|  |  |
| --- | --- |
| **UTN INSPT**  Instituto Nacional Superior del Profesorado Técnico |  |

|  |
| --- |
| TRABAJO PRÁCTICO |
| TÍTULO: “Sistema de reservas deportivas” |
| PROFESORA: Paula Ithurralde |
| MATERIA: Laboratorio |

|  |  |
| --- | --- |
| ALUMNOS: Adrian Rinaldo  Amelia R. Rodriguez    Martín Zembo | CURSO:  1° año |

|  |
| --- |
| FECHA DE REALIZACIÓN: 1/07/2023 |
| FECHA Y FIRMA DE APROBACIÓN DEL TP: |

**Introducción:**

El presente proyecto tiene como objetivo el desarrollo de un Sistema de Reservas Deportivas que permita a los usuarios realizar reservas de canchas en un club deportivo. El sistema brindará una interfaz intuitiva y amigable para que los usuarios puedan registrarse, acceder a información relevante sobre las reservas y consultar otros servicios relacionados con las actividades deportivas. Esta solución busca mejorar la experiencia de los usuarios al agilizar el proceso de reservas y proporcionar información actualizada sobre la disponibilidad de canchas y horarios.

**Marco teórico o conceptual:**

El sistema se basará en los principios de programación estructurada y se implementará utilizando el lenguaje de programación (pseudocódigo) *PseInt*. Se considerarán conceptos clave como el registro de usuarios, la gestión de reservas, la visualización de la disponibilidad de canchas y horarios, así como también la integración de un sistema de socios y métodos de pago. Se investigarán antecedentes relacionados con sistemas similares para asegurar una solución eficiente y acorde a las necesidades de los usuarios.

**Objetivos generales y específicos:**

* Objetivo general: Desarrollar un Sistema de Reservas Deportivas que permita a los usuarios realizar reservas de canchas de manera eficiente y obtener información actualizada sobre la disponibilidad y los servicios relacionados.
* **Objetivos específicos:**
  1. Implementar un módulo de registro de usuarios para permitir su acceso al sistema.
  2. Diseñar un menú principal que ofrezca opciones como reservar cancha, ver disponibilidad, consultar descuentos, sistema de socios y métodos de pago.
  3. Crear submenús para cada opción del menú principal, brindando una navegación intuitiva y facilitando el acceso a la información requerida.
  4. Desarrollar un módulo para mostrar el layout del club y las canchas mediante una demostración con dibujos.
  5. Implementar un sistema de gestión de reservas que permita a los usuarios reservar canchas y consultar las reservas existentes.
  6. Integrar un sistema de socios para brindar beneficios adicionales a los usuarios registrados.
  7. Incluir un módulo de métodos de pago que muestre las tarjetas aceptadas por el club y los posibles descuentos disponibles.

**Metodología:**

La metodología seleccionada para el desarrollo del Sistema de Reservas Deportivas será la programación estructurada utilizando el lenguaje (pseudocódigo) *PseInt*. Se empleará una estrategia metodológica basada en el análisis de requisitos, el diseño modular y la implementación paso a paso de cada funcionalidad. Se utilizarán técnicas de recopilación de información como encuestas y entrevistas con los usuarios para obtener datos relevantes sobre sus necesidades y preferencias. Los recursos disponibles incluirán el entorno de desarrollo PseInt y bibliotecas relevantes para el desarrollo de la interfaz gráfica. Se realizará un análisis exhaustivo de los datos recopilados para garantizar la calidad y eficiencia del sistema resultante.

**Primer Entrega (Anotaciones):**

* Código de reserva
* Buffet, consume con código de reserva
* Máximo de 1000 clientes por cancha
* Hora? Buscar si en PseInt puede linear hora real o preguntar el horario al cliente

**Hora:**

Para el tema de la hora buscamos si existe alguna función que pueda tomar la hora de la máquina para saber si reservo la cancha antes que otro usuario entonces tendría prioridad.  
Encontramos que en PseInt ya existe una función prehecha llamada FechaActual() que nos retorna un solo número entero en formato AAAAMMDD, a partir de ahí podemos cambiarla para darle el formato que nosotros queramos.

También existe la HoraActual() que retorna un solo número entero en formato HHMMSS.

**CRUD:**

Una idea que se nos ocurrió es confeccionar un CRUD que es el acrónimo de "Crear, Leer, Actualizar y Borrar" también conocido como ABMC (alta, baja, modificación y consulta).

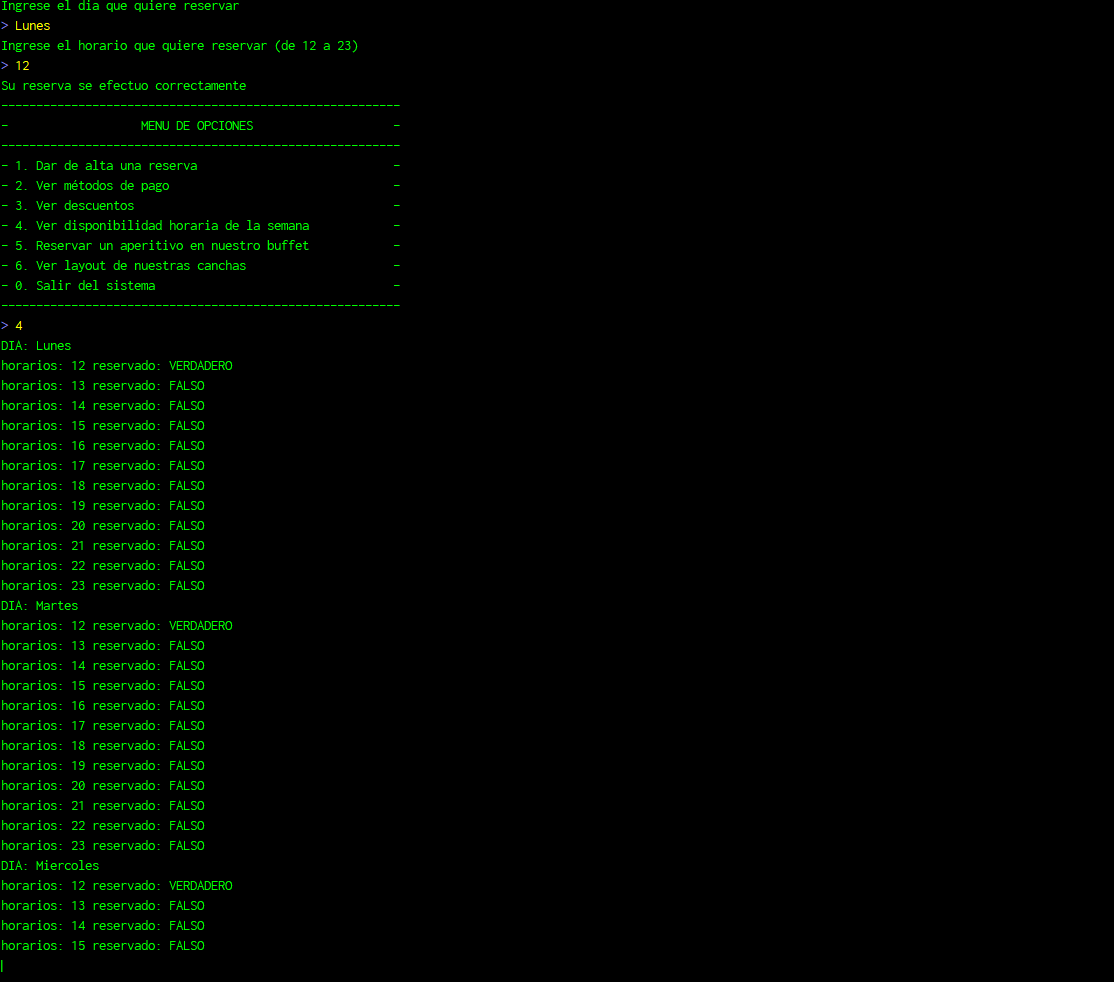
Esto nos permitiría tener una función que dé de alta, otra de baja, otra de modificación y una de consulta.

El alta sería efectuar la reserva, la baja sería eliminar la reserva, la modificación sería cambiar la reserva por otra disponible y la consulta sería ver la disponibilidad horaria de las canchas.

Esto nos facilita el trabajo ya que es una forma de las funciones básicas de la persistencia de bases de datos y nos ayudaría a estar más organizados a la hora de programar.

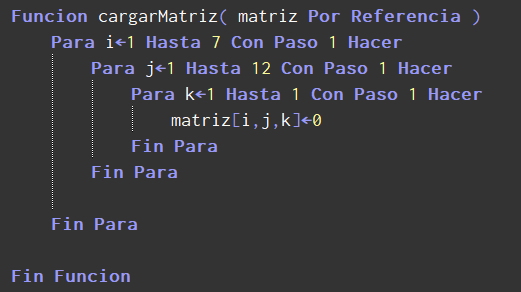
**Dificultades:**

A medida que empezamos a programar nos encontramos con algunos errores, creamos 3 arreglos uno para los días de la semana, otro para los horarios y un tercero que tiene el estado de la reserva (Verdadero o falso). El problema es que cuando efectúa el usuario una reserva y cambia el estado a Verdadero, en vez de cambiar solo para el dia elegido cambia todos los días ese horario a reservado, a continuación una imagen de ejemplo:

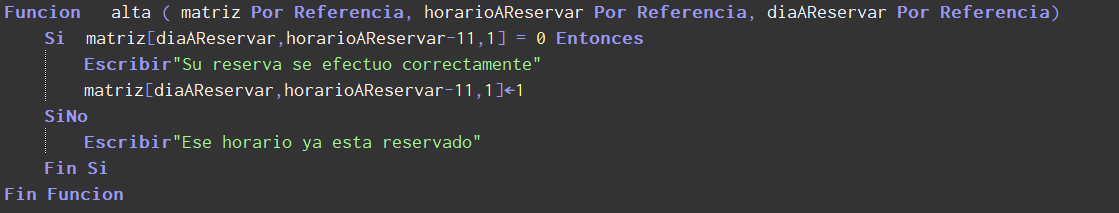


Creemos que para solucionar esto tenemos que crear una matriz tridimensional que tenga el día asociado al horario y estado de la reserva. El tema es que no sabemos bien cómo cargar la matriz para tener los valores predefinidos.

**Solución:**

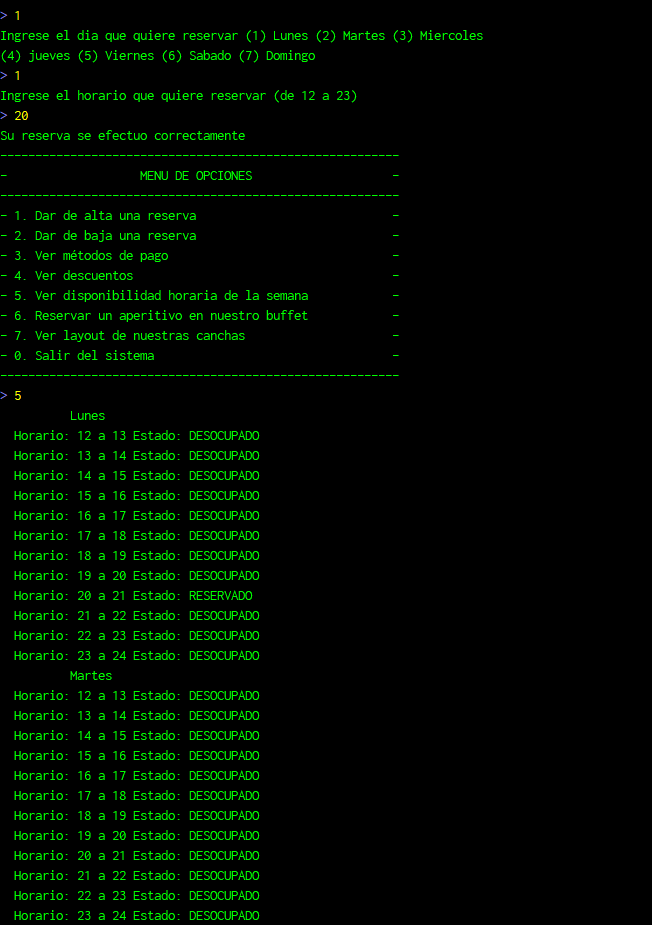
Se creó una función que inicializa la matriz tridimensional en 0 para todos sus campos.  
****

Luego para realizar una reserva(alta) se verifica que en ese día a ese horario sea = 0 porque sino ya está reservado, lo igualamos a 1 y mostramos que su reserva se efectuó correctamente.



En la siguiente imagen pueden ver cómo reservamos el día 1 (Lunes) a las 20 hs, el sistema nos muestra que nuestra reserva se efectuó correctamente ya que ese horario en ese día no estaba ocupado.

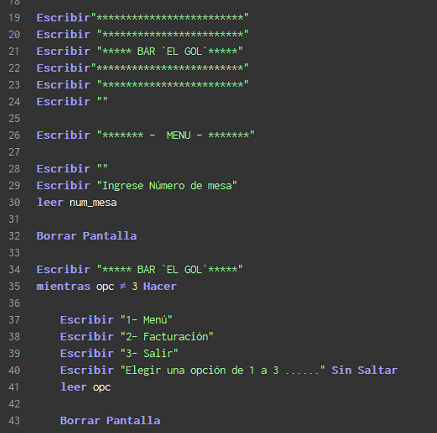
Luego ingresamos la opción 5 del menú (disponibilidad horaria) y nos muestra correctamente el Lunes a las 20hs RESERVADO y los demás días DESOCUPADO.

****

***Buffet:***

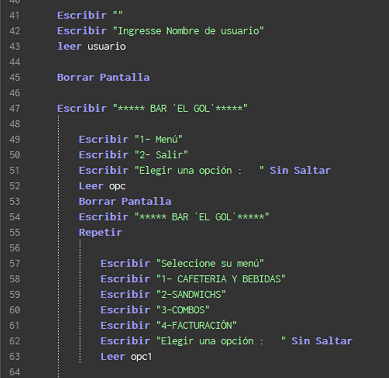
**Dificultades:**

Se realiza el boceto del programa de un buffet, se pensó como un lugar físico, dónde existan mesas y personal de atención, un “Bar”.



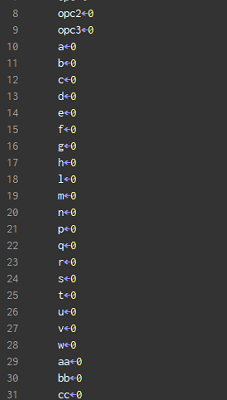
Luego se lleva a la forma de un “Candy Bar”, reserva en línea con el nombre de usuario.

Debemos resolver en grupo sí sólo se realiza un caso, o se podría incluir ambas formas de pedido.



Luego en la realización del boceto del programa vamos incluyendo demasiadas variables, las cuales no deben coincidir con las variables de la matriz.

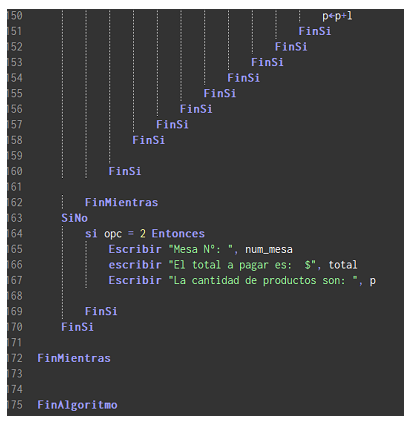
Al quedarnos sin alfabeto introducimos la repetición de las letras (aa, bb….etc).



En una primera instancia, se piensa el programa como un “Si anidado”, con pocas opciones del menú para ir probando si el programa corría y realizaba lo que se necesitaba.

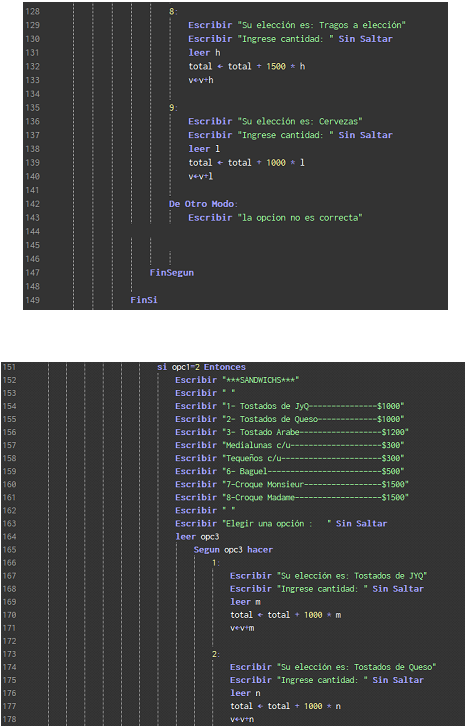


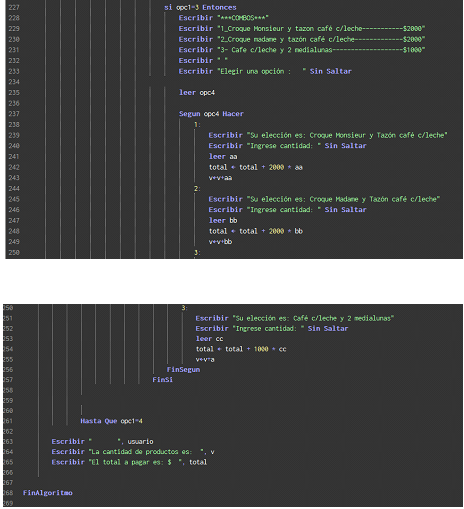
Y así sucesivamente, lo que no hacía el programa era, por ejemplo, si la persona quería agregar algo más al pedido antes de facturar, sí o sí se debía “salir”.



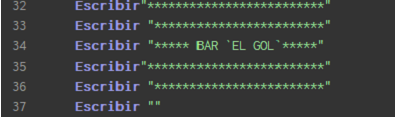
Con estos inconvenientes que presentaba el boceto del Buffet, se plantea de darle una estructura del tipo “Según”, con esta nueva estructura se comienza a agregar más ítems en el menú de elecciones.







Otra de las dificultades que presenta el boceto es sobre el diseño, se tiene que darle más visión gráfica para que capte la atención del usuario.



El nombre del Buffet todavía no fue conversado, al igual que las formas de pago o medios a utilizar.

El proyecto del Buffet sigue en la etapa de planificación.

***Registro e inicio de sesión:***

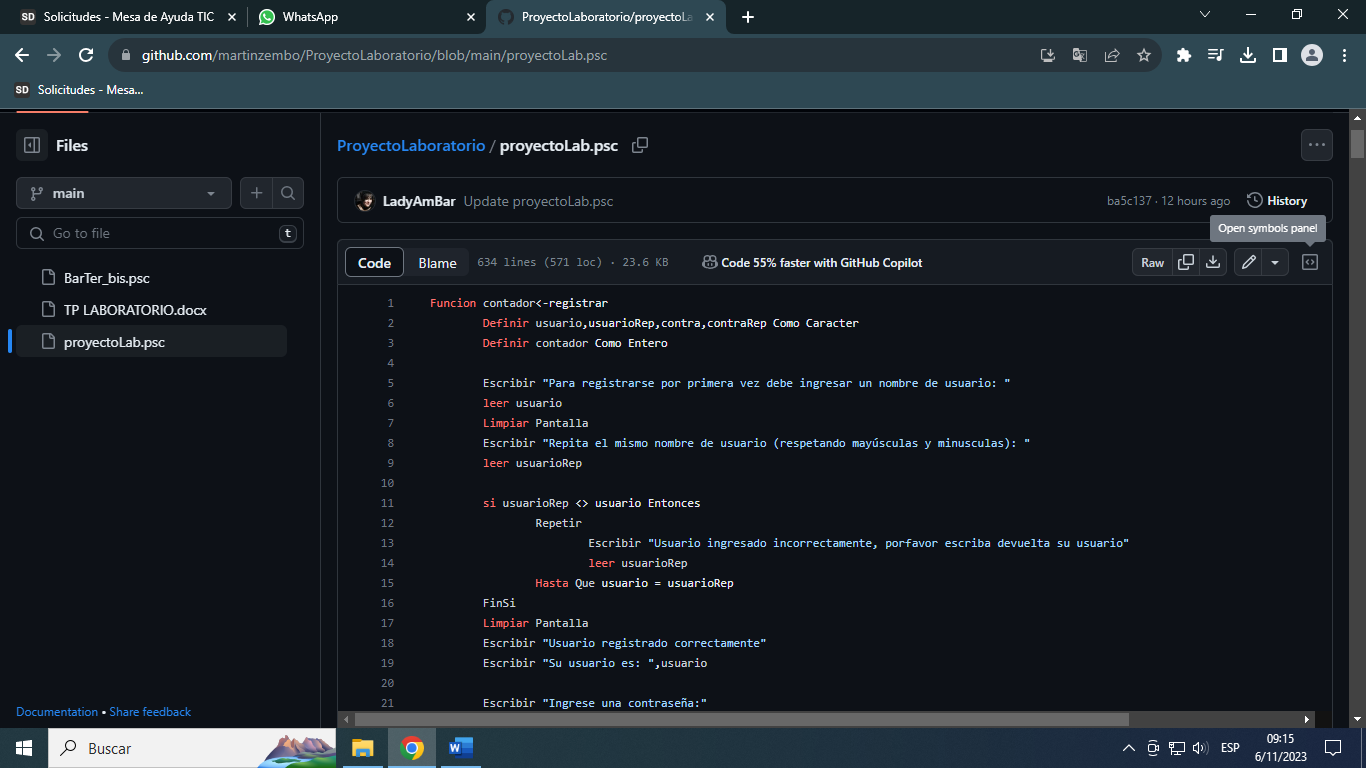
Para registrarse, una vez iniciado el programa, el sistema le pedirá al usuario un nombre de usuario, luego que repita el mismo nombre (respetando mayúsculas y minúsculas). En esta instancia no ponemos un limite de intentos ya que nos pareció mejor que genere el usuario si o si y que luego al iniciarse con ese usuario si tenga un máximo de 5 intentos.

Luego lo mismo para registrar su contraseña, y una vez registrado el sistema le pide al usuario que ingrese el usuario previamente creado (máximo de 5 intentos) y luego la contraseña también con un máximo de 5 intentos.

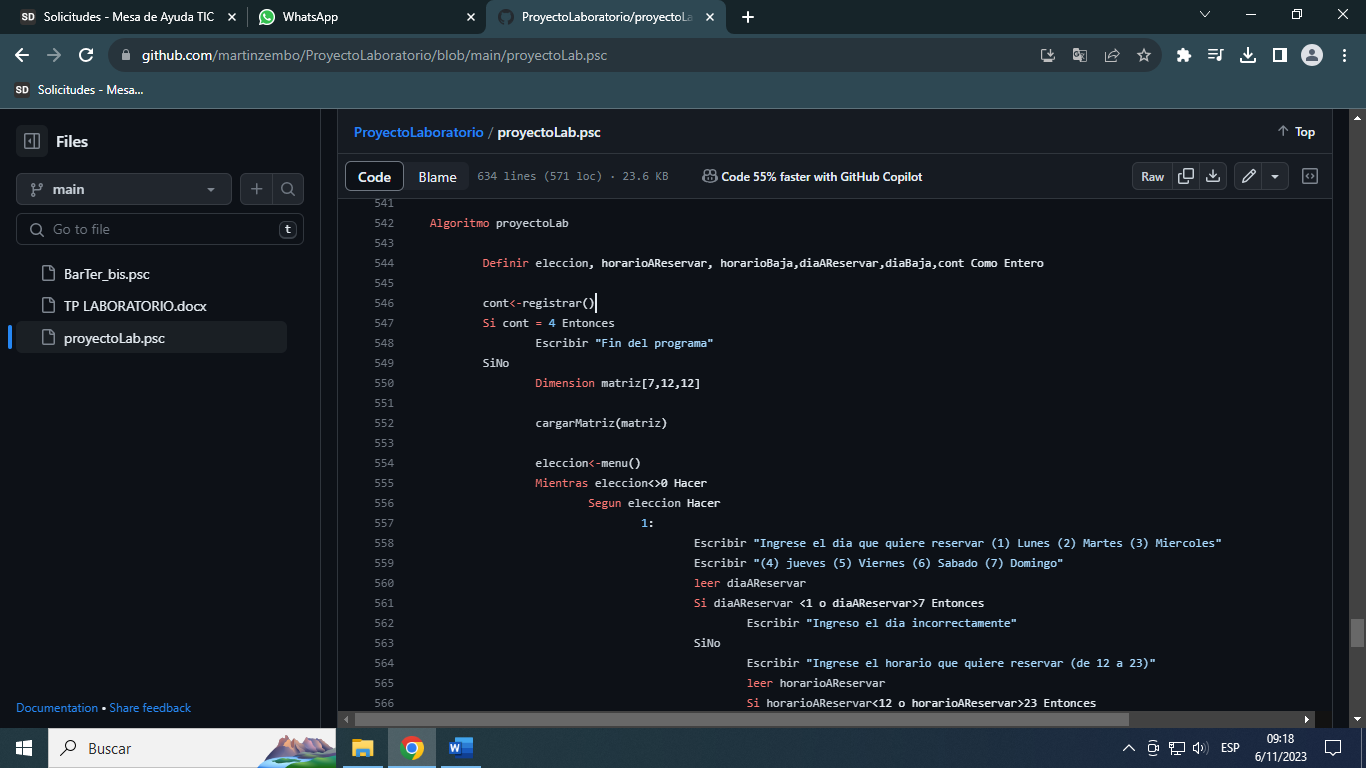
**Dificultades:**

Las dificultades que se nos presentaron fueron mas que nada al agregar los intentos del usuario, ya que una vez finalizados los intentos el programa debe largar un mensaje de error y terminar su ejecución.

Esto se arregló con un contador en la función para contar la cantidad de intentos fallidos del usuario, pero para finalizar con la ejecución tengo que tener la cantidad de intentos en el main(algoritmo), por lo tanto, la función misma debe devolver el contador para que en el main verifique si llegó a los 5 intentos, si llegó el programa finaliza, sino sigue con su ejecución.



En el algoritmo principal:



Y continúa con el código principal.