FitHub - sistema de gestión de gimnasios: documentación, problemáticas y realización

por Patricio Tubio ([@soypato](http://github.com/soypato)), Ignacio Acuña ([@imonety](http://github.com/imonety)), Enzo Guerci ([@zitowastaken](http://github.com/zitowastaken)) y Juan Manuel Ciardi ([@JuaniiC07](http://github.com/JuaniiC07))



# Narrativa, estructuras, problemas enfrentados y Trello

## Índice

[**Narrativa, estructuras, problemas enfrentados y Trello 1**](#_ktabx2box6ag)

[Índice 1](#_1jtrqsm4xvzy)

[**Qué hace nuestro programa 3**](#_qsldl7gnuuvo)

[Gestión de Clientes: 3](#_oe9ygu8xcept)

[Alta de Clientes: 4](#_rsi7tdpyfljz)

[Modificación de Datos: 4](#_eiydrhjfry92)

[Baja y Restauración de Clientes: 4](#_kxaekkwo7zeb)

[Cambio de Plan: 4](#_csw9p33ucspe)

[Visualización Integral: 4](#_l4chd5d27ijv)

[Seguridad y Privacidad: 4](#_srmywvo39djl)

[Gestión de Empleados 4](#_swgh9icf8rwk)

[Ingreso de Empleados 5](#_ikifcfb7j81)

[Visualización de Empleados 5](#_11z8x56iy1cf)

[Modificación de Datos 5](#_c02j06faaexf)

[Despido y Recontratación 5](#_xkh5gcnyjzao)

[Ordenación Personalizada 5](#_mhnhfl3y21vj)

[Seguridad y Privacidad 5](#_4liqzm75faf2)

[Gestión de pagos 6](#_zg8jl3ubtpml)

[Generación de Facturas: 6](#_spqrnxb5a0dy)

[Visualización de Facturas: 6](#_o04r1pb2u6uh)

[Filtrado por Monto y DNI: 6](#_kkgv8lvr10t3)

[Facturas Dadas de Baja: 6](#_3gqhfr86i7mk)

[Quitar y Reactivar Facturas: 6](#_38kverhg9480)

[Seguridad y Registro: 6](#_21eihs4qiyti)

[Gestión de inventario 7](#_b4cxm8ds014a)

[Cargar Objeto: 7](#_e5fl9k5qzld7)

[Mostrar Inventario 7](#_82ze14e5c79o)

[Desapilar Último Objeto 7](#_yahftf6rm9ow)

[Buscar Objeto 7](#_4z30z4mo2vs4)

[Modificar Objeto 7](#_vajq1c3vt433)

[Mostrar Stock Total 7](#_niqta95zlkos)

[**Estructuras utilizadas 8**](#_a4g9ai70gj64)

[Clientes y Planes de Entrenamiento 8](#_qvme0f8qn59)

[stArchivo: 8](#_9txr0ocihwnk)

[Propósito: 8](#_i37ef7nlj3zz)

[Campos Principales: 8](#_gktlxc9584x2)

[stCeldaPlanes: 8](#_qykug460qxsj)

[Propósito: 8](#_brskvrg8wk6n)

[Campos Principales: 8](#_axg0tlpaqojw)

[nodoArbol: 8](#_7b84wk7ao911)

[Propósito 8](#_uet6lir6lvbk)

[Campos Principales: 8](#_ts184vp8pwc0)

[stCliente: 8](#_ei8gcaowwn65)

[Propósito: 8](#_fb9tccqzypw6)

[Campos Principales: 8](#_rchobaj4y1vc)

[Gestión de Empleados 9](#_q4p1zira1qw8)

[stEmpleado: 9](#_qj8uh6cdkzmw)

[Propósito 9](#_p212ilpjiu1z)

[Campos Principales: 9](#_og9odqi69hts)

[nodoListaEmpleados: 9](#_l2729ffttilc)

[Propósito: 9](#_gfsu8ou9jxgu)

[Campos Principales: 9](#_ze1iv6ezrzct)

[Gestión Financiera 9](#_85xioeco6rd1)

[stPago: 9](#_qihal5lgfv0q)

[Propósito: 9](#_6pu4k8mfu6ns)

[Campos Principales: 9](#_guph92z66od)

[nodoListaPagos: 9](#_2w11h9xn8nqj)

[Propósito: 9](#_mgfw5xe26h5t)

[Campos Principales: 9](#_fpqe6mm6g940)

[Inventario 10](#_5seksbh6tyi)

[stObjeto: 10](#_wo9qbrjfmbih)

[Propósito: 10](#_eg8tbod1cwuc)

[Campos Principales: 10](#_1jb1zuqpuon8)

[nodoListaInventario: 10](#_1rogri8lfd39)

[Propósito: 10](#_gkvxbc9pgqgk)

[Campos Principales: 10](#_l7q55e442u8q)

[pila: 10](#_wzl4mnvdxvq7)

[Propósito: 10](#_4o9mq58a2vg)

[Campos Principales: 10](#_43dwwtkrdsk)

[Resumen 10](#_3bqd7tkqhb4p)

[**Problemáticas y Soluciones 11**](#_msjnqjkt1308)

[Integración de Pagos y Manejo del Tiempo 11](#_u612vxaga6cr)

[Problemas en la Integración de TDAs 11](#_5fzwxs5bvrvc)

[Problema: 11](#_w9vv8dmjwo8d)

[Solución: 11](#_ma3ppb5k5qp1)

[Problemas en la ABML de Clientes en Archivos 11](#_f4djywilxts8)

[Problema: 11](#_pmusxx5xpb20)

[Solución: 11](#_tbjsl0ruha13)

[Integración de Cambios de Planes 11](#_pq9baek1atjp)

[Problema: 11](#_r4nd2l2ipa1e)

[Solución: 12](#_hwa0dhhh4bgn)

[Coherencia en la Integración de Nuevos Clientes 12](#_783f3clje57g)

[Solución: 12](#_zgz5brvctrlv)

[Problemas de Sobreescritura en Archivos 12](#_ph6m999bhozq)

[Problema: 12](#_7dz4m59twbg3)

[Solución: 12](#_em2lnzef318w)

[**Herramientas utilizadas 13**](#_xf08mklzztpi)

[Git 13](#_n8nu69pigruy)

[Trello 13](#_osn4qliia1un)

[Listas Principales 13](#_n1cvdre5k7wd)

[Por Comenzar: 14](#_evnmcdjsaf77)

[En Progreso: 14](#_ivxy1dc607gs)

[Finalizadas: 14](#_l14cijcd7cn6)

[Listas Específicas 14](#_gxdjqm7ex9ov)

[Diario: 14](#_sx3uuchlvnc1)

[Listas de Ideas 14](#_kod5jrlrjbha)

[Ideas Generales: 14](#_cmznwtqz5htu)

[Uso de Etiquetas y Asignación de Tareas 14](#_ynbk8m7usk5k)

# Qué hace nuestro programa

## Gestión de Clientes:

Una de las características destacadas de nuestro sistema es la robusta y eficiente gestión de clientes, diseñada para satisfacer las necesidades específicas de tu gimnasio. Este componente clave del software permite a los usuarios realizar y administrar fácilmente una variedad de operaciones relacionadas con los clientes, brindando una experiencia intuitiva y centrada en el usuario.

#### Alta de Clientes:

* Facilita el ingreso de nuevos clientes al sistema, recopilando información crucial como nombre, apellido, DNI, domicilio, edad, peso, estatura, y más.
* Asigna a cada cliente un plan específico, adaptado a sus metas y preferencias, simplificando así la planificación y seguimiento de sus actividades en el gimnasio.

#### Modificación de Datos:

* Permite realizar ajustes y actualizaciones en la información de los clientes, como cambios en el plan de entrenamiento, días de asistencia, peso, etc.
* Proporciona una interfaz fácil de usar para garantizar la precisión y consistencia de los datos.

#### Baja y Restauración de Clientes:

* Ofrece la capacidad de dar de baja temporal o restaurar clientes según sea necesario.
* Facilita la administración de la membresía de los clientes, brindando flexibilidad a los usuarios y garantizando una base de datos ordenada.

#### Cambio de Plan:

* Permite a los empleados gestionar cambios en los planes de entrenamiento de los clientes.
* Proporciona una interfaz intuitiva para seleccionar y asignar nuevos planes, adaptándose a las necesidades cambiantes de los clientes.

#### Visualización Integral:

* Ofrece una vista completa y detallada de la información de cada cliente, facilitando un seguimiento efectivo de su progreso y estado actual.
* Presenta estadísticas y datos relevantes para respaldar decisiones informadas sobre la planificación y personalización de rutinas de entrenamiento.

#### Seguridad y Privacidad:

* Garantiza la seguridad de la información del cliente mediante un sistema de inicio de sesión con contraseñas exclusivas para cada equipo, preservando la privacidad de los datos.

Con esta sólida gestión de clientes, nuestro sistema proporciona a los usuarios las herramientas necesarias para optimizar la experiencia en el gimnasio y lograr resultados excepcionales

## Gestión de Empleados

Nuestro sistema cuenta con un eficiente módulo de gestión de empleados, diseñado para optimizar la administración del personal en tu gimnasio. Este componente integral del software proporciona las herramientas necesarias para llevar un control preciso de los empleados, garantizando una operación fluida y eficaz.

#### Ingreso de Empleados

* Simplificamos el proceso de ingreso de nuevos empleados al sistema. Recopilamos información esencial como nombre, apellido, legajo, edad, DNI y categoría. Además, ofrecemos la flexibilidad de asignar a cada empleado a una categoría específica, ya sea administrativo, entrenador o recepcionista.

#### Visualización de Empleados

* Proporcionamos diversas opciones para visualizar la información de los empleados, permitiendo acceder a datos ordenados según distintos criterios. Puedes explorar la lista completa de empleados activos, ordenarlos por legajo, DNI o edad, y visualizar aquellos que han sido dados de baja temporalmente.

#### Modificación de Datos

* Facilitamos la actualización de la información de los empleados. Desde cambios en el nombre y apellido hasta ajustes en el legajo, edad, DNI o categoría, nuestro sistema proporciona una interfaz intuitiva para garantizar la precisión y consistencia de los datos.

#### Despido y Recontratación

* Ofrecemos la capacidad de dar de baja temporal a un empleado o de reactivar a aquellos que han sido previamente despedidos. Esta funcionalidad brinda flexibilidad en la gestión de la plantilla, permitiéndote ajustar el personal según las necesidades de tu gimnasio.

#### Ordenación Personalizada

* Nuestro sistema permite ordenar la lista de empleados de acuerdo con criterios específicos, como legajo, DNI o edad. Esta característica facilita la identificación y seguimiento de empleados, optimizando la toma de decisiones relacionadas con la planificación de recursos humanos.

#### Seguridad y Privacidad

* Priorizamos la seguridad de la información del empleado mediante un sistema de inicio de sesión con contraseñas exclusivas. Esto garantiza la confidencialidad y privacidad de los datos, cumpliendo con los estándares más rigurosos en cuanto a protección de información sensible.

Con este robusto módulo de gestión de empleados, nuestro sistema proporciona a los usuarios las herramientas necesarias para administrar eficazmente el personal del gimnasio y garantizar un entorno laboral organizado y eficiente.

## Gestión de pagos

Nuestra aplicación ofrece un sólido sistema de gestión de pagos diseñado para simplificar y optimizar el proceso de manejo financiero en tu gimnasio. Este componente esencial del software proporciona una variedad de funciones para gestionar eficientemente las transacciones y facturación, garantizando una experiencia fluida y precisa.

#### Generación de Facturas:

* Permite la generación fácil y rápida de facturas para los clientes.
* Recopila información clave, como nombre y apellido del cliente, DNI, monto del pago, método de pago, y fecha.
* Facilita la asignación de un identificador único a cada factura para un seguimiento efectivo.

#### Visualización de Facturas:

* Ofrece la posibilidad de mostrar todas las facturas almacenadas en el sistema.
* Proporciona una vista detallada de la información de cada factura, incluyendo nombre del cliente, monto, método de pago y fecha.

#### Filtrado por Monto y DNI:

* Permite la visualización de facturas filtradas por monto, facilitando la identificación de transacciones específicas.
* Ofrece la opción de mostrar facturas filtradas por número de DNI, optimizando la búsqueda de información de clientes.

#### Facturas Dadas de Baja:

* Presenta una sección dedicada a mostrar las facturas que han sido dadas de baja temporalmente.
* Proporciona una visión clara de las transacciones inactivas, ofreciendo flexibilidad en la gestión financiera.

#### Quitar y Reactivar Facturas:

* Permite dar de baja o reactivar una factura según las necesidades del gimnasio.
* Facilita la administración y control de las transacciones, brindando flexibilidad en la gestión financiera.

#### Seguridad y Registro:

* Implementa un sistema seguro de registro de pagos, garantizando la integridad y privacidad de la información financiera.
* Incluye medidas de seguridad, como contraseñas exclusivas para el acceso, para proteger la confidencialidad de los datos.

Con esta potente gestión de pagos, nuestro sistema proporciona a los usuarios las herramientas necesarias para administrar eficazmente las transacciones financieras del gimnasio, garantizando precisión y transparencia en el manejo de las finanzas.

## Gestión de inventario

La gestión de inventario en nuestro sistema es una característica esencial diseñada para proporcionar a los usuarios una herramienta eficiente y organizada para el control de los objetos almacenados en el gimnasio. Este módulo clave del software facilita diversas operaciones relacionadas con el inventario, asegurando un seguimiento detallado y una experiencia intuitiva para los usuarios.

#### Cargar Objeto:

* Simplificamos el proceso de ingreso de nuevos objetos al sistema. Recopilamos información esencial, como el nombre del objeto, peso en kg y cantidad disponible. Esta funcionalidad permite mantener actualizado el inventario de manera rápida y sencilla.

#### Mostrar Inventario

* Proporcionamos opciones para visualizar de manera ordenada y estructurada la lista completa de objetos en el inventario. Puedes explorar y revisar la información según distintos criterios, facilitando la identificación y gestión eficiente de los objetos almacenados.

#### Desapilar Último Objeto

* Ofrecemos la capacidad de retirar el último objeto agregado al inventario. Esta funcionalidad permite corregir posibles errores o ajustar la información del inventario según las necesidades del gimnasio.

#### Buscar Objeto

* Facilitamos la búsqueda de objetos en el inventario mediante la opción de buscar por nombre. Esta funcionalidad agiliza la localización de objetos específicos, proporcionando información detallada sobre su peso y cantidad disponible.

#### Modificar Objeto

* Permitimos realizar ajustes y actualizaciones en la información de los objetos almacenados. Desde cambios en el peso hasta ajustes en la cantidad disponible, nuestro sistema proporciona una interfaz intuitiva para garantizar la precisión y consistencia de los datos en el inventario.

#### Mostrar Stock Total

* Ofrecemos una visualización completa del inventario acumulado y el total de objetos disponibles. Esta característica presenta una tabla detallada con información sobre el nombre del objeto, la cantidad y el acumulado, facilitando la toma de decisiones relacionadas con la planificación de recursos y la gestión del inventario.

#### 

Con este módulo de gestión de inventario, nuestro sistema proporciona a los usuarios las herramientas necesarias para mantener un control eficiente y ordenado de los objetos en el gimnasio, optimizando la experiencia de gestión y asegurando un entorno organizado.

# Estructuras utilizadas

## Clientes y Planes de Entrenamiento

#### stArchivo:

##### Propósito:

* Almacena información detallada de cada cliente, incluyendo datos personales y detalles del plan de entrenamiento.

##### Campos Principales:

* idCliente, nombre, apellido, DNI, edad, domicilio, peso, estatura.
* bajaPasiva, díasConcurridosEstaSemana, idDePlan, plan, díasDelPlan.

#### stCeldaPlanes:

##### Propósito:

* Contiene datos sobre los planes de entrenamiento, con un árbol binario de búsqueda que organiza a los clientes asociados a cada plan.

##### Campos Principales:

* idPlan, plan, díasDelPlan, árbol (árbol binario de clientes asociados).

#### nodoArbol:

##### Propósito

* Nodo en el árbol binario de búsqueda, organizando clientes por número de DNI.

##### Campos Principales:

* cliente (estructura de información del cliente), izq, der (punteros a nodos hijos).

#### stCliente:

##### Propósito:

* Contiene información personal de un cliente.

##### Campos Principales:

* nombre, apellido, domicilio, DNI, edad, peso, estatura.
* Baja Pasiva, días Concurridos Esta Semana.

## Gestión de Empleados

#### stEmpleado:

##### Propósito

* Almacena detalles individuales de cada empleado.

##### Campos Principales:

* nombreYApellido, legajo, edad, DNI, categoría (Administrativo, Entrenador, Recepcionista), bajaPasiva.

#### nodoListaEmpleados:

#### Propósito:

* Nodo en una lista enlazada que organiza empleados de manera eficiente.

#### Campos Principales:

* empleado (estructura de información del empleado), siguiente (puntero al siguiente nodo).

### Gestión Financiera

#### stPago:

##### Propósito:

* Almacena información detallada de transacciones financieras realizadas por clientes.

##### Campos Principales:

* DNICliente, nombreApellido, montoPago, idMetodoDePago, metodoDePago, bajaPasiva, fechaDePago.

#### nodoListaPagos:

#### Propósito:

* Nodo en una lista enlazada de transacciones financieras.

#### Campos Principales:

* pagoCliente (estructura de información de pago), siguiente (puntero al siguiente nodo).

## Inventario

#### stObjeto:

##### Propósito:

* Almacena información detallada de cada objeto en el inventario.

##### Campos Principales:

* nombreObjeto, pesoEnKg, cantidad.

#### nodoListaInventario:

##### Propósito:

* Nodo en una lista enlazada utilizada para implementar una estructura de pila.

##### Campos Principales:

* objeto (estructura de información del objeto), siguiente (puntero al siguiente nodo).

#### pila:

##### Propósito:

* Estructura de pila que gestiona objetos en el inventario.

##### Campos Principales:

* listaDePila (puntero al último nodo agregado a la pila).

## Resumen

Las estructuras de FitHub están diseñadas para trabajar de manera conjunta, proporcionando una base sólida y eficiente para gestionar clientes, empleados, finanzas y el inventario en tu gimnasio. Estas estructuras permiten una gestión intuitiva y centrada en el usuario, mejoran la eficacia operativa y contribuyen al éxito global del negocio. Con enfoque coherente y diseño bien pensado, FitHub ofrece una experiencia fluida tanto para clientes como para el personal, respaldando la eficiencia operativa y el éxito del gimnasio.

# Problemáticas y Soluciones

## Integración de Pagos y Manejo del Tiempo

La implementación de pagos presentó desafíos, especialmente en la integración del componente temporal. Afortunadamente, la reutilización de la librería ducktime.h simplificó la gestión del tiempo, permitiéndonos enfocarnos en el desarrollo de la funcionalidad de pagos.

## Problemas en la Integración de TDAs

En la fase de desarrollo, encontramos dificultades al intentar integrar los Tipos de Datos Abstractos (TDAs), específicamente al trabajar con listas. Inicialmente, decidimos almacenar la información de pagos en una estructura de nodo de lista simple dentro de manejoDePagos.h. Posteriormente, nos dimos cuenta de que las declaraciones de funciones relacionadas con listas deberían estar en su propia librería, listasTDA.h.

#### Problema:

* Al realizar esta separación, surgieron numerosos errores que impedían la compilación del programa.

#### Solución:

* Después de varios intentos, optamos por incluir las declaraciones de las estructuras en listasTDA.h y dejar las funciones relacionadas con pagos en manejoDePagos.h. Este enfoque solucionó los errores y permitió una integración más efectiva de las listas en nuestro sistema.

## Problemas en la ABML de Clientes en Archivos

#### Problema:

* Enfrentamos ineficiencias en la operación de Alta, Baja, Modificación y Listado (ABML) de clientes en archivos. El código existente sobrescribía datos repetidamente, consumiendo recursos de manera innecesaria.

#### Solución:

* Tras identificar el problema, revisamos y reestructuramos nuestro código para mejorar la eficiencia de la ABML en archivos. Implementamos técnicas que evitan la sobreescritura innecesaria de datos, optimizando así el uso de recursos.

Estos ajustes y soluciones contribuyeron al desarrollo exitoso del proyecto y mejoraron significativamente su eficiencia y modularidad.

## Integración de Cambios de Planes

La integración de cambios de planes para clientes presentó desafíos en la manipulación de datos y la actualización correspondiente en el archivo.

#### Problema:

* Al modificar el plan de un cliente, se generaban clientes con datos basura.

#### Solución:

* Revisamos la lógica de modificación de planes y corregimos el código. Al realizar la modificación, garantizamos que los datos de los clientes se actualicen correctamente sin generar información no válida.

## Coherencia en la Integración de Nuevos Clientes

Al agregar un cliente a un nuevo plan, era esencial garantizar que los datos se copiaran correctamente en el archivo.

#### Solución:

* Implementamos la función altaClienteNuevoPlan para manejar la transición de clientes entre planes. Esta función garantiza que los datos del cliente se almacenen adecuadamente en el nuevo plan y se eliminen del plan anterior, manteniendo la coherencia en el archivo.

## Problemas de Sobreescritura en Archivos

La manipulación de archivos presentó retos, especialmente al modificar información existente.

#### Problema:

* Se observó que la sobreescritura de datos en archivos no estaba gestionada adecuadamente, lo que podría llevar a inconsistencias.

#### Solución:

* Implementamos prácticas para gestionar la escritura en archivos de manera segura, evitando la pérdida de datos y asegurando la integridad de la información almacenada.

Estas soluciones abordaron eficazmente los desafíos encontrados durante el desarrollo, mejorando la estabilidad y la funcionalidad del proyecto

# Herramientas utilizadas

## Git

Utilizamos Git y GitHub para ir programando de par a par, e ir creando conflictos para acomodar el programa.

## 

1. **Control de Versiones:** Git proporciona un sistema robusto de control de versiones que permite seguir la evolución del código a lo largo del tiempo. Cada cambio realizado en el código queda registrado, lo que facilita la identificación de modificaciones, la resolución de problemas y el seguimiento del progreso del proyecto.
2. **Colaboración Eficiente:** En un equipo de cuatro personas, la colaboración es esencial. Git permite a cada miembro trabajar en su propia rama (branch) sin interferir con el trabajo de los demás. Luego, estas ramas se pueden fusionar de manera controlada, facilitando la integración de nuevas características y la resolución de posibles conflictos.
3. **Respaldo y Seguridad:** Git actúa como un respaldo eficaz. Al almacenar el código de forma distribuida, incluso si un miembro del equipo pierde localmente su versión del código, siempre hay copias completas en el repositorio central. Esto garantiza la seguridad y la disponibilidad del código en todo momento.
4. **Rastreo de Problemas:** Git, combinado con plataformas como GitHub o GitLab, proporciona herramientas adicionales para el seguimiento de problemas (issues) y la asignación de tareas. Esto facilita la comunicación entre los miembros del equipo, permitiendo un enfoque estructurado para abordar problemas y mejoras.

## Trello

Trello fue una herramienta fundamental en la gestión y organización de tareas durante el desarrollo del proyecto. Facilitó la colaboración, el seguimiento de avances y la coordinación entre los miembros del equipo. A continuación, veremos cómo Trello fue utilizado para organizar las distintas etapas del proyecto:

### Listas Principales

Utilizamos el modelo de Kanban para las listas muy básicas.

#### Por Comenzar:

En esta lista se incluyeron todas las tareas pendientes que aún no se habían iniciado. Esto permitió visualizar de manera clara y ordenada cuáles eran las actividades que debían abordarse próximamente.

#### En Progreso:

Las tareas que se encontraban actualmente en desarrollo fueron movidas a esta lista. Proporcionó una vista instantánea del trabajo en curso y ayudó a evitar la duplicación de esfuerzos dentro del equipo.

#### Finalizadas:

Una vez que una tarea estaba completa, se movía a esta lista. Esto no solo sirvió como un registro de los logros alcanzados, sino que también proporcionó un historial claro de las actividades completadas.

### Listas Específicas

En esta lista se detallaron las estructuras de datos específicas que se planificaban utilizar en el proyecto. Cada estructura se abordó como una tarea independiente, lo que facilitó su implementación y seguimiento.

#### Diario:

Se destinó una lista exclusiva para registrar el diario del equipo. Aquí se documentaron los avances diarios, los problemas encontrados y las soluciones propuestas. Esta lista funcionó como un registro continuo del progreso del proyecto.

### Listas de Ideas

#### Ideas Generales:

En esta lista se recopilaron todas las ideas generales relacionadas con el proyecto. Cada idea se convirtió posteriormente en una tarea específica que podía ser asignada y desarrollada por algún miembro del equipo.

### Uso de Etiquetas y Asignación de Tareas

* **Etiquetas:** Se utilizaron etiquetas para categorizar las tareas según su naturaleza (implementación, corrección de errores, documentación, etc.), lo que facilitó la identificación de las áreas clave del proyecto.
* **Asignación de Tareas:** Cada tarea se asignó a un miembro del equipo responsable. Esto permitió una distribución equitativa de las responsabilidades y una mayor visibilidad sobre quién estaba trabajando en cada aspecto del proyecto.

Trello se convirtió en una herramienta integral para gestionar tareas, comunicarse sobre el progreso y mantener al equipo organizado. Facilitó la colaboración y ayudó a mantener un enfoque claro en los objetivos del proyecto.