



Día 10 de mayo

Objetivo:

Iniciar el proyecto de gestión de alquileres de autos.
Familiarizarse con Git y Visual Studio para la colaboración y desarrollo del proyecto.

Tareas Realizadas:

Reunión Inicial:

Nos reunimos como equipo para discutir el alcance y los requisitos del proyecto.
Establecimos las funcionalidades principales que debe incluir la aplicación, como la gestión de personas, vehículos y alquileres.
Discutimos la estructura del proyecto y las tecnologías a utilizar, así como la unificación de criterios para el código.

Investigación para Metodología de Trabajo:

Investigamos las mejores metodologías para iniciar y llevar a cabo el proyecto en general.
Creación de las primeras clases y métodos básicos.
Investigación sobre herramientas y prácticas recomendadas de Git.
Configuración de Entorno de Desarrollo:
Instalamos Visual Studio en nuestros equipos y configuramos el entorno de desarrollo.

Programación de Reuniones Regulares:

Programamos reuniones periódicas para revisar el progreso y discutir posibles problemas o bloqueos en el desarrollo.

División de Tareas para Archivos Específicos:

Asignamos roles y responsabilidades para el desarrollo del proyecto.

Dividimos las tareas iniciales entre los miembros del equipo:

Desarrollo de las estructuras de datos para personas, vehículos y alquileres.
Desarrollo de funciones relacionadas con el menú y la gestión de cuentas estadísticas de precios y evaluaciones de la administración de la renta.

Tareas Pendientes para el Próximo Día:

Continuar con el desarrollo de las estructuras de datos y funcionalidades básicas.
Explorar características avanzadas de Git, como ramas, pull requests y revisión de código.
Dedicar tiempo a familiarizarse con Git y Visual Studio.
Investigar las mejores prácticas para la gestión de ramas, resolución de conflictos y colaboración en Git.
Crear un repositorio en una plataforma de alojamiento Git (por ejemplo, GitHub, GitLab, Bitbucket) y agregar a todos los miembros del equipo como colaboradores en el repositorio.

Día 12 de mayo

Objetivo:

Ajustar la estructura de datos y aprender a utilizar GitHub para la gestión del proyecto.

Tareas Realizadas:

Revisión de la Estructura de Datos:

Inicialmente utilizamos un arreglo dinámico declarado en el main.
Decidimos utilizar variables globales para facilitar la modularización del código.



Aprendizaje de GitHub:

Nos enfrentamos a problemas al hacer commits y push, ya que a veces sobrescribíamos el trabajo de otros. Realizamos un mini taller interno sobre los comandos básicos de Git y GitHub, como clone, commit, push, pull, y cómo resolver conflictos de merge. Estudiamos la documentación oficial de GitHub y realizamos algunos tutoriales en línea para familiarizarnos con las mejores prácticas.

Tareas Pendientes para el Próximo Día:

Continuar la práctica con GitHub, aplicando lo aprendido.
Implementar las variables globales y ajustar las funciones para utilizarlas.
Seguir avanzando en el desarrollo de las estructuras de datos y funcionalidades básicas del proyecto.

Día 15 de mayo

Objetivo:

Implementar la gestión de hilos y mutex, y mejorar la colaboración en GitHub.

Tareas Realizadas:

Gestión de Hilos y Mutex:

Tuvimos conflictos al intentar utilizar la función gotoxy desde múltiples hilos. Implementamos mutex para asegurar que solo un hilo a la vez acceda a gotoxy. Investigamos sobre sincronización de hilos y utilizamos pthread_mutex_t para proteger la llamada a gotoxy.

Subida y Baja de Versiones en GitHub:

Experimentamos problemas con versiones desactualizadas del código al trabajar simultáneamente en diferentes partes del proyecto. Establecimos una política de hacer pull antes de empezar a trabajar y push después de completar una tarea, asegurándonos de resolver conflictos de inmediato. Discutimos y acordamos un flujo de trabajo más estructurado.

Tareas Pendientes para el Próximo Día:

Continuar trabajando en la sincronización de hilos y pruebas de las funciones implementadas.
Asegurarse de seguir la nueva política de manejo de versiones en GitHub.
Revisar el código y realizar pruebas para identificar posibles errores o mejoras.

Día 18 de mayo

Objetivo:

Integrar los módulos desarrollados y resolver errores de compilación.

Tareas Realizadas:

Integración de Módulos:

La integración de los módulos desarrollados por diferentes miembros del equipo no funcionaba correctamente. Realizamos una sesión de revisión de código en conjunto para entender mejor las interdependencias y ajustamos las interfaces de las funciones. Nos reunimos todos y discutimos el flujo de datos y cómo se interconectaban los módulos, realizando ajustes necesarios en las funciones y sus llamadas.

Errores de Compilación:

Se presentaron múltiples errores de compilación debido a dependencias no satisfechas entre los módulos. Utilizamos un sistema de compilación makefile para gestionar las dependencias y asegurarnos de que todos los módulos se compilen en el orden correcto.



Investigamos sobre la creación de makefiles y realizamos pruebas hasta que logramos un proceso de compilación limpio.

Tareas Pendientes para el Próximo Día:

Continuar con la integración de módulos y pruebas de las funciones implementadas.
Mejorar la documentación del código para facilitar futuras integraciones.
Seguir ajustando el makefile para que incluya todas las dependencias y módulos necesarios.

Día 22 de mayo

Objetivo:

Resolver problemas de fuga de datos y manejar adecuadamente los tipos de datos en las estructuras.

Tareas Realizadas:

Fuga de Datos:

Detectamos problemas de fuga de datos al manejar arreglos dinámicos, especialmente al agregar y eliminar vehículos y usuarios.
Utilizamos llamadas a free para liberar memoria y evitar fugas.
Revisamos el código y añadimos free en las funciones donde se realizaban operaciones con arreglos dinámicos, asegurándonos de liberar memoria correctamente después de cada uso.

Tipos de Datos en Estructuras:

Tuvimos que convertir entre tipos de datos para validar la entrada del usuario, como convertir de int a char y viceversa.
Implementamos funciones de conversión que permitían leer los datos como cadenas de caracteres y luego convertirlos a su tipo original para las estructuras.
Identificamos que la validación directa en tipos numéricos no permitía una entrada flexible, por lo que se decidieron los cambios de tipo temporalmente para facilitar las comprobaciones.

Tareas Pendientes para el Próximo Día:

Continuar con las pruebas para asegurar que no haya más fugas de datos.
Revisar y mejorar las funciones de conversión de tipos de datos.
Seguir avanzando en el desarrollo de funcionalidades básicas del proyecto.

Día 25 de mayo

Objetivo:

Mejorar la responsividad de los menús y resolver conflictos en GitHub.

Tareas Realizadas:

Menús No Responsivos:

Los menús no respondían bien a las entradas del usuario.
Estos conflictos también existían en la captura de datos de los hilos, que respondían a una tecla y realizaban una acción específica.
Rediseñamos la lógica de los menús para mejorar la captura y manejo de las entradas del usuario. Implementamos funciones que limpian el buffer de entrada para asegurar que las teclas sean correctamente interpretadas.
Realizamos pruebas de usabilidad y ajustamos el código de los menús según las observaciones hechas durante estas pruebas, añadiendo también las funciones para limpiar el buffer de entrada.

Inconsistencias en GitHub:

Tuvimos incidentes de eliminación accidental de ramas importantes.
Implementamos una política de revisión y aprobación de pull requests antes de fusionar cambios significativos.



Después de un incidente donde se eliminó una rama crucial, decidimos implementar revisiones de código más estrictas y capacitar al equipo en la restauración de ramas eliminadas.

Problema: Validación de Ingreso de Datos para la Patente:

Se enfrentó dificultades al validar correctamente el ingreso de datos para la patente de los vehículos.

Se decidió modularizar la carga de datos de la patente en funciones individuales para facilitar la gestión y modificación específica de cada dato necesario.

Desarrollo de la función modificar vehículo, desglosada en funciones según el dato a modificar, con opción de mantener los datos previos si el campo se deja vacío.

Tareas Pendientes para el Próximo Día:

Continuar con las pruebas de los menús y ajustar según sea necesario.

Seguir aplicando la política de revisión y aprobación de pull requests en GitHub.

Avanzar en el desarrollo de las funciones asignadas y revisar el código para identificar posibles mejoras.

Día 28 de mayo

Objetivo:

Resolver conflictos de concurrencia y mejorar la gestión de ramas en GitHub.

Tareas Realizadas:

Concurrencia en la Gestión de Alquileres:

Conflictos al intentar alquilar y devolver vehículos simultáneamente.

Utilizamos más mutex para asegurar la consistencia de los datos en la gestión de alquileres.

Analizamos las secciones críticas del código y agregamos mutex en las áreas donde se accedía a los datos compartidos.

Uso Inadecuado de GitHub:

Hubo problemas con la gestión de ramas, algunas ramas quedaron obsoletas o no se fusionaron correctamente.

Implementamos una estrategia de ramas más estricta, utilizando ramas de desarrollo (development), producción (main), y ramas de características (feature).

Al estudiar las mejores prácticas de GitHub, decidimos aplicar un modelo de desarrollo basado en ramas.

Se subió el branch(rama) correspondiente al repositorio de GitHub, asegurando así la integración y colaboración efectiva con el equipo.

Desarrollo de Validaciones para la Carga de Datos del Vehículo

Necesidad de implementar validaciones efectivas para garantizar la correcta carga de datos del vehículo, destacándose la complejidad en la validación de la patente.

Se aplicaron funciones isdigit() e isalpha() para verificar la validez de los caracteres ingresados en la patente.

Los datos de tipo int/float se ingresaron inicialmente como char para realizar la validación adecuada y luego se convirtieron a su tipo original después de la validación.



Tareas Pendientes para el Próximo Día:

Continuar con las pruebas de concurrencia y asegurar que no haya más conflictos.
Seguir aplicando la nueva estrategia de gestión de ramas en GitHub.
Avanzar en el desarrollo de las funciones asignadas y revisar el código para identificar posibles mejoras.

Día 30 de mayo

Objetivo:

Optimizar el rendimiento del programa y mejorar la persistencia de datos.

Tareas Realizadas:

Optimización del Rendimiento:

El programa se ejecutaba lentamente al manejar grandes cantidades de datos.
Mejoramos la eficiencia de los algoritmos de búsqueda y ordenación.
Analizamos los puntos críticos de rendimiento y optimizamos los algoritmos y estructuras de datos.

Persistencia de Datos:

Fallos al guardar y cargar datos desde archivos.
Revisamos y corregimos las funciones de lectura y escritura de archivos.
Realizamos pruebas de persistencia y corregimos los errores en la lógica de manipulación de archivos.

Tareas Pendientes para el Próximo Día:

Continuar con la optimización del rendimiento y realizar pruebas adicionales.
Seguir revisando y mejorando las funciones de persistencia de datos.
Avanzar en el desarrollo de las funciones asignadas y revisar el código para identificar posibles mejoras.

Día 2 de junio

Objetivo:

Mejorar la interfaz de usuario y añadir validaciones de entrada.

Tareas Realizadas:

Interfaz de Usuario:

Rediseñamos la interfaz de usuario para hacerla más intuitiva y fácil de navegar.
Implementamos menús más claros y accesibles, añadiendo descripciones detalladas de cada opción.

Validaciones de Entrada:

Añadimos funciones de verificación de datos para evitar entradas incorrectas por parte de los usuarios.
Implementamos comprobaciones para asegurar que los datos ingresados sean válidos, como formatos de fechas, números de identificación válidos, y restricciones de longitud en los campos de texto.
Realizamos pruebas exhaustivas para asegurarnos de que las validaciones funcionen correctamente.

Tareas Pendientes para el Próximo Día:

Continuar con la mejora de la interfaz de usuario, basándonos en el feedback recibido.
Seguir añadiendo y ajustando las validaciones de entrada según las pruebas y necesidades detectadas.



Día 5 de junio

Objetivo:

Implementar funciones de verificación y comprobación de datos para evitar duplicados y errores.

Tareas Realizadas:

Verificación de Datos de Usuarios:

Implementamos funciones para verificar si un usuario ya existe en la base de datos antes de agregar uno nuevo.

Añadimos una funcionalidad para reactivar usuarios eliminados en lugar de duplicarlos en la base de datos.

Realizamos pruebas para asegurarnos de que estas funciones funcionen correctamente y que los datos de los usuarios se mantengan consistentes.

Gestión de Vehículos:

Implementamos funciones de verificación para evitar la duplicación de vehículos en la base de datos.

Añadimos comprobaciones para asegurarnos de que los vehículos eliminados puedan ser restaurados sin crear duplicados.

Tareas Pendientes para el Próximo Día:

Continuar con las pruebas de las funciones de verificación y ajustes según los resultados.

Seguir mejorando la lógica de gestión de vehículos y usuarios.

Día 8 de junio

Objetivo:

Optimizar el manejo de archivos y mejorar la documentación del código.

Tareas Realizadas:

Manejo de Archivos:

Optimizamos las funciones de lectura y escritura de archivos para mejorar el rendimiento.

Implementamos funciones de backup automático para asegurar la integridad de los datos en caso de fallos.

Revisamos y mejoramos el formato de los archivos de datos para facilitar su manejo.

Documentación del Código:

Añadimos comentarios detallados en el código para mejorar su legibilidad y facilitar el mantenimiento.

Creamos una documentación técnica del proyecto, describiendo las principales funciones y estructuras de datos utilizadas.

Realizamos una revisión de la documentación con todo el equipo para asegurarnos de que sea clara y completa.

Tareas Pendientes para el Próximo Día:

Continuar optimizando el manejo de archivos y realizar pruebas adicionales.

Seguir mejorando la documentación del código y actualizarla conforme se realicen cambios en el proyecto.



Día 11 de junio

Objetivo:

Implementar funciones avanzadas de estadísticas y reportes.

Tareas Realizadas:

Estadísticas de Alquileres:

Desarrollamos funciones para generar estadísticas sobre los alquileres, como vehículos más alquilados, ingresos generados, y tiempos promedio de alquiler. Implementamos gráficos simples en la interfaz de usuario para visualizar estas estadísticas.

Reportes:

Añadimos la funcionalidad de generar reportes detallados de usuarios, vehículos y alquileres.

Implementamos opciones para exportar estos reportes en formatos comunes, como PDF y CSV.

Realizamos pruebas para asegurar que los reportes se generen correctamente y que los datos sean precisos.

Tareas Pendientes para el Próximo Día:

Continuar desarrollando y mejorando las funciones de estadísticas y reportes.

Realizar pruebas adicionales y ajustar la funcionalidad según sea necesario.

Día 14 de junio

Objetivo:

Mejorar la integridad de los datos del sistema.

Tareas Realizadas:

Verificación de Datos:

Implementamos funciones de autenticación y autorización para controlar el acceso a diferentes partes del sistema.

Añadimos cifrado a los archivos de datos para proteger la información sensible.

Realizamos pruebas de seguridad para identificar y corregir posibles vulnerabilidades.

Integridad de Datos:

Añadimos funciones de verificación de integridad para asegurarnos de que los datos no se corrompan.

Implementamos comprobaciones para detectar y corregir inconsistencias en la base de datos.

Realizamos pruebas exhaustivas para asegurar que los datos se mantengan consistentes y seguros.

Tareas Pendientes para el Próximo Día:

Continuar mejorando las medidas de comprobación y realizar pruebas adicionales.

Seguir trabajando en las funciones de integridad de datos y ajustarlas según los resultados de las pruebas.



Día 17 de junio

Objetivo:

Finalizar las pruebas y preparar el proyecto para su lanzamiento.

Tareas Realizadas:

Pruebas Finales:

Realizamos pruebas exhaustivas de todas las funcionalidades del sistema para asegurar su correcto funcionamiento.

Detectamos y corregimos varios errores menores en la lógica de negocio y la interfaz de usuario.

Hicimos una revisión final de la documentación técnica y de usuario para asegurar que esté completa y actualizada.

Preparación para el Lanzamiento:

Preparar un plan de despliegue detallado, incluyendo copias de seguridad y procedimientos de recuperación.

Configuramos el entorno de producción y realizamos pruebas de despliegue para asegurar una transición suave.

Coordinamos una sesión de capacitación para los usuarios finales para familiarizarlos con el sistema.

Tareas Pendientes:

Realizar un último conjunto de pruebas de aceptación con usuarios finales.

Ejecutar el plan de despliegue y monitorear el sistema para asegurar su correcto funcionamiento después del lanzamiento.

Continuar proporcionando soporte y mantenimiento según sea necesario.

Conclusiones Finales

El proyecto de gestión de alquileres de vehículos ha avanzado significativamente desde su inicio el 10 de mayo. A través de una serie de reuniones, discusiones, y colaboración efectiva, hemos logrado implementar una amplia gama de funcionalidades, desde la gestión de usuarios y vehículos hasta la implementación de medidas de validación y la generación de reportes avanzados. A lo largo del proceso, hemos aprendido y mejorado nuestras habilidades en Git y Visual Studio, superando diversos desafíos técnicos y organizativos. Ahora, nos preparamos para el lanzamiento final del sistema, seguros de que proporcionará un servicio robusto y eficiente para la gestión de alquileres de vehículos.