prototipo elaborado por Santiago Arango Rodriguez

4.1.5 Informe Técnico, Plan de Trabajo

Para el frontend vamos a trabajar con React.js junto a TailwindCSS, apoyándonos en Figma para los prototipos de diseño. En el backend usaremos Node.js con Express.js, siguiendo el patrón MVC en capas. La base de datos estará en MySQL o PostgreSQL, gestionada con Sequelize.

El control de versiones se llevará en GitHub, mientras que la gestión de tareas y los sprints la haremos en Trello siguiendo Scrum. Para el despliegue se usará un servidor VPS con contenedores Docker, lo que nos da flexibilidad y escalabilidad.

El plan arranco con la planificación del proyecto, donde se definio el alcance, los requerimientos y el backlog. Después pasamos a la fase de diseño: arquitectura del sistema, modelo de base de datos y prototipos de la interfaz.

El desarrollo lo se organizo en sprints de dos semanas. En el primero actualmente estamos construyendo el login, registro y gestión de usuarios. En el segundo trabajaremos la vista de equipos y control de tiempos. En el tercero implementaremos facturación, cobros y reservas. En el cuarto desarrollaremos la tienda, y en el quinto los reportes y estadísticas. El sexto sprint será para la gestión de sitios, tarifas y fidelización. Finalmente, el séptimo servirá para pulir detalles y hacer las pruebas de aceptación.

Cuando el sistema esté listo, realizaremos pruebas unitarias, de integración y de usabilidad para garantizar que todo funcione como debe. Luego vendrá el despliegue en el servidor y la entrega final.

La meta es que el sistema tenga tiempos de respuesta menores a dos segundos, con una disponibilidad mínima del 95 %. La interfaz debe ser intuitiva y validada con

usuarios. El código se escribirá de manera modular y documentada para facilitar su mantenimiento. Además, la solución debe permitir escalar el número de equipos y usuarios sin afectar el rendimiento.

Este plan de trabajo marca la ruta que vamos a seguir para construir Gaming Network. Con la metodología Scrum y las tecnologías seleccionadas, podemos avanzar de manera ordenada y entregar un sistema que realmente responda a las necesidades de la sala de internet. El enfoque ágil nos permitirá ir ajustando en el camino y asegurarnos de que el producto final sea funcional, confiable y fácil de usar.

5. Conclusiones

Al terminar este trabajo siento que logré organizar de una manera clara los avances de mi proyecto. Me doy cuenta de la importancia de ir documentando cada fase, porque así se entiende mejor cómo evoluciona la idea y cómo se va transformando en algo más completo. Los diagramas, las historias de usuario y los prototipos no solo sirven como evidencia de lo que voy construyendo, sino también como una guía para mantener la coherencia en el desarrollo.

Me parece muy útil haber trabajado solo lo correspondiente a esta fase, ya que me permitió enfocarme en lo que realmente estoy avanzando en este momento sin tratar de abarcar más de lo necesario. Además, aprendí que este tipo de documentación no es solo un requisito, sino un apoyo para tener claridad y organización en el proceso. Con esto siento que tengo una base más sólida para seguir avanzando en el proyecto y llegar a un resultado final bien estructurado.

6. Referencias

Redalyc, S. (2013). *Revisión de metodologías ágiles para el desarrollo de software*.

Obtenido de Redalyc: <https://www.redalyc.org/pdf/4962/496250736004.pdf?>

Wikipedia. (s.f.). *Wikipedia*. Obtenido de Producto viable mínimo:

https://es.wikipedia.org/wiki/Producto_viable_m%C3%ADnimo#:~:text=El%20product%20m%C3%ADnimo%20viable%20es,m%C3%ADnimo%20no%20se%20utilizan%20formulaicamente.